imc STUDIO 5.0

Manual

Doc. Rev.: 3.7 - 2017-05-15

© 2017 imc Meßsysteme GmbH

imc Meßsysteme GmbH • Voltastraße 5 • 13355 Berlin • Germany

Foreword

Thank you for deciding to purchase our product. We wish you total success in accomplishing your measurement assignments with the help of your hardware and software. If you have any open questions about our products, please contact our Hotline (hotline@imc-berlin.de).

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 gratefully accept any suggestions for improvements, please contact our Hotline (hotline@imc-berlin.de).

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

Copyright

© 2017 imc Meßsysteme GmbH, Berlin, Germany

This documentation is the intellectual property of imc Meßsysteme GmbH. imc Meßsysteme 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".

imc Software and Microsoft® Windows

imc software runs on the Microsoft® Windows operating system.

GPL Sources

Some components of our hardware use software, that is licensed under GNU General Public License (GPL). A description can be found at the imc STUDIO setup DVD in folder "*Products\imc DEVICES\OSS*".

1 imc STUDIO - Help ... 12
2 General Notes ... 13
2.1 Before you Start ... 13
2.2 Notes / Quality Management / Customer Support ... 13
2.2.1 Quality Management ... 13
2.2.2 imc Warranty ... 13
2.2.3 imc Customer Support / Hotline ... 13
2.2.4 Product Improvement / Change Requests ... 14
2.3 imc Software License Agreement ... 14
2.4 Documentation ... 19
3 Setting Up ... 20
3.1 System requirements ... 20
3.2 Installation - Preparation ... 21
3.2.1 Update with the help of the existing database ... 23
3.2.2 Update without using the existing database ... 23
3.2.3 Notes and Troubleshooting ... 25
3.3 Installation Step by Step ... 26
3.3.1 Demo and Auto ... 29
3.3.2 User-defined ... 31
3.3.3 Starting the Installation ... 35
3.4 Product Configuration / Licensing ... 36
3.5 Start ... 39
3.5.1 Device Connection / Network / Firewall ... 41
3.5.2 Plug-ins / Main Windows ... 42
3.5.3 Important Settings ... 43
3.5.4 Command Line Parameters ... 43
3.6 Info / Version Information ... 45
3.7 Information and Tips ... 45
3.7.1 Installing Additional Languages ... 45
3.7.2 Recommended Virus Scanner Settings ... 46
3.7.3 Update and Compatibility ... 46
4 imc STUDIO - General ... 51
4.1 Ribbon ... 51
4.1.1 Project Menu ... 51
4.1.2 Edit Menu ... 56
4.1.3 View Menu ... 57
4.1.4 Extra Menu ... 58
4.2 Navigation Pane ... 67
4.3 Tool Windows ... 67
4.3.1 Operation ... 68
4.3.2 Logbook ... 69
4.4 User Administration and Access rights ... 73
4.4.1 User Administration ... 74
4.4.2 Startup Behavior ... 76
4.4.3 Logging On, Logging Off and Switching Users ... 77
4.4.4 Setting Access Rights ... 79
4.5 Views ... 80
4.5.1 Exporting/importing views ... 82
4.6 Information and Tips ... 83
4.6.1 Projects, Experiment templates and Experiments ... 83

4.6.2 Data Management ... 84
4.6.3 What is saved where? ... 85
5 Setup - Device Configuration ... 87
5.1 Device Overview ... 88
5.2 Ribbon ... 89
5.2.1 Control ... 90
5.2.2 Configuration ... 98
5.3 Tool Windows ... 119
5.3.1 Details ... 119
5.3.2 Sensors ... 122
5.4 Operation ... 129
5.4.1 Tabular Display ... 129
5.4.2 Dialogs ... 137
5.5 Performing a measurement: Procedure ... 139
5.5.1 Device Search - Known and Selected ... 140
5.5.2 Firmware ... 142
5.6 Configuring Device - Setup pages ... 147
5.6.1 Documentation (Experiment Description) ... 148
5.6.2 Configuring Devices ... 148
5.6.3 Configuring Channels and Variables ... 159
5.6.4 Channel balance ... 177
5.6.5 Trigger and Events ... 182
5.6.6 TEDS - Sensors ... 201
5.6.7 Additional Pages ... 205
5.7 Metadata and Additional Columns ... 206
5.7.1 Metadata Assistant ... 206
5.7.2 Additional Columns ... 211
5.7.3 Table description and complete layout ... 228
5.8 Information and Tips ... 229
5.8.1 Incremental Encoders ... 229
5.8.2 Parameter set ... 239
5.8.3 Enumeration Classes ... 254
5.8.4 Transferring an experiment to other devices ... 255
5.8.5 Import of an imc DEVICES Experiment ... 257
5.9 Tutorial ... 259
5.9.1 Simple measurement - First Steps ... 260
5.9.2 Subsequent saving of measured data ... 265
5.9.3 Pre-defined saving of measured data ... 268
5.9.4 Simple Triggering ... 272
5.9.5 Simple Triggering - Enhanced ... 279
5.9.6 Artificial sine ... 281
5.9.7 Triggered measurement ... 285
5.9.8 Thermocouple measurement ... 291
6 Setup - Connect the device ... 295
6.1 Connection via LAN ... 295
6.2 Connecting via LAN in four steps ... 296
6.3 Special options for connecting to the device ... 301
6.3.1 Devices and TCP/IP ... 301
6.3.2 Device connection via LAN ... 307
6.3.3 Device connection via WLAN ... 313
6.3.4 Device connection via Modem (TCP/IP with PPP) ... 321
6.4 Optimize the network ... 335
6.5 Complex Networks ... 336

7 Setup - Advanced Device Functions ... 337
7.1 Device Overview ... 337
7.2 Fieldbusses ... 338
7.2.1 General Notes on Field-Busses ... 338
7.2.2 ARINC-Bus Interface ... 344
7.2.3 AFDX-Bus Interface ... 369
7.2.4 CAN-Bus Interface ... 382
7.2.5 EtherCAT-Interface ... 458
7.2.6 Ethernet-Bus Interface ... 469
7.2.7 FlexRay-Bus Interface ... 482
7.2.8 imc Application Module ... 499
7.2.9 imc HiL: Hardware in the Loop ... 500
7.2.10 J1587-Bus Interface ... 501
7.2.11 LIN-Bus Interface ... 518
7.2.12 MVB-Bus Interface ... 539
7.2.13 PROFIBUS Interface ... 558
7.2.14 RoaDyn Interface ... 569
7.2.15 SPI-Bus Interface ... 574
7.2.16 XCPoE-Bus Interface ... 580
7.2.17 GPS ... 586
7.3 Storage Options and Directory Structure ... 586
7.3.1 Storage Options - Overview ... 587
7.3.2 Data Processing and Storage in the Device ... 592
7.3.3 Data Processing and Storage on the PC ... 594
7.3.4 Saving Interval ... 598
7.3.5 Saving Data on the Network ... 601
7.3.6 Data Storage and Directory Structure ... 603
7.3.7 Saving - Miscellaneous ... 607
7.3.8 Controlling Data Storage ... 610
7.3.9 Viewing Saved Measurement Data ... 616
7.4 Device Hard Disk, removable drive ... 616
7.4.1 Internal Storage Media ... 617
7.4.2 Data Transfer ... 620
7.4.3 Partition and File System ... 624
7.4.4 Notes on the use of the Storage Media ... 626
7.5 Synchronization ... 629
7.5.1 What's the purpose of Synchronization? ... 629
7.5.2 Definitions ... 630
7.5.3 Clock types ... 632
7.5.4 Phase offset correction ... 640
7.5.5 Master/Slave setup ... 640
7.5.6 Separate synchronization of all devices ... 644
7.5.7 Synchronized start with multi-master-devices ... 649
7.5.8 Working with time zones ... 649
7.5.9 Application notes ... 655
7.6 imc Display Editor ... 664
7.6.1 Overview ... 664
7.6.2 Compatibility Mode ... 665
7.6.3 Display Editor in imc STUDIO ... 666
7.6.4 System Menu ... 667
7.6.5 Keyboard ... 668
7.6.6 Diskstart ... 669
7.6.7 Graphical Display Elements ... 670
7.6.8 Function Keys on the Display ... 676
7.6.9 Firmware Update ... 682
7.7 imc Messaging ... 682

7.7.1 Overview ... 682
7.7.2 Message Types ... 684
7.7.3 Reliability ... 685
7.7.4 Configuration ... 687
7.7.5 Special characteristics of using network bits ... 699
7.7.6 UDP Status Monitoring ... 700
7.7.7 WakeOnLAN (Magic Packets) ... 702
7.7.8 Possible error sources ... 702
7.7.9 Technical Specifications imc Messaging ... 704
7.7.10 Templates ... 704
7.7.11 Step by Step ... 706
7.8 Configure via FTP ... 707
7.8.1 How does configuration via FTP work? ... 709
7.8.2 Syntax of command call and response ... 713
7.9 imc REMOTE WebServer ... 720
7.9.1 System requirements ... 721
7.9.2 Activation ... 722
7.9.3 Calling from the Browser ... 723
7.9.4 User Role ... 725
7.9.5 Experiment ... 726
7.9.6 Current values ... 727
7.9.7 Files ... 730
7.9.8 Panel ... 731
7.9.9 Miscellaneous ... 738
7.9.10 Event history ... 739
7.10 imc Online FAMOS ... 739
7.10.1 imc Online FAMOS - Overview ... 740
7.10.2 Starting imc Online FAMOS ... 747
7.10.3 Main Window Elements ... 748
7.10.4 Brief Introduction ... 750
7.10.5 Variables and Syntax ... 751
7.10.6 Variables and Syntax with Control Commands ... 756
7.10.7 Types of Variables ... 761
7.10.8 Menu ... 768
7.10.9 Context Menu ... 773
7.10.10 Formula Assistant ... 775
7.10.11 Function Dialog ... 777
7.10.12 Information and Tips ... 781
7.10.13 Function Reference ... 792
7.11 Miscellaneous ... 792
7.11.1 Tuning, Tips and Tricks ... 792
7.11.2 FAQ ... 797
7.11.3 Troubleshooting ... 807
7.11.4 Glossary ... 809
7.11.5 File Extensions ... 811
8 Panel ... 813
8.1 Ribbon ... 814
8.1.1 Control ... 814
8.1.2 Design ... 814
8.2 Tool Windows ... 815
8.2.1 Properties (general) ... 816
8.2.2 Widgets ... 817
8.2.3 Data Browser ... 818
8.2.4 Page templates ... 828
8.2.5 Repository ... 828
8.3 Context Menu ... 829

8.4 Design Mode ... 832
8.5 Widgets - Operation and Properties ... 833
8.5.1 Insert Widget ... 834
8.5.2 Context menus of the Widgets ... 834
8.5.3 Operation ... 835
8.5.4 Properties - Widget ... 843
8.6 Curve Window ... 854
8.6.1 Introduction ... 854
8.6.2 "File" menu ... 855
8.6.3 "Edit" menu ... 864
8.6.4 "Configuration" menu ... 909
8.6.5 "Options" menu ... 1044
8.6.6 "Extras" menu ... 1060
8.6.7 Greek texts in comments, markers and axis labels ... 1069
8.6.8 Curve Window Context Menu ... 1072
8.6.9 Copy&Paste - Transfer configuration ... 1072
8.6.10 Drag and Drop, mouse wheel ... 1072
8.6.11 Axes Navigation Bar ... 1073
8.6.12 Last Changes ... 1073
8.7 Special Widgets ... 1074
8.7.1 Text input for report channel ... 1074
8.7.2 Audio ... 1076
8.7.3 Curve Window - "Free-floating" ... 1077
8.8 Pages ... 1079
8.8.1 Insert - Dialog / Report ... 1079
8.8.2 Properties - Pages ... 1081
8.8.3 Adapt/automatically adapting the size of pages and Widgets ... 1083
8.8.4 Lock and unlock page ... 1084
8.8.5 Aligning to grid ... 1085
8.8.6 Zooming Panel pages ... 1085
8.8.7 Fullscreen ... 1086
8.9 Variable Linkage ... 1089
8.10 Navigation Bar ... 1093
8.10.1 Menu ... 1094
8.10.2 Linking a Widget ... 1095
8.10.3 Slider ... 1096
8.10.4 Linking variables ... 1097
8.10.5 Navigation by marker ... 1098
8.10.6 Find in Variables ... 1099
8.10.7 Post-processing mode ... 1101
9 Automation ... 1102
9.1 Structure of the plug-in Automation ... 1103
9.2 How to create a operating plan? ... 1103
9.3 How does it work? ... 1105
9.4 Operation ... 1107
9.4.1 Operation in the Task Management ... 1107
9.4.2 Operation in the Task and Block Editor ... 1108
9.5 Source Code - Find and Replace ... 1111
9.6 Trace Info ... 1113
9.7 Tool window Automation controls ... 1113
9.7.1 Templates ... 1113
9.7.2 Repository ... 1137
9.7.3 Resources ... 1137

9.7.4 Functions ... 1138
9.8 Automation Editor ... 1140
9.8.1 Task Management ... 1140
9.8.2 Task Editor ... 1149
9.8.3 Block Editor ... 1155
9.9 Tutorial ... 1158
9.9.1 Blinking LED using timer ... 1159
9.9.2 Blinking LED using Cycle Time ... 1167
9.9.3 Traffic light ... 1173
9.9.4 Limit value monitoring ... 1179
10 Sequencer ... 1189
10.1 Tool window Commands ... 1190
10.2 Sequence table ... 1191
10.2.1 Display in the sequence table ... 1191
10.2.2 Groups ... 1192
10.3 Context menu ... 1193
10.4 Creating a sequence of commands ... 1193
10.4.1 Adding commands by means of Drag&Drop ... 1194
10.4.2 Adding a command by means of the context menu ... 1194
10.4.3 Deleting, copying, moving commands ... 1195
10.5 Running Sequencer ... 1196
10.6 Events ... 1197
10.6.1 Panel ... 1197
10.6.2 Project ... 1198
10.6.3 Sequencer ... 1199
10.6.4 User-defined events ... 1199
10.7 Tutorial ... 1202
10.7.1 Starting the Sequencer ... 1202
10.7.2 Output box ... 1204
10.7.3 Scrolling through the Panel pages ... 1205
10.7.4 Dialog ... 1208
10.7.5 Use of imc FAMOS ... 1213
10.7.6 Conditional branches ... 1216
10.7.7 Example: Wind Power ... 1220
11 Monitor ... 1235
11.1 Ribbon ... 1235
11.2 System Prerequisites and Limitations ... 1235
11.3 Tool Window ... 1236
11.3.1 Devices / Channels ... 1236
11.4 Information and Tips ... 1237
11.5 Tutorial ... 1238
11.5.1 Monitoring - First Steps ... 1238
12 Data Processing ... 1241
12.1 Ribbon ... 1242
12.2 Inline FAMOS ... 1243
12.2.1 Comparison: imc Online FAMOS / imc Inline FAMOS ... 1244
12.2.2 System Prerequisites and Limitations ... 1247
12.2.3 Scope of Functions ... 1248
12.2.4 Operation ... 1249
12.2.5 Context Menu ... 1252
12.3 Power Quality ... 1253

12.3.1 Opening Power Quality ... 1254
12.3.2 User Interface ... 1254
12.3.3 Introductory Example ... 1257
12.3.4 Output Parameters ... 1262
12.3.5 Performing Calculations ... 1264
12.3.6 Context Menu ... 1265
12.3.7 Basics ... 1265
12.4 Bus Decoder ... 1269
12.5 Powertrain Monitoring ... 1269
12.5.1 Assistant ... 1270
12.5.2 Editor ... 1272
12.5.3 Base Diagnosis ... 1272
12.5.4 Advanced Diagnosis ... 1278
12.5.5 Visualization ... 1287
13 Video ... 1292
13.1 Setup Configuration ... 1292
13.1.1 Saving ... 1294
13.1.2 Video measurement with triggering ... 1295
13.2 Displaying Video-Channels on a Panel Page ... 1296
13.3 Export/Import of video files ... 1301
13.4 Information and tips ... 1301
13.4.1 Specifications and Notes ... 1301
13.4.2 Video Compression ... 1303
13.4.3 Camera: GoPro Hero 4 ... 1305
13.4.4 Integrated imc STUDIO-PC with video ... 1307
14 Scripting ... 1309
14.1 Menu bar and context menu ... 1310
14.1.1 Create scripts ... 1312
14.1.2 Edit scripts ... 1317
14.1.3 Run scripts ... 1317
14.1.4 Export and Import ... 1318
14.1.5 Regenerate proxies ... 1319
14.1.6 Debugging ... 1320
14.2 Script (general) ... 1323
14.2.1 Device functions ... 1323
14.2.2 Channel functions ... 1327
14.2.3 Panel and Widget functions ... 1334
14.2.4 Datapool functions ... 1338
14.2.5 Curve window ... 1342
14.2.6 Logbook ... 1344
14.2.7 imc FAMOS functions ... 1346
14.2.8 Inline FAMOS functions ... 1349
14.2.9 Commands ... 1351
14.2.10 Additional functions ... 1354
14.3 Script types ... 1356
14.3.1 Script ... 1356
14.3.2 Panel Script ... 1358
14.3.3 Context Script ... 1365
14.3.4 Type Library Script ... 1369
14.3.5 Event Script ... 1372
14.3.6 Third Party Device Script ... 1374
14.4 EClassID - Parameters ... 1374
14.4.1 Device Parameters ... 1374
14.4.2 Channel Parameters ... 1377

14.4.3 Experiment Parameters ... 1383
14.4.4 Finding parameters and setting values ... 1385
14.5 Script options ... 1390
14.6 Tutorial ... 1391
14.6.1 Device and channel settings ... 1392
14.6.2 Channel calculation ... 1397
14.6.3 Curve Window and channel selection ... 1398
14.6.4 RS232-Interface ... 1401
14.6.5 Using the script clipboard ... 1402
14.7 Important Information ... 1403
15 Third Party Device Interface ... 1407
15.1 Getting started ... 1407
15.1.1 Introduction ... 1407
15.1.2 Add a new script ... 1407
15.2 Serial interface ... 1409
15.2.1 Device control ... 1409
15.3 Third Party Device script ... 1410
15.3.1 Add a device ... 1410
15.3.2 Add a channel ... 1410
15.3.3 Action column (eAction) ... 1412
15.3.4 Data transfer ... 1413
15.4 3rd Party Device Management ... 1414
15.4.1 Device Scripts ... 1415
15.4.2 Device script templates ... 1424
15.5 Tutorial ... 1435
15.5.1 Extension of the SimplePollDevice ... 1435
16 Shared Components ... 1440
16.1 imc Online/Inline FAMOS Function Reference ... 1440
16.1.1 Operators and Symbols ... 1440
16.1.2 Functions (alphabetic) ... 1445
16.1.3 Functions with Control Commands ... 1572
16.2 Command Reference ... 1592
16.2.1 Flow control ... 1594
16.2.2 User interacion ... 1599
16.2.3 Read/Write data ... 1608
16.2.4 Open experiment/Start measurement ... 1615
16.2.5 External calls ... 1616
16.2.6 Panel ... 1620
16.2.7 Parameter set ... 1626
16.2.8 Setup ... 1635
16.2.9 Scripts/imc FAMOS ... 1637
16.2.10 Variables ... 1645
16.3 Command Sequence ... 1650
16.3.1 Event Dialog ... 1651
16.3.2 Creating a sequence of commands ... 1652
16.4 Placeholders ... 1655
16.4.1 List of placeholders ... 1657
16.4.2 Formatting ... 1668
16.4.3 Environment variables of the OS ... 1670
16.5 User-Defined Variables ... 1671
16.5.1 Creating variables ... 1672
16.5.2 Report Channel ... 1675

17 Technical Data Sheets 1678
17.1 imc STUDIO 1678
17.1.1 Licensing 1679
17.1.2 System Requirements 1680
17.2 Plug-in imc STUDIO Setup 1681
17.3 Plug-in imc STUDIO Panel 1685
17.4 Plug-in imc STUDIO Sequencer 1687
17.5 Plug-in imc STUDIO Automation 1688
17.6 Plug-in imc STUDIO Scripting 1689
17.7 Plug-in imc STUDIO Third Party Device Interface 1690
17.8 Plug-in imc STUDIO DataProcessing 1691
17.9 Functions Package imc Inline FAMOS 1692
17.10 Functions Package Power Quality 1693
17.11 Functions Package Bus Decoder 1694
17.12 Functions Package Powertrain Monitoring 1694
17.13 Plug-in imc STUDIO Project Management 1695
17.14 Plug-in imc STUDIO Video 1696
17.14.1 Specifications and Notes 1696
17.14.2 Supported Cameras 1698
17.14.3 Frame Grabber and Compression 1701
17.15 imc STUDIO Monitor 1701
17.16 Additional imc Software Products (optional) 1702
17.17 imc Online FAMOS 1703
17.18 imc Online FAMOS Professional 1706
17.19 imc Online FAMOS Class-Counting 1707
17.20 imc Online FAMOS Order-Tracking Analysis 1708
18 Last Changes 1709
18.1 in Doc. Rev. 3.7 1709
18.2 in Doc. Rev. 3.6 1709
18.3 in Doc. Rev. 3.5 1712
18.4 in Doc. Rev. 3.3 1712
18.5 in Doc. Rev. 3.2 1715
18.6 in Doc. Rev. 3.1 1716
18.7 in Doc. Rev. 3.0 1718
18.8 in Doc. Rev. 2.2 1727
18.9 in Doc. Rev. 2.1 1729
Index 1730

1 imc STUDIO - Help

Welcome to imc STUDIO

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

imc STUDIO represent a modular system. For this reason, the descriptions of the components are presented in separate chapters:

- imc STUDIO (general) 51
- Setup 87
- Setup - Connect the device 295
- Setup - Advanced Device Functions 337
- Panel 813
- Sequencer 1189
- Automation 1102
- Video 1292
- Monitor 1235

Customer Support / Hotline

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

Questions or problems? Contact our Customer Support / Hotline 13.

Copyright

© 2017 imc Meßsysteme GmbH. All rights reserved.

2 General Notes

2.1 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 Hotline 13.
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 Hotline.
3. Please contact our Hotline 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 14. By using the software, you agree to the terms and conditions of the license agreement.

2.2 Notes / Quality Management / Customer Support

2.2.1 Quality Management

imc Meßsysteme GmbH holds DIN-EN-ISO-9001 certification since May 1995.

You can download the CE Certification, current certificates and information about the imc quality system on our Webpage: www.imc-berlin.com/about-imc/quality-assurance.

2.2.2 imc Warranty

Subject to the general terms and conditions of imc Meßsysteme GmbH.

2.2.3 imc Customer Support / Hotline

If you have problems or questions, please contact our Customer Support/Hotline:

Germany:

imc Meßsysteme GmbH

Phone: +49 (0) 30 / 467 090 - 26

Fax: +49 (0) 30 / 463 15 76

WWW: www.imc-berlin.com

E-Mail: hotline@imc-berlin.de

For our international partners see www.imc-berlin.com/our-partners/distributor.

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. Thank you!

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

2.2.4 Product Improvement / Change Requests

Please help us to improve our documentation:

- What terms or descriptions are incomprehensible?
- What additions and enhancements you suggest?
- Where have material mistakes slipped in?
- Which spelling or typing errors have you found?

Responses and other feedback should be directed to the Hotline 13 (phone / e-mail) or by writing to:

imc Meßsysteme GmbH

Voltastrasse 5

D - 13355 Berlin

Germany

2.3 imc Software License Agreement

imc Meßsysteme GmbH
Voltastrasse 5
13355 Berlin

Commercial register: Berlin-Charlottenburg HRB 28778
Managing director: Dr.- Ing. Dietmar Sprenger

imc Meßsyteme GmbH Software

Terms and Conditions

Governing the Use of imc Meßsysteme GmbH Software

As of: June 1, 2013

§ 1
Objects of Agreement

(1) In addition to the "General Terms and Conditions Governing Meßsysteme GmbH Deliveries and Services to Customers", these terms and conditions apply to all contracts concluded with imc Meßsysteme 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.

(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
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.

§ 6
License Key

(1) The Licensor may, after serving the customer 3 months written prior notice, terminate this contract after 12 months at the earliest. This contract can be terminated by either party with immediate effect if important grounds exist. This is especially relevant to noncompliance with the stipulations in § 2 (Scope of Use) and § 3 (Protection of Licensed Material).

(2) Upon receipt of notice of termination, irrespective of its date and grounds, the Customer is obligated to return the original and all copies whether complete or partial of the licensed material to the Licensor. In the case of licensed material copied onto machine-decodable data carriers which are the property of the Customer, the Customer is obligated to destroy completely the copies rather than to return them.

§ 7
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) All disputes arising from or in association with this agreement are to be resolved by an arbitration tribunal under exclusion of the public courts. The arbitration tribunal also rules on the valid interpretation of this arbitration clause. The arbitration tribunal consists of three arbitrators. The arbitrators, which are to be appointed by the parties, must be qualified to hold the office of a judge in the Federal Republic of Germany. The chairman must be an active or retired judge of the Federal Republic of Germany. With regard to appointing the arbitrators, § 1035 of the German Code of Civil Procedure (ZPO) applies. Multiple plaintiffs or defendants may only mutually appoint an arbitrator. The place of the arbitration proceedings is the registered seat of imc. The language of the proceedings is German. In addition, the provisions of the Tenth Book of the ZPO are applicable to the arbitration proceedings. The arbitrators receive customary remuneration as well as compensation for expenses. As soon as the participation of the public courts is necessary, the competent courts at imc's registered seat have jurisdiction.

(5) 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.

2.4 Documentation

Help

All imc STUDIO products come with help (CHM format). In the menu bar, click on the symbol to open the help. 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.

Notes on the descriptions and the screenshots

- 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 73. 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.

Documentation in PDF format

Certain selected parts of the documentation may additionally be present in PDF (Portable Document Format) format. This format is better adapted to printout. To read PDF files, the Adobe Reader by Adobe Inc. is required. This software is already installed on some PC systems. The newest version of the Adobe Reader for your PC operating system can be downloaded for free from Adobe Inc. (www.adobe.com).

3 Setting Up

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).

Start here

- System requirements 20
- Installation / Uninstall 21
- Product Configuration / Licensing 36
- Device connection / Network / Firewall 41
- Version Information 45

Customer Support - Hotline

Questions or problems? Contact our Customer Support / Hotline 13.

3.1 System requirements

Supported operating systems
Windows 10
Windows 8 / 8.1
Windows 7 (32 bit; recommended: 64 bit)
Windows Vista (32 bit) as of SP1
Windows XP (32 bit) as of SP3

Minimum requirements for the PC [1]	Recommended configuration for the PC [2]
Hyper-threading or Dual Core processor with 2 GHz clock speed	Quad Core processor with 2 GHz clock speed or higher
2 GB RAM / 4 GB RAM for Windows 7 and 8 (64 bit)	3 GB RAM (32 bit) / ≥ 8 GB RAM (64 bit)
10 GB free hard disk space (NTFS format)	10 GB free hard disk space (NTFS format)
For installation of the software via DVD an appropriate drive is needed	For software installation via DVD an appropriate drive is needed
Color graphics (16 bit color depth)	True color graphics (32 bit color depth)
Display resolution 1280 x 768	Display resolution: 1280 x 1024 or more
	64 bit operating system

1 A system with minimum requirements is not adequate for connection with multiple devices and complex design tasks with the imc STUDIO Developer. Use such systems preferably only for data monitoring purposes.

2 The requirements for the PC's configuration increase with the number of devices involved and the scope of the Data Processing-calculations to be performed.

Supported Measurement Devices

Which devices you can use in imc STUDIO is described in the documentation on the *Setup* > *Device Overview* 88, as well as in the *Technical Data Sheet* 1678. For the purpose of connecting with imc STUDIO Monitor, the devices must additionally have at least 32 MB of internal (interface) device memory available (which is assured for all current device models, except for a limited number of imc CRONOS-PL and imc CRONOS-SL devices dating back to before 2007).

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 36).

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 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 25.

Restarting the computer in the process of installation

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

Notes on 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. 5.0R5), 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 WAVE and imc STUDIO, which must be performed in a single step.

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.

You can keep both versions installed in parallel, as long as the version numbers are different (e.g. 5.0R3 and 5.0R5). The new version can be installed in the same folder (default case: "C:\Program Files\imc" for 32-bit Win7). 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)

If you wish to continue using the existing database, the possible settings will be adopted. 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 23.

Adopting the settings without using the existing database

You can apply the settings, such as the views, without using the database. To do this, save and import the appropriate settings.
See: Update without using the existing database 23.

Notes on the update

Please also observe the notes regarding the update and compatibility 46!

The subsequent procedure - an overview

Follow the installation program instructions (see: "Installation step by step 26").

- 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, you can use the products.

Uninstalling

To uninstall the product, use the Windows Control Panel/Settings and select the respective entry:

- Control Panel / Add or Remove Programs (Windows XP) or
- Control Panel / Programs (Uninstall a Program) (Windows Vista / Windows 7)
- Settings / Apps & features (Windows 10)

3.2.1 Update with the help of the existing database

If you continue to use the existing database, whatever settings are possible will be adopted. This includes Views, any Setup-columns created by the user, the user administration (as of imc STUDIO 4.x and higher) and all experiments.

Notes on the database

- It is impossible to use the imc STUDIO database by both versions in parallel.
- If the same path in the new imc STUDIO version is selected, you will be notified. A dialog for performing conversion appears. There, among other things, you can convert the database or have it copied beforehand.

Once the database has been converted, it can be used by the current imc STUDIO version.

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! (necessary for 3.x!, if you do not wish to use the simplified menu guidance 40).
- 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 new views are provided. Select the Complete-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 imc STUDIO documentation under *imc STUDIO (general)* > *Views* 80.

3.2.2 Update without using the existing database

In the older imc STUDIO version (3.x), various settings were saved in the so-called Application Settings. These are mainly the view settings: user-defined views, column configurations (e.g. metacolumns).

As of imc STUDIO version 4.x, these settings are saved with the respective project and apply to all the experiments it contains.

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.

Backup settings in imc STUDIO 3.x

In order to save the current view settings, proceed as follows:

- Setup the view as desired
- Save the current view as *<default>*: to do so, open the *View* menu, choose command *Save current view as <default> in application settings*
- Alternatively, you can save the current view under a name of your choice
- Open the *File* menu, choose the *Import / Export* command
- Choose option *Export* and then the list item *Export views, metacolumns, sensors, ...*
- Choose an appropriate file name and path for the backup file

Backup settings (in current version)

In order to save the current view settings, proceed as follows:

- Setup the view as desired
- Save the current view as *<default>*: to do so, open the Ribbon *View*, choose command *Save View As*
- Alternatively, you can save the current view under a name of your choice
- Open the Ribbon *Project*, choose the *Import / Export* command
  - 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

Restore settings

In order to import the View settings to the current project, proceed as follows:

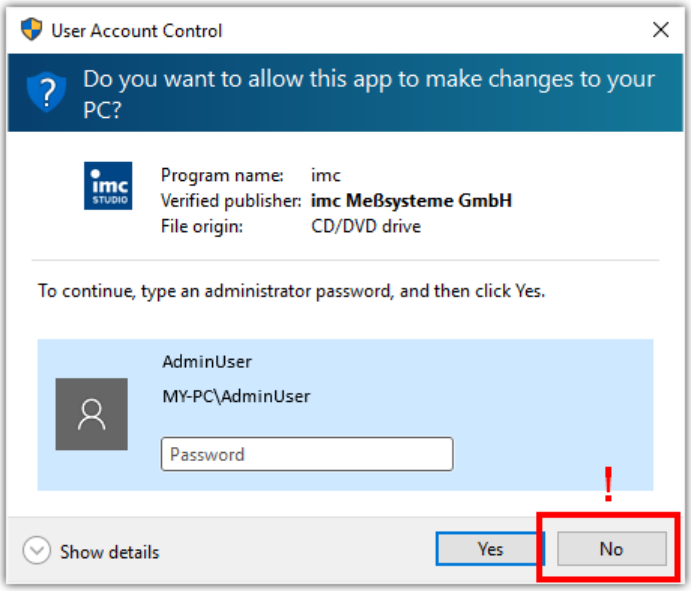
- Open the Ribbon *Project*, choose the *Import / Export* command
  - Choose option *Import* and then the list item *Import views, meta data columns, ...*
  - In the file selection dialog, selected the saved view settings file

Thereafter you will see a dialog allowing you to load either all or individual view settings.

3.2.3 Notes and Troubleshooting

Notes on Windows User Account Control

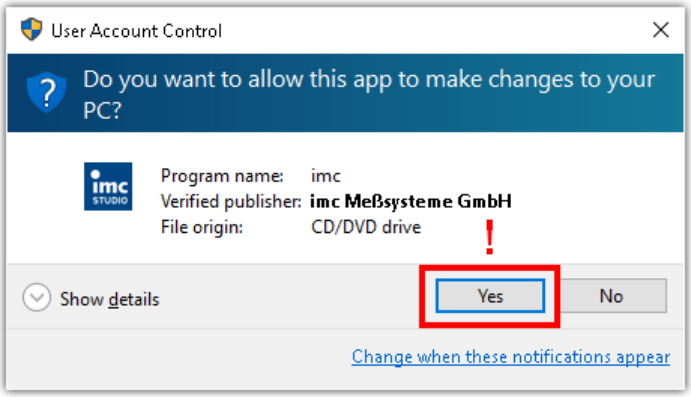
Do not change the user account



Example of a password request.
Select "No".
Do NOT try to continue the installation.

With Windows 10, 7 or Vista, 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 below. 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" as shown below.

Notes on the security software: McAfee

As of McAffe version 8.8 the rules listed below, cause installation problems.

Anti-spyware Maximum Protection

Prevent installation of new CLSIDs, APPIDs and TYPELIBs

Prevent all programs from running files from the Temp folder

Prevent execution of scripts from the Temp folder

Anti-virus Maximum Protection

Prevent alteration of all file extension registrations

Common - Maximum Protection

Prevent programs registering to autorun

Prevent programs registering as a service

Prevent creation of new executable files in the Windows folder

Prevent creation of new executable files in the Program Files folder

Please deactivate these rules for the duration of the installation procedure. By default these rules are active when you select "Maximum protection" in McAfee.

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).

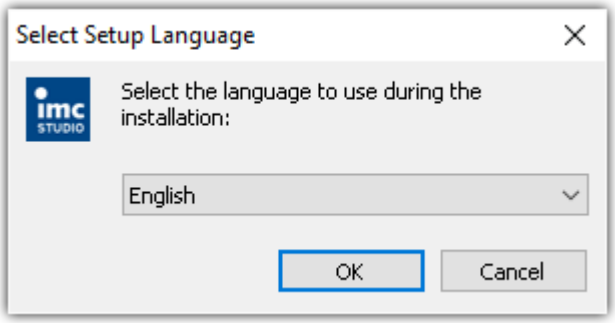
Installation from the product DVD

- Close all programs.
- Insert the product DVD.
- In usual cases, the Setup program will automatically start after a brief delay. Otherwise use the Windows Explorer and start the installation program from the DVD.

Installation after Download

- If you have obtained the product electronically (download, Email), simply start the installation program manually.

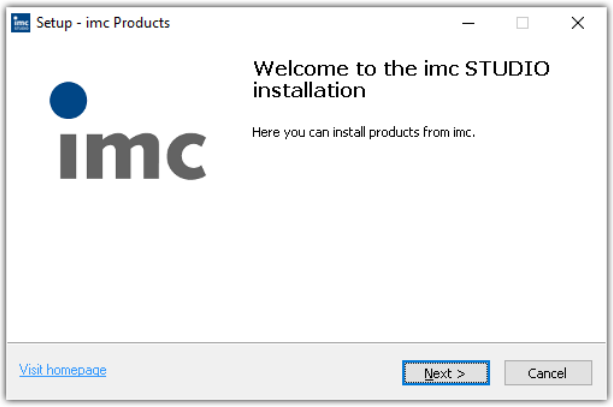
Installation language settings (language after starting the software)



Selecting the language during installation

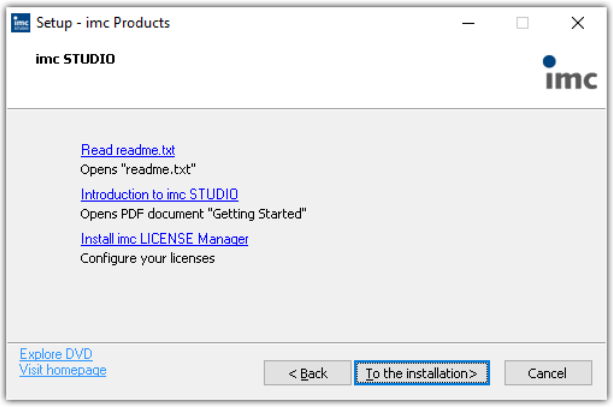
When you start the installation, you first see a dialog for selecting the installation language.

Performing installation



Installation setup welcome page

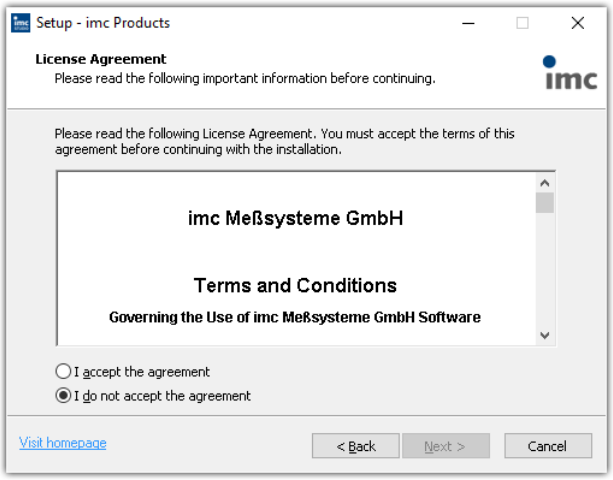
Once you have selected the language, the installation setup starts with a welcome page:



Before starting installation

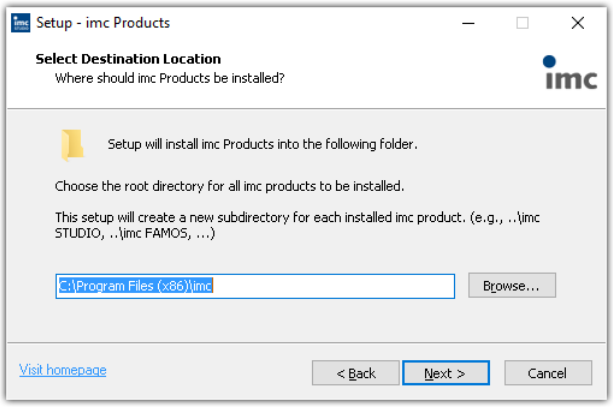
On the installation setup's next page, you are offered the following choices:

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



License agreement

You must accept the license agreement to continue the installation.



Entering the installation path

In the next step, select the file path where the products should be installed. For each imc product, a separate subfolder is created in this path (e.g. imc STUDIO 5.0R5). Therefore it is recommended that the path should end with "..\imc".

Select the products / Installation type

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

- Demo 29
- Auto 29
- User-defined 31

However independent from the selected installation type the products that are always included are:

- imc LICENSE Manager for managing the licensing, and
- imc Shared Components, proving common components of the imc products, such as the curve window.

Details on the types and the respective subsequent installation steps are provided in the associated sections below.

Note on re-configuring imc STUDIO

Regardless of which variant of the installation is selected, when imc STUDIO is installed, all components/plug-ins are included. This way, after successful installation it is possible to adapt the configuration at any time by means of the product configuration 36 and to select the appropriate edition.

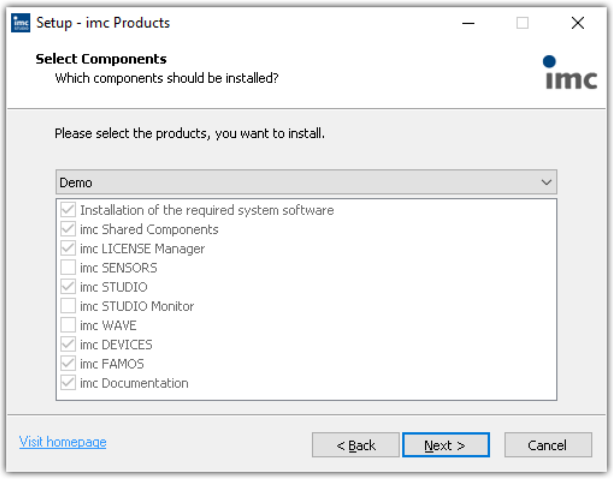
3.3.1 Demo and Auto

The installation types Demo and Auto are only slightly different. For both types the installation settings are already configured. If you need further settings please choose the user-defined installation type 31.

Demo

Who should select the Demo installation

Use the installation type Demo in order to test the full scope of imc STUDIO for 30 days. No other products which require a password will be installed.



Selecting Demo installation

The installation type "Demo" is installed along with imc Shared Components and imc LICENSE Manager

- imc STUDIO Developer (Demo),
- imc DEVICES [1],
- imc FAMOS Reader + Enterprise (Demo) and
- imc Format Converter.

[1] As a package of device drivers and firmware used by imc STUDIO.

The associated Demo license must be activated after concluding installation by means of imc LICENSE Manager, otherwise it is not possible to start imc STUDIO. Once the demo trial period has elapsed, use the product configuration 36 to covert the edition to the license you have purchased.

The imc STUDIO edition and the necessary components/plug-ins can be adapted at any time subsequent to successful installation by means of the product configuration 36.

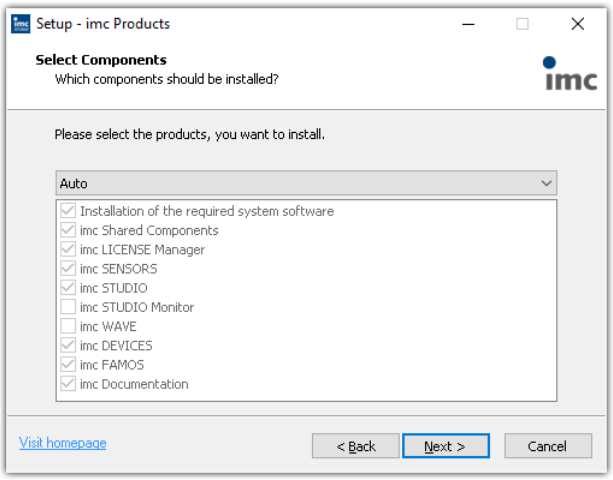
The imc FAMOS Reader is freeware which also needs to be registered using imc LICENSE Manager.

Installation start follows directly (see: "Starting the installation 35").

Auto

Who should chose the Auto installation

Use the installation type Auto in order to install the standard imc STUDIO edition including all necessary components. Optionally, the password-protected sensor management system imc SENSORS is installed.



Selecting Auto installation

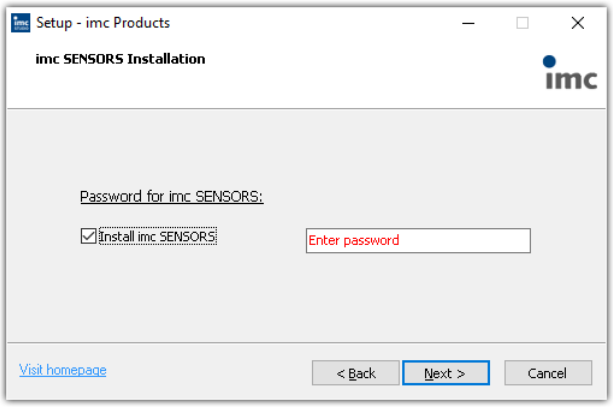
The installation type "Auto" is installed along with imc Shared Components and imc LICENSE Manager

- imc STUDIO Standard,
- imc DEVICES [1],
- imc FAMOS Reader
- optionally imc SENSORS and
- imc Format Converter.

[1] As a package of device drivers and firmware used by imc STUDIO.

The edition imc STUDIO Standard requires the purchase for license. By contrast, imc FAMOS Reader is freeware. Both products must be activated by means of imc LICENSE Manager, otherwise the programs can not be used.

The imc STUDIO edition and the necessary components/plug-ins can be adapted at any time subsequent to successful installation by means of the product configuration 36.



Password prompt for imc SENSORS

Optionally, imc SENSORS is installed, which is a tool for administering sensors. This product requires a password for installation. If you do not wish to install imc SENSORS, you can de-select it.

This additional step follows immediately upon selection of this installation type. After entering the correct password, or selection of the imc SENSORS installation, the installation starts (see: "Starting the installation 35").

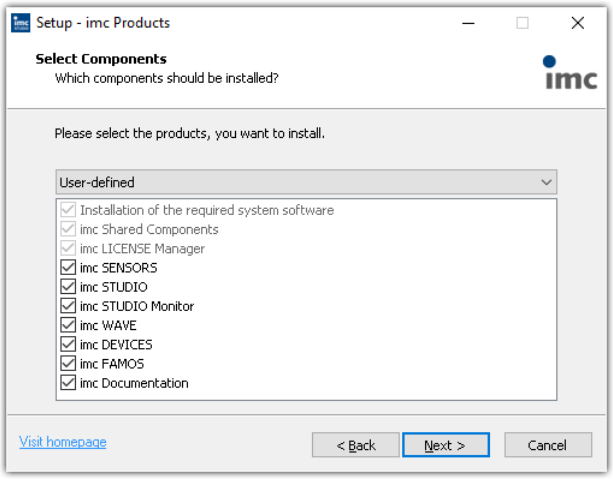
3.3.2 User-defined

This installation type enables the user to choose a detailed configuration. Based on your selection of components desired, different installation steps are displayed.

Who should select User-defined installation

Use the installation type User-defined if you wish to configure the products individual components. With this type, it is possible to install password-protected components. Further, you can install and configure additional imc products such as imc FAMOS (to different degrees). Note that some components can require a separate license. For imc STUDIO, it is additionally possible to install all available language packages.

Selecting components



Selecting User-defined installation

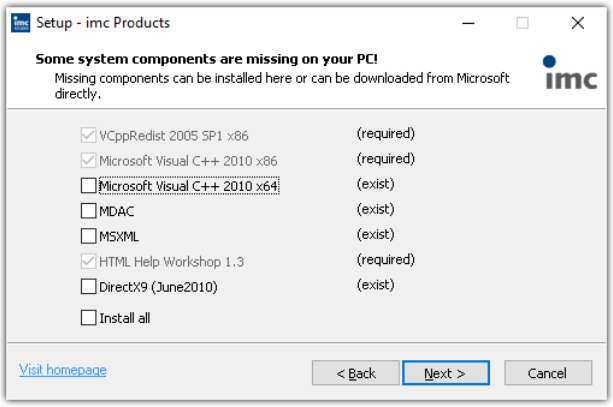
Once you have selected the installation type *User-defined*, it is possible to select/de-select the desired components in the box at the bottom.

imc LICENSE Manager and
imc Shared Components are included, as well as the necessary system component, otherwise the installed products will not function properly.

Selection	Description
Required system components	System components require using of imc products.
imc LICENSE Manager	imc LICENSE Manager enables the display and activation of all imc software products requiring a license.
imc Shared Components	Shared components of the imc products, such as the curve window.
imc SENSORS	imc SENSORS is a turnkey database program applicable across different measurement devices, for administering and editing freely definable sensor information.
imc STUDIO	imc STUDIO is modular software platform handling all aspects of modern measurement technology from simple data acquisition to sophisticated test configurations.
imc STUDIO Monitor	License-required component for imc STUDIO
imc WAVE	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 DEVICES	Package of device drivers and firmware used by imc STUDIO and imc WAVE. Additionally, imc DEVICES alternatively offers a limited operation software which allows the setting of all measurement parameters, real-time functions, trigger machines and data saving modes. imc DEVICES must be installed in order to use imc STUDIO or imc WAVE.

Selection	Description
imc FAMOS	imc FAMOS is a program for analyzing, evaluating and documenting measurement results.
imc Format Converter	Converts imc measurement data to other formats such as EXCEL and ASCII.

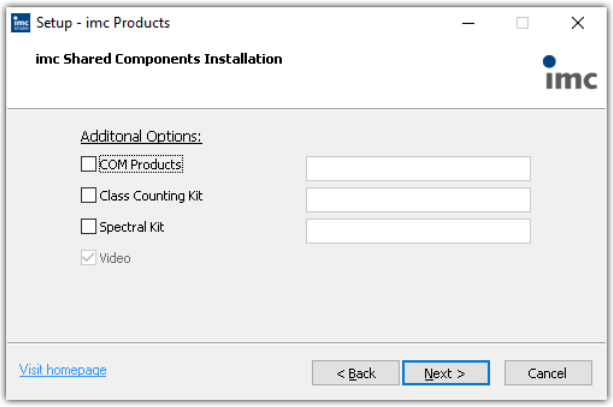
Required system components



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 below are intended as an example only.

Configuring imc Shared Components

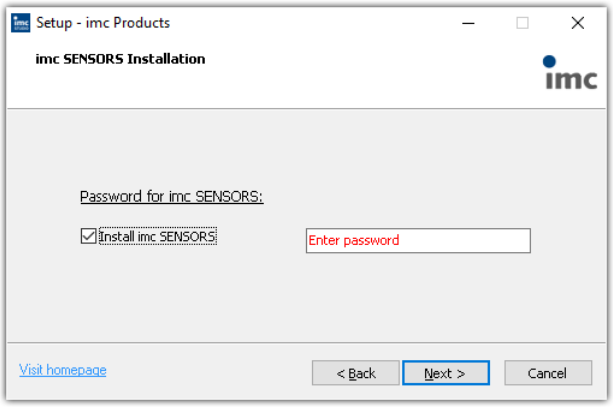


Configuring imc Shared Components

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
Video	

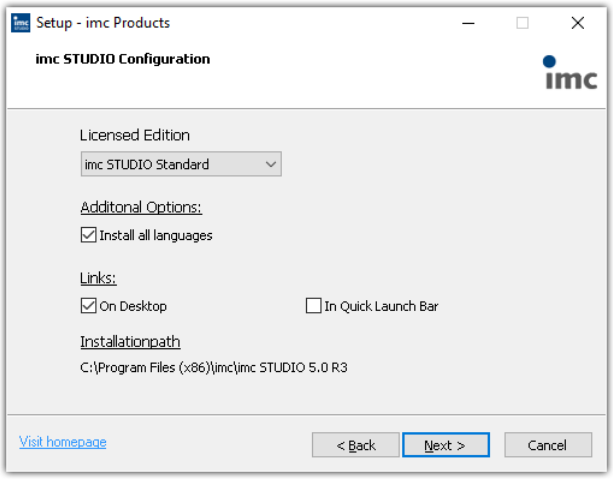
Password for imc SENSORS



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

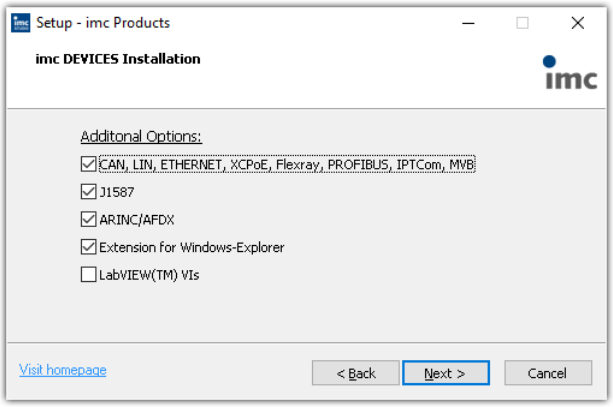


Configuring imc STUDIO

It is possible to make settings for your licensed product edition here. In consequence, later product configuration 36 can be omitted.

Option/Component	Description
Licensed edition	Information on the editions is provided in the Technical datasheet for imc STUDIO.
Install all languages	This option installs imc STUDIO in all available languages. If you do not need any extra languages, you can shorten the installation process. Please note that some functions require other languages. E.g. foreign-language parameter sets can only be imported if the corresponding language is installed. If the option is de-selected, then English and the operating system's language (if available) are installed automatically. If desired, all other supported languages can be installed later (see section Installing additional languages 45).
Links	Program starting links can be created either on the Desktop or in the Quicklaunch bar.

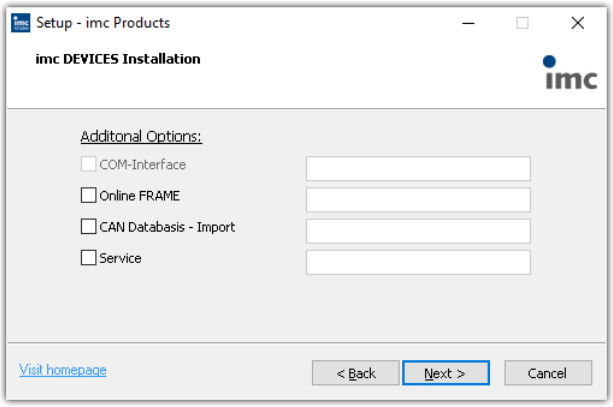
Configuring imc DEVICES



Configuration imc DEVICES

Here, you can install optional components of the imc DEVICES device driver package, which are used by imc STUDIO and imc WAVE.

Option/Component	Description
CAN, LIN, ETHERNET, XCPoE, Flexray, PROFIBUS, IPTCom, MVB	For measurement devices with one of these Fieldbus terminals
J1587	For measurement devices with a J1587-Bus terminal
ARINC/AFDX	For measurement devices with a ARINC- or AFDX-Bus terminal
Extension for Windows-Explorer	Access to the device hard drive via the MS Windows-Explorer
LabVIEW(TM) VIs	Library for accessing the imc devices via LabVIEW(TM)

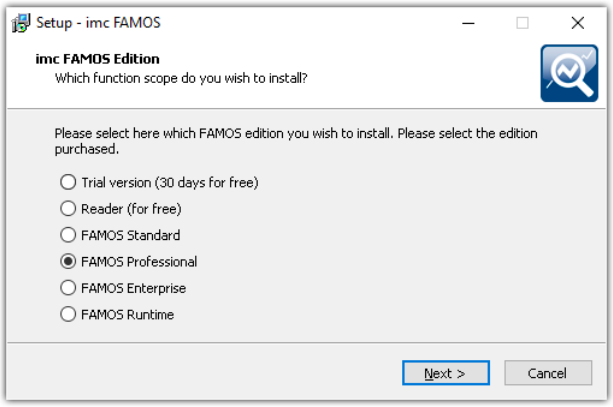


Configuration imc DEVICES (password-protected)

In this step, you can install the password-protected components of the device driver package.

Option/Component	Description
COM-Interface	Installation of the ActiveX components
imc Online FRAME	Installation of the imc Online FRAME Workbench with Compiler
CAN Database - Import	Vector database linkage, DBC-Format (see Device properties 116)
Service	Service support. Install this component only if requested by Customer Support.

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.

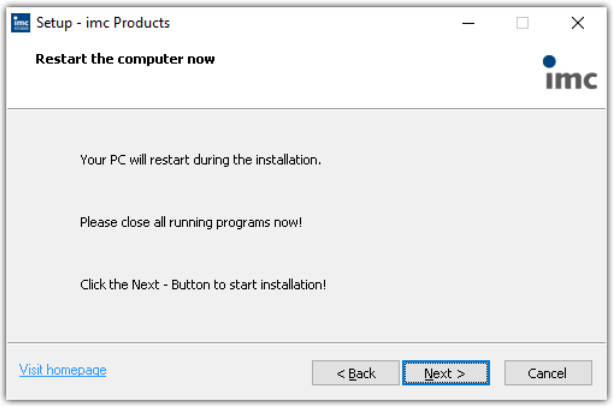
Follow the Assistant's instructions and select your preferred language for the help files and example files like projects, sequences and dialogs. Select any optional components which you wish to install along with imc FAMOS. Additionally, a folder for sample files is needed.

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.

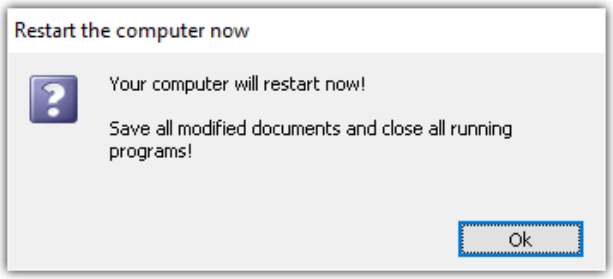
3.3.3 Starting the Installation



Concluding the installation setup and starting installation

You are notified that restart of the system is necessary for successful installation. For this reason you should close any running programs.

Next, the necessary system components are installed.



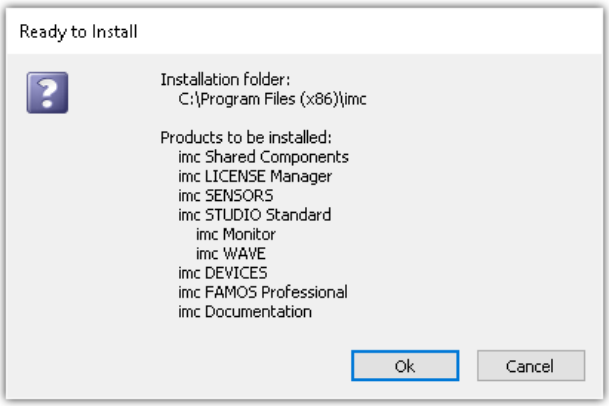
Restarting the system

Once this part of the installation process has been concluded, the system must be restarted, so the following message is posted.

As soon as you click on "OK", the restart is performed.

Warning

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.



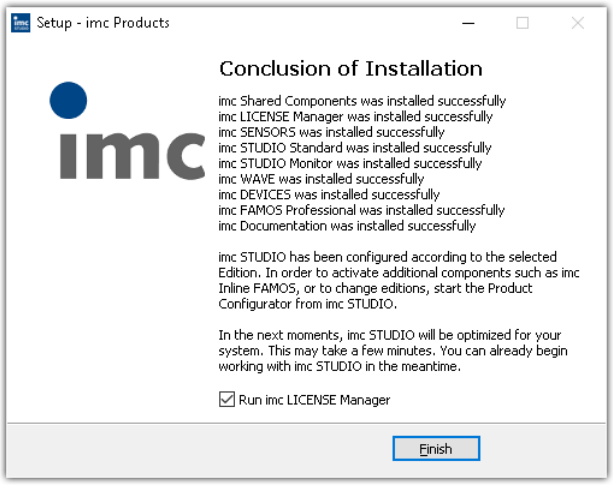
List of components to install, following restart (example)

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

Next, installation of the individual imc products begins.

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 www.imc-berlin.com. For a list of all possible editions and components (plug-ins), see the *Technical datasheet* 1678 or contact our Hotline 13.

Editions and components

The following editions are available for imc STUDIO, and each include a certain basic package of components (plug-ins)/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

Additional optional or individually licensable components (plug-ins) 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 freeware version. You can register this edition in the imc LICENSE Manager. There, select the option "Trial versions and freeware".

Licensing (imc LICENSE Manager)

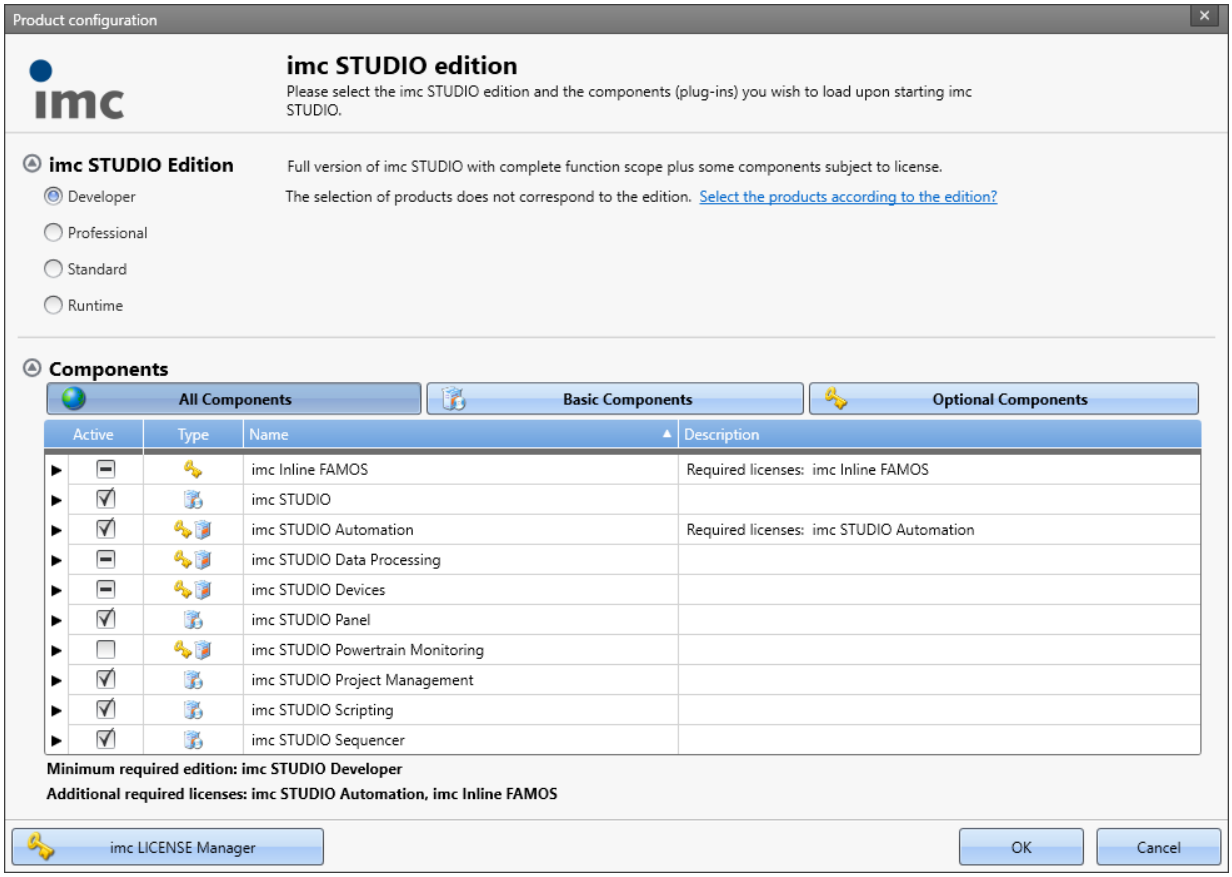
The software licenses are managed with imc LICENSE Manager. After concluding installation, you can start imc LICENSE Manager directly in order to configure your license (e.g. by means of the corresponding entry in the Windows Start menu: for Windows 7: *All Programs* > *imc* > *imc LICENSE Manager*). 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 which describes imc LICENSE Manager in detail. 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 (upgrade/downgrade) 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

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

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.

Changing the components

Additional components (plug-ins) for the edition selected are available to choose. Some of these plug-ins 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 Spec Sheet.

Activate/deactivate the individual components in the column "Active". Below the list, the information on which additional licenses are required is presented.

Note: imc STUDIO Automation

The component Automation has its own independent license and for this reason always appears in the list when selected. If you possess an imc STUDIO Developer license, you also have a Automation-license.

FAQ

Why are there sometimes two components for the same product? E.g. "Sequencer Editor" and "Sequencer 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 Sequencer Engine, you can run the Sequencer in the Standard edition. However, the Editor (Sequencer Editor) is only available in the Professional 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.

Whenever programs reach the boundaries of their capabilities, unforeseen effects can occur, up to and including a system crash.

If this is a concern to you, you can deactivate any components which are not needed.

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

3.5 Start

Starting imc STUDIO

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 and - depending on your choice during installation 26 - a symbol in the quick launch toolbar.

If no icon was set up (installation option), then open the Windows Start menu. There you find under "Programs" the folder "imc" and in it a link to start the product.

If product configuration / licensing 36 is missing, the Product configuration starts automatic.

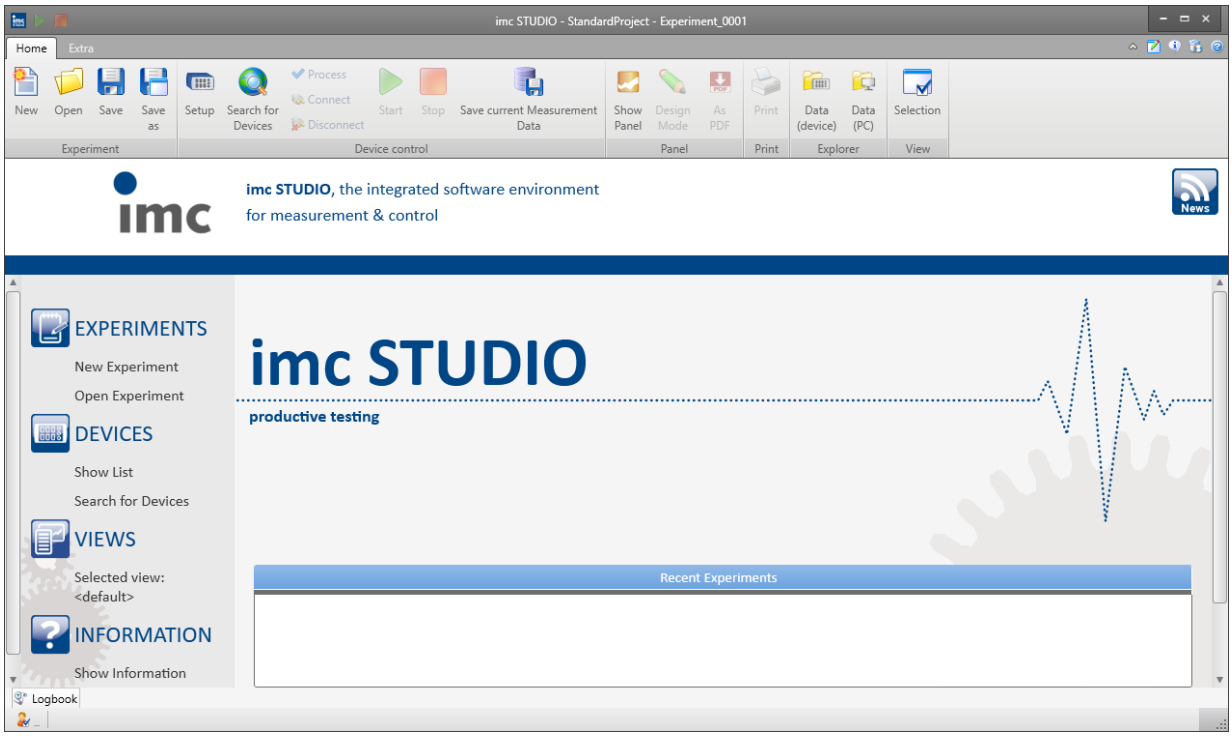
Splash screen

Loading the software components after start

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.

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:



Welcome 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.

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. 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 42.

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.

In order that this view also be used the next time imc STUDIO is started, save the view as <default>:

- Ribbon *View* > *Save as*. Select from the list: *<default>*.

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 *Setup - Connect the device (configuring the network)* 295.

Skipping the network settings

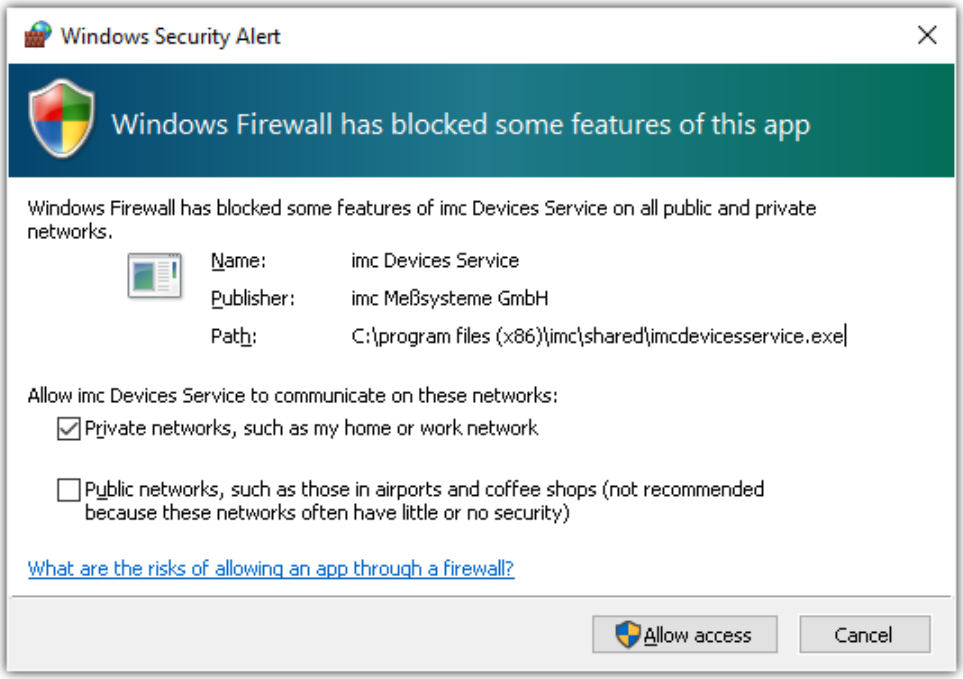
If you have used the predecessor device control software imc DEVICES in the past, you can skip these steps, as the network configuration of the PC and the devices need not be changed for imc STUDIO.

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:

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



Example of the firewall warning under Windows 10

Windows XP: For undisturbed operation of imc STUDIO, you must select "Unblock".

Windows 10, 8, 7 and Vista: Here, there is a distinction between private and public networks. 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.

3.5.2 Plug-ins / Main Windows

imc STUDIO provides the framework for *plug-ins*. Plug-ins appear within imc STUDIO as *main windows*. A plug-in 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 67: Select this link to go to further information on operation of the Navigation Pane.

Notes on the use of the simplified view

- When you use the simplified view 40, 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 68, or change the view 80.

Tool window

Most plug-ins 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.

Tool window 67: 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 73.

Options

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

- Select the item *Options* 58 in the *Extra* menu
- To change the folder path, select *Project Management* > *HDD settings*
- Here, you can edit the path int he database.

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.

3.5.4 Command Line Parameters

Opening an Experiment file

An experiment file 83 (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:\ProgramData\imc\imcStudio\Applications\_2\imcStudio.exe" "C:\Users\Me\Documents\imcStudio\MyExperiment.imcstudio"` The experiment file `MyExperiment.imcstudio` is started in the imc STUDIO installation under `Applications\_2\`.	

"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, if this component was installed along with it.

One possibility is to specify the path in absolute terms:

```
"C:\Users\Public\Documents\DB\StandardProject\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`.

```
"C:\Users\Me\Documents\imcStudio\MyExperiment.imcstudio" /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]`.

```
"C:\Users\Me\Documents\imcStudio\MyExperiment.imcstudio" /do[StartMeasurement]
```

or

```
"C:
\Users\Me\Documents\imcStudio\MyExperiment.imcstudio" /fullscreen /do[StartMeasur
ement]
```

/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]`.

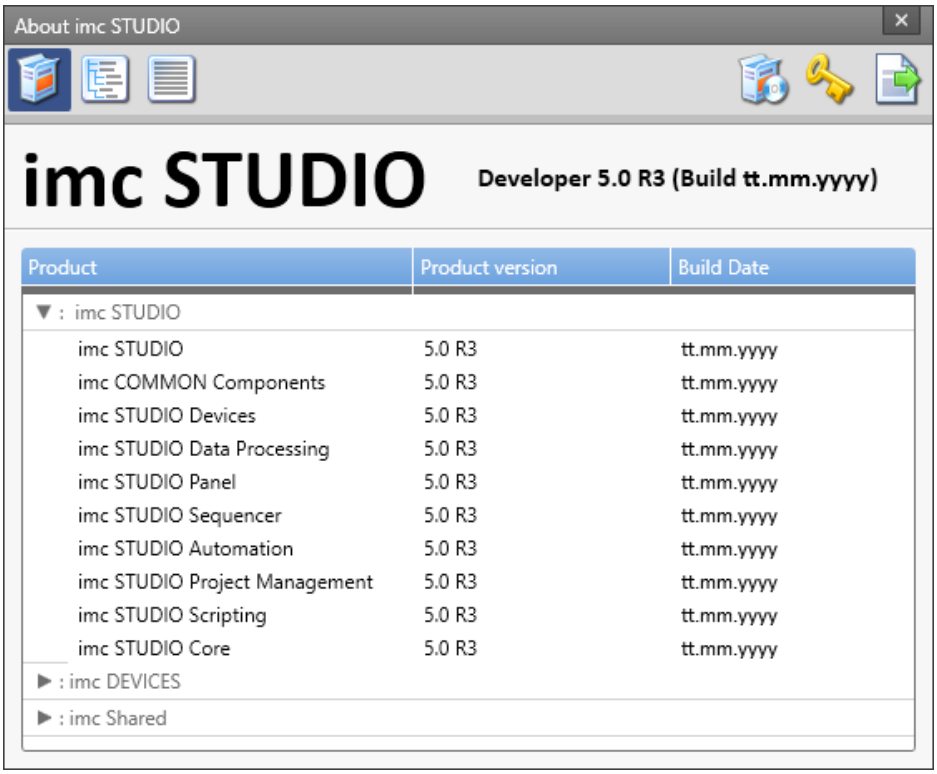
```
"C:\Users\Me\Documents\imcStudio\MyExperiment.imcstudio" /do[StartSequencer]
```

or

```
"C:
\Users\Me\Documents\imcStudio\MyExperiment.imcstudio" /fullscreen /do[StartSequen
cer]
```

3.6 Info / Version Information

imc STUDIO consist of multiple components (plug-ins). 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, you can change the list's depth of detail and how it is sorted.

With the buttons on the right side, you can start the assistant for the product configuration 36, start the imc LICENSE Manager 36 or export the product selection.

While performing the export, you can save the contents as a file in XML format. For inquiries for Hotline, it may be necessary to send it as an Email (see also chapter Customer Support / Hotline 13).

3.7 Information and Tips

3.7.1 Installing Additional Languages

If in the installation type User-defined 31 the option *Install all languages* was not selected, then upon conclusion of the installation, imc STUDIO is available in English plus, if provided, in the operating system's language. In all other cases all supported languages are already installed.

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

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:\Programme(x86)\imc\imc STUDIO 4.0 R1), 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.

Instruction notes (not applicable to Windows XP)

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)

3.7.2 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.

- Add the program imc.Studio.exe to the list of low-risk programs
  The default path is: C:\Program Files (x86)\imc\imc STUDIO...
- Add the program imcDevices.exe to the list of low-risk programs
  The default path is: C:\Program Files (x86)\imc\imc DEVICES...
- Add the database path along with all subfolders to the list of low-risk folders.
  The default path is: C:\Users\Public\Documents\DB
- Ensure that its data are not searched for viruses during reading or writing

3.7.3 Update and Compatibility

Update if imc FAMOS Version 6.2 or older is installed

On this DVD the current version of imc FAMOS is available. This version is installed automatically if the type "Demo" or "Auto" for installation is chosen. Before the installation starts, check if your license key is valid for this imc FAMOS version. If not, choose the installation type "user-defined" and deselect imc FAMOS.

Update from imc STUDIO V3.0R5 or older

Import of the device descriptions necessary

The administrations of the device descriptions has been revised. In consequence, some devices which were previously known no longer appear in the device list.

Please run a new search for the devices after installation or save the device description files before the de-installation. Following the installation, the saved device description files can be imported again.

To find the device description files, proceed as follows:

The easiest way to reach the new path is:

- Hold down the <WIN> (Windows) key and press "r" (or "Start"/"Run").
- Next, in the input box, enter the following shell command and click on OK.

  For imc STUDIO V3.0R4 07.07.2011 or older

  - shell:common appdata\imc\imc Devices\Device

  For imc STUDIO V3.0R4 19.09.2011 or newer

  - shell:programfiles\imc\imcStudio_3_0_R4\Devices

Save all device description files (*.umi). After installation, import the saved files.

- Menu item "Project" > "Import / Export"
- Under "Import", select the item: "Import device descriptions" and confirm with "OK"
- Select the saved device description files (*.umi) and confirm with "Open"

There is a new handling of variable types to be observed (e.g. Numeric or Text)

If a variable takes the type: Numeric, it is not possible to write text to that variable.

In Version 3.0R4, this constraint did not yet exist. Please check after converting your experiments whether the user-defined variables created are of the correct type. If any imc FAMOS variable have been created, this must also be taken into account.

Automatic creation of user-defined variables has been eliminated (e.g. by running commands...)

The handling with variables has been revised. Variables are no longer created automatically in the process of setting them. Instead, variables need to exist already when they are used. For instance, the command Import Parameter Set is no longer able to generate variables, but can only import already existing variables.

To create variables, please use the Data Browser (button: "Create Variable" or "Load Variable") or the command: "Load Variable".

When updating the experiments, you previously (including up until imc STUDIO 3.0R4) needed to perform parameter set export as .dat, which saves all user-defined variables. With imc STUDIO 4.0R1, you load the experiment and next load the missing variables with "Load Variable".

Next, select all loaded variables, open the context menu and select: "Edit selected variable". Set the property "Validity range" to an appropriate range, e.g. "Experiment".

The variables will then be saved in the experiment and do not need to be created again. Then, the experiment works in the familiar way.

Parameterset commands have been revised (for example, it is no longer possible to create a .dat file)

The handling of Parameterset commands has been revised. For the purpose of exporting and importing, there are now "Variables"-commands. The Parameterset commands are responsible for setting properties.

In consequence, it is only possible to create and import .csv files by means of the parameter set commands. When converting an experiment, a message in the logbook provides notification of this fact. Please use the appropriate commands:

- for variables: The Variables commands
- for settings (e.g. channel settings): The Parameterset commands

Import of old application settings

Saved views from older versions can be imported. Note that the menu and the window arrangements have been revised. In consequence, these settings are not imported. A notification in the logbook makes not of this.

Update from imc STUDIO V4.0R1

Product configuration settings are not adopted

If an already existing imc STUDIO V4.0R1 is overwritten by this version, the new version does not adopt the product configuration of the version before. After the installation is finished successfully, the product configuration has to be rechecked.

Simplified views

imc STUDIO starts with a new, simplified view offering all important functions for measurement and visualization. This view does not show neither Sequencer nor Automation. For experiments, that use these plug-ins, change the view to "Complete" or open the Navigation pane via Extras > View > Toll Windows. In the new default view, also the available actions in the ribbon are reduced. If the complete menu is required, switch also to the view "Complete".

Sequencer

During the Sequencer is running, canceling dialogs no longer stop the Sequencer. If that behavior is still desired, the result of the respective dialog need to be evaluated. If the dialog response is "Cancel", the command "Stop Sequencer" must be inserted. The execution of commands connected to any events or Widgets is still stopped when a dialog is canceled.

Scripts

Scripts in experiments, that are created with version 4.0, shall be compiled after conversion. Detaillied notes on this topic can be found in section "Important notes" in the "Scripting" help.

Update from imc STUDIO V5.0R1

SQL Placeholder

Such SQL placeholders used as: SETUP.SQL, EXPERIMENT.SQL, MEASUREMENT.SQL may need to be converted manually. The associated information is always outputted when an old experiment is loaded.

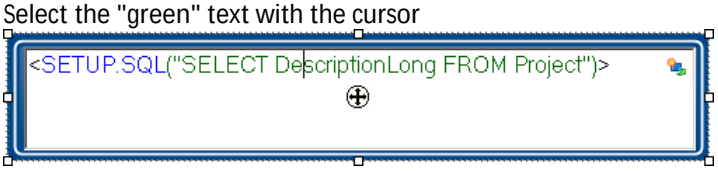
Background:

Column identifier with spaces in the name could not be resolved. In order to make this possible, the syntax needed to be revised. Column identifier are now always expected in apostrophes (' '). Old, saved placeholders (without the appropriate brackets) may in some cases fail to be resolved.

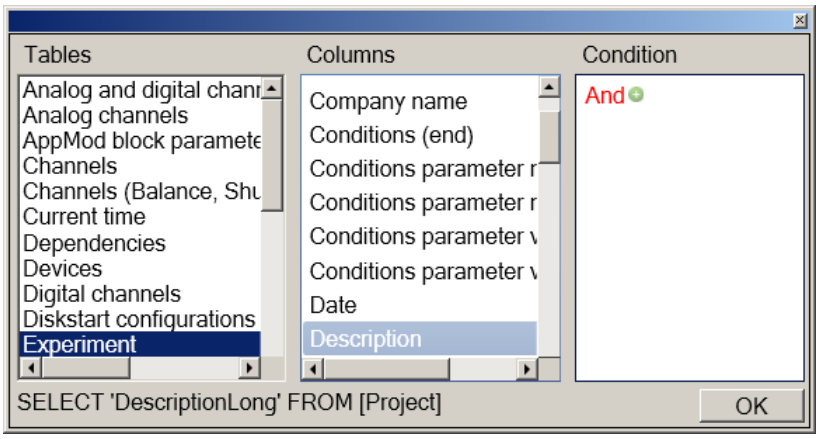
Correction:

If an SQL-placeholder is not resolved correctly, navigate to the position where the placeholder is used.

- Select the "green" text with the cursor



- Press the keys: <CTRL> + <Space>
- In the Assistant which opens consequently, the corrected syntax already appears. Simply click on the button: OK



The automatically corrected syntax is used:

imc Application Module - ExternalEditor

"If a dedicated "ExternalEditor" was written for an application, it needs to be revised upon updating to imc STUDIO 5.0R3.

In the class inherited from API_ReturnValue_V1, the following lines of code must be appended:

```
//new in the imc STUDIO 5.0R3 Version
public API_Vx GetAPI<API_Vx>() where API_Vx : class, IReturnValueBase
{
    return this as API_Vx;
}
```

In total, this would appear as follows:

```
class ChangedAppModZip : API_ReturnValue_V1
{
    public imc.Common.Interfaces.Logbook.API_LogbookEntry_V1[] Error
    {
        get { return new imc.Common.Interfaces.Logbook.API_LogbookEntry_V1[]{}; }
    }

    public bool HasChanges
    {
        get { return true; }
    }

    public bool HasErrors
    {
        get { return false; }
    }

    //new in the imc STUDIO 5.0R3 Version
    public API_Vx GetAPI<API_Vx>() where API_Vx : class, IReturnValueBase
    {
        return this as API_Vx;
    }
}
```

4 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).

Start here

The imc STUDIO framework includes common functions:

- Projects, Experiment templates and Experiments 83
- Data management 84
- Open / Save experiments 51
- Saving / Loading views 80
- User administration and Access rights 73
- Navigation Pane 67, Logbook 69

Customer Support - Hotline

Questions or problems? Contact our Customer Support / Hotline 13.

4.1 Ribbon

4.1.1 Project Menu

The Project menu is available in every plug-in.

Some of the dialogs offer an extended view 58. By default, the dialogs are displayed without project view and experiment templates.

Project

	Menu item	Description
	Manage Projects 53	Managing projects and experiment templates
	Save Project	Saves the current project

Experiment

	Menu item	Description
	New 55	Creates a new experiment
	Open	Opens an existing experiment
	Save	Saves the current experiment
	Save as 55	Saves the current experiment under a new name

Im-/Export

	Menu item	Description
	Import / Export 55	

Measurement Data

	Menu item	Description
	Save current Measurement Data Save current Measurement Data as	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 - Advanced Device Functions: *Storage Options and Directory Structure > Controlling Data Storage > Targeted Data Saving, or Saving Subsequent to Measurement* 610

Note: Without imc STUDIO Project Management

In the product configuration, the component imc STUDIO Project Management can be deactivated. This is always the case in imc STUDIO Monitor, for example.

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 Spec 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*".
	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

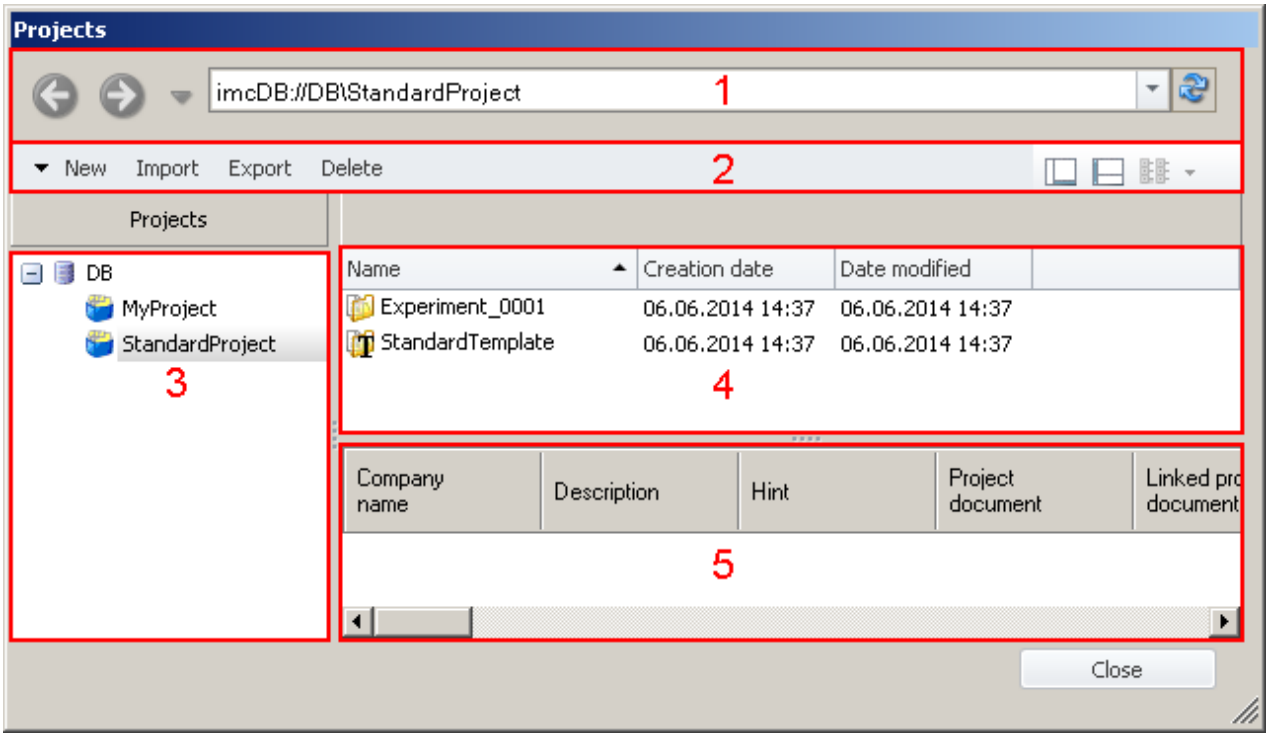
4.1.1.1 Dialogs: Project and Experiment

The Project Management dialogs are described below:

- with Project view and
- with Experiment templates.

By default, these options 62 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 five regions (from top to bottom):

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

By default, the regions 1,2 and 4 are displayed; the others can be activated if desired.

Region 1: Address bar

The address bar displays only the names of the database and the project selected. The bar has no additional functions.

Region 2: Menu bar

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 84 (a project must be selected). Experiment templates can be generated from a variety sources: • From present settings • From existing experiment • From default settings
Import	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	Exports the selected entries to a file. Whole projects and individual/multiple experiments and experiment templates can be exported to a file.
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 3 (list of all projects)
/	Show/Hide Region 5 (metadata information)

Regions 3 and 4: List of all projects/experiments and 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 3 is only displayed if the project view is activated and the region is shown by means of the menu button .

Region 4 only shows the experiment templates if these are activated.

Region 5: 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 61 *Metadata > Experiment - Metadata > Setup page*, you can select which source to use for the metadata.

Only shown if the region is shown by means of the menu button .

Dialog: New experiment

See: Create Experiment 55

Dialog: Manage Project

Preferred experiment template

When you create a new experiment, it is generated from an experiment template (see experiment template 84).

When displaying experiment templates 62, 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.

4.1.1.2 Creating/saving 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 62 is activated, you can additionally select the target project.

New Experiment

When you create a new experiment, it is generated from an experiment template (see experiment template 84).

- If experiment templates are displayed 62, select an experiment template.
- If they are not displayed, the preferred experiment template 55 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.

4.1.1.3 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* 1630 *and* *Export parameters* 1626.
Views, metadata columns, (sensors,) ... 82	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.
imc DEVICES experiment 257	Imports the devices configurations of an old imc DEVICES experiment or an imc DEVICES device configuration for a device.

Option	Description
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 105, in which you can export in the experiment existing supplemental files. See the description: *Setup* > *Ribbon* > *Configuration* > *Supplemental Files* 105.
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 255	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 82.

4.1.2 Edit Menu

The content of this menu depends on which plug-in is currently open.

Objects (e.g. on the Panel) are deleted without a confirmation prompt. If you have deleted an object accidentally, use the menu item *Edit* > *Undo*.

4.1.3 View Menu

Views

	Menu item	Description
	Save View 80	Saves current view - overwrites the used view in the project settings.
	Save View as 80	Saves current view in the project settings under a new name freely specified by the user, e.g. "Test Drive 1".
	View selection 80	Loads view from the project settings.
	Delete View 80	Deletes a saved view from the project settings.
	Restore 80	Restores views from the factory settings.

Layout

	Menu item	Description
	Reset Window Arrangement 57	Restores the GUI of the currently open main window.
	Customize Ribbon Menu 57	Customizes the menu ribbon.
	User-defined buttons	Creates user-defined buttons which cause commands to be carried out.
	Choose Tool Windows 68	Opens the dialog: Tool windows chooser in order to show/hide the tool windows.

4.1.3.1 Layout

Reset Window Arrangement

Menu item *View > 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

Menu item *View > Customize Ribbon Menu*

By means of this function the menu ribbon can be customized.

Choose Tool Windows

Menu item *View > Choose Tool Windows*

Opens the dialog: *Tool windows chooser* 68 in order to show/hide the tool windows.

4.1.4 Extra Menu

The content of the menu Extra depends on what plug-ins are included.

Options

	Menu item	Description
	Options 58	General options
	Metadata Assistant 206	Using the Metadata Assistant, you can save metadata from the Setup along with a saved measurement or also save them directly to the channel file.

User Administration

	Menu item	Description
	Login 77	Logs in a user
	Logout 77	Logs a user out
	Administration 74	Sets up a user and assigns them to imc user groups
	User access rights 79	Assigns rights to imc user groups
	Startup behavior 76	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 1189
	Stop (Sequencer)	Stops the Sequencer. For more info see the description of the plug-in Sequencer 1189

4.1.4.1 Options

In this dialog you can make fundamental settings for the individual plug-ins.

FAQ 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.

Command Options

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
Address	Sender address (your email address).
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 email.
Sender name	The name of the sender of the email (your name).
User	User name for logging in.

Data Browser

General options

Options - Automatic loading on demand	Description
Automatic loading on demand	Measurements are loaded automatically if your variables are linked to a Widget on a Panel page.

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. Additionally, 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".

Options - Filter list	Description
Display filter list	Enables filters to be defined which filter according to the channel name and to metadata. The filters defined are listed in the Filter List in the Data Browser and are available for selection there.

Options - Measurement number	Description
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.

Dialog name	Description
Measurement running - reconnect not possible	Stop and Connect
Overwrite autostart configurations	OK
Overwrite existing file	Yes No Cancel
Project has changed	Yes No
Connection failed - transfer configuration	Yes No
Disconnect from running measurement	Disconnect Stop and Disconnect
No new devices found	Yes No
Measurement running	Reconnect Stop and Connect
Devices deselected	Discard Transfer
Experiment has changed	Yes No

Logbook

Logbook

Options - Show logbook at	Description
Category of the message (Errors, Information, Warnings)	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 54).
Setup page	Here you define what Setup page to export when saving the experiment.

Panel

Navigation

Options - Options	Description
Always link new Widgets to Navigation bar	If this option is activated, newly positioned Widgets are automatically linked with the Navigation bar.
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 through all Panel pages	Activates the mode in which the Navigation bar navigates through all Panel pages, including invisible pages.
Post-processing mode as default	Automatically activates the Post-processing mode.
Widgets navigation mode	One can choose which value the Widgets shall show. There exist following possibilities: • Automatic: During the measurement is running or while working on the current measurement, the Widgets show the current value. They show the value at the slider's position, whenever one is working on saved measurement data. • Always show navigation value: The Widgets show always the value at the slider's position. • Always show current value: The Widgets show always the current value.

Options - View	Description
Datacut tools	Shows the datacut tools. These tools make it possible to setup a datacut, mark an interval and process it.
Post-processing tool	Shows the post-processing tool for activating Postprocessing Mode, which enables you to insert tags in the Report channel at navigation time.
Time controls	Shows advanced time controls on the Navigation Bar. The visible range of the linked curve windows is displayed and can be set for each of the two sides separately.

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
Load measurements	If this option is activated, saved measurements in the Data Browser are displayed.
Project view enabled	To create more projects, you must activate the Project view. This option is deactivated by default. (Information on Projects 83). There is an extended view available of some dialogs 53. 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.
Store in database	If this option is activated, the database is relocated into the imc measurement database: imc SEARCH. Only activate this option if you have installed imc SEARCH. Also observe the other options for configuring the connection to imc SEARCH.
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 83) An extended view of certain dialogs 53 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 84) 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.

imc SEARCH

Options - Login settings	Description
Login mode	Mode used for login process.
User name	User name to use for login.
Password	Password to use for login.

Options - Connection data	Description
IP address	IP-address of the computer on which the imc SEARCH service is running. A valid IP-address is needed for a working connection.
TCP Port	The TCP port must be between 1 and 65536. This is required for connection to the imc SEARCH service.
Connection Mode	

Measurement storage area

Options - Measurement storage area	Description
Use a user-defined measurement storage path	By default, this is the Experiment-path. This denotes the folder in which all measurements are saved (according to the measurement folder structure specified). See also Setup - Advanced Device Functions: *Storage Options and Directory Structure > Controlling Data Storage > User-defined Measurement Storage Area* 612.
Use a user-defined measurement folder structure	This refers to the structure below the measurement storage folder. For instance, folders are set up here according to the settings for Path naming (e.g. Continuous numbers). See also Setup - Advanced Device Functions: *Storage Options and Directory Structure > Controlling Data Storage > User-defined Measurement Storage Area* 612.

Setup

General

Options - Common	Description
Data storage state after download	The state of the data storage after downloading the device configurations. This option does not replace activation of storage for each channel. Instead, this option regulates the condition of "Suspend and Resume data storage 95".
Parameter set export - Expanded mode	If set, you can control which columns will be exported upon parameterset export.

Device options

Options - General	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. • When the setting "Manual" is used, it is always asked, which imc DEVICES version should be used, when devices are selected or an experiment is loaded. • "Automatic" prevents firmware updates, prefers the version that is currently running on the device. • "Always use latest" selects always the most recent imc DEVICES version which is compatible wich this imc STUDIO version
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 - 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.

Options - Virtual device clock	Description
For the purposes of exact timekeeping on the PC and for synchronization of device and PC results, a virtual clock is used on the PC. • When there is no external synchronization source, upon downloading the device clocks and the virtual clock are set according to the PC's clock one initial time. The subsequent behavior is determined by the user's choice for the option: *Synchronize always*. • When external synchronization sources are available, upon downloading the device clocks synchronize to the synchronization source. The virtual clock is set according to the PC clock one initial time. The subsequent behavior is determined by the user's choice for the option: *Synchronize always*. The clients on the PC will always use the virtual clock's time. A client may be, for instance, the clock on a Panel page or a video-signal. 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.	
Minimum accuracy [ms]	The option defines the maximum time in milliseconds that the device clock may deviate from the PC's virtual 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 - FIFOs	Description
Automatic RAM size	If true, the RAM required for the FIFOs is automatically calculated by imc STUDIO.
Maximum RAM size (MB)	Must be a value between 500-4000. Only evaluated if the automatic calculation of the RAM size is deactivated.

Options - Curve window	Description
Automatic RAM size	If true, the RAM required for the curve window is automatically calculated by imc STUDIO.
Maximum RAM size (MB)	Must be a value between 0-200. Only evaluated if the automatic RAM calculation is deactivated.

Save current Measurement 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

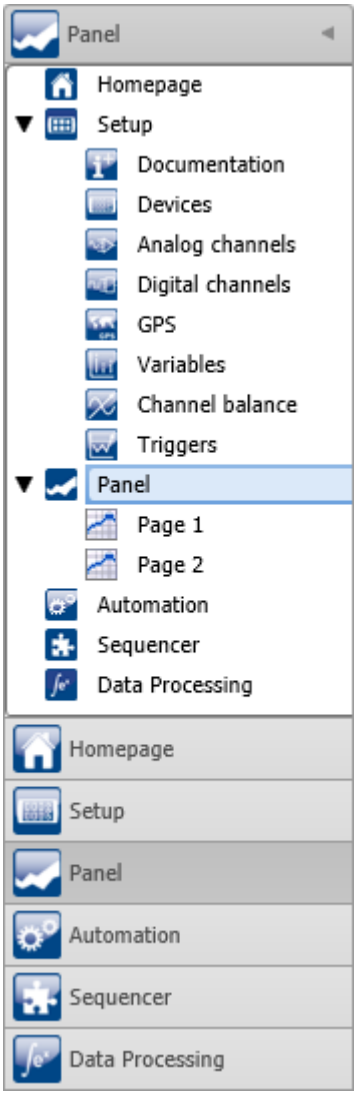
See also Setup - Advanced Device Functions: *Storage Options and Directory Structure* > *Controlling Data Storage* > *Targeted Data Saving, or Saving Subsequent to Measurement* 610.

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.

4.2 Navigation 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. Using the arrow symbols (▶ ▼) which are in front of the main windows, the tree diagram branch can be expanded/collapsed.

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).

To show or hide the Navigation Pane use the dialog: Tool windows chooser 68.

4.3 Tool Windows

Tool windows contain special elements for operating/viewing a main window. (For information on operating the tool window: Operation 68)

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 69

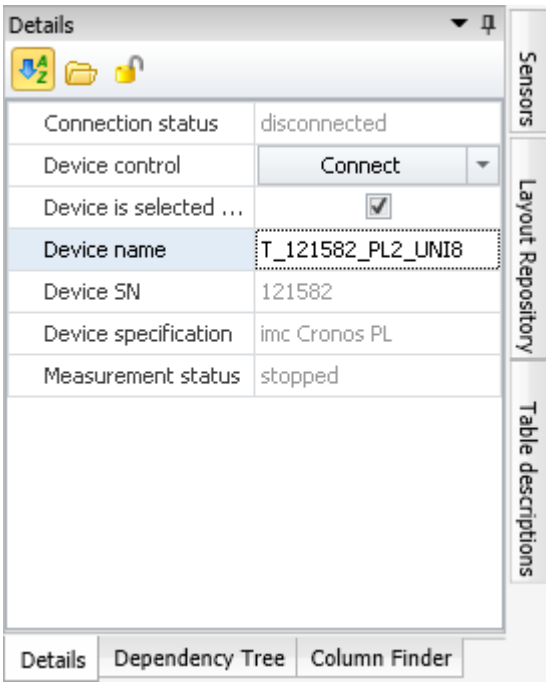
The *Logbook* is always displayed and can be minimized. The other main windows' tool windows (e.g. for the Panel) are displayed as soon as the respective main window is opened.

Not all tool windows are visible

By default, all necessary tool windows are displayed. All others can be displayed 68, if you need them.

4.3.1 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*).

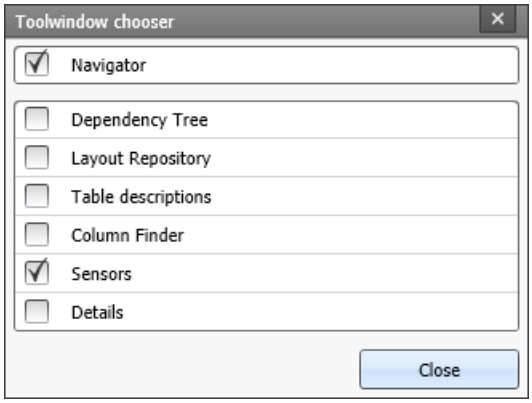
A tool window is opened when the mouse is swiped over it.

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

- use the menu *View* > *Tool windows*, or
- the button (▾) in one of the present tool windows.



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.

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 and drop to the desired position.

Docking the tool window

In order to dock the tool window, use the drag and 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)

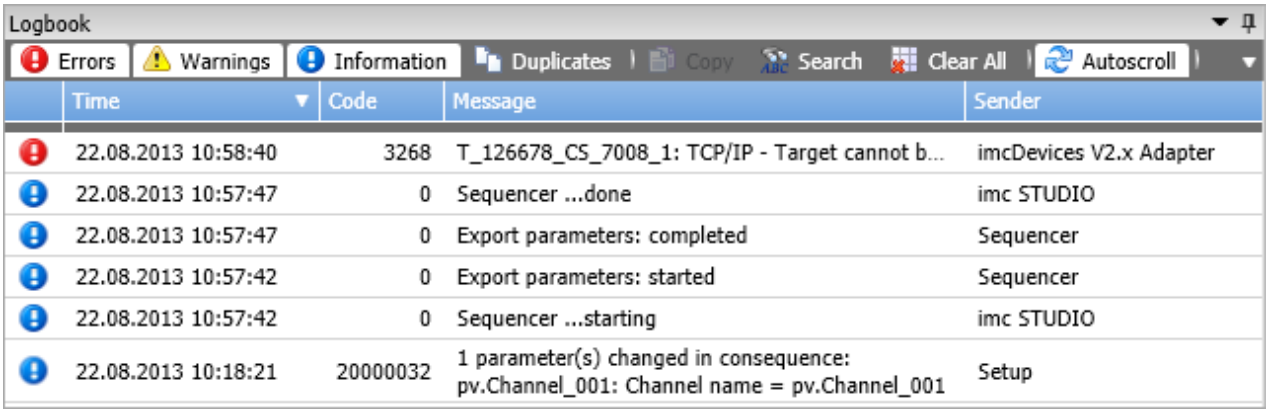
4.3.2 Logbook

In the tool window Logbook, messages of the categories: *Fatal* (), *Error* (), *Warning* () and *Information* () 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.

The logbook is saved

For purposes of tracing history, the logbook is saved daily:

Default path for Windows 7: C:\ProgramData\imc\imc STUDIO\Applications_1\log (example)

Filtering the Logbook

You can filter according to category.

Logbook filter

To do this, click on the Filter-button in the logbook's title bar:

- Errors
- Warnings
- Information

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.

Searching in the logbook

It is possible to search the logbook for entries and text passages.

To do this, click on the *Search*-button in the logbook's 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 *Search* works like a filter and has an effect when a new message appears. For this reason, this message may not be displayed.

Hiding duplicates

It is possible to hide 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. To do this, click on the *Duplicates*-button in the logbook's title bar.

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.

Copying a selected message

It is possible to copy the selected message to the Clipboard.

To do this, click on the *Copy*-button in the logbook's title bar.

The copy contains all information which is visible in the logbook.

Deleting messages

It is possible to delete displayed messages.

To do this, click on the *Clear All*-button in the logbook's title bar.

Note: *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.

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.

Sending Email

You can send the logbook via Email (e.g. to the imc Hotline).

When you click on the *Send Email* -button, your Email-program starts. The logbook is zipped and attached to the Email. The imc Hotline is automatically added as a recipient.

Starting the Logbook-Viewer and opening old logbooks

By means of the Logbook-Viewer, you can look at current, deleted and saved logbook entries.

When you click on the button, the Logbook-Viewer opens. In addition to the familiar logbook-functions, it is possible to load saved logbooks.

Options for the Logbook

From within the logbook, you can open the Options for the Logbook 60.

4.4 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

- The user administration is available as of the product edition imc STUDIO PRO.
- By default, the User Administration is deactivated. In that state, each user has the rights belonging to the group *imc Administrators*.
- The User Administration can be activated in the User Administration 74.

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 experiments (measurements) and check and save the results. However, they would not be able to change the experiments.

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)

The association of users to imc user groups is set in the User Administration 74.

Upon installation, default access rights are assigned. As a *imc Administrators*, you can make very detailed configurations (see the section Setting access rights 79).

When the program starts, it determines to which user group the user logged into the PC belongs (see the section Startup behavior 76).

You can switch users after the program start (see the section Logging on, logging off, and switching users 77).

Note

The user group has extensive influence on the visibility and/or operability of menus, icons, etc..

The rights of the individual user always depend on the which of the groups the user belongs to. There are no special rights for an individual user.

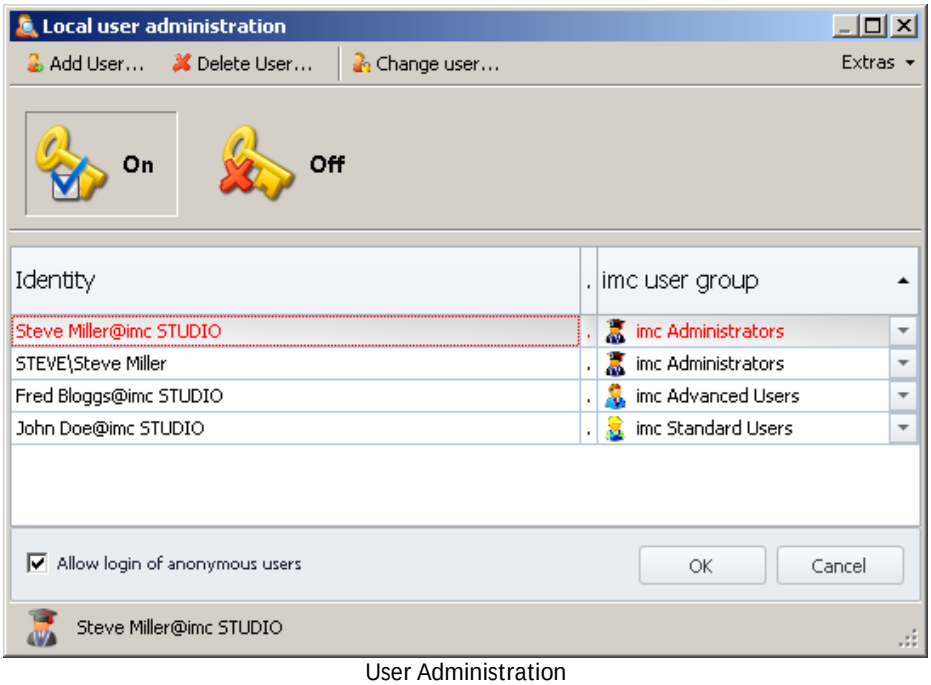
The user administration is saved with the project

- The User Administration settings are saved with the respective project.
- Changes take immediate effect, but are only saved when the project is saved.

4.4.1 User Administration

In this dialog you can activate/deactivate the user administration. You can set up users and assign them to the imc user groups. Thus, each user obtains rights corresponding to their respective group.

Open the dialog by selecting the menu item *Extra > Administration*.



User Administration
Assignment to imc user groups

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:

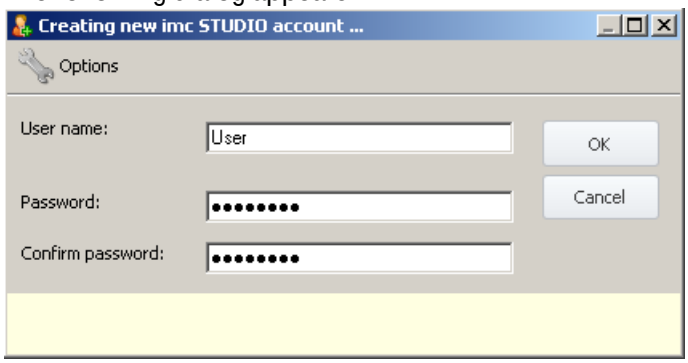
imc STUDIO internal users	Users who can be assigned independently of the particular PC or domain. The user can be protected with a password.
Local computer accounts or groups	Windows or Domain account or group. The group memberships set under Windows or in the domain are not used.
Active Directory domain accounts or groups	The user is protected by the password used for login to Windows 77.

Adding users

Adding a user of the type imc STUDIO internal user:

- click on the button *Add user*
- select the user type *Create an imc STUDIO internal 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*.

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 imc 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 imc 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 76: *Windows Session*).

Prohibiting logging in as an Anonymous User

To prohibit Anonymous Users from logging in,

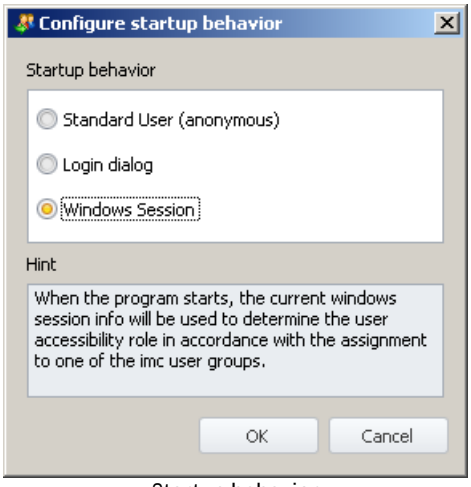
- delete the checkmark next to *Allow login of anonymous users*.

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

4.4.2 Startup Behavior



Startup behavior

In this dialog, the startup behavior is set. Open the dialog by selecting the menu item *Extra > Startup behavior*.

The associated dialog opens offering the following options.

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 77 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 imc user groups. That is the default setting after installation. For the assignment between the users and the imc user groups, see the section User administration 74. If the User Administration is activated, any PC-users not belonging to any imc user group 74 are assigned the rights of the group *imc Standard Users*.

4.4.3 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.

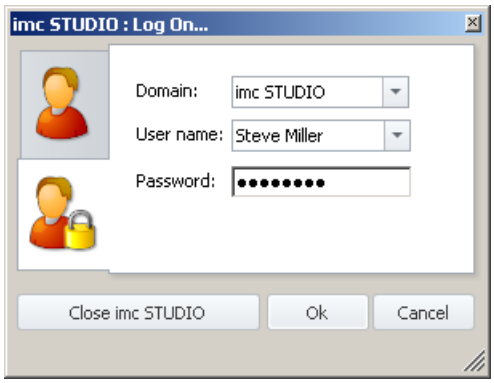
With Startup behaviour: Login Dialog 76 when imc STUDIO starts the Log On window appears. As well, after imc STUDIO has started, you can switch users using the menu item *Extra* > *Login*.

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 74).
  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

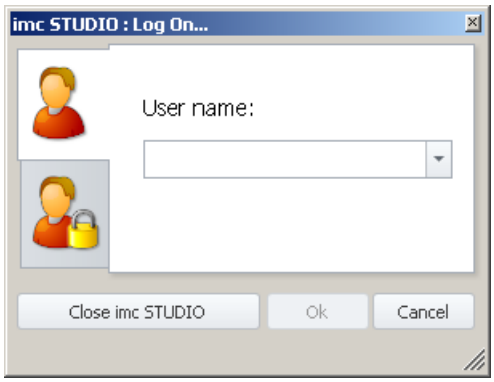
Authentication of the registered user upon logging on references:

- 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

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



Login Dialog
Logging on an anonymous user

Authentication of the anonymous user references the user name.

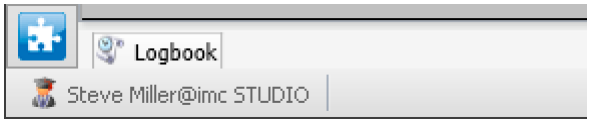
However, any arbitrary name can be used.

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



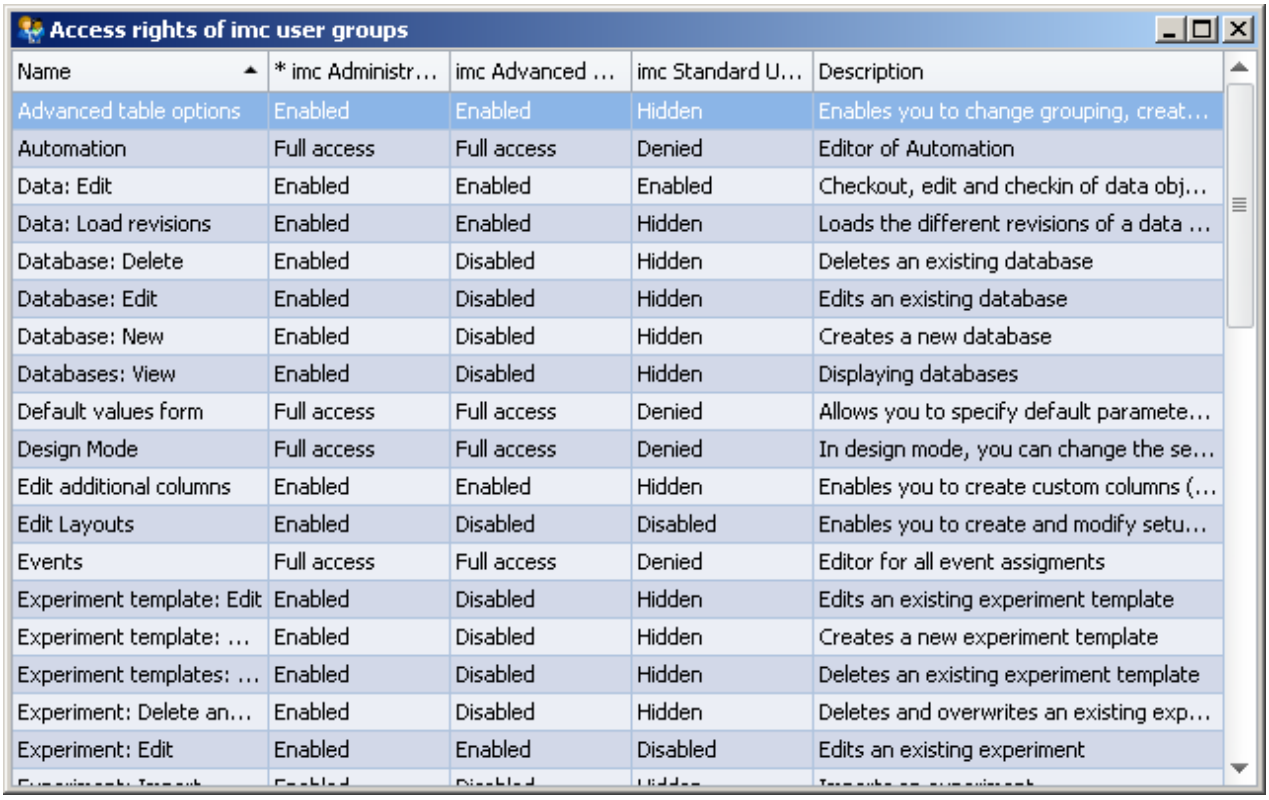
The Status bar at the lower left indicates which user is currently logged on. To the left of the name is the symbol representing the respective imc user group.

	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)

4.4.4 Setting Access Rights

To set the access rights, select the menu item *Extra > User access rights*.

In the dialog *Access rights of imc user group* the rights belonging to the imc user groups are listed and can be edited according to your wishes.



Access rights of imc user groups

Name	* imc Administr...	imc Advanced ...	imc Standard U...	Description
Advanced table options	Enabled	Enabled	Hidden	Enables you to change grouping, creat...
Automation	Full access	Full access	Denied	Editor of Automation
Data: Edit	Enabled	Enabled	Enabled	Checkout, edit and checkin of data obj...
Data: Load revisions	Enabled	Enabled	Hidden	Loads the different revisions of a data ...
Database: Delete	Enabled	Disabled	Hidden	Deletes an existing database
Database: Edit	Enabled	Disabled	Hidden	Edits an existing database
Database: New	Enabled	Disabled	Hidden	Creates a new database
Databases: View	Enabled	Disabled	Hidden	Displaying databases
Default values form	Full access	Full access	Denied	Allows you to specify default paramete...
Design Mode	Full access	Full access	Denied	In design mode, you can change the se...
Edit additional columns	Enabled	Enabled	Hidden	Enables you to create custom columns (...
Edit Layouts	Enabled	Disabled	Disabled	Enables you to create and modify setu...
Events	Full access	Full access	Denied	Editor for all event assigments
Experiment template: Edit	Enabled	Disabled	Hidden	Edits an existing experiment template
Experiment template: ...	Enabled	Disabled	Hidden	Creates a new experiment template
Experiment templates: ...	Enabled	Disabled	Hidden	Deletes an existing experiment template
Experiment: Delete an...	Enabled	Disabled	Hidden	Deletes and overwrites an existing exp...
Experiment: Edit	Enabled	Enabled	Disabled	Edits an existing experiment

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.

4.5 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:

- Window arrangement
  - the imc STUDIO window's position and size
  - the tool window's position and size
- The last main window opened
- Menus
  - layout of the menu ribbon
  - quick access toolbar
- Layout
  - 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

Please note that only the positions of individual elements is saved. The existence and configuration, for instance, of the Setup-columns (tables- and column descriptions) is saved along with the respective project.

When you save a view, the whole project is also saved. This includes the current configuration of the columns.

Reference

See also:

- Ex- and Import of views 82
- Ex- and Import of Setup table descriptions
- Ex- and Import of Setup column descriptions

Saving views

In order to save the current view, select the menu item *View* > *Save* or *Save As*.

All settings belonging to the current view are saved. These include the items named above.

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.
- Along with the project, additional settings are saved along with it, which affect the views:
  - Setup table descriptions and
  - Setup 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)

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.

Deleting views

To delete a saved view,

- select the menu item *View* > *Delete View*.
- 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

The factory settings come with a copy of all views created as part of the program installation.

These views are not write-protected. But they can be restored to their original state.

To restore a view from the factory settings,

- select the menu item *View* > *Restore View*.
- In the dialog which then opens, select the desired views. They are imported from the factory settings.
- Conform your selection by clicking on *OK*.

4.5.1 Exporting/importing views

When you export the view settings, a variety of project settings are stored in the export file.

- saved views 80
  - 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)
- user-defined characteristic curves/sensors (tool window: Sensors)

Export of views, metadata columns, sensors, ...

In order to save the current view settings, proceed as follows:

- Setup the view as desired
- Save the current view as <*default*>: to do so, open the Ribbon *View*, choose command *Save View As*
- Alternatively, you can save the current view under a name of your choice
- Open the Ribbon *Project*, choose the *Import / Export* command
  - 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

Import of views, metadata columns, ...

In order to import the View settings to the current project, proceed as follows:

- Open the Ribbon *Project*, choose the *Import / Export* command
  - 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 load either all or individual view settings.

Reference

See also:

- Export/import of Setup table descriptions
- Export/import of Setup column descriptions
- Import of sensors 56

4.6 Information and Tips

4.6.1 Projects, Experiment templates and Experiments

This chapter describes the relationships between projects, experiment templates and experiments. How the individual components are saved is discussed in the chapter Data Management 84.

Experiment

imc STUDIO saves the settings in so-called experiments. These comprise, among other things:

- the experiment file (filename extension: *.imcStudio),
- measurement files
- metadata
- various backup files
- 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 55.

Project

A project contains:

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

You can set up multiple projects containing various experiments.

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

Project view

To create additional projects, you must activate the Project view. This option is deactivated by default.

- Select from the menu *Extra* > *Options* 58
- In the Options dialog, select *Project Management* > *General options*

Note

- Only activate the Project view if you wish to use and manage many experiments in separate projects.
- If this function is deactivated, some dialogs appear in simplifier form (e.g. Save experiment and Open experiment without project selection, among other things).

Experiment Template

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

To be able to use experiment templates, you must first make experiment templates visible. This option is deactivated by default.

- Select from the menu *Extra* > *Options* 58
- In the Options dialog select *Project Management* > *General options*

Note

- 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 without experiment template selection, among other things).

4.6.2 Data Management

imc STUDIO uses a "*Database*" for saving projects. The database's path can be freely specified.

- Select from the menu *Extra* > *Options* 58
- Select in the Options dialog *Project Management* > *HDD settings*

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	
| | logo.png | | MyProject.imcAppSettings | | project.pcf |	Project properties
| \---templates | | MyTemplate.imcStudioTemplate | | StandardTemplate.imcStudioTemplate |	Experiment templates
| \---~MyProject | \---2012-11-19.bak | MyProjekt~001.imcAppSettings |	Project backup files
\---MyExp	Experiment Name
+---00000001 | 129977989468984375.ms.lnk | Chanel_01.raw | meta.de.csv.lnk | +---00000002 | 129977989628330078.ms.lnk | Kanal_01.raw | meta.de.csv.lnk |	Saved measurements (the data structure is set in the device settings under Storage 150) • *.raw: measured data • *.ms.lnk: link to the appropriate experiment setting • *.csv.lnk: link to the saved metadata
+---config	

`|  |  MyExp.imcStudio` `|  |`	Experiment-file (*.imcStudio) and internal files

`|  +---.ms`
`|  |  129977989468984375.ms`
`|  |  129977989628330078.ms`
`|  |`
	Saved experiment (configuration) containing the saved measurements

`|  \---~MyExp`
`|  \---2012-11-19.bak`
`|  MyExp~001.imcStudio`
`|  MyExp~002.imcStudio`
`|`
	Experiment's backup files

`\---Meta`	

`\---.mm`
`+---0`
`|  |  meta.de.csv`
`|  |`
`|  \---meta.de.data`
`|  \---DevSetup`
`\---1`
`|  meta.de.csv`
`|`
`\---meta.de.data`
`\---DevSetup`
	Saved metadata belonging to the saved measurements

Note

Please note that not all files are listed here. For the internal administration of the database, several internal files are necessary.

4.6.3 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	Exp:	Experiment
Proj:	Project	Tab.d:	Table description
Seq:	Sequencer	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	Appl						
Repository (Automation)								File
Repository (Panel)								File
Sequencer	Sequence table in the Sequencer				Exp			
Sequencer-Events (Experiment)	Events which apply to the loaded experiment				Exp			

Name	Description	Appl	Proj	Seq	Exp	View	Tab.d	File
Sequencer-Events (Sequencer)	Events which apply to the loaded experiment			Seq				
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					
Menu ribbon (customizing)	Which button appears where					View		
User defined buttons (collection)	What buttons exist and their significance		Proj					
User defined buttons (customizing)	Which button appears where		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					
Table descriptions	Aggregation of the normal and additional columns		Proj					
Normal columns	What columns exist and their significance						Tab.d	
Additional columns	What columns exist and their significance						Tab.d	
Column arrangement	What columns are displayed where					View		
Complete layouts	Completely designed pages		Proj					

5 Setup - Device Configuration

imc STUDIO Setup is the imc STUDIO plug-in for uniform configuration and control of imc measurement devices.

Start here

- Devices supported 88 (Device Overview)
- Ribbon 89
- Operation 129
- Setup pages
  - Documentation - Experiment description 148
  - Configuring Devices 148 (saving, synchronization, timer start, Diskstart)
  - Configuring Channels and Variables 159 (amplifiers, filters, sampling...)
  - Adjusting channels 177
  - Configuring triggers 182

The Setup includes functions with which to control the measurement process (the experiment):

- Performing a measurement: Procedure 139

When using the device configuration for the first time, you should first become familiar with the operation 129 concepts.

With the Setup pages , you can configure one or more measurement quickly and easily. During preparation of a measurement, the software recognizes the device's firmware, so that the device is ready to measure immediately.

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.

If the documentation mentions a device, the comments apply similarly to other devices as well. All screenshots were created using the *Complete* view.

Customer Support - Hotline

Questions or problems? Contact our Customer Support / Hotline 13.

5.1 Device Overview

imc STUDIO connects with the imc measurement devices e.g. via the local area network (LAN). In order to be used e.g. via the LAN, the devices must normally be prepared (see Setup - Connect the device 295).

Some of the capabilities discussed in this manual only pertain to certain device models. The associated device groups are indicated at the respective locations in the manual. The groups are shown in the following table which can be used with imc STUDIO.

— not available • standard ▯ optional

CRFX imc CRONOS*flex* CRC imc CRONOS*compact* CRPL imc CRONOS-PL

Device	Interface protocol / Bit/s		Supported data carrier			RAM		Rate [1] [kHz]	Short description
	Std.	MBit/s	CF [2]	PCMCIA	Hard drive	Data [MB]	Interface [MB]		
Group 2: SN12XXXX									
imc CRPL -2, -3, -4, -8, -13, -16 imc CRSL-2, -4	TCP/IP	100	—	•	▯	14	16 (32 from 2007	400	Modular system to identify on date of manufacture (as of summer, 2003)
Group 3: SN12XXXX									
imc C1 imc C-SERIES	TCP/IP	100	—	•	—	14	32	400	
Group 4: SN13XXXX									
imc BUSDAQ imc BUSDAQ*flex*	TCP/IP	100	•	—	▯	16	32	400	Field bus data logger
imc SPARTAN	TCP/IP	100	•	—	▯	16	32	400	Modular system
Group 5: SN14XXXX									
imc SPARTAN-R	TCP/IP	100	•	—	▯	16	32	400	Modular system
imc CRC-400 imc CRFX-400	TCP/IP	100	•	—	▯	16	32	400	Modular system
imc miniPOLARES	TCP/IP	100	•	—	—	16	32	400	
imc C1-N imc C-SERIES-N	TCP/IP	100	•	—	—	16	32	400	

Device	Interface protocol / Bit/s		Supported data carrier			RAM		Rate [1] [kHz]	Short description
	Std.	MBit/s	USB [2]	Express Card [2]	Hard drive	Data [MB]	Interface [MB]		
Group 6: SN16XXXX									
imc CRFX-2000	TCP/IP	100	•	•	▯	16	512	2000 via EtherCAT else 400	Modular system

Device	Interface protocol / Bit/s		Supported data carrier			RAM		Rate [1] [kHz]	Short description
	Std.	MBit/s	USB [2]	CFast [2]	Hard drive	Data [MB]	Interface [MB]		
Group 7: SN19XXXX									
imc CRC-400GP imc CRC-2000G imc CRFX-2000G(P)	TCP/IP	1000	•	•	▯	16	256	2000 via EtherCAT else 400	Modular system

1 Max. aggregate sampling rate (see data sheet)

2 We recommend storage media that are tested by imc (please consider the hotline for a current list)

Groups	Storage media
Group 4-6	For the purpose of onboard data storage, devices within those groups, are equipped with CF-Card, ExpressCard slot. imc CRONOS-devices within those groups can be equipped with an internally fixed hard drive available as an option.
Group 7	For the purpose of onboard data storage devices within this group are equipped with an CFast-Card slot. Devices within this group can be equipped with an internally fixed hard drive available as an option.

5.2 Ribbon

Reference

Some of the commands in this menu are compatible with the corresponding functions belonging to the predecessor software imc DEVICES.
Complete documentation on these functions is presented in the separate document
Setup - Advanced Device Functions 337.

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.

5.2.1 Control

Device Control

	Menu item	Description
	Process configuration 90	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 69).
	Connect 91	imc STUDIO connects itself with all selected measurement devices (normally via LAN).
	Download 93	Changed settings are analyzed and completely imported by the device.
	Reconfigure 93	All settings are analyzed and completely imported by the device.
	Start measurement 94	The measurement is started for all devices.
	Stop measurement 94	The measurement is stopped for all devices.
	Suspend data storage 95	If data transfer to storage is activated, it is suspended for all devices.
	Resume data storage 95	If data transfer to storage is activated, it is resumed for all devices.
	Disconnect 91	imc STUDIO trennt sich von allen Geräten.
	Search for devices 96	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 96	Search for devices by a user-defined IP address or DNS name.
	3rd Party Device Management 96	Opens the 3rd Party Device Assistant to add Third Party Devices to the device list.
	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.
	Show Start Trigger	Shows/hides the configuration of the Start trigger (Trigger_48). In most case, the configuration does not need to be modified. You can view the configuration on the Setup page: "Trigger".

5.2.1.1 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

5.2.1.2 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 142 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

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

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 91 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.

5.2.1.2.1 Disconnect/Connect - with a measurement running

Normal disconnection from and re-connection to running measurements

After *Disconnect* has been selected, the device resumes the measurement.

Upon disconnection, a disconnection file is generated, which is imported upon subsequent re-connection. If the re-connection is to the same computer from which the device was originally disconnected, this can proceed unnoticed.

Note

- If you start a measurement with computer A, then disconnect it from the device and later connect the device to computer B, this will not work.
- If the experiment is deleted form the PC with the disconnection file, then re-connection with the running measurement is no longer possible.

Disconnecting from the device

- If imc STUDIO is disconnected from the device during a running measurement, then a disconnection file is saved on the PC.

Re-connecting with the measuring device

- When connecting to a device, the system determines its current working status (experiment name, experiment settings, memory location for the experiment and of the disconnection file).
- Upon re-connection to a device running a measurement, the system uses the specifications referring to the current working status to determine whether the corresponding experiment or disconnection file is available on the operator PC.

Note

- As long as imc STUDIO is not connected to the device, it is not possible to view the current measurement data.
- If a gap in the data is detected upon re-connection (e.g. RAM buffer duration has been exceeded, or data points are missing), then saving of measured data to the PC-HD continues in a new folder.
  - In this case, the data from the gap are lost, unless they have been separately saved in the device.
- 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.

Note

If the selected data storage location is the device's *removable drive*, then in addition to the Diskstart-configuration, the imc STUDIO-experiment (*.imcStudio) is saved to the device's removable drive.

In this case, it is possible to connect with a device running a measurement, even without an experiment and disconnection file.

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.

5.2.1.3 Download 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:

- Reconfigure: All settings are analyzed and completely imported by the device.
- Download: 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 > Download ( )	Complete
Setup-Control > Download ( )	Complete
Setup-Control > Reconfigure ( )	Complete

Note

- Under some circumstances, there is a previous configuration preparation 90, unless this has already taken place.
- For preparation and reconfiguration, the PC must previously have set op a connection to the device (see Connect and Disconnect 91).

FAQ

Question: Why don't I have the buttons "Download" 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").

Question: What happens when multiple "writers" each try to set their own value during the initialization (Download)? 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: (as of firmware 2.8R7) 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.

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 the imc Hotline 13.

5.2.1.4 Start and stop measurement

The measurement is started/stopped for all devices.

Note

- It may be necessary to carry out the preparation 93, 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 91)

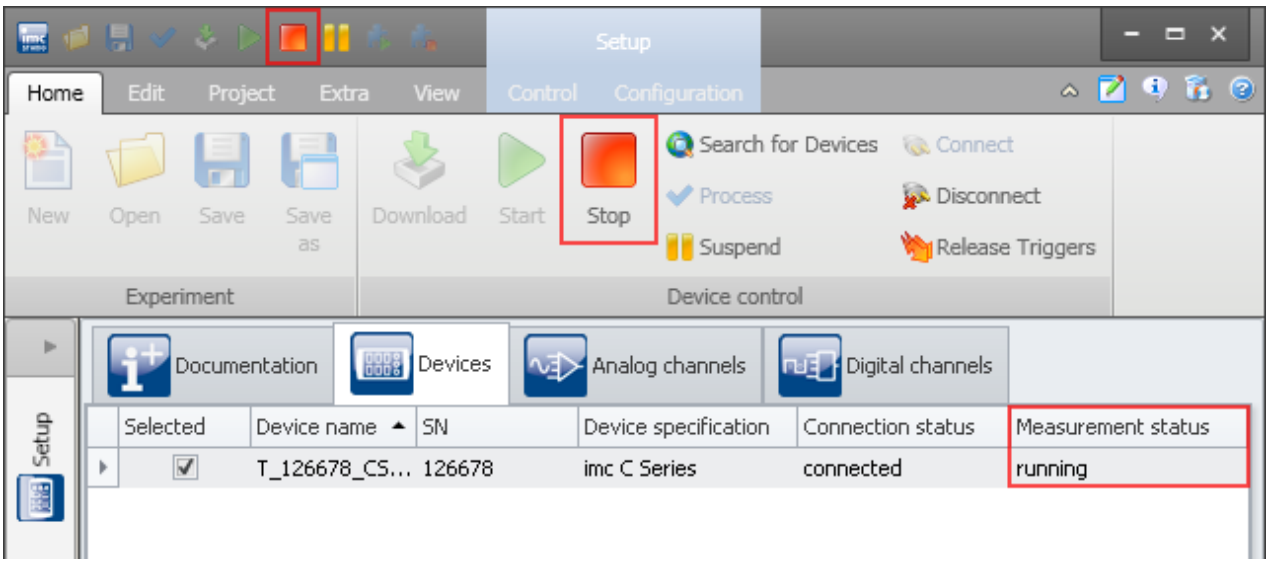
Ribbon	View
Home > Start ( )	all
Setup-Control > Start ( )	Complete
Home > Stop ( )	all
Setup-Control > Stop ( )	Complete

Measurement status (running, stopped)

The measurement's status is denoted e.g. by the stop-symbol (if you are connected with the device):

	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"

5.2.1.5 Suspend and Resume data storage

If data transfer to storage 174 is activated, it is suspended/resumed for all devices.

Ribbon	View
Home > Suspend data storage ( )	all
Setup-Control > Suspend data storage ( )	Complete
Home > Resume data storage ( )	all
Setup-Control > Resume data storage ( )	Complete

Reference

For a detailed description, see the manual:
Setup - Advanced Device Functions > Storage Options and Directory Structure > Controlling data storage > Suspend and Resume data storage 611.

Note

- 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.
- Data storage on both the PC's and the device's hard drive is suspended/resumed.
- The saving of Video files is not affected by this function. Video files are always recorded.

5.2.1.6 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 149 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 149 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:
  "*Setup - Connect the device* 295"
- For more information on selecting the device for the experiment, see the chapter: "*Device Search - Known and Selected* 140"

5.2.1.7 Search for devices by IP/DNS (TCP/IP, PPP via a router)

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 149 and to establish a connection.

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

Reference

More information is presented in the chapter: "*Setup - Connect the device* 295" > "*TCP/IP, PPP via a router* 312"

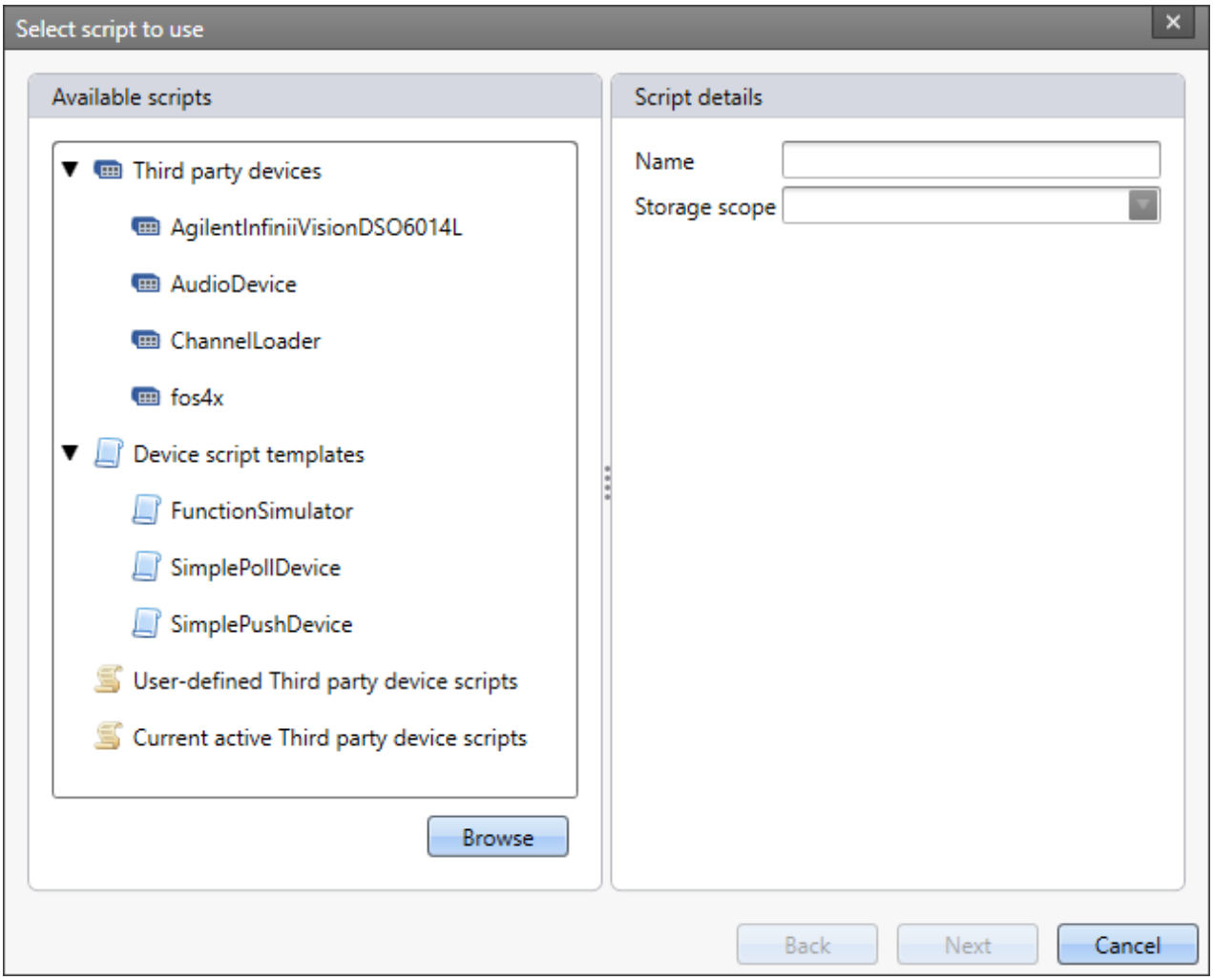
5.2.1.8 3rd Party Device Management

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 Agilent Scope and the fos4x are also available. The last of the two device require 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 necessary.
- The AudioDevice and the ChannelLoader can be used without any 3PDI license.

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*.

Verweis Reference

For more information, see Scripting - Third Party Device 1407.

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 1415.

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 1416.

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, ...).

5.2.2 Configuration

Assistants

	Menu item	Description
	imc Online FAMOS	This starts the imc Online FAMOS dialog. For a detailed description, see the manual: *Setup - Advanced Device Functions* > *imc Online FAMOS* 739.
	imc Display Editor	This starts the imc Display Editor dialog. For a detailed description, see the manual: *Setup - Advanced Device Functions* > *imc Display Editor* 664.
	imc CANSAS 103	This starts the imc CANSAS dialog.
	Assistants	Here, a variety of setup assistants are offered. For a detailed description, see the manual: *Setup - Advanced Device Functions* > *Fieldbusses* 338.

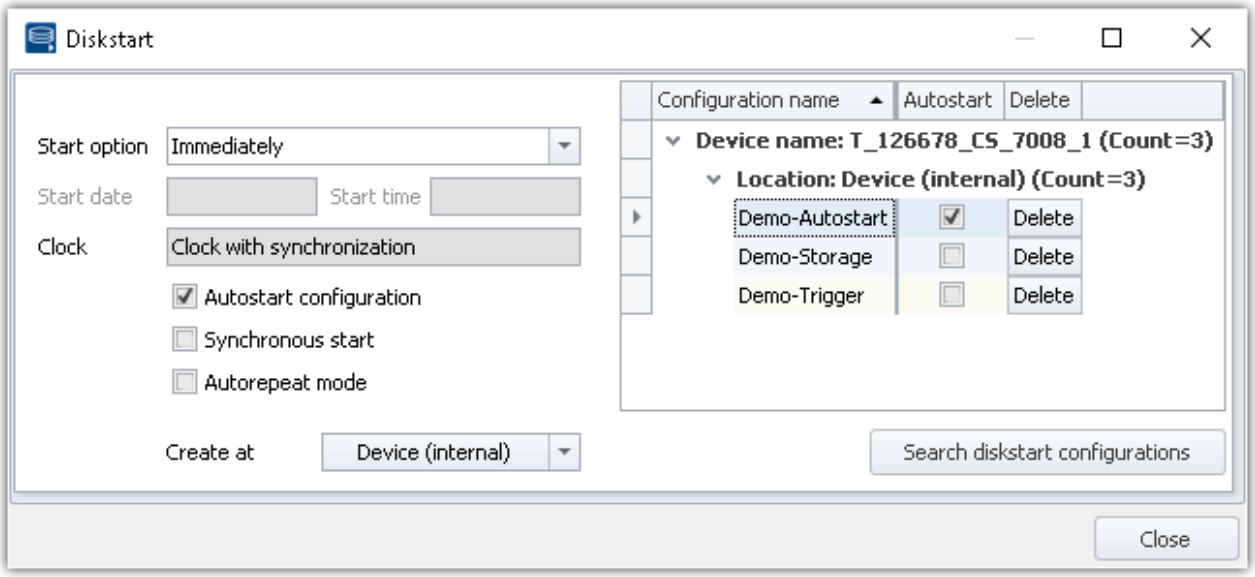
Synthesizer

	Menu item	Description
	Synthesizer Configuration and Synthesizer Control Panel	Operation of the Synthesizer is described in a separate document.

Device configuration

	Menu item	Description
	Export Configuration 241	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 242	This function enables the import of parameters. You can also select from which file to import which variable's values and/or settings.
	Statistic	Displays the dialog which shows various statistics on Setup parameters.
	Default Values 107	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 307	Displays the dialog for configuring the devices' (network-) interfaces.
	Diskstart 99	Displays the dialog for creating Diskstart and Autostart experiments.
	Device Properties 113	Displays the dialog showing the device's properties.
	Module Properties 117	Displays the dialog showing the properties of the device's amplifiers/modules.
	Supplemental Files 105	Shows the dialog to manage supplemental files.

5.2.2.1 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*.

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 Timestart 154 (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 153.

Observe the *maximum time to wait*, which is set in the device properties 115.

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.

Note on Timestart with Autorepeat

In Autorepeat mode, the duration between the measurements is not defined. For this reason, the Timestart 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.

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.

5.2.2.1.1 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.

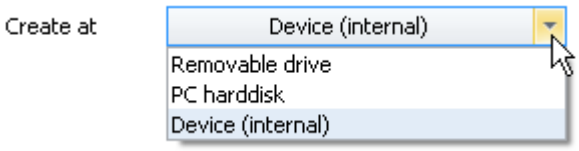
Preconditions for the storage locations: *Device (intern)* and *Removable drive*:

- A connection must be established between the device and PC.
- The device must have sufficient memory.

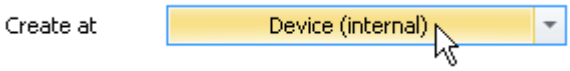
Creating a Diskstart configuration

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* 154.
- Activate the option: *Autostart configuration* 99 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:

- *.ume.zip (without Autostart) or *.ums.zip (with Autostart) (Diskstart configuration file for the device)
- *.imcStudio (a corresponding imc STUDIO experiment file)

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 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
```

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)

5.2.2.1.2 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 *Search diskstart configuration*. 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.

5.2.2.1.3 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.

5.2.2.1.4 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: Connect and Disconnect 91.

5.2.2.2 imc CANSAS

This starts the imc CANSAS dialog.

Note

To see this entry it is necessary that

- the imc CANSAS configuration software is at least from version 1.4Rev 5 and
- the measurement device is equipped with a CAN interface of type CAN2.

Reference

- 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 copied from the setup-CD for imc CANSAS.

5.2.2.2.1 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

- 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.
- When a sampling rate is set in the imc STUDIO settings dialog, this does not automatically affect the imc CANSAS software. Thus if you accidentally make some setting, an error message appears. The result is that you must open the imc CANSAS software in order to make the desired setting.
- When changing the "X-axis" (time stamp, sampling rate), the previously mentioned error message will also appear.

5.2.2.2.2 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.

5.2.2.3 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.

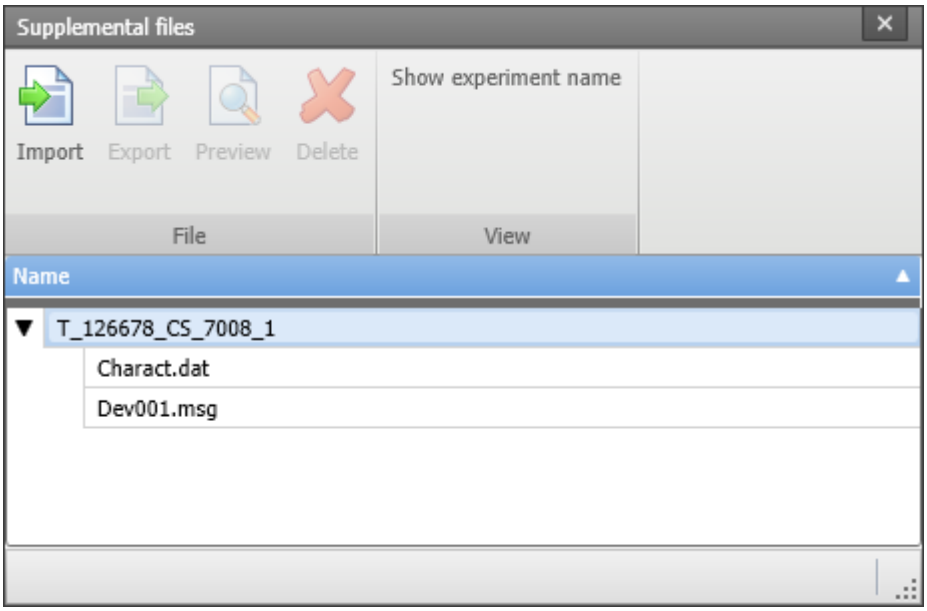
The following supplemental file type are available:

- 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)

The Supplemental files dialog

To open the dialog for managing the supplemental files, select the menu ribbon item *Setup-Configuration > Supplemental files*. 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* 56)

Imported supplemental files are saved in the Experiment-file. When the measurement is downloaded, the supplemental files are copied to the device memory. In this way, it is possible for imc Online FAMOS, for example, to access the file without needing to be connected with the PC.

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.

5.2.2.4 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 Field-bus 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.

The dialog for managing the default values is opened via the menu ribbon *Setup-Configuration > Default Values*. In this dialog, all available default values are listed.

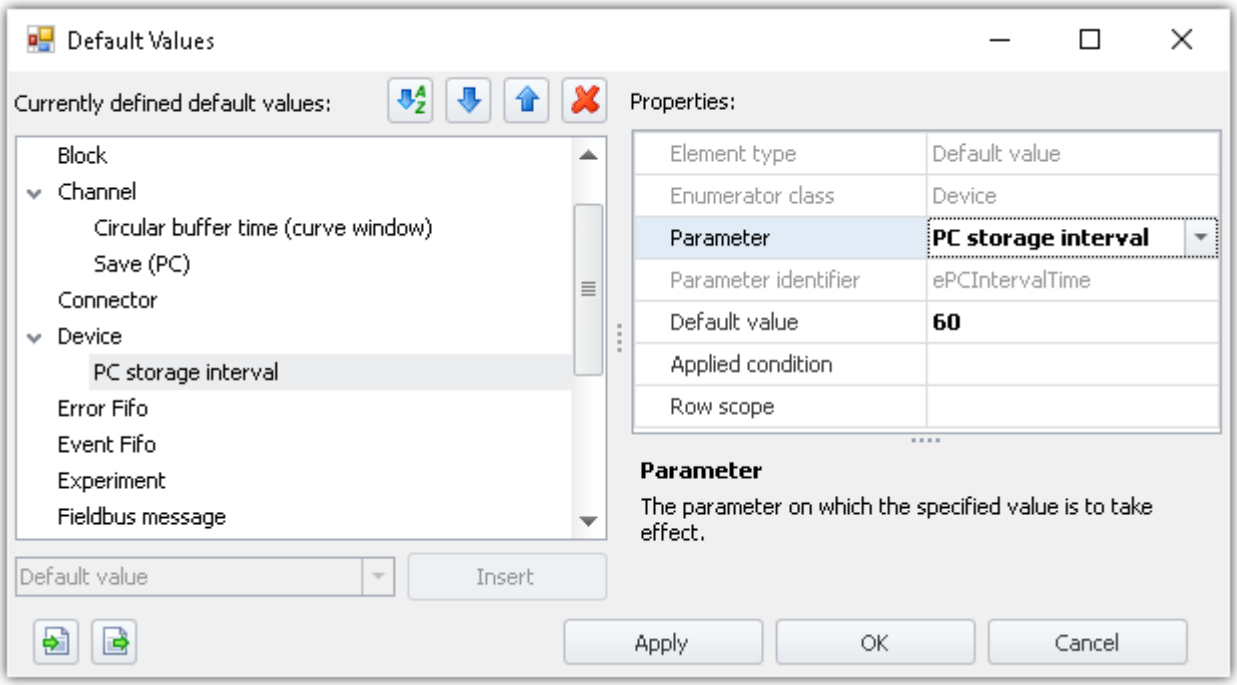
Example:

- 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.

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 254").
- 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* 109".
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.
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).

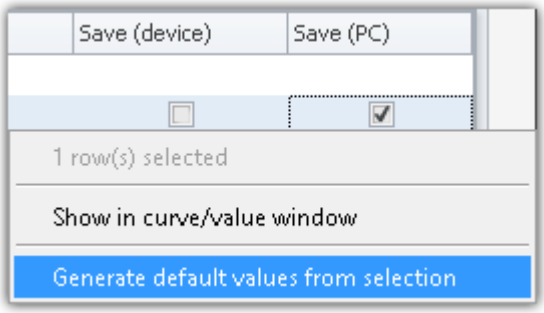
Dependencies can be defined either for each default value separately or for a whole group of default values. In this case, add a "Default value group" and add the respective parameters to the group. Define the application condition for the group. The individual default values do not require any application 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:

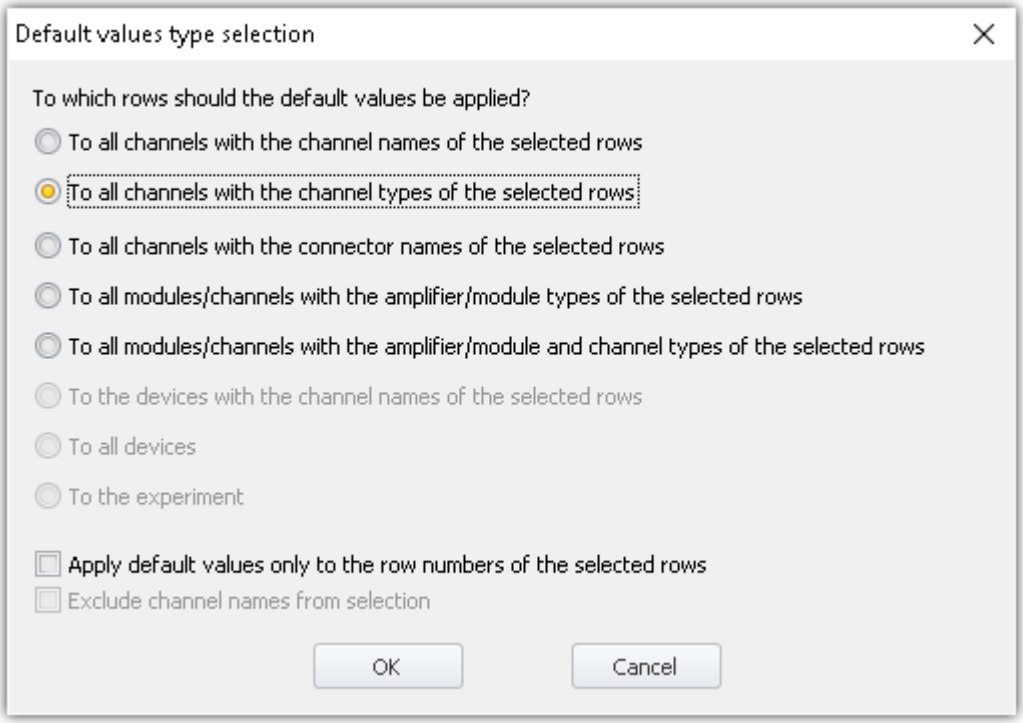
Make the following parameter settings for an analog channel:

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

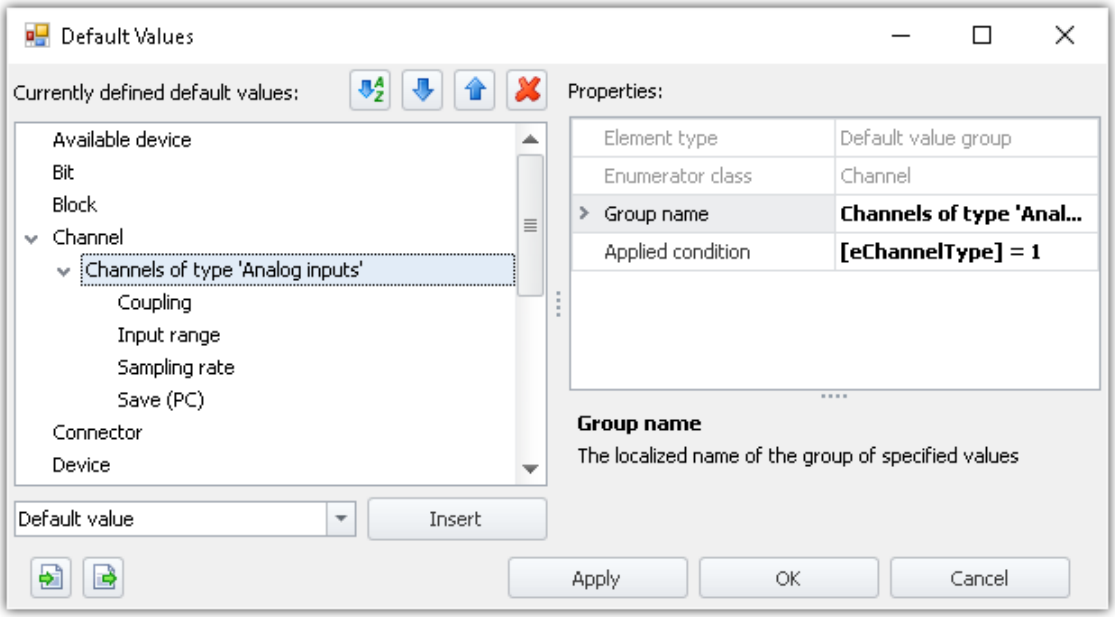
These settings are also found in the channel table in various columns. If you generate default values from a combined column 213 (one with multiple parameters), all of the column's parameter are set as the default value. In this case, either you have the desired parameter displayed 133 as a normal column 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.

Saving the default values

The configuration of the default values is saved with the respective installation (application). Thus, the default values are available in all experiments which were generated with that installation of imc STUDIO.

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.

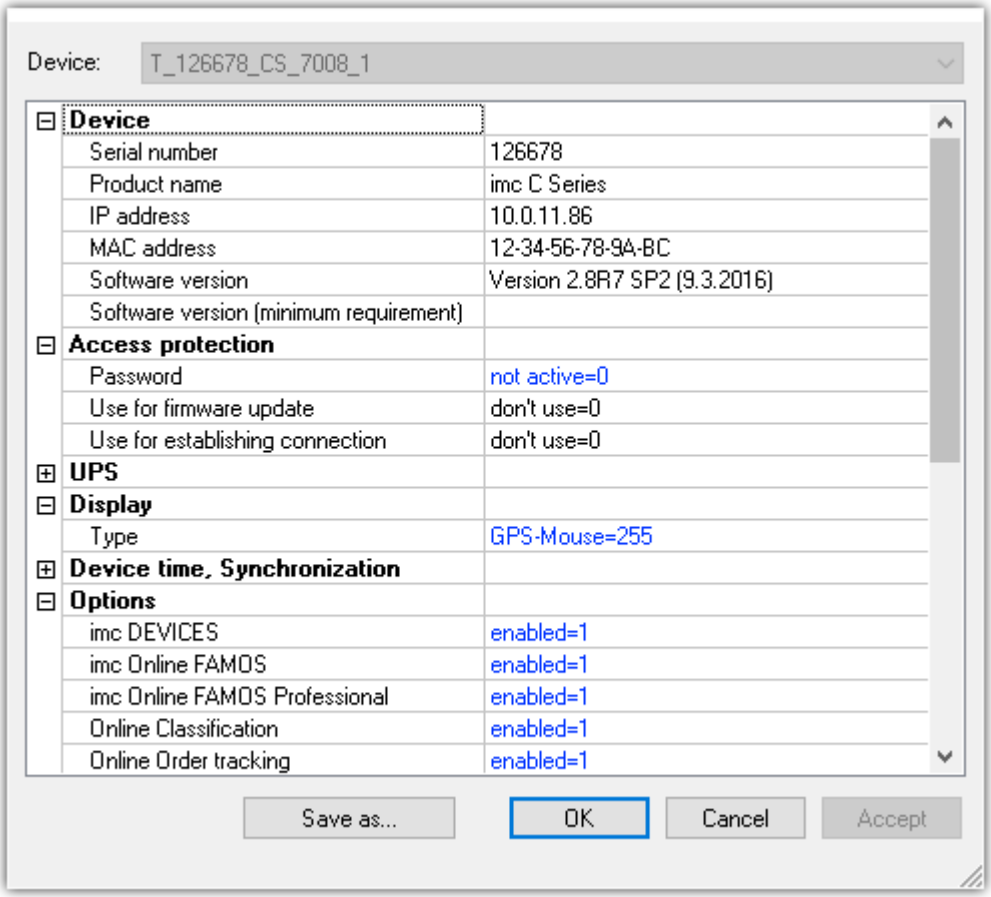
5.2.2.5 Device properties

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. But this dialog doesn't make a distinction. Technically an internal display is a "normal" display with a mounted cable. So even for these devices the display type can be changed.

A message is warning for wrong settings when opening the dialog. Please contact the imc Hotline 13 before changing the settings.

Note

All changes of the settings need a restart of the system!



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.
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.

Property: Access protection	Description
Password	For protection against firmware update or connection setup. To delete the password, enter the valid password again.

Property: Access protection	Description
Use for firmware update	Enable and Disable of firmware update 146.
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, imc REMOTE SecureAccess).
Certificate	The certificate needed for setup of a connection via imc REMOTE SecureAccess. Here, select the certificate "imcCert.cpt", which is in the installation path under *Firmware\Idif*, e.g. "*C:\Program Files (x86)\imc\imc_DEVICES_2.8R3\Firmware\Idif*". The certificate is always valid for one year and must be replaced upon expiration. The certificate is obtained from imc Hotline 13.
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. Changes take effect after rebooting the device. The alphanumerical display is not supported for all devices. 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 the imc Hotline 13. After switching to GPS, all GPS signals can be received be the process vector variables.

Property: Device time and synchronization	Description
Time zone	*none*: Compatibility to imc µ-MUSYCS / imc SPARTAN. Devices behave as in versions prior to imc DEVICES 2.7R3. The device seems to apply the time source's time zone. With download by the PC: the device runs in the PC's time zone With synchronization: the device seems to run in the synchronization source's time zone If a time zone is set, then when the device clock is set the time source is converted to the device's local time. For this purpose, the time source's time zone must be known. When setting the device clock, the PC tells its own time zone to the device. With all sources which send NMEA strings to the device's RS232 interface, it is assumed that the time is expressed in UTC. All time information at the Sync socket (DCF/IRIG B) is expected to conform to the time zone set for the device!
Daylight Saving Time	activated: the device automatically switches to/from daylight saving time. deactivated: the device uses standard time all year round!

Property: Device time and synchronization	Description

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.

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.

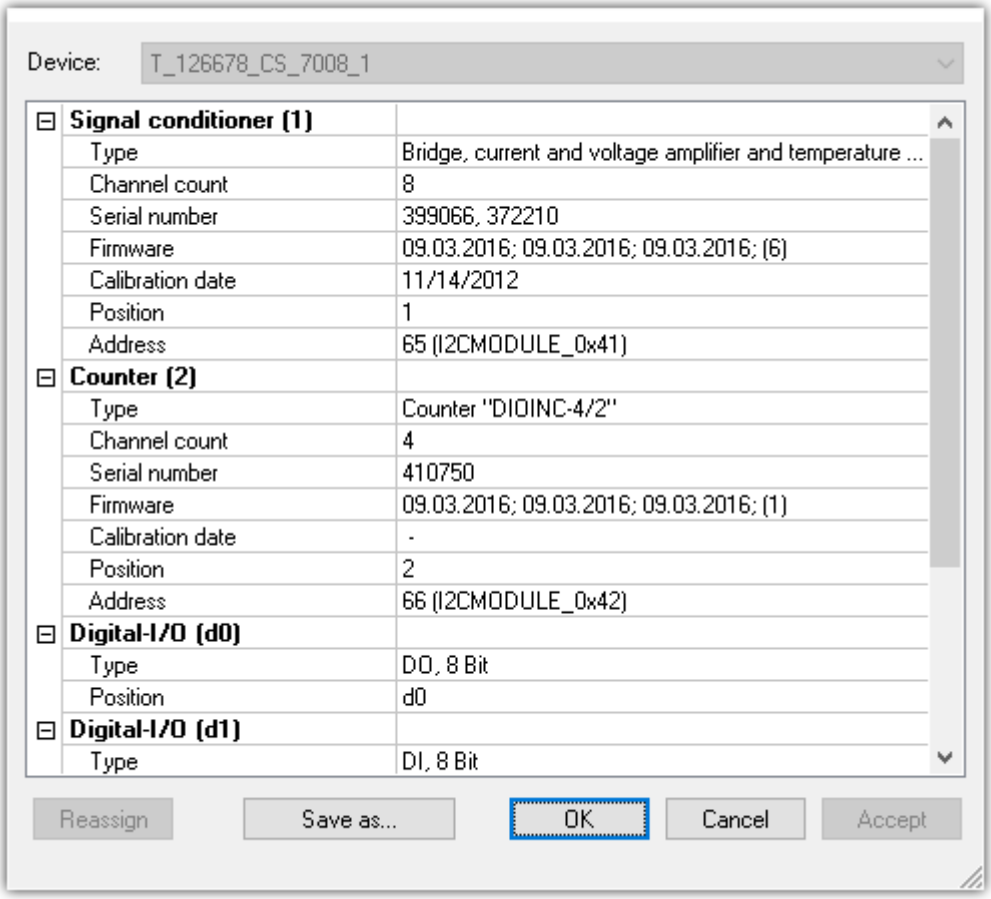
Property: Device time and synchronization	Description
Synchronization signal input	Specifies a default clock to be used upon activation. If the value for synchronization signal is n.s.=0, the settings remain intact after activating the device. 0 none: not defined. 1 no synchronization signal: Synchronization signal input not evaluated 2 DCF synchronization signal: only DCF or IRIB-B is evaluated 3 GPS synchronization signal: only GPS receiver evaluated 4 DCF or GPS synchronization signal: depending on which signal is applied first upon deactivation, either DCF or GPS is utilized 5 NTP synchron signal: Network Protocol used These setting are applied in the experiment when connected with the device and there are no other settings regarding Synchronization 153 had been specified for the experiment.
Synchronization signal output	Specifies whether the device time is outputted as a DCF-77 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 DEVICES	Enabling and disabling of operation under imc STUDIO.
imc Online FAMOS	Enable Online processing with imc Online FAMOS 739 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 739).
Online classification	Enables functions of Online Class-counting Kit (see the chapter on imc Online FAMOS 739).
Online-Order tracking	Enables functions of Order Tracking Kit (see the chapter on imc Online FAMOS 739).
CAN-Database Import	Enables Vector-database access, DBC-format; order option */VEC-DATB (license per device) •
imc CANSAS	Total parameterization of imc CANSAS modules connected at the CAN-interface. The availability of the option, however, depends on the particular hardware configuration and is determined upon connecting to the device.
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 739 and the imc Online/Inline FAMOS Function Reference 1440).
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.

Property: Options	Description
imc REMOTE	Activation of imc REMOTE LinkSecure, imc REMOTE SecureAccess 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 SecureAccess*: Enables imc REMOTE SecureAccess for direct and secure access to a device, even with activated Firewall. • *imc REMOTE WebServer*: Enables imc REMOTE WebServer 720 which provides platform-independent remote access to imc measurement systems.

5.2.2.6 Module properties

The dialog lists all amplifiers equipped with the device.



Modul configuration

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.

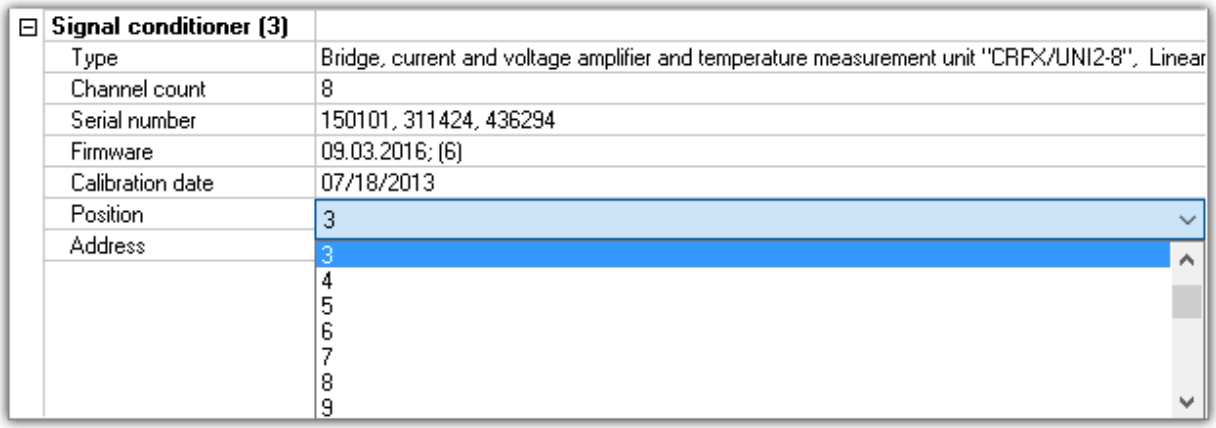
By using *Save as...* a file is created which lists the connected amplifiers

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 modules 161) are integrated with a master (e.g. imc CRONOS*flex* 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 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 imc CRONOS*flex* modules

If an CRFX 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 imc CRONOS*flex* modules

The device description identifies an CRFX module by its serial number. After exchanging an CRFX 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.

Notes on assigning the module number

- If an *flex module* is alternately used with different imc CRONOS*compact* 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.

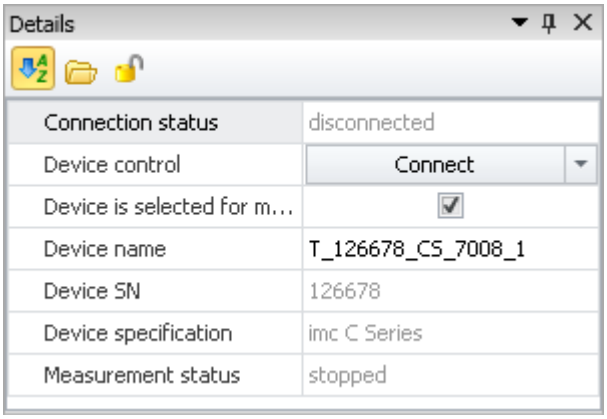
Notes on changing the device configuration

- 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")!

5.3 Tool Windows

5.3.1 Details

For purposes of Setup, the tool window *Details* is provided:



Tool window "Details" in the plug-in "imc STUDIO Setup" (example)

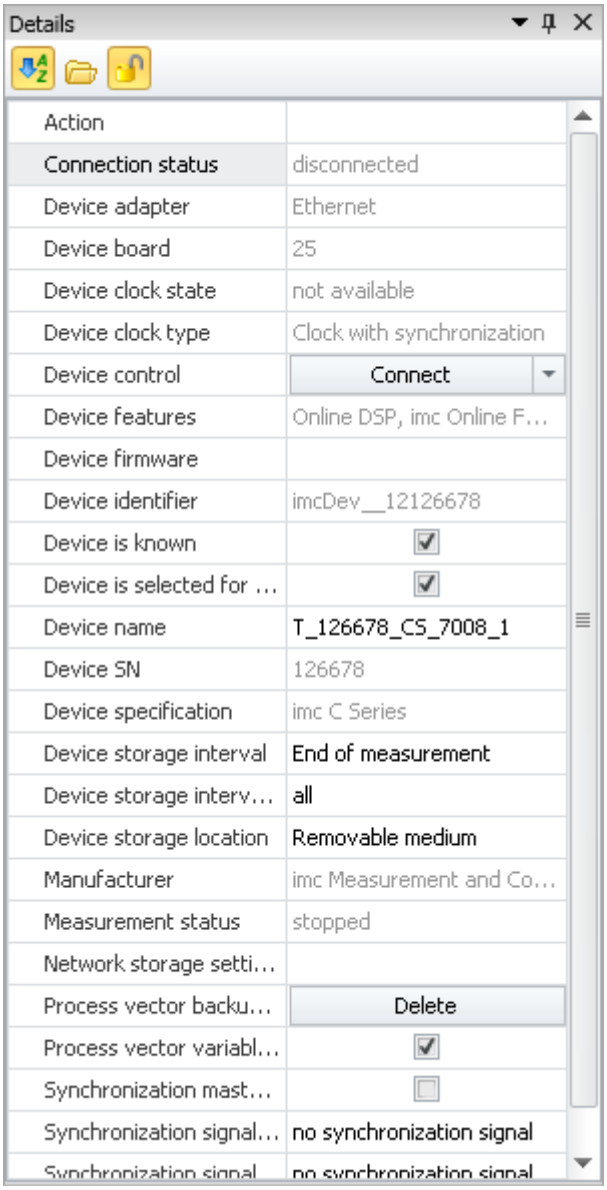
The content of the *Details* window depends on which page (see Operation 129) is selected. As in the tabular display, you can also directly edit 138.parameters in this tool window.

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

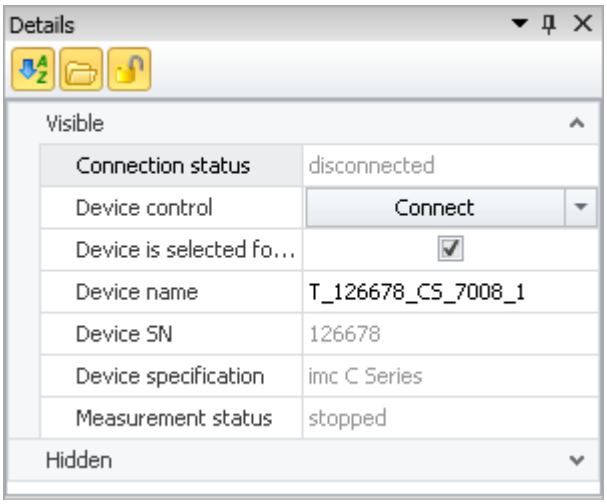
In this display, you can also set parameters which are not available either in the tabular display or in the dialogs.

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

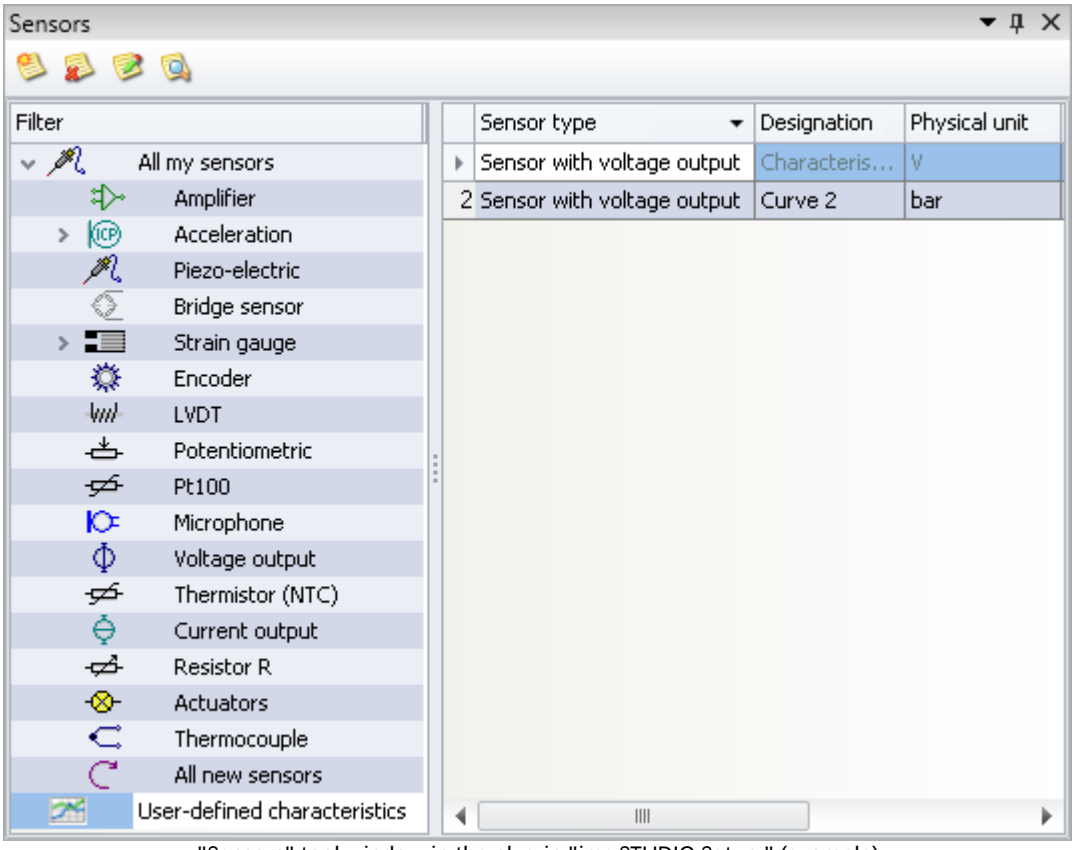
The groups can be opened or closed individually (symbols:).

To sort the parameters alphabetically, click on the sorting symbol ( ). Click again to change the sorting order.

5.3.2 Sensors

The tool window Sensors presents all information on existing sensors and characteristic curves (both referred to below as "Sensors" in most cases).

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.



"Sensors" tool window in the plug-in "imc STUDIO Setup " (example)

Structure

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. An exact description of the sensors is presented in the separate manual on 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.

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. Here, information on the sensors are edited. This applies in particular to the specs in the spec sheets and the calibration values.
User-defined 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. A list of amplifiers/devices 127 is presented at the end of this chapter.
Filter	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.
imc SENSORS 128	imc SENSORS is a tool for managing and editing information on sensors.

Writing sensor information to the channel

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.

Notes

- 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 amplifiers/devices 127 which support sensors having characteristic curves is presented at the end of this chapter.
- A list of modules supported is presented in the chapter : *List of supported sensors (TEDS/imc SENSORS)* 204.

The sensor connected is now also located under: "*Connected sensors*".

Deleting sensor information from a channel

In order to delete the sensor information from a channel, the Setup page: TEDS 201 is needed.

How to add a page is described in the chapter: *Configuring Device - Setup pages* > *Additional Pages* 205 .

- Open the Setup page: *TEDS*.
- 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.

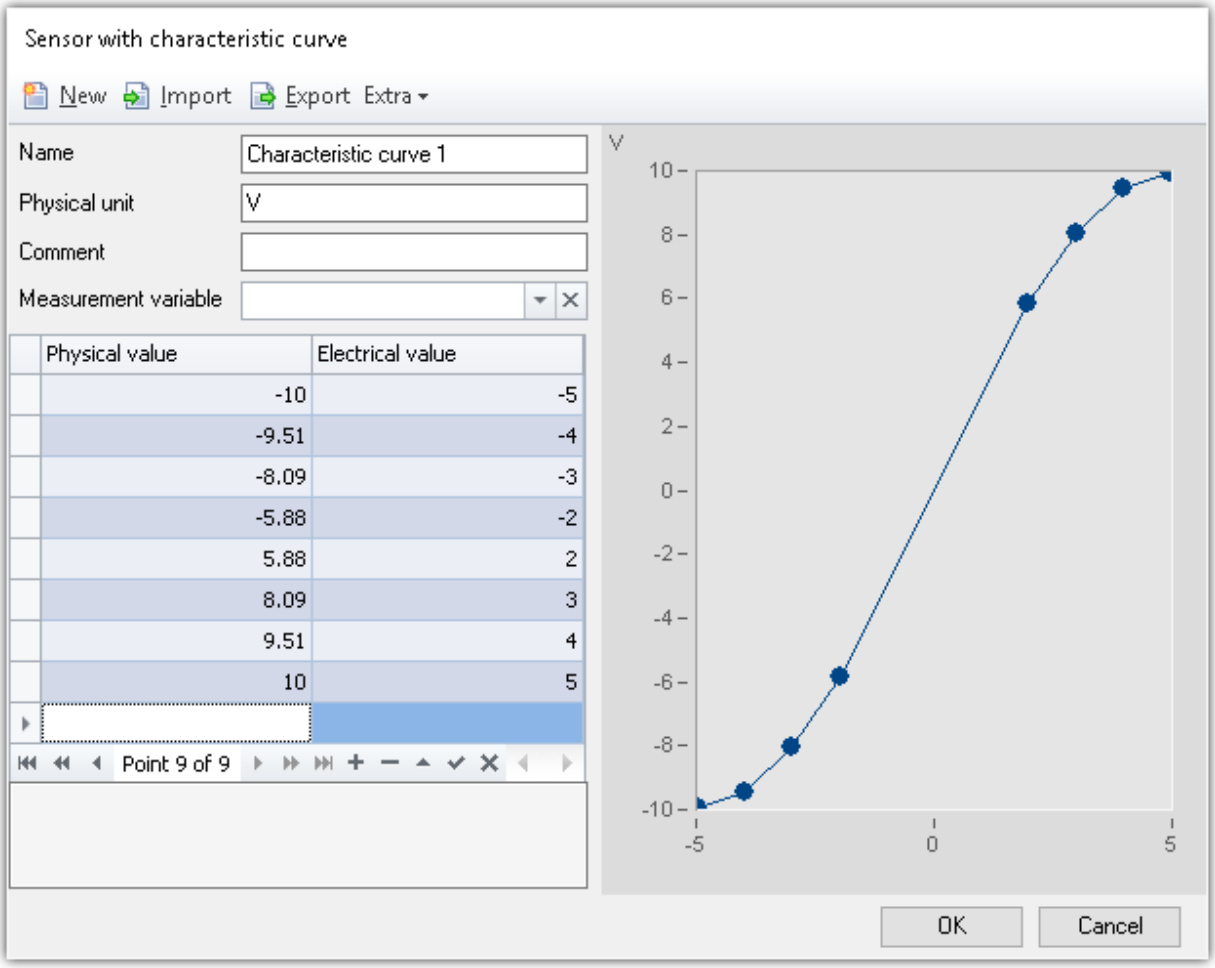
Toolbar and context menu

	Menu item		Description
	Create a new characteristic curve 125		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 128		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	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.

	Menu item		Description
			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.

Creating a new sensor

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.



Example of a user-defined characteristic curve

	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.

Points for the characteristic curve

A characteristic curve always consists of at least two points. In the table, you can enter the values for the characteristic curve, which will be displayed accordingly in the curve window.

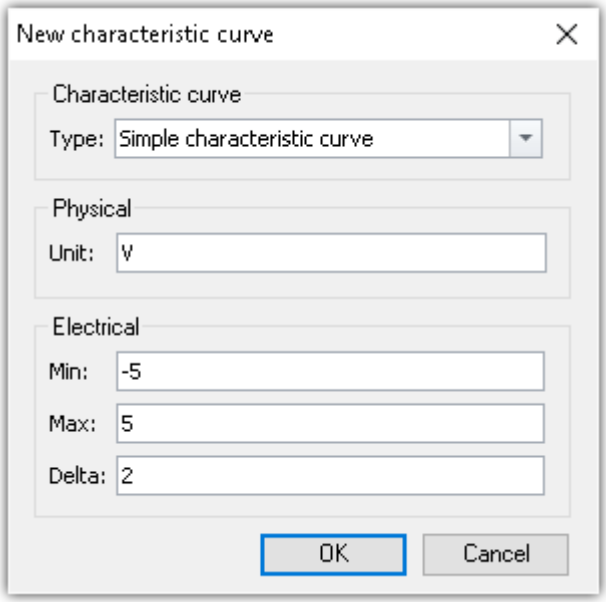
Values for the characteristic curve:

- Electrical value: the reading measured by the sensor
- Physical value: the value to be indicated in response to the corresponding electrical reading.

You can enter or import values ( ), or generate a line using the Assistant ( ), and later modify these as desired.

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.

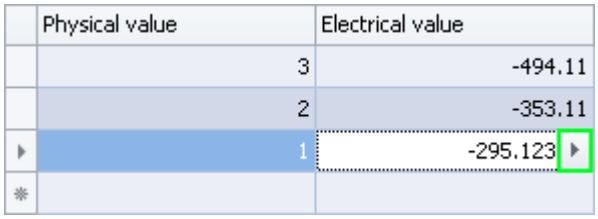


In this example, a straight line is generated between -5 and +5 with a point-to-point interval of 2. In this way, you are able to generate many points quickly, which can later be easily modified.

Importing the values via the measurement variable

The electrical values can be read from the sensor directly. Select the sensor's associated pv-variable as the measurement variable. 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:



Physical value	Electrical value
3	-494.11
2	-353.11
1	-295.123

Acquiring a new electrical value via the measurement variable

FAQ

Question: Which amplifiers support sensors with characteristic curves?

Support of sensor characteristic curves is enabled for following devices:

Amplifier/Device	CRPL/SL	Firmware	CRC	Firmware	CRFX	Firmware
ICPU-8	●	2.7R3	---		---	
DCB-8	●	2.7R3	---		---	
LV2-8	●	2.7R3	---		---	
UNI-8	●	2.7R3	---		---	
ISO2-8	●	2.7R3	●	2.7R3	●	2.8R5
ISOF-8	---		---		●	2.9R2
UNI-4	●	2.8R7	●	2.7R3	●	2.8R5
SC2-32	●	2.7R3	●	2.7R3	---	
ICPU2-8	Ø		●	2.8R7	●	2.8R5
UNI2-8	Ø		●	2.8R7	●	2.8R5
DCB2-8	Ø		●	2.8R7	●	2.8R5
LV3-8	Ø		●	2.8R7	●	2.8R5

Device	Feature	Firmware
Cx-41xx-N	●	2.8R7
SPAR-U	●	2.8R7

●: Feature supported

Ø: Feature currently not supported

---: Amplifier not available for this device series

imc STUDIO 3.0R4 included firmware 2.7R3

imc STUDIO 4.0 included firmware 2.8R3

imc STUDIO 5.0R1 included firmware 2.8R5

imc STUDIO 5.0R3 included firmware 2.8R7

imc STUDIO 5.0R6 included firmware 2.9R2

5.3.2.1 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* tool window lists all sensors in the database.

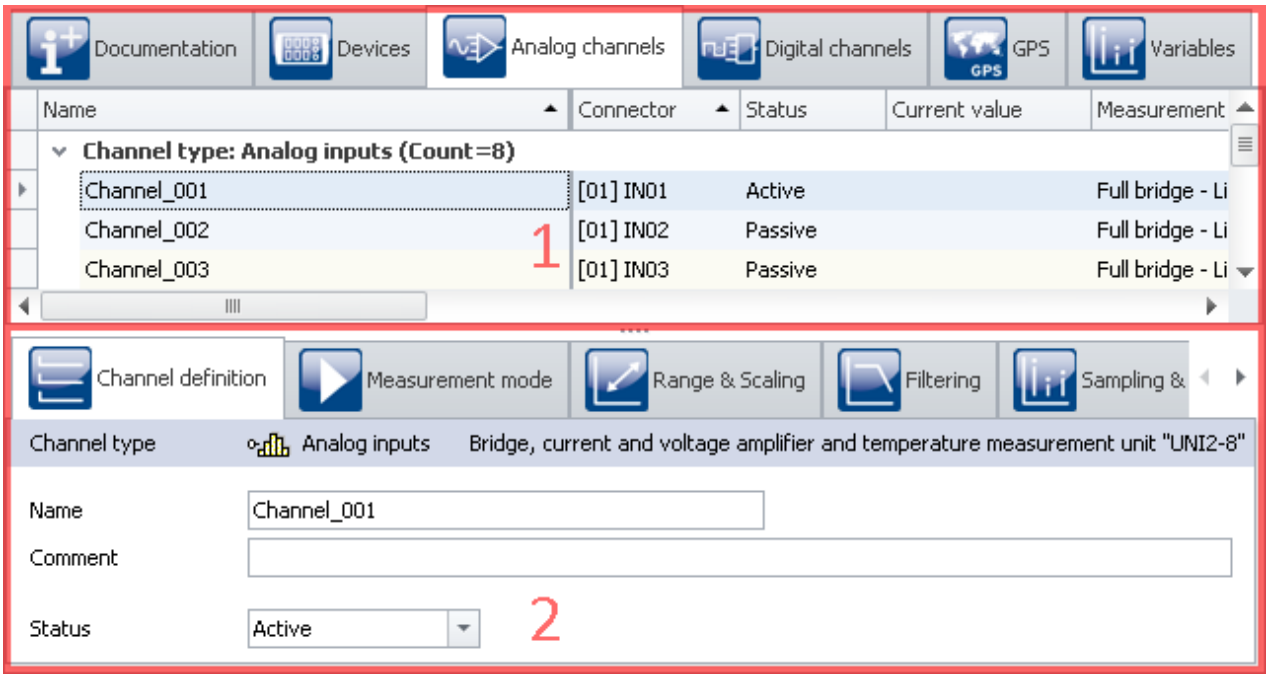
How to transfer the sensor information to the channel is presented in the description of the tool window "Sensors".

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 and select *Add to imc SENSORS.* The channels settings remain unaffected.

5.4 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:



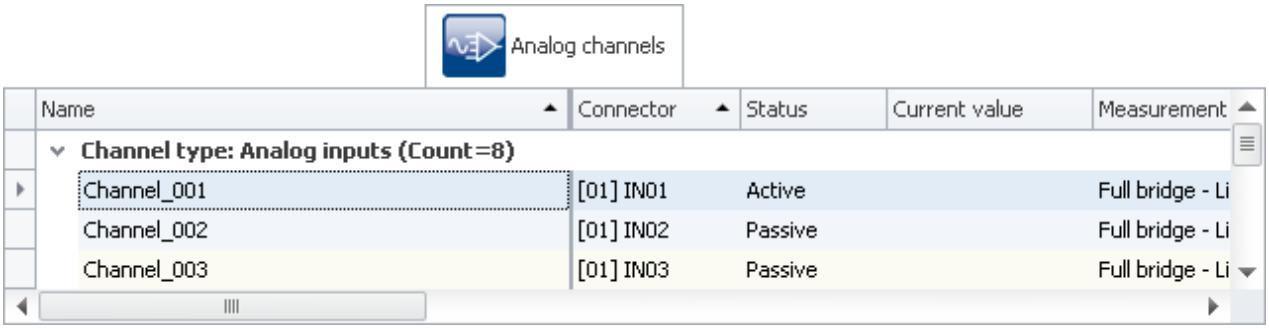
Tabular display and dialog
(example)

		Description
1:	Tabular display 129	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 137	The lower window shows special dialogs 137 for the most important parameters.

Editing: You can modify the parameters by means of in-place editing 132 in the table, as well as in the dialog 138. To edit the height of the window, use the mouse to drag the dashed separator line.

5.4.1 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 132 both individual parameters and sets of multiple parameters (see Selection 130) 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 133 to include the desired parameters in the table. Also, you can configure complex columns using the function Additional columns 211.

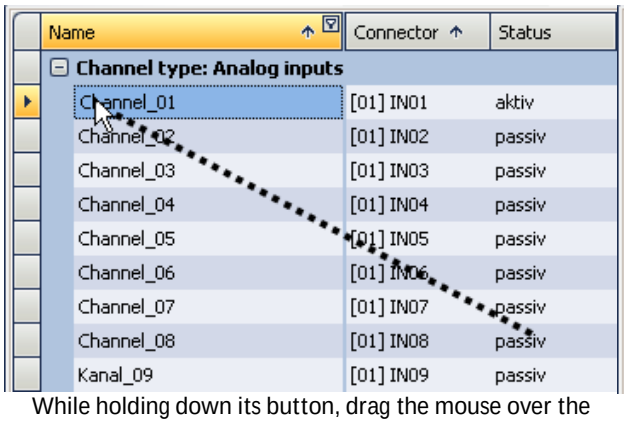
5.4.1.1 Selection

In the tabular display, you can edit both individual parameters as well as sets of multiple parameters directly in table cell.

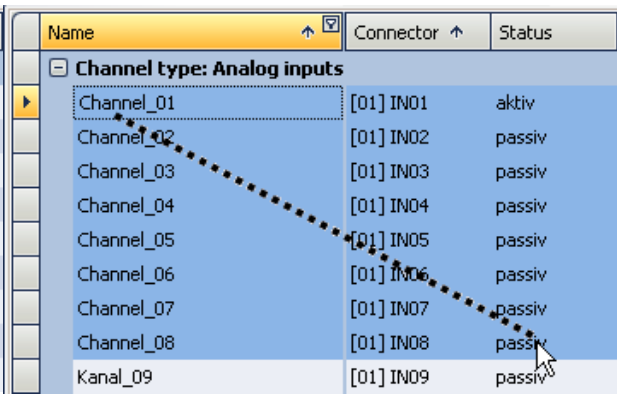
To do this, first select the row(s) which contain the parameters.

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



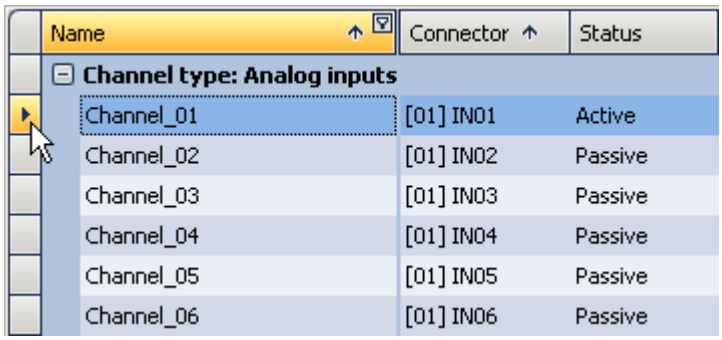
Release the mouse button at the position desired

Selection with mouse and keyboard

To do this, use the left margin column, as shown in the first image.

Selecting individual rows

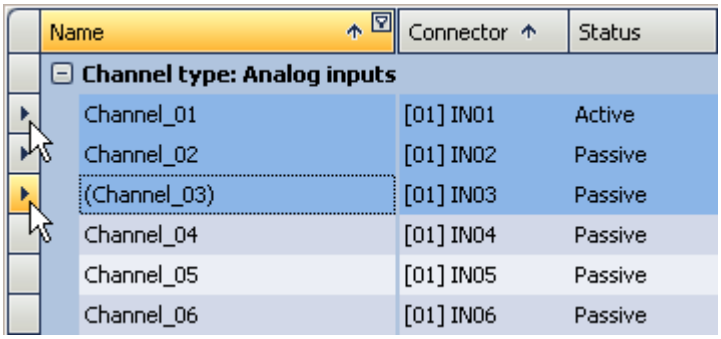
Click on the left edge of the desired entry:



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.

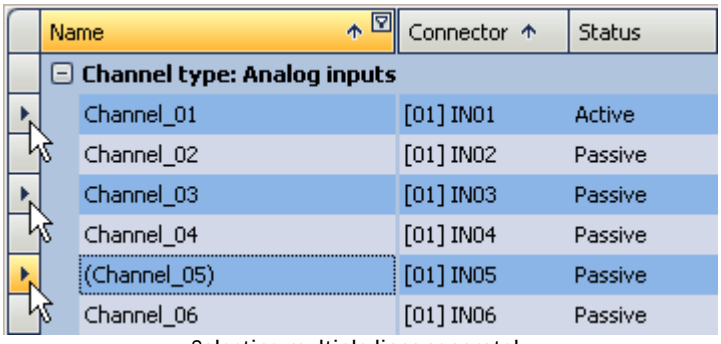


Selecting multiple consecutive lines
(hold down <Shift>-key)

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.

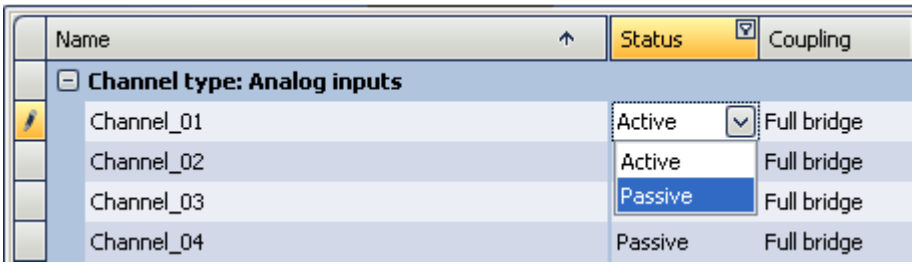
Cancel selection

To cancel a multiple selection, click on the left side of a single entry in the table

5.4.1.2 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 137 for selected parameters.

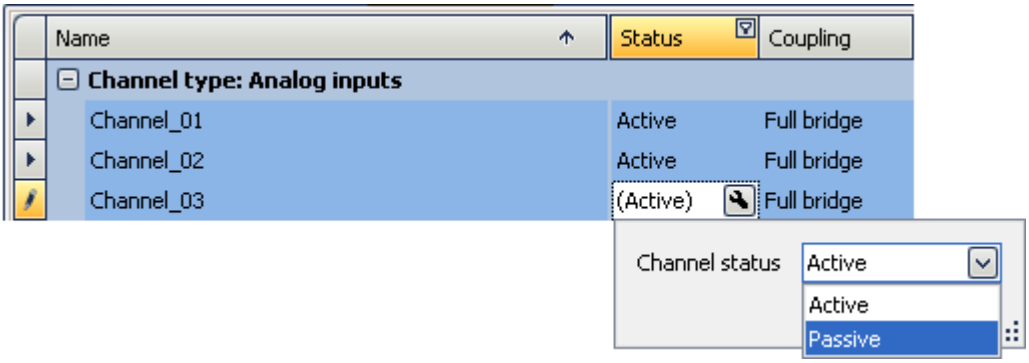
Editing single parameters



Editing single parameters:
click on the row

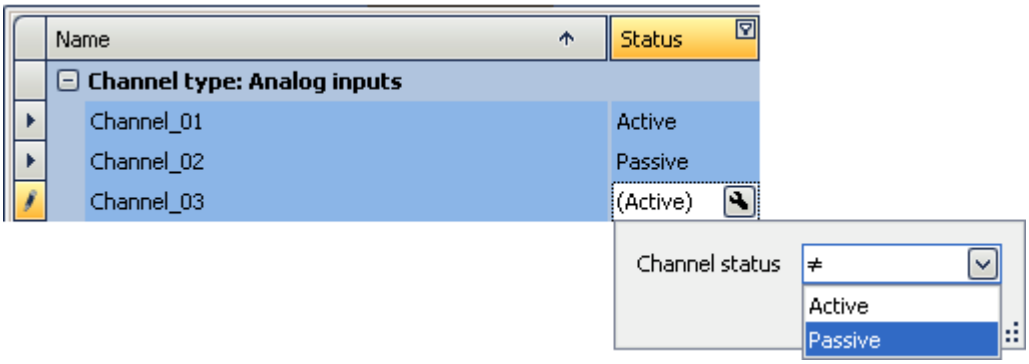
In order to edit individual parameters, click directly in the associated cell of the table. It is not necessary to select the line beforehand. After clicking, a control appears with which you can edit the parameters. The control's type depends on the particular parameter concerned. In the image above, 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 130). 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:

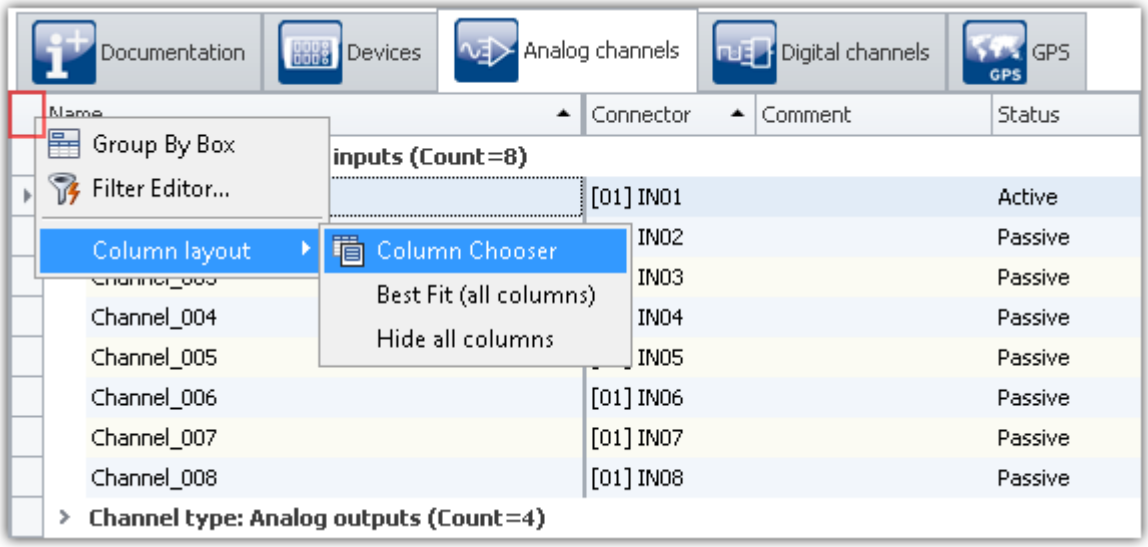


Different parameters in a multiple selection procedure
(example)

5.4.1.3 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. The column configuration is saved/opened with the view configured (see the manual imc STUDIO (general) > chapter Views 80).

Opening "Column Chooser"

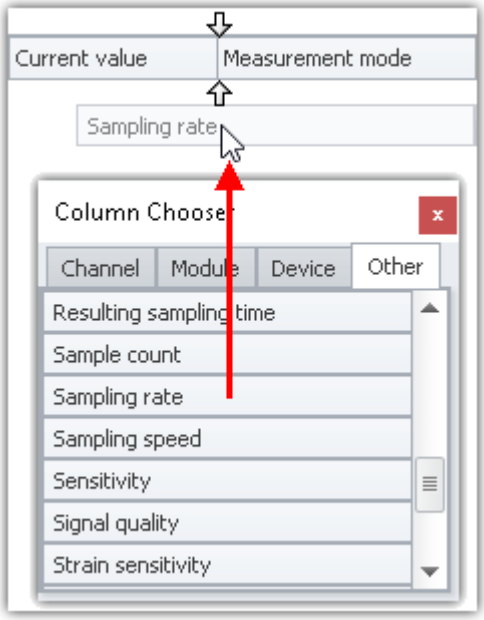


Opening the table's context menu

In order to insert columns, open the *Column Chooser* dialog. To do this, right-click over the upper left corner of the table and select the corresponding context menu entry.

Inserting columns

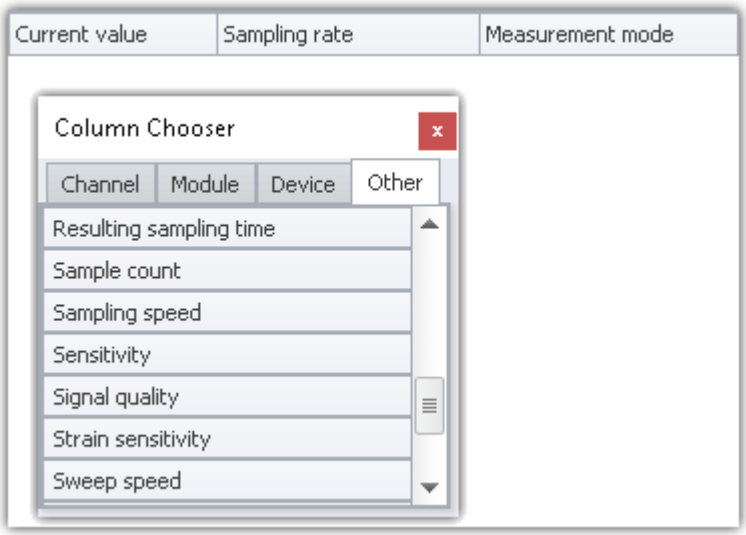
The dialog *Column Chooser* contains a list of al columns which can be inserted. In order to insert a column, drag it from the list to the column's header, as shown in the image below.



Dragging a column

How the frame is aligned (before the column has been placed) indicates whether the column is to be inserted to the left or the right of the existing column. 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, simply use Drag&Drop to drag its title bar to 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.

Saving / restoring a configuration

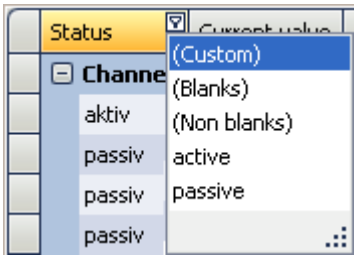
In order to save a column layout, use the functions belonging to the View menu (see the manual imc STUDIO (general) > chapter Views 80).

5.4.1.4 Filtering columns

User-defined filters

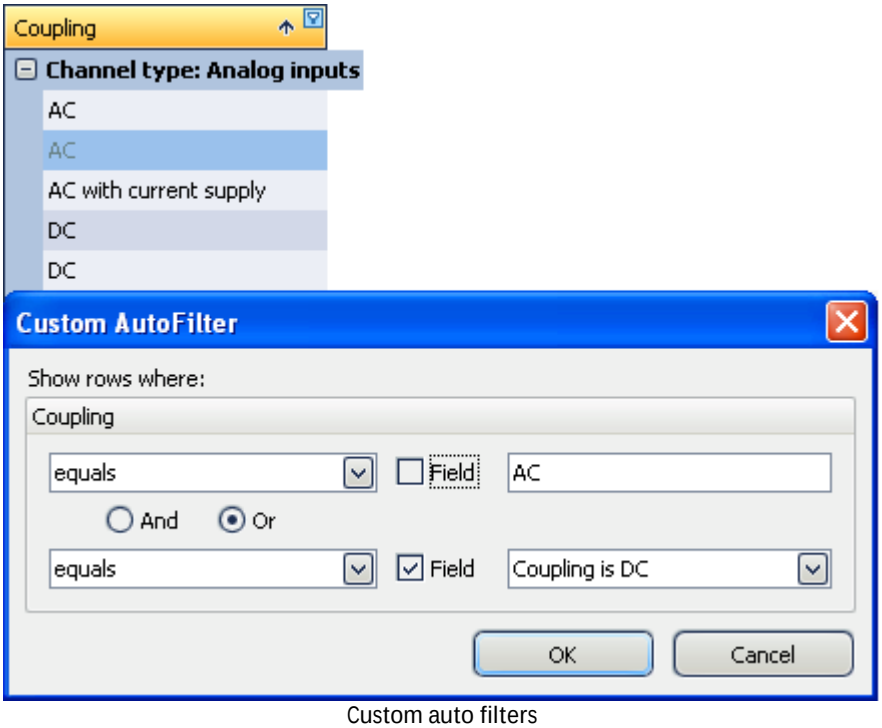
To start the filter, you must click the left mouse button in the column to which you wish to apply the filter, on the "filter-symbol" in the column's upper right corner.

A selection menu opens. Here, click on "(Custom)"..



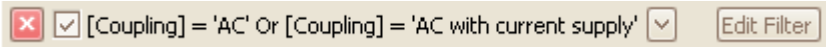
Filter menu
(example "Status" column)

The "Custom Auto Filter" opens. After setting the filter, click on OK. The "Filter status indicator" opens.



Custom auto filters

Filter status bar

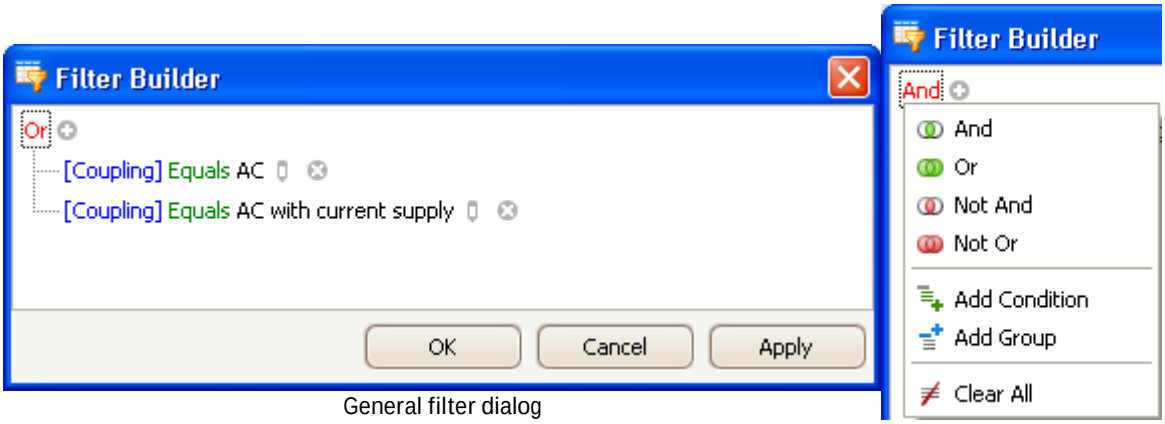


Filter status bar

Editing filters

To define complex filters, click in the filter status bar on the button *Edit Filter*.

A dialog as shown below then opens:

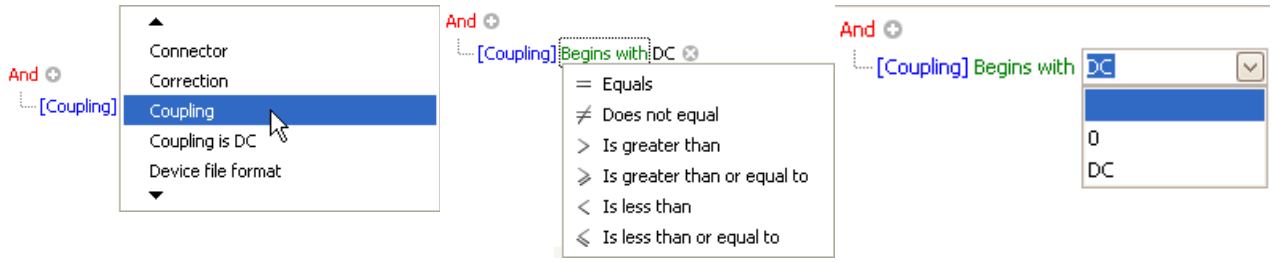


General filter dialog

With this dialog, you can formulate complex combinations of filter conditions.

The filter conditions are defined using the logical operators AND, OR etc. Using the symbol ⊕, it is possible to add multiple filter conditions. Note that the correct syntax for a filter condition is like filling three columns, with the first having the coupling, for instance, the second having the condition operator, and the third the prevalent "state".

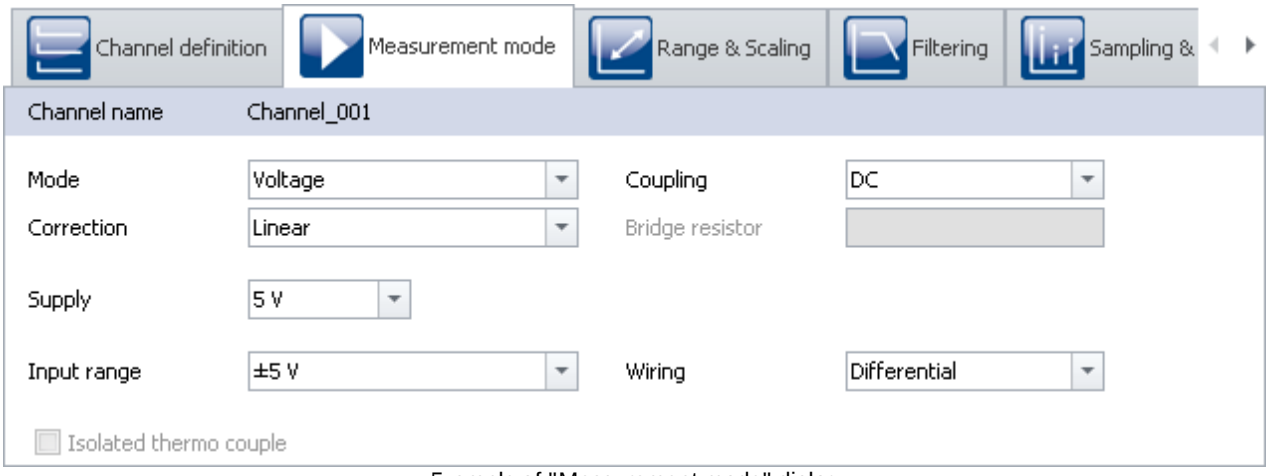
By means of the tool ⊗, filter conditions can also be deleted.



The filter conditions are saved along with the program data (not with the experiment). To activate one, open the drop-down list in the Filter Status bar.

5.4.2 Dialogs

In this section, the most important parameters of special dialogs are offered.



Example of "Measurement mode" dialog

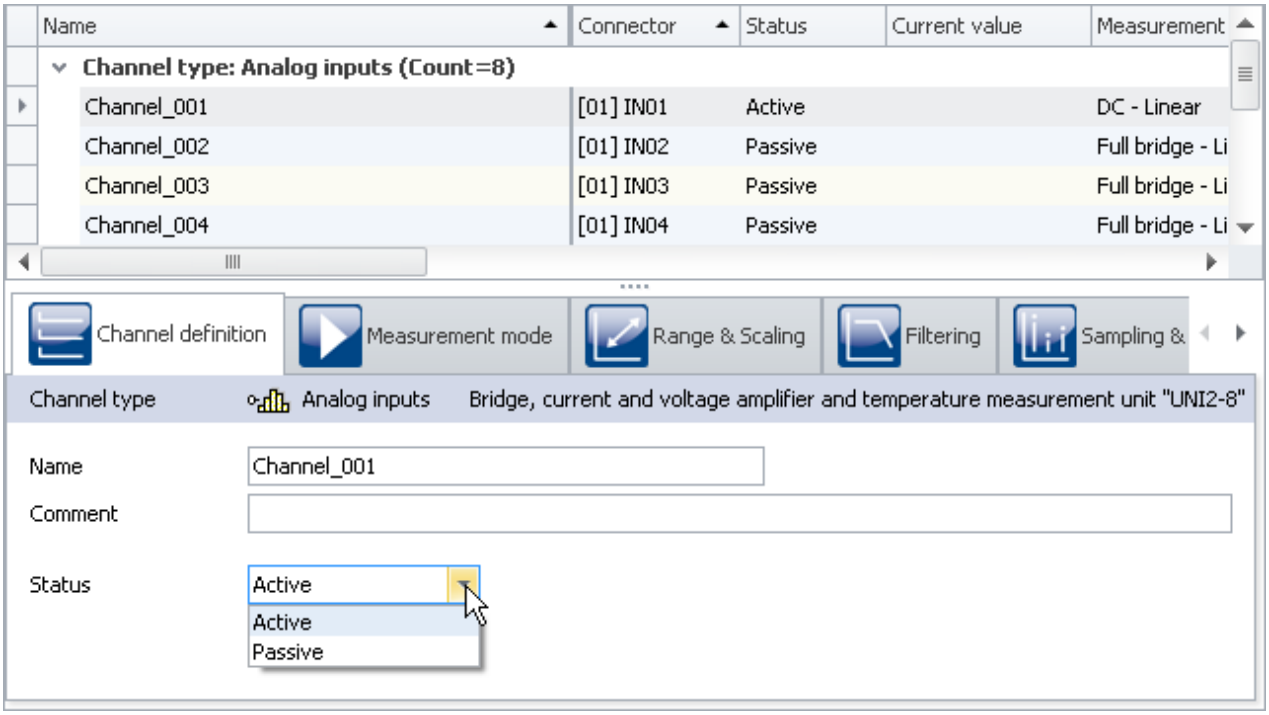
Selected parameters are presented here, distributed over multiple dialogs. The dialogs are designed in a clearly structured manner. As with the tabular display, you can also directly edit 138 parameters in the dialogs.

Availability and selection of the dialogs and the settings depends on what page is open and on the selection in the table.

5.4.2.1 Direct editing

You can directly edit (change) all parameters displayed in the dialogs.

Editing individual parameters



Editing individual parameters (example)

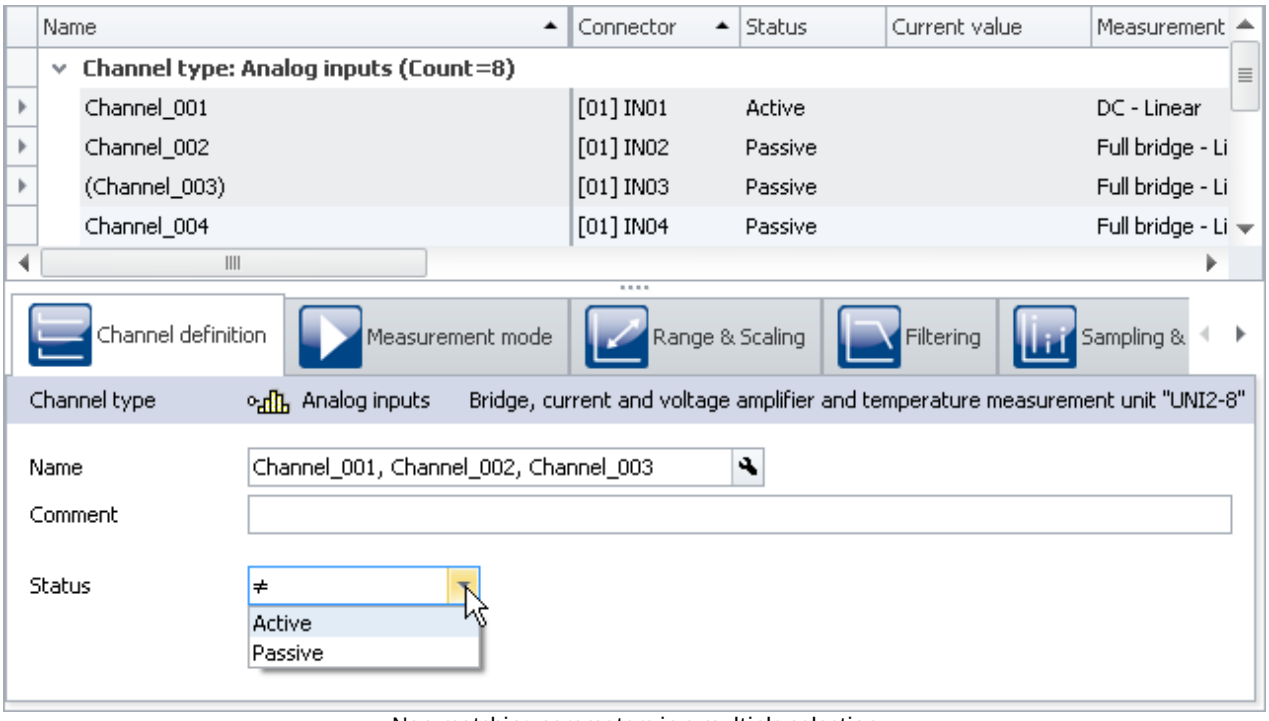
To edit a parameter, first select the corresponding table cell in the tabular display (see Selection 130). The dialogs associated with the selection are indicated. Open the dialog in which the parameter to be changed is found.

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)

5.5 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 89.

		Description
	Connect signal leads	
	Searching for and selecting devices 140	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 142	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 147	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 90	Current configuration is inspected for errors and made available to all the plug-ins. No connection with the device is required for this.
	Connect 91	The program 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.
	Scandisk	Scan the internal storage medium on a regular basis if used.
	Download 93	The current configuration is exported to the device.
	Start measurement 94	The measurement is started for all devices.
	Stop measurement 94	The measurement is stopped for all devices.

5.5.1 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 149 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.

Select the Ribbon Home (or Setup-Control) > Search for devices ( ). The system then searches for devices in the network (see Device Search 96).

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)

Note

The devices listed now are not yet known!

To make a device which has been found "known", you can do either of two things:

- *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 119).

Once the device has been made *known*, its setup is determined.

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.

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.

Note

You are not yet connected with the device!

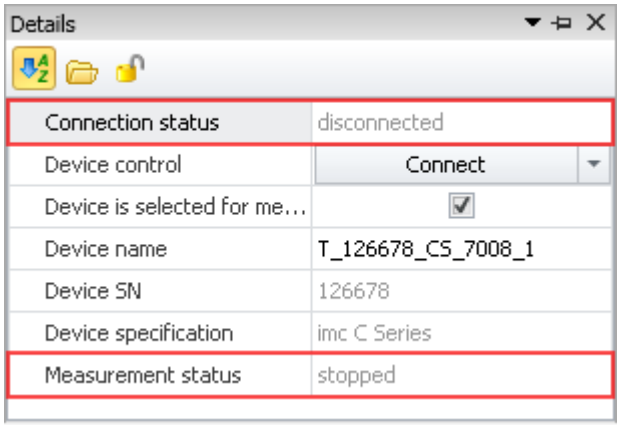
If your device was not previously known, it is made *known* the first time the it is selected.

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

5.5.2 Firmware

The device software (imc STUDIO) always ships with a matching firmware version. 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 143 is carried out.

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 (Ribbon *Extra > Options - Setup > Device options > Selection of the imc DEVICES version*):

Selection	Description
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.

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: There is a clear assignment regarding the compatibility with the firmware/firmware group:

imc STUDIO 3.0 is assigned to the firmware group 2.7

imc STUDIO 4.0 is assigned to the firmware group 2.8R3

imc STUDIO 5.0R1 is assigned to the firmware group 2.8R5

imc STUDIO 5.0R3 is assigned to the firmware group 2.8R7

...

As of imc STUDIO 4.0, the applicable rule is: the assigned group and all older groups up to 2.8R3 are compatible with the imc STUDIO-version used.

Example: the following firmware groups are compatible with imc STUDIO 5.0R1: 2.8R3 and 2.8R5.

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 148, a column can be added: Device firmware
- In the Device properties 113.

5.5.2.1 Firmware Update

If the firmware on the device does not match the firmware on the PC, the Firmware-Update is performed.

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

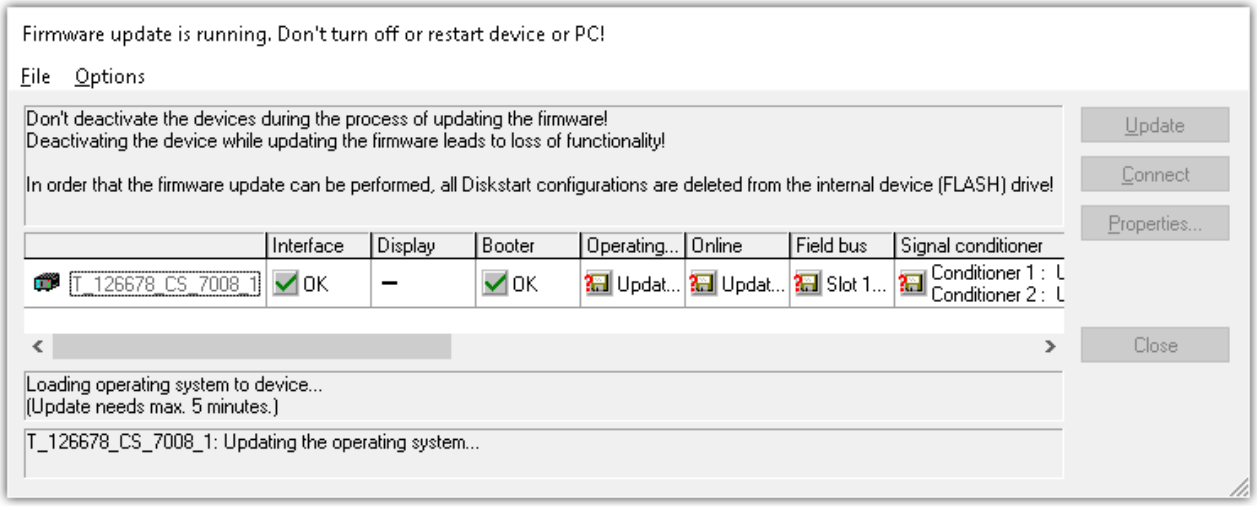
Note

- The firmware update is only required if the software was obtained as an update. If you obtained your measurement with the software, no firmware update is necessary.
- Before transfer to the device module is completed, the system checks whether the firmware already in the device does not match the software used. The firmware update for the device module is only performed if it does not.

Warning

Do not under any circumstances deactivate the device during the firmware update.
If any error messages appear during the firmware update, do not switch the device off, but contact the imc Hotline. The firmware update may be continued with guidance from the Hotline.

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 682
Field bus	Field bus
Signal conditioners	Amplifiers

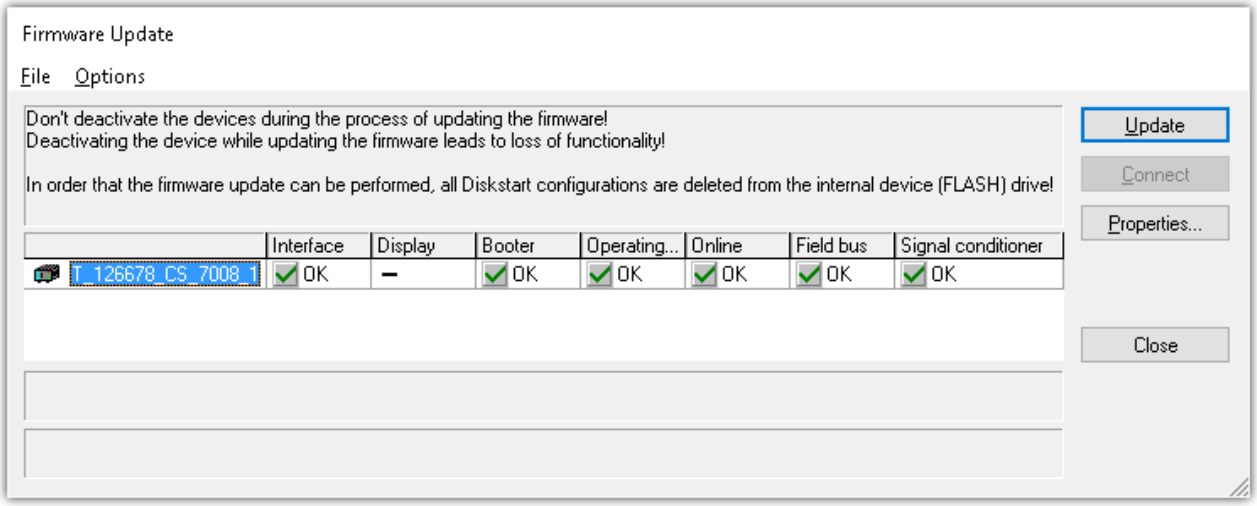
The following symbols for the individual firmware components appear in the list:

	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 amount 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.

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!

- 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 FW-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, access SAP or Novell in parallel, which are often running on 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 the imc Hotline. It may be possible to continue the firmware-update under the guidance of the Hotline using a service program.

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 imc-Hotline.

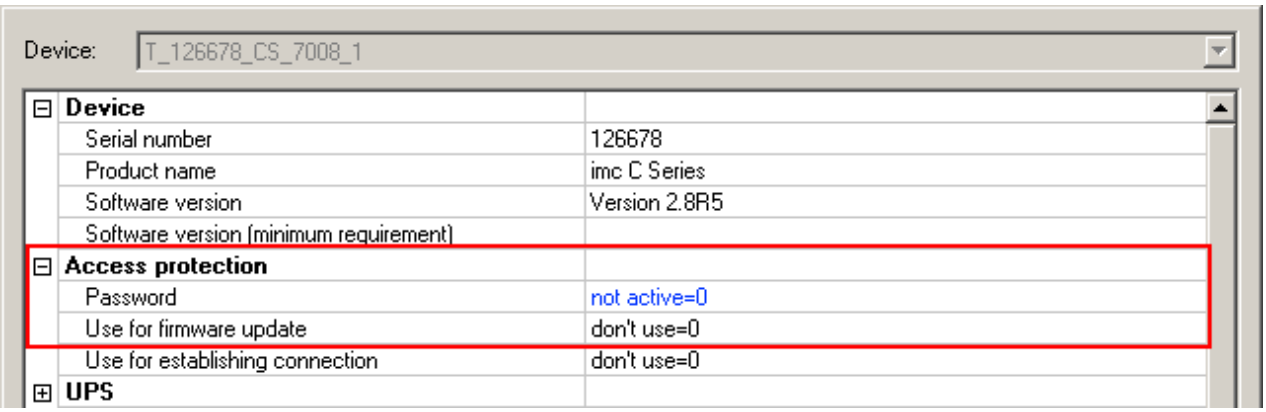
Note

- A message box will appear to notify you if and when the firmware-update has been completed successfully.
- The File menu offers a function for working with the 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*
- The firmware belonging to the amplifiers installed in the device is only updated if there was a firmware change along with the respective imc STUDIO software version.

5.5.2.1.1 Enable and Disable

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 113 dialog (Ribbon *Devices-Configuration* > *Device properties*).

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.



Blocking / Unblocking of the firmware update

Blocking

The firmware update is blocked with a freely defined password.

Unblocking

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

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.

5.6 Configuring Device - Setup pages

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.

	Documentation 148	
	Devices 148	
	Analog channels 159	
	Digital channels 159	
	GPS	
	Variables 159	
	Channel balance 177	
	Triggers 182	
	TEDS 201	
	HiL + Application module 205	

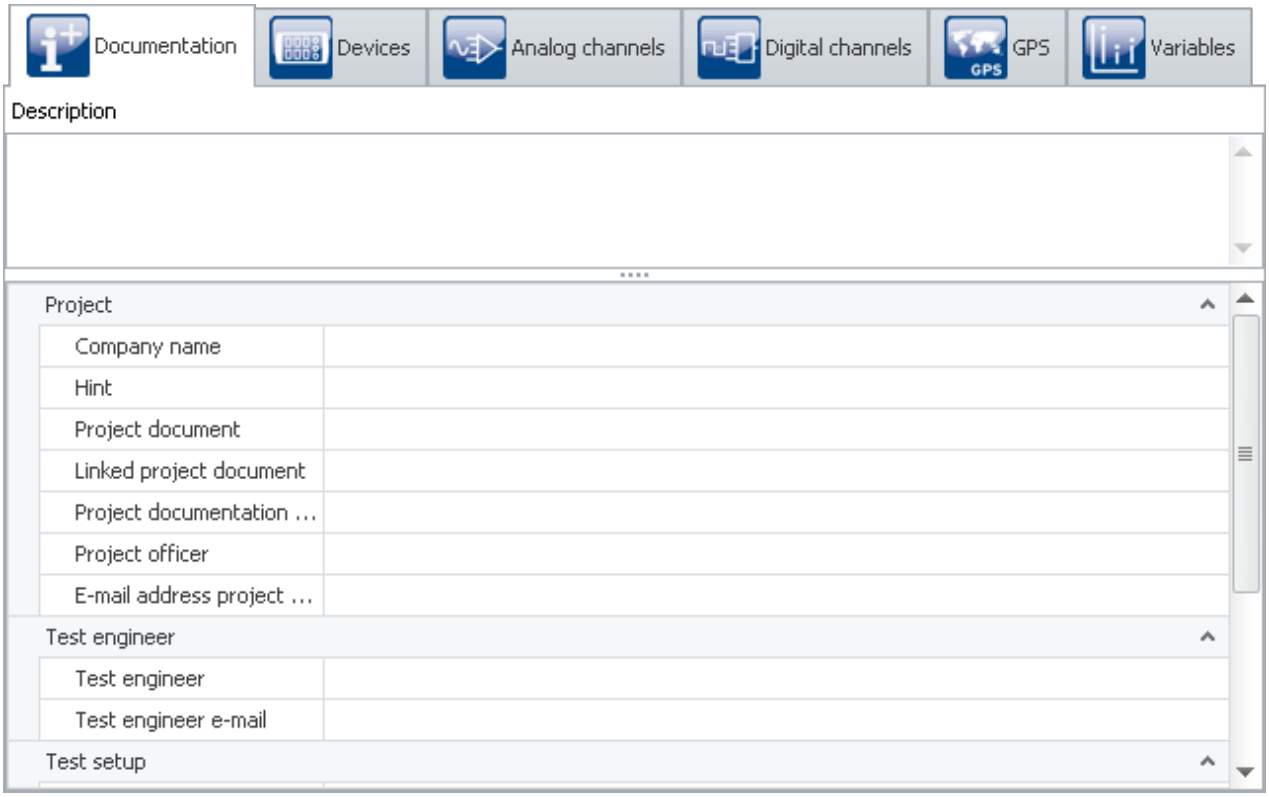
The menu ribbon offers various Assistants and additional dialogs 98 for configuration purposes.

Reference: Advanced Device Functions

For some settings, there are advanced descriptions in the manual: Setup - Advanced Device Functions 337. At the respective locations, you will find special references to the extra descriptions.

5.6.1 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.

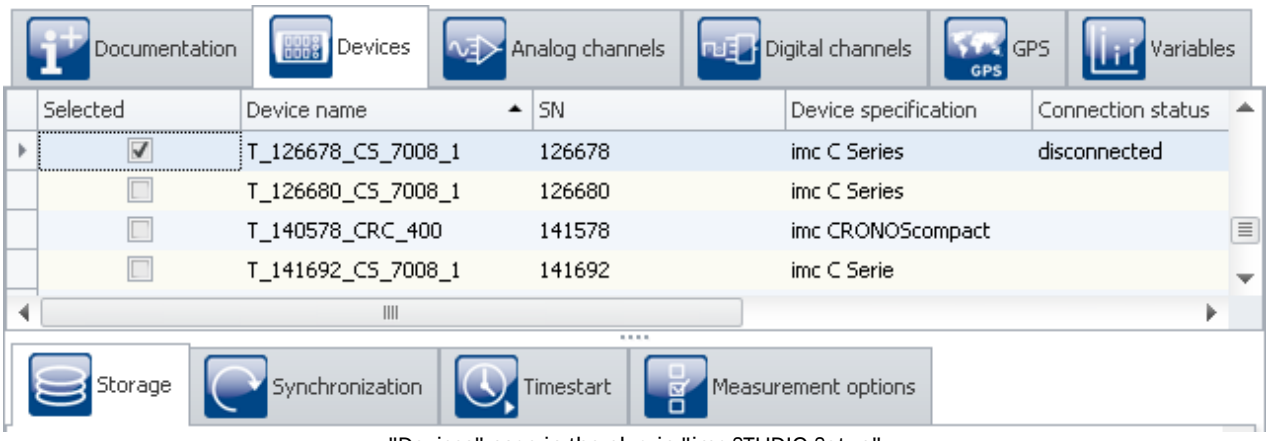
In the table, you can enter information pertinent to the experiment.

The table can be filled with multiple columns (see Configuring columns 133). 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.

5.6.2 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:

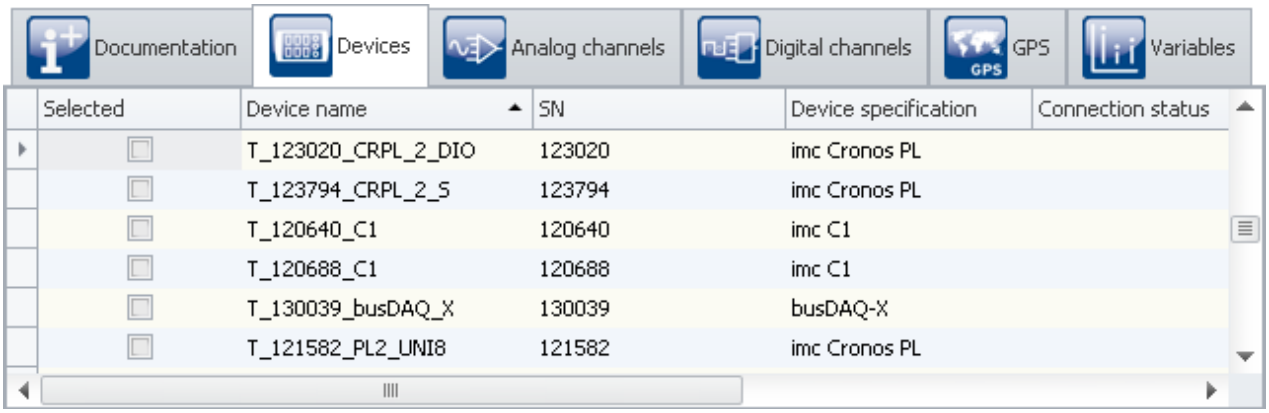
	Page	Description
	Storage 150	Settings for saving measured data in the device and on the PC.
	Synchronization 153	Settings for synchronization to a clock or between multiple measurement devices
	Timestart 154	This enables you to start the measurement at a defined point in time.
	Measurement options 157	Various additional settings.
	Video 158	Settings for a Video device. This page is only visible if an appropriate camera is selected.

See also additional info on:

- Operation and configuration of the tabular display 129 and the dialogs 137
- Configuring and adding columns 133

5.6.2.1 Device Table

In this area, the known 140 devices are displayed in a structured table, as seen in the example image below.



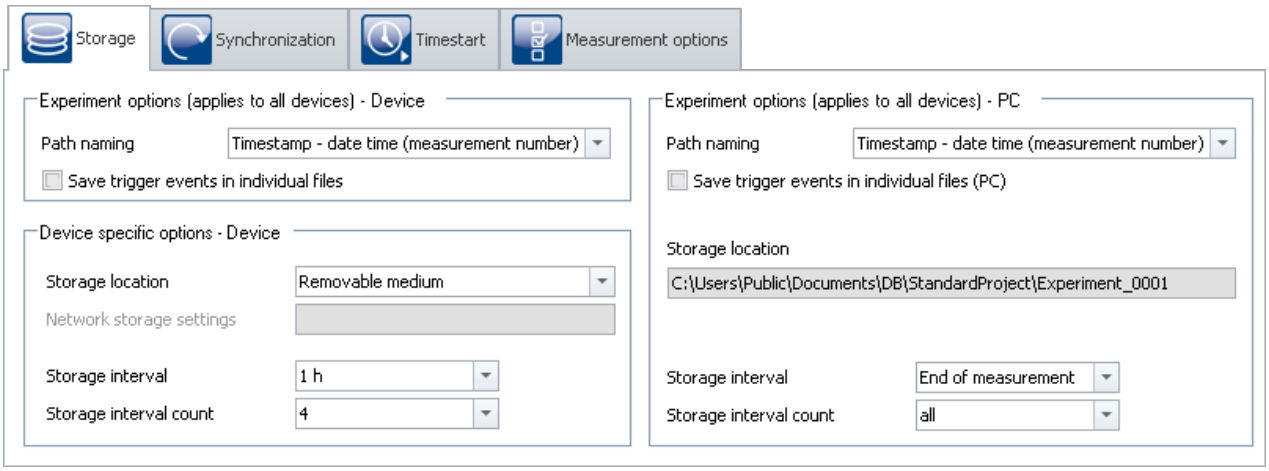
Device-Table (example)

To be able to use a device for the current experiment, you must first find and select 140 it. To adapt the devices' settings, select one or more devices. After that, you either edit the table cells 132 directly or select the respective dialog 137 for the desired parameters.

5.6.2.2 Storage

Reference

This chapter contains a brief introduction to saving channel signal data. More detailed descriptions of all functions are presented in the chapter: *Setup - Advanced Device Functions* > *Storage Options and Directory Structure* 586.



"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
Path naming	Using this parameter, you set the folder structure for the measured data (see: Designation of the Measurement Data Folders 603). • Timestamp - date time (measurement number) The path name is formed from the start time • Continuously numbers
Save trigger events in individual files	Every triggered event is saved to the same file. If multiple events are saved (see: data transfer 174 dialog) it becomes an event-based data set, otherwise, the previous one is deleted. If you select the property Save trigger events in individual files, a new file is created for each event.
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: Save trigger events in individual files

- 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.

Note: Storage interval count

- Once the measurement is done, you will normally obtain one save interval more of data than was set.
- When interval storage is active, all folders are deleted except for the specified number of intervals!
- See the other notes in the chapter: *Setup - Advanced Device Functions* > *Storage Options and Directory Structure* 586 > *Intervall-Speicherung* 598.

5.6.2.2.1 Storage with the device

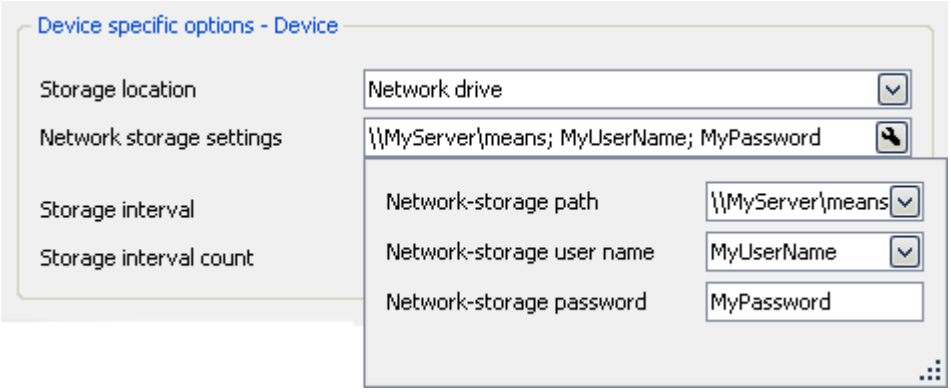
With the storage with the device, you can also specify a network drive as the target. I.e. the device can independently save the measured data to the network without that the program is running on the PC. However, the device must support network data storage, otherwise the option is not available. (Devices as of group 4 88)

In this dialog you only can make settings for the directories and their paths. The settings refer to the device. The storage settings pertaining to each channel are described in the chapter *Configure channels > Data transfer* 174.

Using these options, you can optimize the data storage according to various criteria (performance, data security).

Network storage settings

Only works for devices of group 4 and higher 88.



"Storage" dialog
Network storage settings

The network storage path must be stated in UNC notation (\\server\share\). Ensure that write access to the network storage path (clearance, "share") is possible with the specified user name and password.

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.

Example

You have entered a network path as shown in the image above

\\MyServer\means

The device is named:

imcCronosPL_123368

If you save the experiment under the name:

Test1

and start the measurement, you find the measured data on the path:

\\MyServer\means\imcCronosPL_123368\Test1.

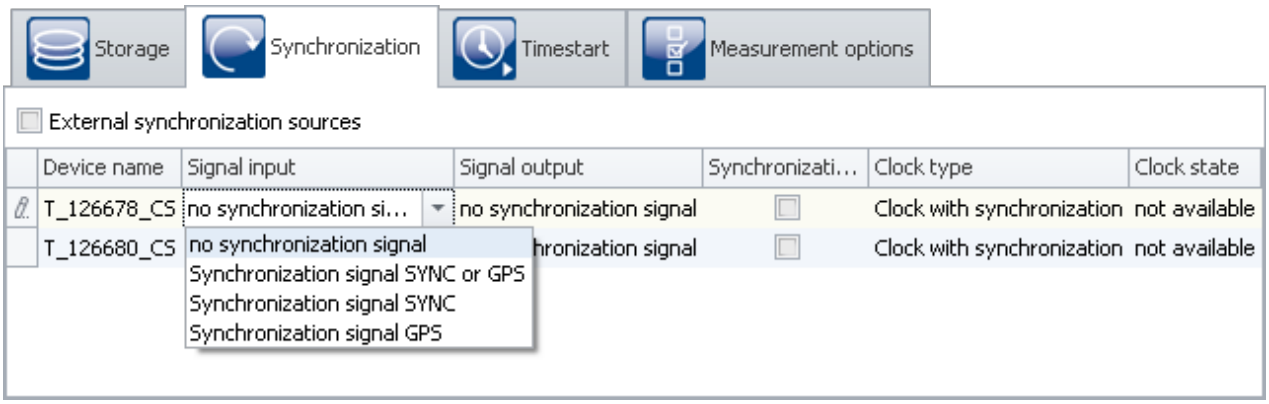
Note on network names

DNS or only IP address possible

5.6.2.3 Synchronization

In this dialog you ca set the synchronization of the internal device clocks. To synchronize the devices to each other, you must connect their SYNC terminals using appropriate cables.

Additionally, these require a synchronization signal. This can be an external clock (SYNC (e.g. DCF77) or GPS) or a device whose internal clock serves as the master. In this case, the master device provides its internal time as a converted DCF77 signal to the other devices (see Synchronization 629).



Parameter	Description
External synchronization sources	The devices are synchronized to an external synchronization source (e.g. DCF77, IRIG-B or GPS). They have their own *signal output* and thus can not be *synchronization signalers*.
Synchronization master	If no external synchronization source is used, you can use one of the devices as the synchronization signaler. This device has DCF77 as the *Signal output*. Devices synchronized to this device must have the same value (SYNC) for the property *Signal input*.
Output	Here, select the synchronization signal (DCF77) to which the other devices are synchronized
Input	Here, select the incoming synchronization signal (GPS and SYNC)

Note

- Channels of synchronized devices have correct absolute time. However, the devices still start serially.
  To prevent this, select
  - synchronous start 157 under Timestart 154
  - an appropriate "Start options 154". E.g. "Automatic timed start".
- If there is no reliable synchronization signal, there are various cases to observe, see the chapter Synchronization 629.

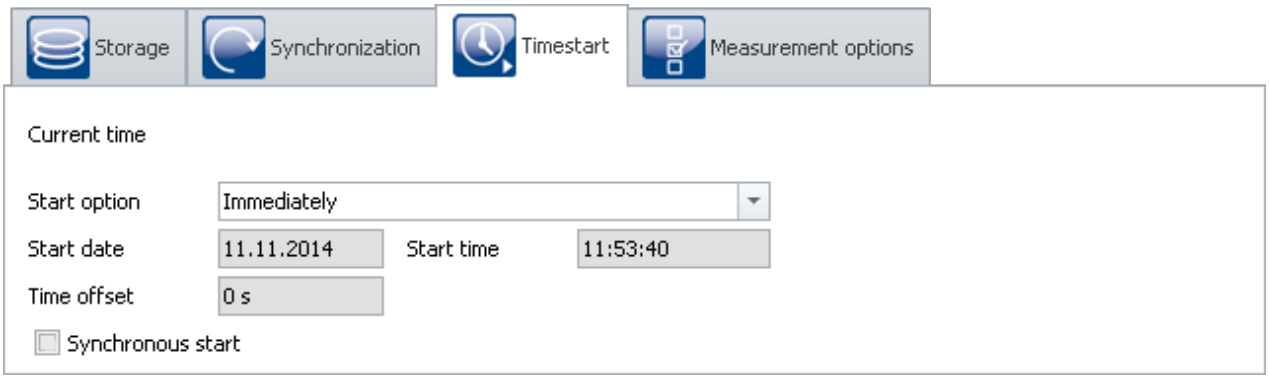
Reference

Regarding synchronization, observe the notes in the manual: *Setup - Advanced Device Functions* > *Synchronization* 629.

5.6.2.4 Timestart

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".



"Start options" are available both for the normal start (in response to the Start button), as well as for the Diskstart 99 (the limitations and conditions are delineated).

Parameter	Description
Start option 154	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 155	Time by which to delay the start in order to ensure that all devices can start simultaneously. Only possible with "Automatic timed start".
Synchronous start	This option ensures that the start only happens once the devices have been synchronized 153. It does not ensure that the devices start simultaneously 157.

5.6.2.4.1 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.

Even if the device has an internal radio clock, it is not used if the start option set is Immediately.

Note

- The time needed for downloading 93 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.
- The start option immediately is not possible with a synchronous start 157 (simultaneous start of all devices).
  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

The measurement starts at the time set.

If the device comes with a internal radio clock, which is activated and synchronized, the start is synchronized with the radio clock. (see: Using the (optional internal) Radio Clock 155)

With a timed start of multiple devices, the software only checks the first device's clock. If any of the other n clocks isn't receiving or wasn't synchronized, no notification is provided.

Note

- At defined time is a setting only appropriate to a one-time start. For each additional start, the clock time must be re-adjusted.
- 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!

An automatic timed start ensures that all devices participating in an experiment start synchronously. Here, the absolute time of the measurement's start isn't important. The precision of synchronization with the DCF-signal is ± 1 ms.

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.

5.6.2.4.2 Using the (optional internal) Radio Clock

Operation

To start measurement with a DCF77 clock, it must be synchronized. This means that it must receive a valid signal from a DCF77-transmitter or via GPS for several minutes. If the device is activated and the antenna on its front panel (antenna jack) is hooked up, it takes at least 4 minutes for the receiver in the device to get synchronized. The device evaluates the DCF-signal by itself. For this purpose, imc STUDIO does not need to be started.

In the dialog Timestart 154,, it is possible to set a time offset, which ensures that all devices are ready by the start time automatically computed (only Start option 154: *Automatic timed start*). If the time you set is too small to fulfill its purpose, a message is posted announcing that time has run out.

Displaying the clock status

In the dialog Synchronization 153, there is a readout of the clock type present and its current status.

Note

- The clock's status is automatically queried in the background.
- If a very weak signal is received (clock reads out "no reception"), check the antenna's orientation. External sources of interference such as monitors can also influence the signal significantly. In this case a suitable location for the antenna must be found by means of experimentation.
- With synchronized devices and the start option 154 Immediately, an offset of up to 1s is still possible! The GPS or DCF77 signal is precise to one second; higher time resolution is only provided within the device, as of the measurement's start. Therefore it's absolutely necessary for all the devices to start on the same full second in order to achieve synchronized measured data, and this is ensured by, for example, with the start option 154 Automatic timed start.

Grounding

The device must be grounded in order to ensure reception of the radio signal.

Devices supplied with AC 230 V voltage are automatically grounded via the protective ground conductor.

Interrupted reception

If the clock signal reception is interrupted, the clock pulse is simulated for up to 1000 h.

The maximum deviation from the true time is 0.012 s per 100 min (= ± 2ppm), provided that the device is not exposed to dramatic temperature changes.

If a clock pulse is lost for up to 2 h, the time is slowly corrected at a maximum rate of 10 µs/s. The internal times are also adjusted accordingly.

If a clock pulse is lost for 2 h to 1000 h, the time is not slowly corrected. Instead, it is corrected immediately by adding or subtracting the necessary number of pulses. The internal times are no longer correct and the measurement therefore proceeds asynchronously.

If the deviation is less than 0.5 s, the device time is slowly adjusted and will be correct after a maximum of 5 min. The measurement continues synchronously.

If the deviation is greater than 0.5 s, then the time is simply rounded to the nearest second. The device clock synchronizes with the second pulse of the radio clock receiver, but a time offset is added to the measurement data. This cannot be regulated by the unit automatically. To eliminate the time offset, switch the device off and on again.

5.6.2.4.3 Synchronous start

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.

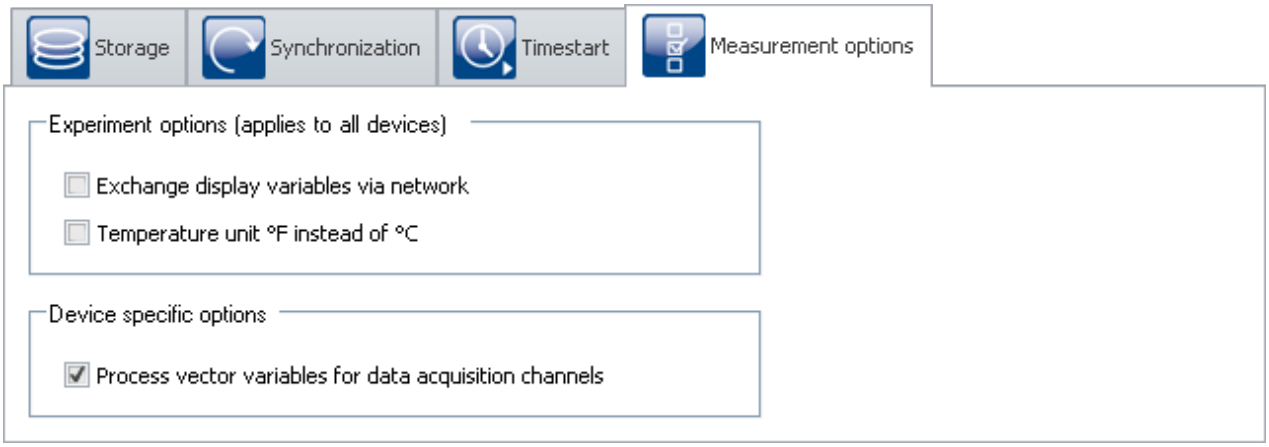
- The option: "Synchronous start" must be activated.
- Appropriate "Start options 154" must be selected. E.g. "Automatic timed start".
- For all devices, a clock signal for the signal input 153 must be selected.

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 - Advanced Device Functions > *Synchronization* 629 > *What happens when?* 655

5.6.2.5 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.

Note

- The system checks every 10 ms for changes to the display variables. For this reason, rapid changes (on channels with high sampling rates) are not taken into account.
- Checking and sending of the display variables burdens the measurement device's and the Ethernet's resources. Therefore, only only activate this function if you want to do operations on two devices' data jointly.

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 Field busses 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 data acquisition channels, process vector variables 161 are created. This option is activated by default.

Note

Even when the process vector variables are not used, they demand a portion of the device's available power. For this reason, less than the entire aggregate sampling rate of your device is actually usable.

For each device, you can deactivated process vector variables for the data acquisition channels. Remove the checkmark from the selected 130 device. This makes the entire aggregate sampling rate available. Additional process vector variables (e.g. GPS data) are not affected.

5.6.2.6 Video

This dialog is only shown when video devices are connected.

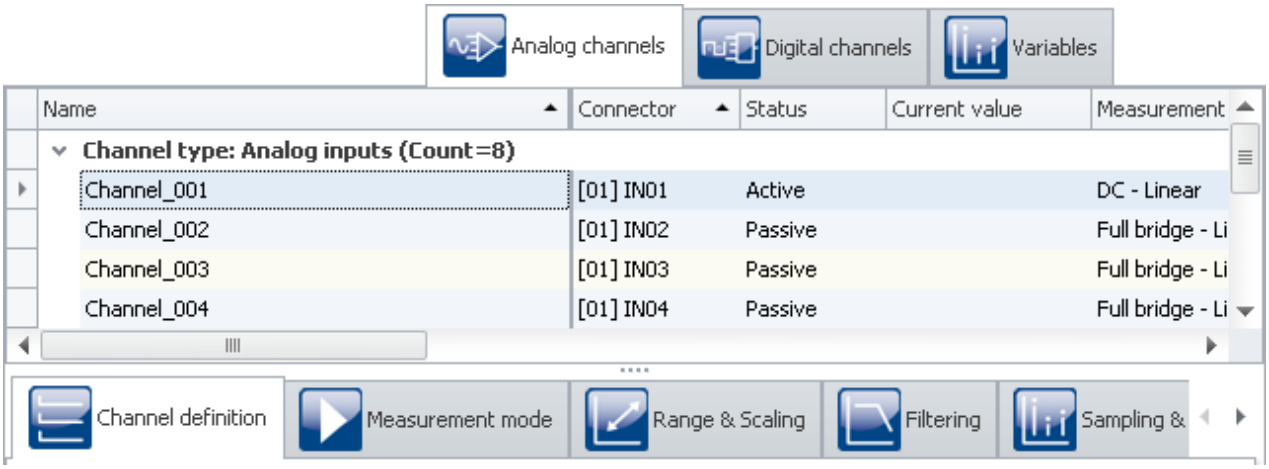
In this dialog, you can adjust the video settings. What settings are available depends on the device.

Reference

For a detailed description, see the manual: Video 1292.

5.6.3 Configuring Channels and Variables

On the pages Analog channels, Digital channels and Variables you can set all channel-specific parameters.



The three 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 140. 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:

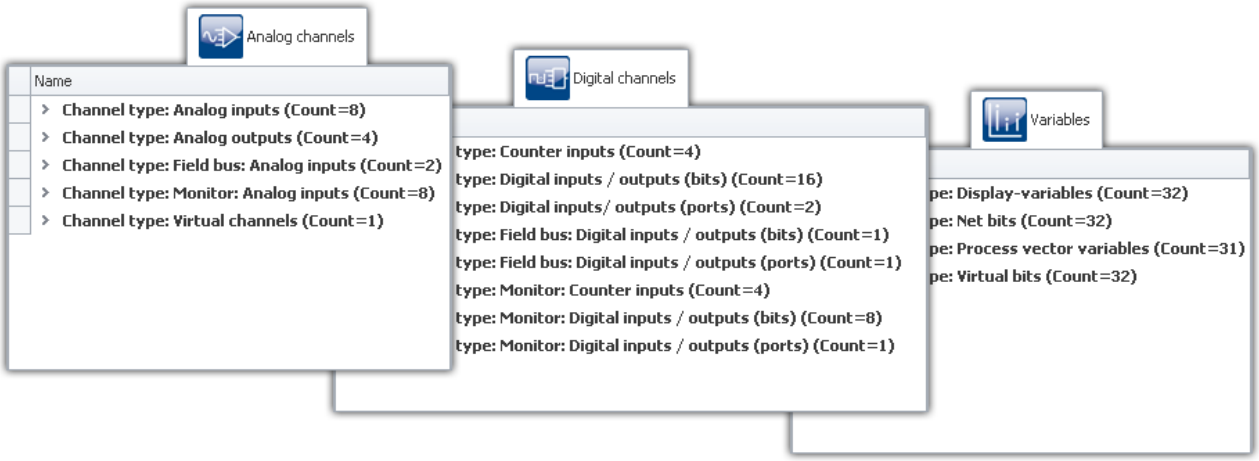
	Page	Description
	Channel definition 163	
	Measurement mode 164	
	Range & Scaling 165	
	Strain gauges 167	Only for devices having strain gauge channels
	Encoder 168	Only for devices having incremental encoder
	Video 171	Only for video devices
	Filtering 171	
	Sampling & Preprocessing 172	
	Data transfer 174	
	Histogram / Rainflow 175	Only certain virtual channels. E.g. Histogram or Rainflow

See also additional info on:

- Operation and configuration of the tabular display 129 and the dialogs 137
- Configuring and adding columns 133

5.6.3.1 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 by clicking on the plus symbol (⊞). Then select one or more channels. Subsequently, you can either edit table cells 132 directly, or open the dialog 137 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 inputs	Analog data acquisition channels
Field bus: Analog inputs	CAN-interface required (CAN, LIN, J1587 etc.)
Virtual channels	Channels created with imc Online FAMOS
Monitor: <xyz>	Digital copies of the input channels which have the same name. By creating monitor channels, it's 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 encoder inputs	ENC4 or INK4 required
Field bus: Digital inputs / outputs (bits and ports)	CAN-interface required (CAN, LIN, J1587 etc.)
Monitor: <xyz>	Digital copies of the input channels which have the same name. By creating monitor channels, it's possible to show slowly-sampled signals recorded before the trigger event.

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.
Net bits 162	Up to 32 net bits are available for Ethernet devices. Events can be assigned to net bits and used as triggers in an Ethernet.
Process vector 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.

5.6.3.1.1 imc CRONOSflex Modules

The device software lists the imc CRONOS*flex* (CRFX) 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 amplifier channels. Channels, recorded with Flex 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 CRFX modules connected is provided in the dialog Module properties 117. A CRFX 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 117.

Notes on the 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.
- Up to 16 digital modules can have an address within a system. 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 *Digital CRFX Modules DIO, Pulse Counter and DAC*.

5.6.3.1.2 Net 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 Net Bits. One way to think of them is as 32 hard-wired digital lines running parallel to the Ethernet, which connect all imc STUDIO mutually. Thus the device has an Ethernet-DIO port, in addition to the physical DIO and the virtual DIO. These digital inputs and outputs can be used for triggering just like the unit's physical DIO-bits.

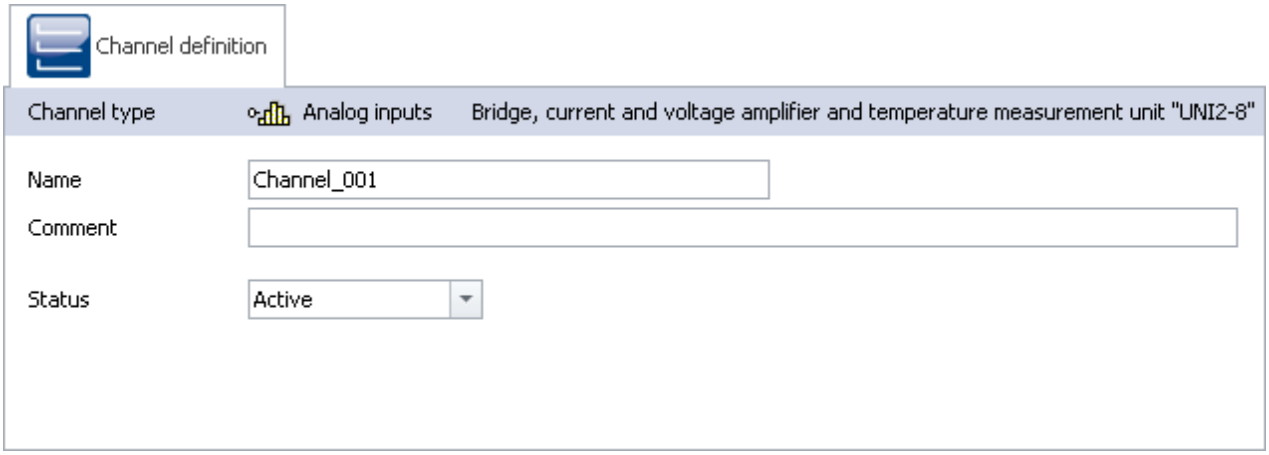
In this context, note the following:

- Net bits are only available with Ethernet devices.
- Net bits only take effect within the scope of one protocol: i.e. NetBEUI-devices do not trigger TCP/IP-devices and vice-versa.
- The net 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

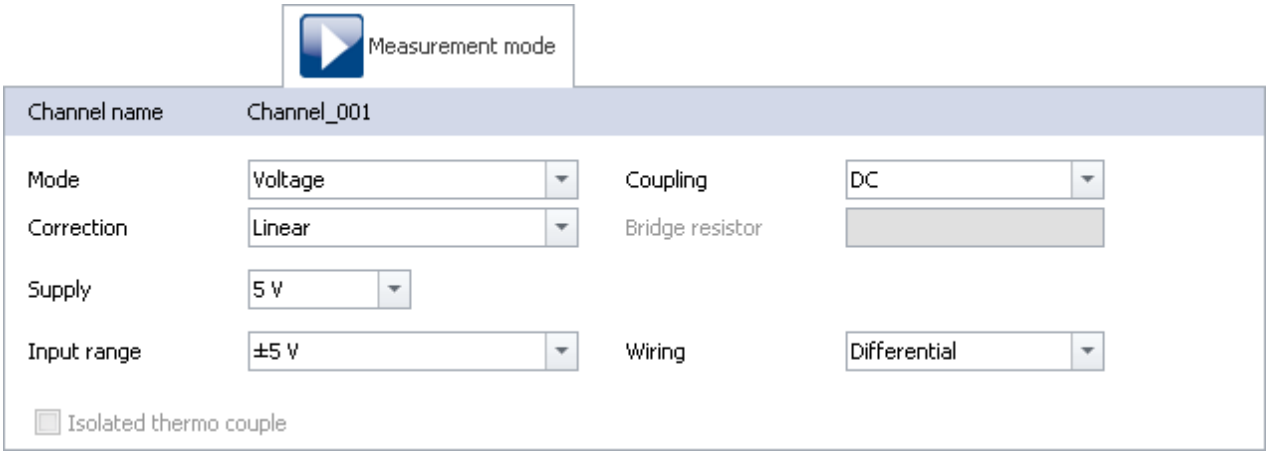
When a measurement is prepared (Downloaded), all bits, including the Ethernet bits, are reset. This also affects devices on the network which are currently performing measurements. Any Ethernet bits which had been set in the process of the running measurements will be reset when any device in the network is prepared

5.6.3.2 Channel definition



Parameter	Description
Name	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. Note: If a channel name begins with a number or contains special characters such as "\?+!" or a space, in imc Online FAMOS the channel name must be set into curly braces. Example: Channel name = "123 My channel 100% display" in imc Online FAMOS = "{123 My channel 100% display}"
Comment	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	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.

5.6.3.3 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.

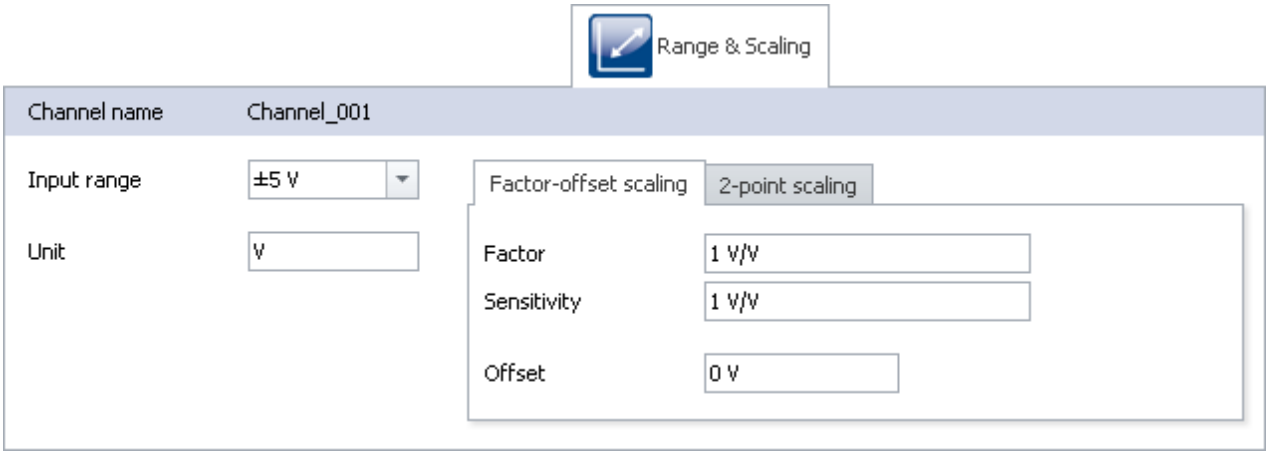
Parameter	Description
Mode	Basic settings for the measurement types: *Voltage*, *Current* or *Strain gauge* (Strain gauge: for further settings, a new dialog appears: Strain Gauge 167).
Coupling	Depending on the amplifier and on the mode set, DC- or bridge measurement settings are possible. If the Strain Gauge mode is selected, a variety of possible applications is listed.
Coupling is DC	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	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 configuration 182) Also, with the bridge resistor, it is possible to determine the cable resistance even if no Sense line is connected.
Correction	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 122 is used [1] For amplifiers produced after November 2012.
Supply	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) 176

Parameter	Description
Input range	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.
Wiring	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.
Isolated thermo couple	Default is isolated thermocouple. When using none isolated thermocouples, that option has to be deactivated, to avoid ground loops.

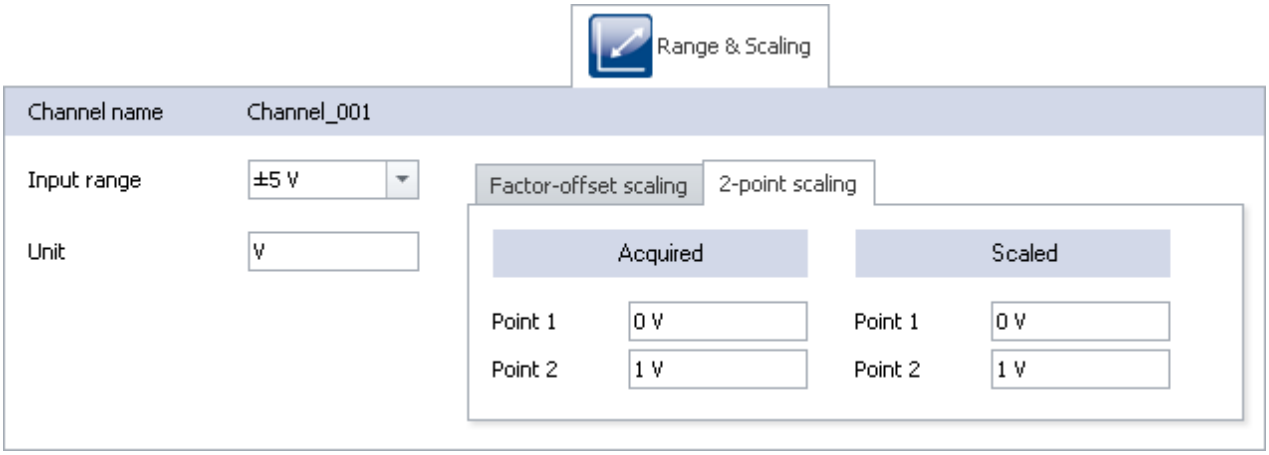
Note

The condition for temperature measurement is that the amplifier is working on mode: voltage (DC).

5.6.3.4 Range & Scaling



Factor-offset scaling



Two-point scaling

Parameter	Description
Input range	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.
Unit	Unit of measurement values. Select SI units without milli, micro etc., which the curve window with do automatically.
Factor	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	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	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 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 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 (see: Two-point scaling 181). By means of this procedure, all uncertainty tolerances are taken into account in the scaling.

Notes on balancing with setted offset

If the sensor is balanced 178 subsequent to making the entries for the gain and offset , the offset entered remains in force, which means that there is no apparent effect. This ensures that any subsequently entered offset 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 178.

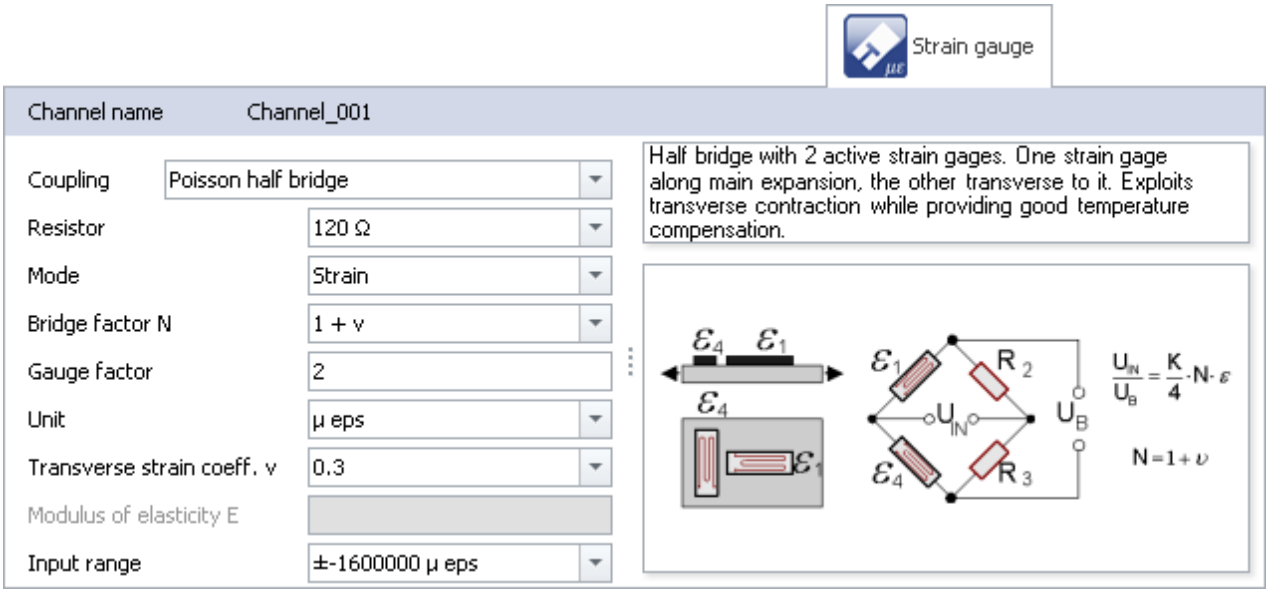
Alternative 1: Two-point characteristic curves can also be applied by means of imc SENSORS (toll window: Sensors). Subsequently, the offset is compensated by means of the Balancing function 178.

Alternative 2: If it is possible to specify two known data points, we recommend Two-point calibration 181, since in this case, the entire measurement chain is taken into account.

5.6.3.5 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.



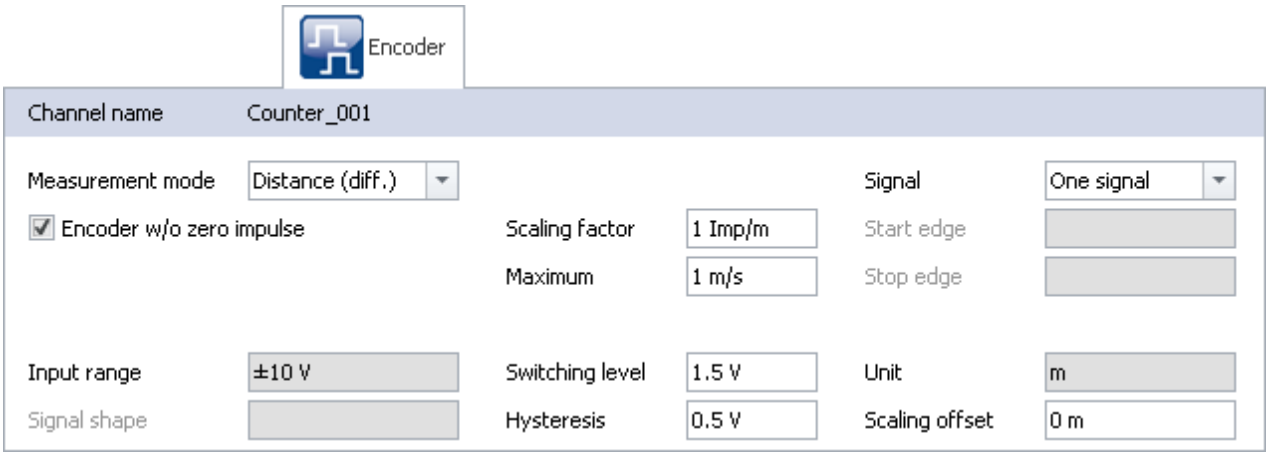
Parameter	Description
Coupling	Lists various usage possibilities. Corresponding descriptions and sketches of the strain gauge are displayed. The strain gauge is pre-set to conform to the coupling-selection. It is also still possible to make appropriate settings.
Resistor (Bridge resistor)	The bridge 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 configuration 182) Also, with the bridge resistor, it is possible to determine the cable resistance even if no Sense line is connected.

Parameter	Description
Mode Bridge mode	*Strain* or *Stress*
Bridge factor N	The available selection depends on the coupling.
Gauge factor	Depends on the part's material
Unit	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	Depends on the part's material
Modulus of elasticity E	Depends on the part's material
Input range	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.

5.6.3.6 Encoder

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
Measurement mode	Measurement types. See Measurement mode - Overview 169
Encoder without zero impulse	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	One signal counters or Two signal counters • One signal counters 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 counters return two impulse sequences, mutually offset by 90°. in this way, the encoder can distinguish in which direction the sensor is moving. This type s especially suited to measurements of velocity, distance and angle.

Parameter	Description
Scaling factor	How many pulses represent one physical unit depends on the measurement mode set.
Maximum	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 two-signal counters.
Start edge	Start edge on time and pulse measurement. Depends on the measurement mode 169.
Stop edge	Stop edge on time and pulse measurement. Depends on the measurement mode 169.
Input range	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	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	Here you set the lower voltage limit interpreted as the HIGH level.
Hysteresis	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 + Hysteresis/2) LOW = Signal < (Switching level - Hysteresis/2)
Unit	Measurement value unit. Depends on the measurement mode set.
Scaling offset	Here you can enter an offset.

Note on the low-pass filter

Set the low-pass filter accordingly (see Filtering 171). 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.

5.6.3.6.1 Measurement mode - Overview

Reference

More information about the measurement modes are described in the chapter: *Information and Tips* > *Incremental Encoders* 229 or in the respective device manual.

As a rule, what type of measurement is performed is determined by your setting for the Measurement mode:

Quantities derived from event-counting:	Quantities derived from time measurements:	Combined measurements
• Events • Distance(differential) • Angle (differential) • Distance (abs.) • Angle (abs.)	• Time • Pulse time (phase-difference)	• Frequency • Speed • RPM

Event counting

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

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.

Note

The mode Pulse time depends on the sampling interval. The entry only appears for some amplifiers if the appropriate sampling interval is set (≤ 1ms or ≤ 100µs; see device manual).

Combined measurement

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 two-signal 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.

5.6.3.7 Video

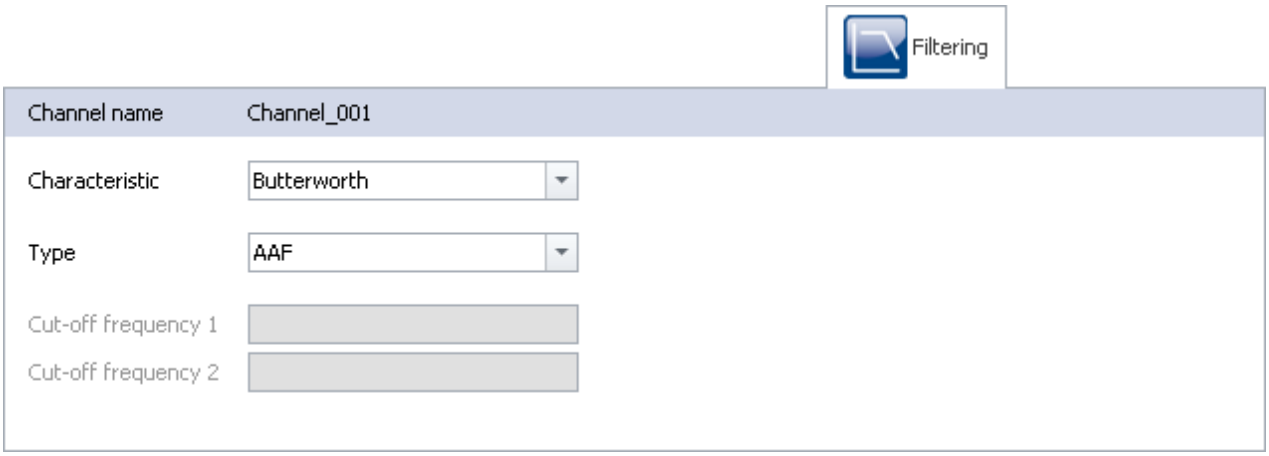
Only enabled when video devices are connected.

Reference

For a detailed description, see the manual: Video 1292.

5.6.3.8 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.

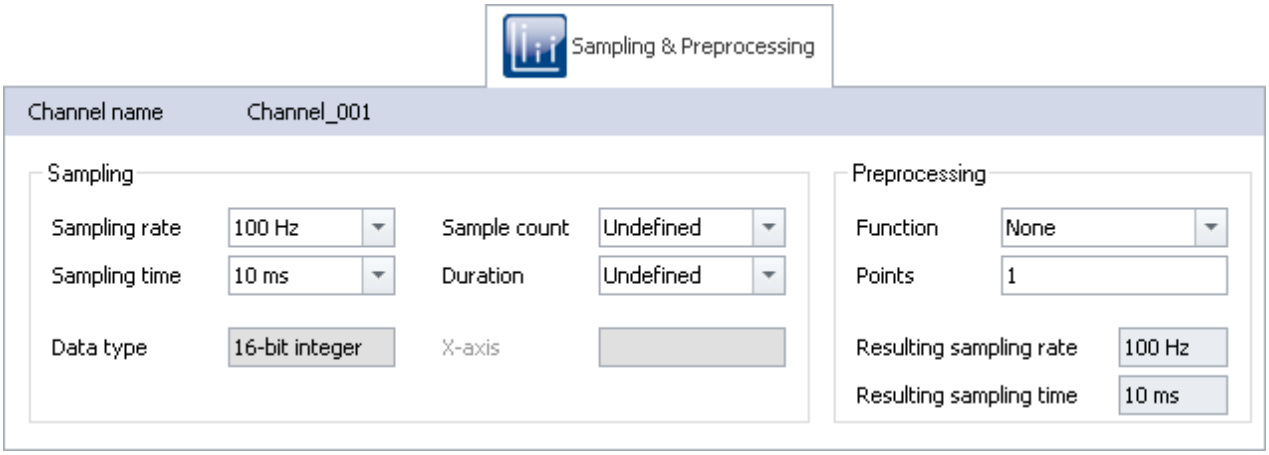
Along with filter types to select by their cutoff frequencies, there is also an anti-aliasing filter (AAF). This is a low-pass filter whose cutoff frequency is set to comply with the requirements of the Sampling Theorem. Thus, frequency components above ½ of the sampling frequency are suppressed.

The default setting is AAF with Butterworth characteristics.

Reference

For a more detailed description of filtering techniques see the device manual.

5.6.3.9 Sampling & Preprocessing



Sampling & Preprocessing

The availability and selection of settings depends on the channel type.

Parameter	Description
Sampling rate	Setting for the sampling time. Sampling time = 1/Sampling rate Active channels can use a maximum of two different sampling times/rates per device.
Sampling time	Setting for the sampling rate. Sampling rate = 1/Sampling time Active channels can use a maximum of two different sampling times/rates per device.
Sample count	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	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.
Function	Simple processing of the measured values. In contrast to the corresponding functions in imc Online FAMOS, it is not possible here to set processing of moving segments of data. Different functions depending on the channel type: • Arithmetic Mean • Minimum • Maximum • RMS • Min-Max (by turns, the minimum and the maximum are returned) • Reduction Reduction: Data reduction for digital inputs: 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.

Parameter	Description
Points	The number of data points (measured values) on which the function is to be carried out (not accounting for reduction). The point count is the same as the reduction factor.
Resulting Sampling rate	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 count of values from the sampling rate/time set and the amount of points. Resulting Sampling time = Sampling time*Points
Measurement mode	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	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 607 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 7 88 and higher, the maximum aggregate sampling rate achievable via EtherCAT is 2000 kHz.

Settings for field bus channels

The data of these channels are captured via a special Field-bus interface. The necessary definitions are made by use of the Field-bus Assistant. Field-bus channels are also included in the trigger machine.

Channel type	Description
Field bus: Analog inputs	The Field-bus channels carry data acquired by sensors which are connected to a field-bus such as a CAN-Bus.
Field bus: Digital inputs / outputs	The Field-bus channels carry digital data acquired by sensors which are connected to a field-bus.

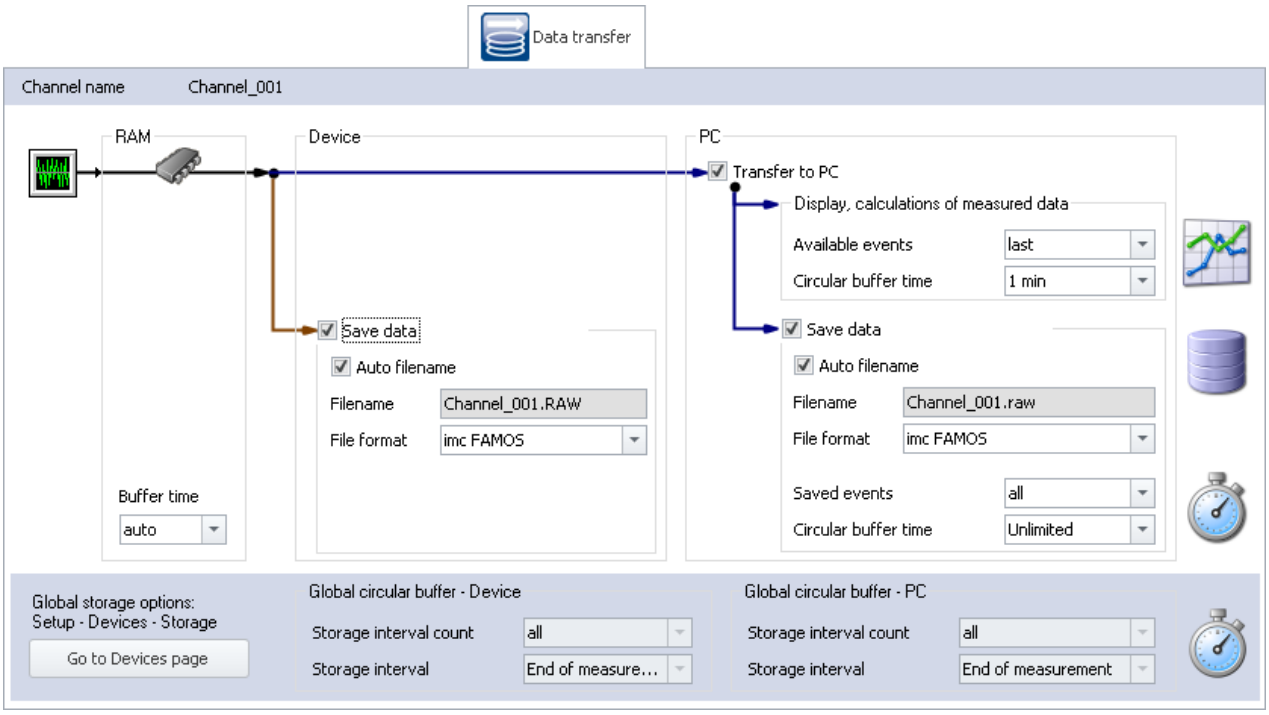
If Field-bus channels are selected in the Channel table (assuming the device is configured correspondingly), there are additional parameters to set:

Parameter	Description
X-axis	• *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 field-bus. 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	If the parameter X-axis is set to *Time stamp*, the duration of the measurement depends only on the setting in Duration.
Assignment	The data can additionally be assigned to a Display variable (only for analog Field-bus inputs).

5.6.3.10 Data transfer

Reference

This chapter contains a brief introduction to saving channel signal data. More detailed descriptions of all functions are presented in the manual:
Setup - Advanced Device Functions > *Storage Options and Directory Structure* 586.



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 150.

Each of the regions is set separately.

Note

Per interval or measurement, one file is created per channel, which contains the respective measurement files

Available / Saved events

With this setting, you determine whether all triggered events are to be saved or only the last one.

If all events are saved, the result is an event-based data set. For storage of the measured data on the PC, each event can be saved to a separate file. (see Device settings/Storage 150)

Circular buffer time

Here you set how long the maximum time segment is. Enter a time in seconds. You can also make a formatted entry. To do this, append one of the following suffixes to the numerical value:

s	seconds
min	minutes
h	hours
d	days

Example

"3 min" = 3 minutes, "2 h" = 2 hours

If you enter *0*, circular buffer operation is deactivated and all data are saved.

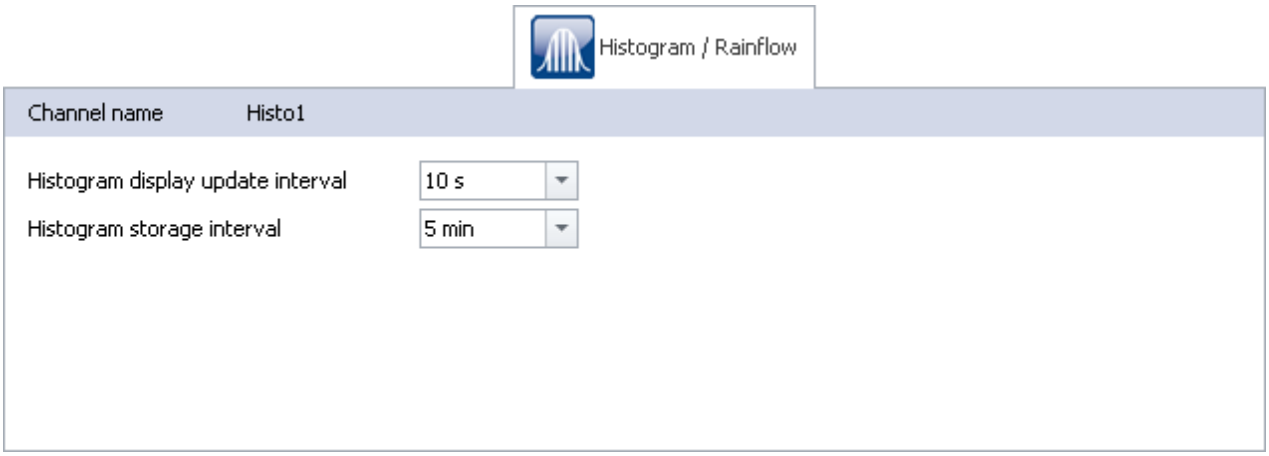
Warning

- If you set the storage duration to unlimited (0), the curve windows use unlimited amounts of RAM
- This settings is absolutely not suitable for long-term measurements

5.6.3.11 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
Histogram display update interval	Time interval for updating data which are derived from counting functions
Histogram storage interval	Time interval for saving data which are derived from counting functions

5.6.3.12 Additional Parameters

Additional parameters are available, which are either already displayed in the channel table, or which you can add manually.

Category: Channel

Parameter	Description
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"
Channel number	Enumerated 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] IN04 | [01] IN05 | [01] IN06 | [01] IN07 | [01] IN08 | [02] IN01 | [02] IN02 | [02] IN03 | [02] IN04 | [02] IN05 | [02] IN06 | [02] IN07 | [02] IN08 | [03] IN01 | [03] IN02 IN001 | IN002 | IN003 | IN004 | IN005 | IN006 | IN007 | IN008 | IN009 | IN010 | IN011 | IN012 | IN013 | IN014 | IN015 | IN016 | IN017 | IN018

Category: Module

Parameter	Description
Bridge supply	Bridge supply (On/Off) [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.

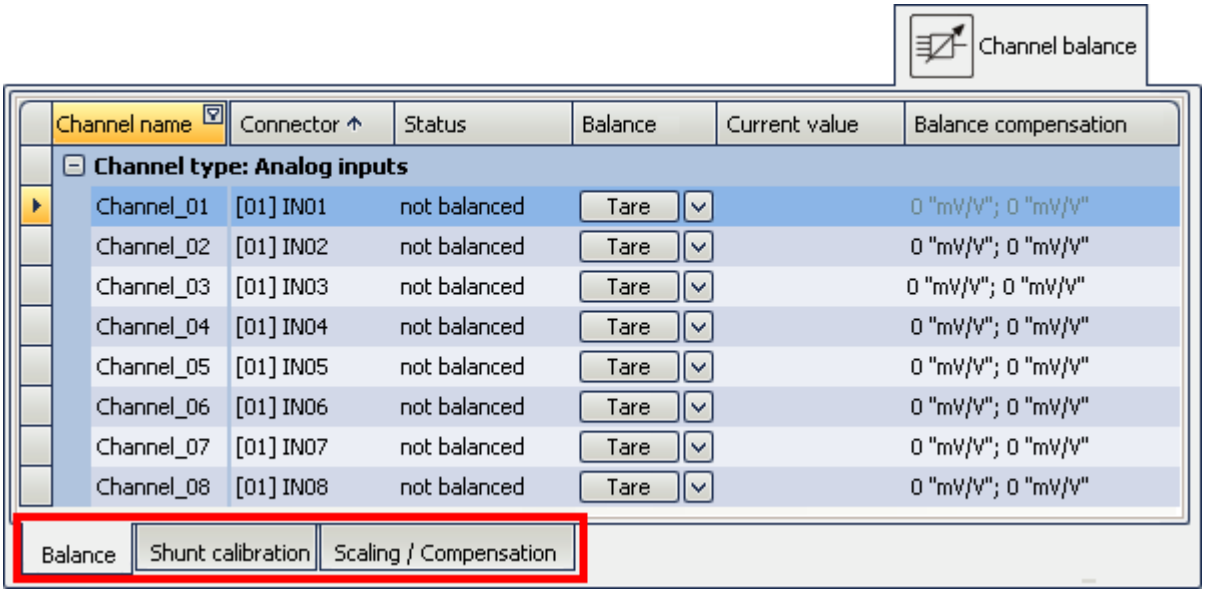
Category: Other

Parameter	Description
Current value	Current value [CurrentValueDisplay] Displays the current measurement value (value of the associated process vector variable). The value is available after initialization (Download) of the measurement. Additionally, color-coding indicates in which region of the measurement range the value is located. If no value is displayed, either there are no process vector variables available or the measurement have not yet been initialized. If the value is displayed in gray, channel settings have been changed since the initialization.

5.6.4 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.

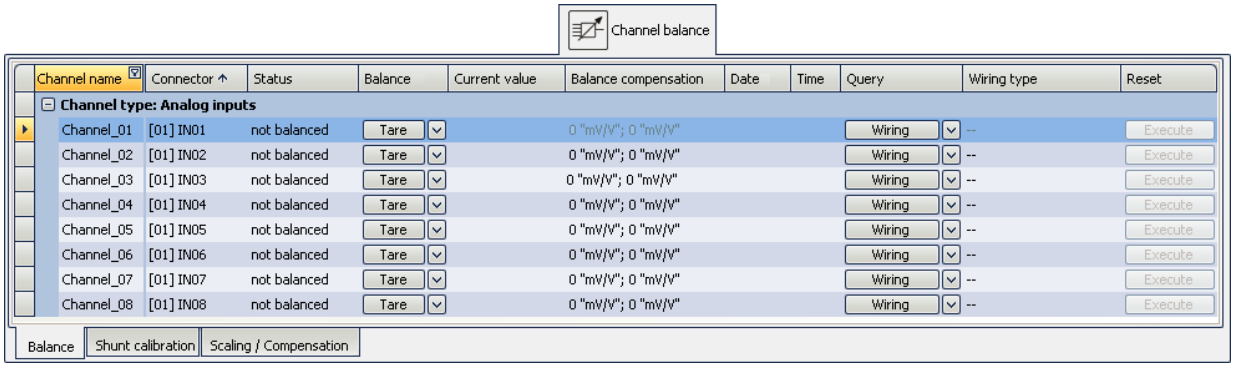


Balance 178	• Tare • Bridge • Query wiring • Reset charge
Shunt calibration 180	• Shunt calibration
Scaling / Compensation 180	• Two-point scaling • Cable compensation

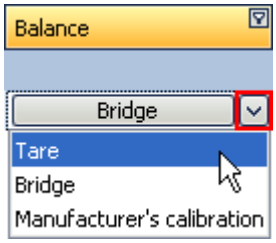
5.6.4.1 Balance - Tare and Bridge

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.
Manufacturer's calibration	Manufacturer's calibration resets all selected channels to the original imc-factory calibration setting

Note

- If a change is made to the input range or to the measurement mode (Voltage - Current - Strain gauge), then any previously performed adjustment or taring is reset.
- However, changing the Y-scaling has no effect on balancing or taring.
  Please observe the notes regarding Offset 167.
- 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. From firmware version 2.8R7 onward, it is possible to perform taring or bridge balancing which a measurement is running. Stopping the measurement is not necessary.

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.

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

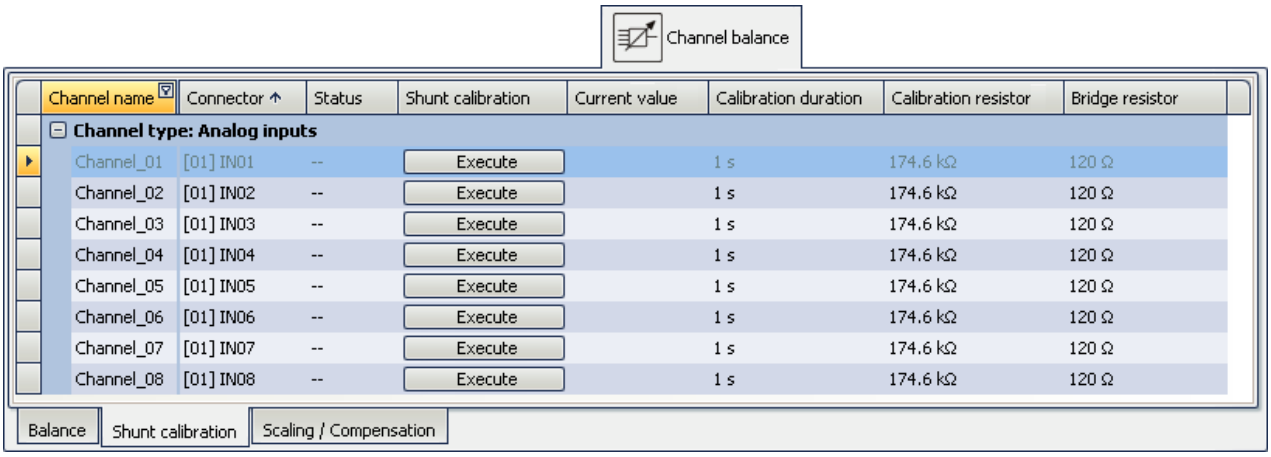
Resetting the charge

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.

5.6.4.2 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.

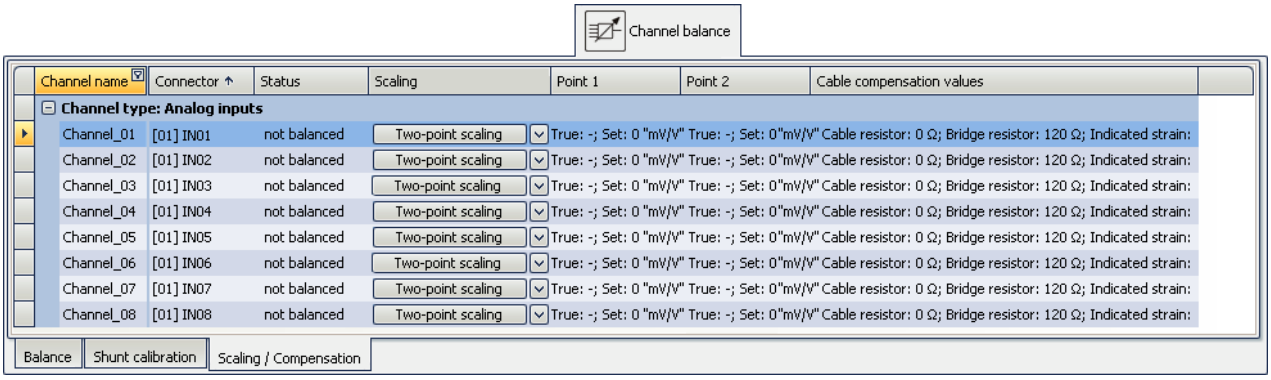
The duration to be set is the minimum duration. It is much longer and depends on the system's load.

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.

5.6.4.3 Scaling / Compensation

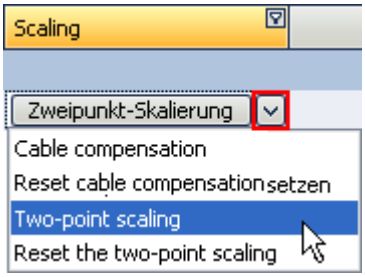
This page enables Two-point scaling 181 and cable compensation without Sense line 182 to be performed.



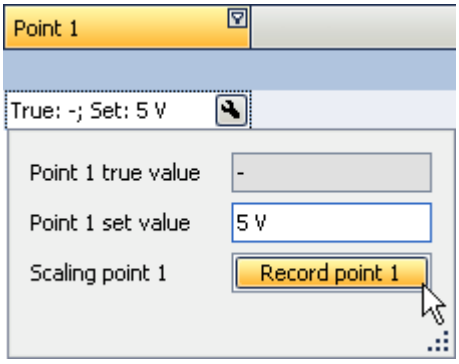
5.6.4.3.1 Two-point scaling

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*.

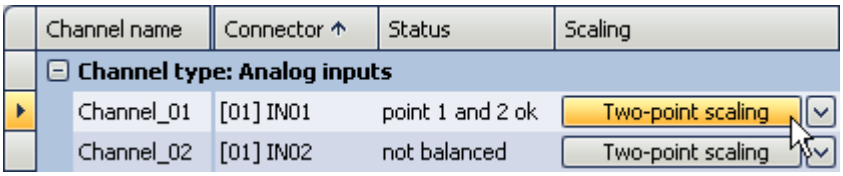


In the columns *Point 1/2*, set the respective target values. Record Point 1 and Point 2 in succession.



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 165's page *Channels (Analog or Digital)*. A description is provided in the FAQ 804.

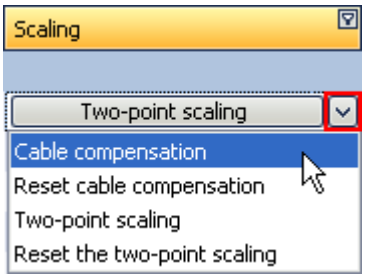
Resetting two-point scaling

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.

5.6.4.3.2 Cable compensation without sense configuration

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.

5.6.4.4 Balance or shunt calibration via Display and imc Online FAMOS

Besides operation by means of the device software, it is also possible to balance or shunt calibration as follows:

- By means of a button on the imc Display which is linked to the Adjustment/Shunt calibration 676.
- By means of the imc Online FAMOS function `DisplaySetButton` 1472, which carries out an balance/shunt calibration if the button is linked with this function 676.

5.6.5 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.

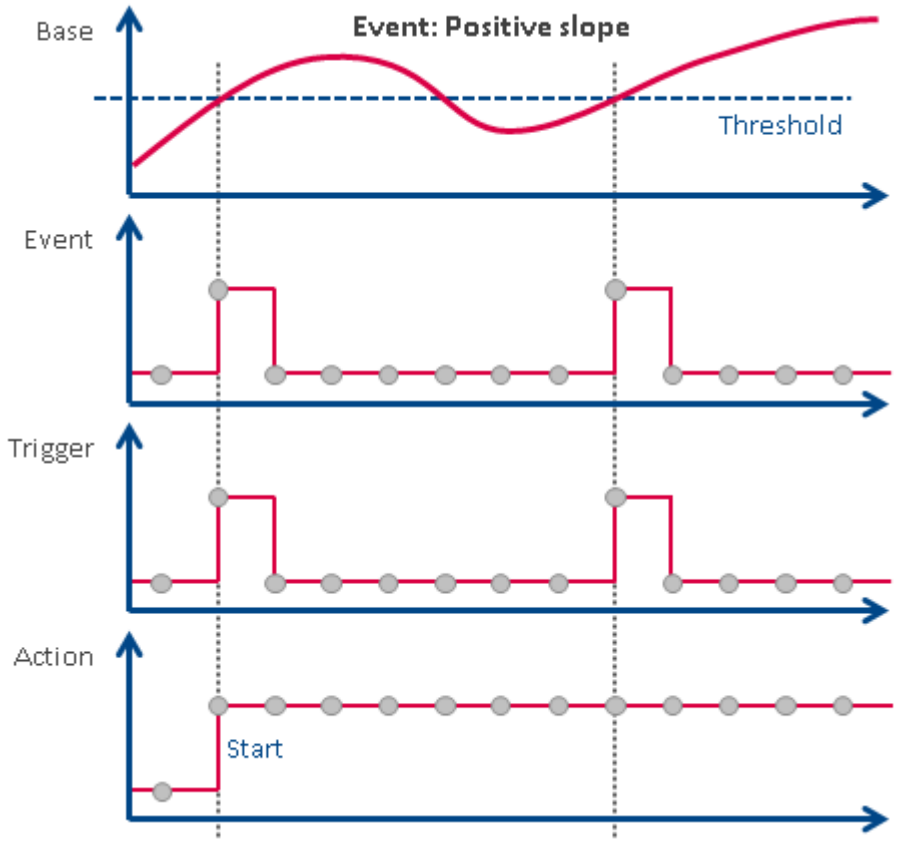
Many kinds of measurements are intended to start only once certain events occur. This applies, for instance, to measurements for which it is known in advance when measurement is to take place.

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.

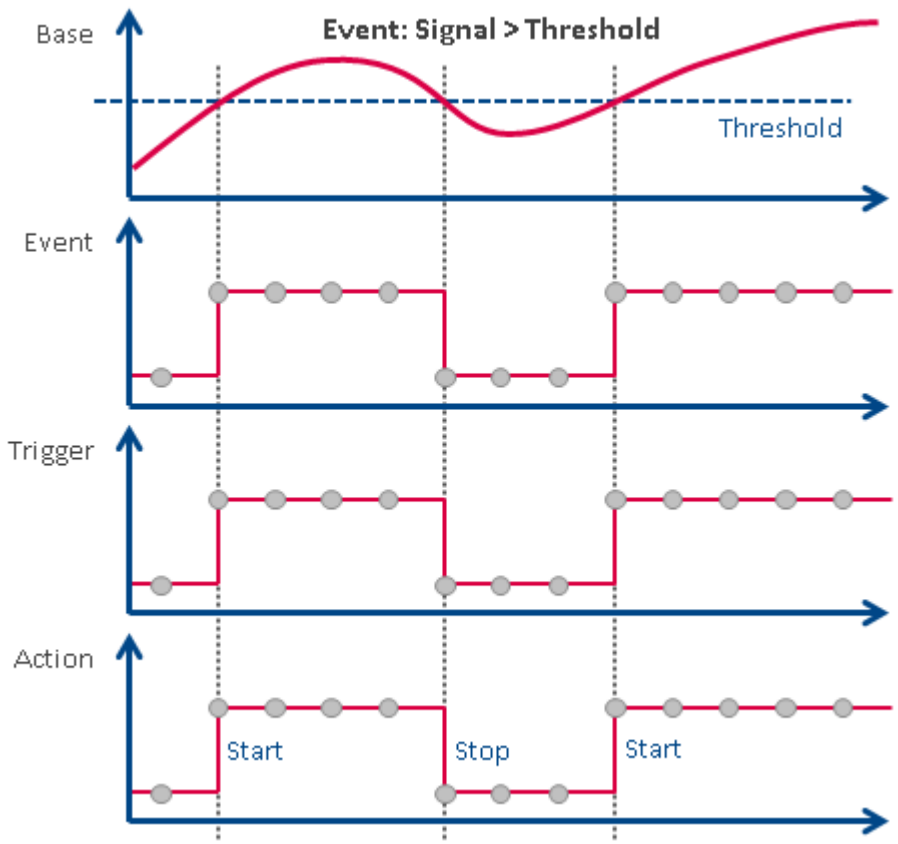
It is possible to continually monitor measurement channels and only to capture data in response to certain events. For this purpose triggers are used which start and stop channels based on the occurrence of certain events.

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

5.6.5.1 Overview

To begin, the "untriggered" measurement are explained. Afterwards, a description of the imc devices' Trigger-Machine 185 is presented:

- Starting and stopping by trigger. What to make note of (see: Sources and events 187)
- Special aspects of repeat measurements (see: Multiple triggering (Multi-Shot operation) 195)
- Triggering with virtual channels (see: Virtual Channels from imc Online FAMOS 198)
- Use of triggers in imc Online FAMOS (see: Online-Trigger 199)
- Slow background measurement with rapid triggering in the foreground (see: Example: Background Measurements 200)
- Event-driven setting of digital outputs (see: Example: Setting event-driven digital outputs 201)
- Measurement Procedure 199

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 ( ).

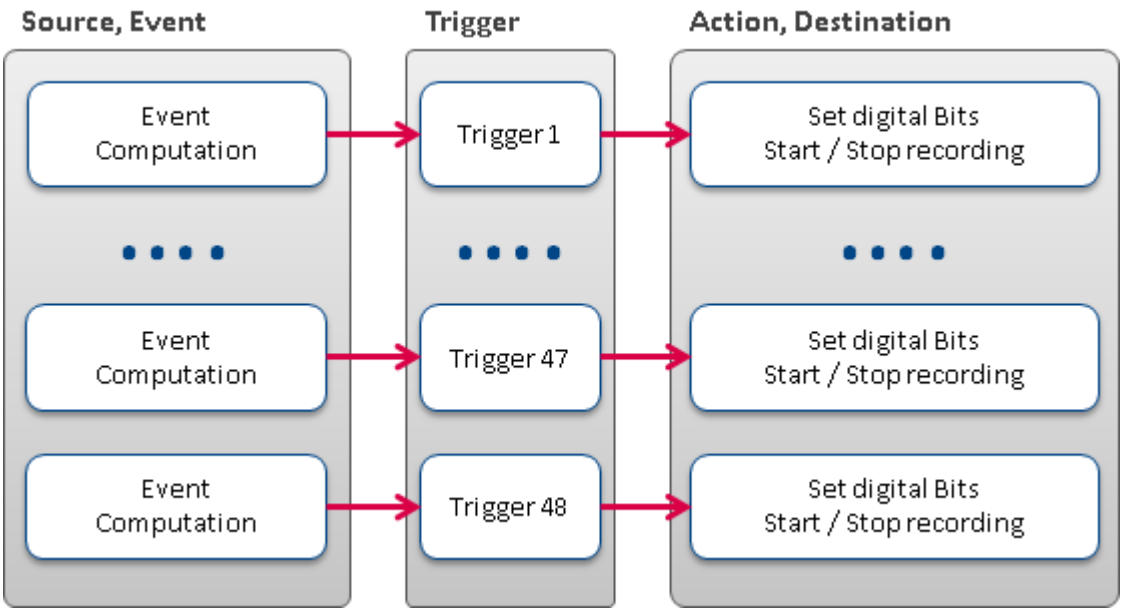
Note

Direct starting of the measurement can be accomplished internally by means of a *1-Trigger*. Even if you have not defined any trigger, the "Trigger_48" is already activated in the Trigger dialog as the *1-Trigger*. For more on this topic, see the description under *Trigger Machine* 185.

5.6.5.2 Trigger Machine

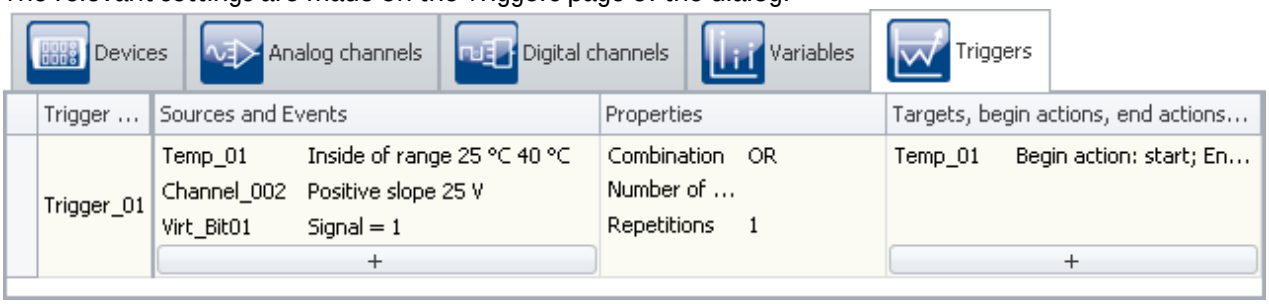
The measurement device implements a complex trigger machine which is capable of accomplishing difficult assignments. It has the following particular features:

- With the trigger machine, up to 48 triggers can each be released independently
- In CRONOS*flex* modules up to 8 trigger events per conditioner can be calculated.
- 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.



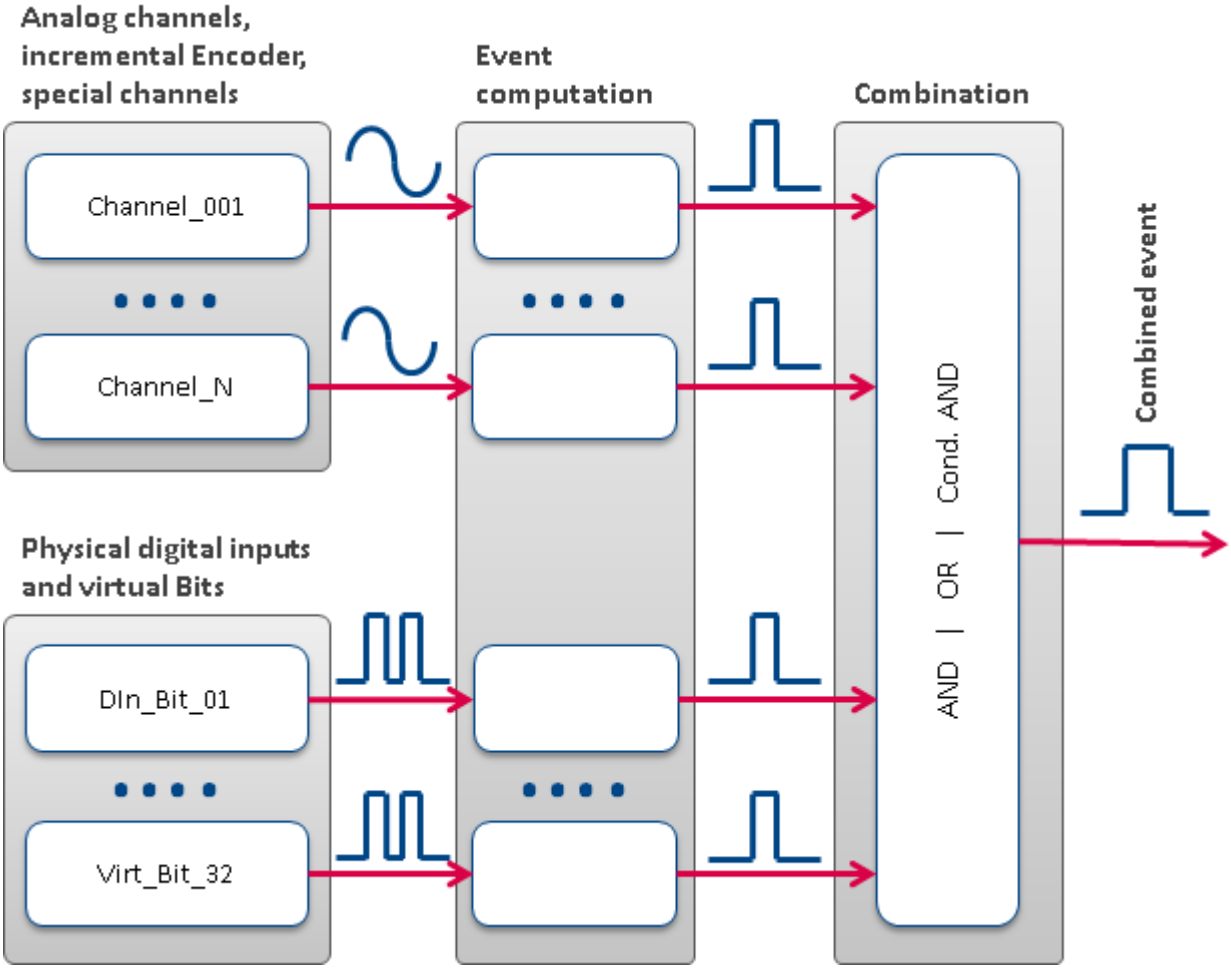
The Trigger dialog page

5.6.5.2.1 Definitions

- 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.
- An Action can be the start or stop of data capture, or setting of a digital output.
- The Trigger is a defined combination of events in response to which an action is performed.
- A Dwell time is a technique for artificially prolonging events.
- A Pretigger serves to record the signal course leading up to an event.

5.6.5.2.2 Sources and events

On the basis of each input channel (analog, incremental counter, Field-bus) 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:

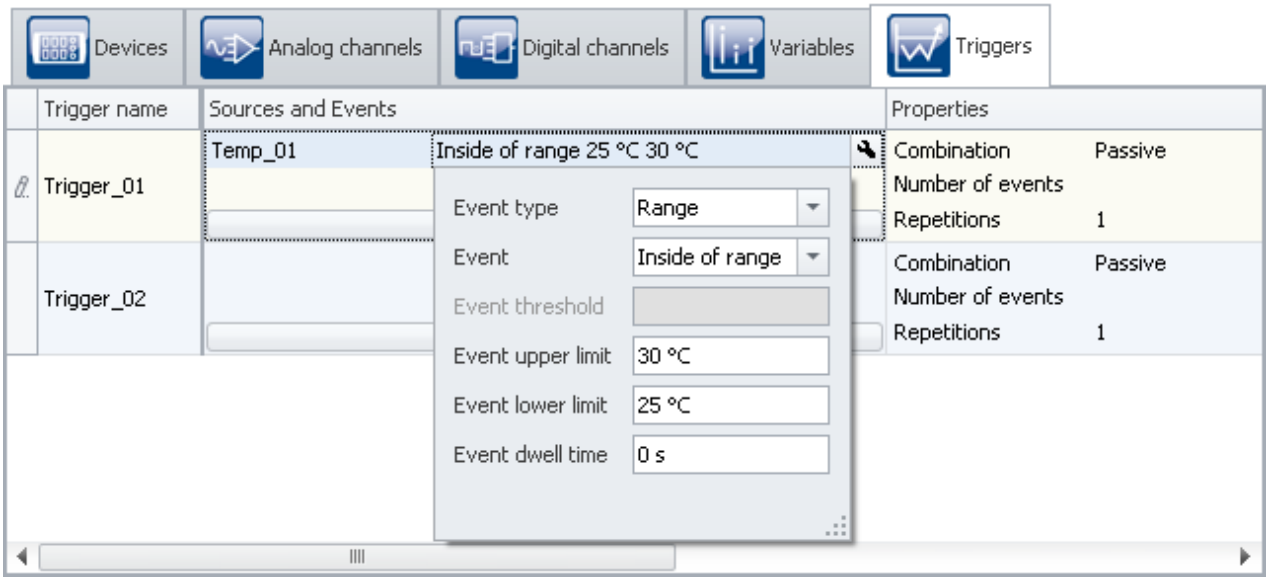


Event computation

Calculation of events from analog variables.

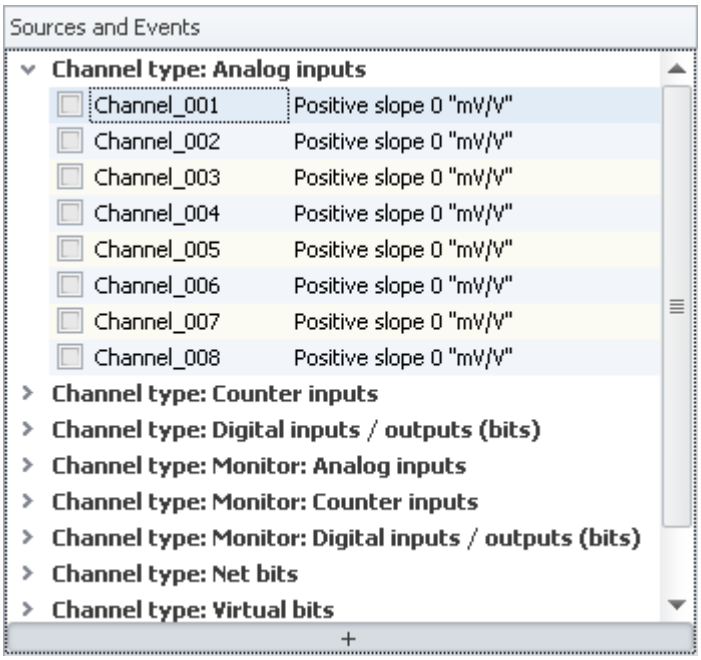
After pre-processing (correction by characteristic curve, scaling, averaging or formation of the RMS), the analog signal's physical value is entered into the event computation, which determines whether an event has occurred, according to the event type set.

The definition is made in the plug-in Setup on the page: Triggers:



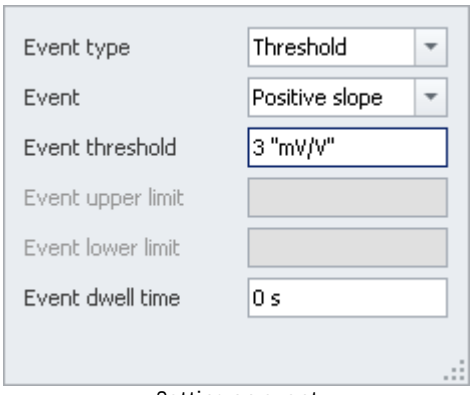
Example: The event occurs when the signal is within the boundaries 25 and 30 °C

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

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 available choices for Event type are *Threshold* and *Range*. The specific event is selected next.

5.6.5.2.3 Event types

Event type: Threshold

The simplest and most common case is the event type threshold value. In the box Dwell time 191, 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 191 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 191, 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 191 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.

Event	Description
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 191, you specify how long the event remains in effect once the condition is no longer met.

The following events are available for selection:

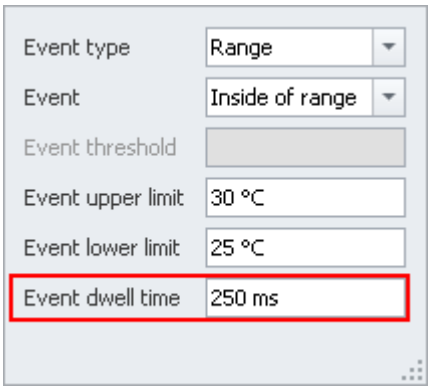
Event	Description
Signal = 1	If the monitored bit's value is *1*, the event is defined to have occurred. This can happen for either a brief instant or for a very long period, depending on the course of the measured signal.
Signal = 0	If the monitored bit's value is *0*, the event is defined to have occurred. The opposite of *Signal=0*.
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 191 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.

5.6.5.2.4 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.



Setting the dwell time

Note

- 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.
- The maximum dwell time for channel-associated events is 2^15 * event-channel's sampling interval. Example: Sampling rate = 1 kHz; Maximum dwell time = 32627 * 1 ms = 32.627 s.
- With virtual bits, the highest sampling rate within the system is used. Example: Channel_01 = 10 kHz, Channel_02 = 500 Hz; Maximum dwell time = 32627 * 100 µs = 3.27 s.
- Digital inputs behave like virtual bits, regardless of whether they are recorded at a different sampling rate.

Example

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

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.

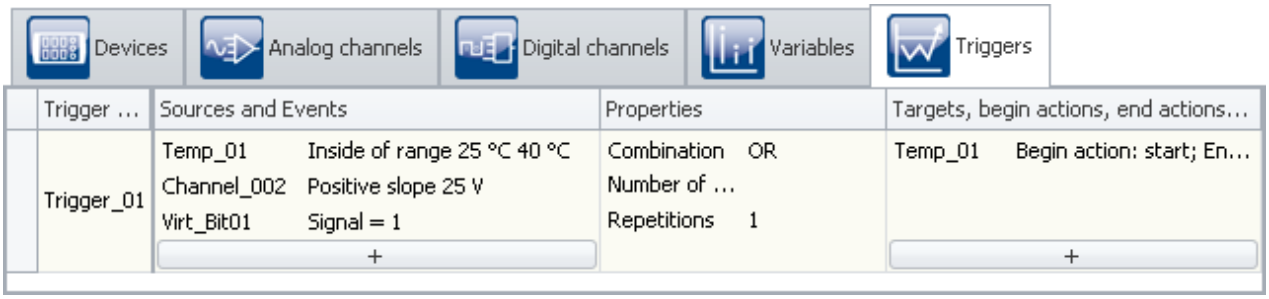
5.6.5.2.5 Combinations of events

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 holdoff time 191.
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) are *true* at the same time, the combined event is *true* and the associated actions are carried out.
1-Trigger	At the start of the measurement, the actions are performed. For this reason, no triggering event can be assigned. *Trigger_48* is by default a *1-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 in the plug-in Setup is done on the 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

5.6.5.2.6 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.

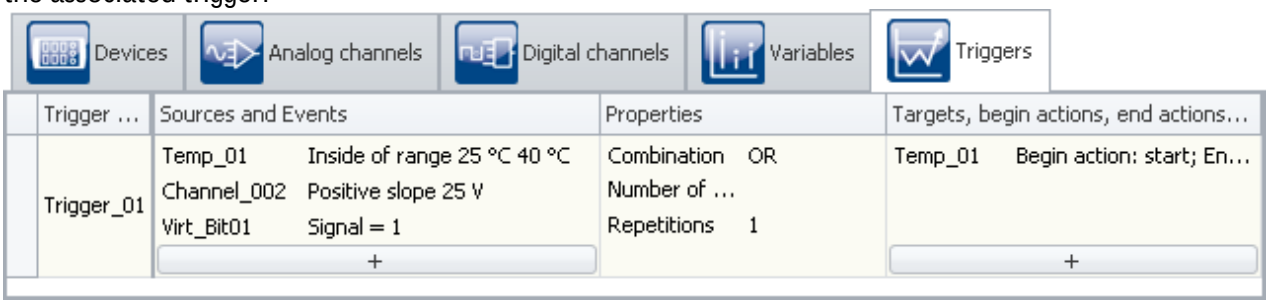
Note

The action: *Set bits* is not dependent on the *Amount* set or on whether a trigger is armed. The action is always performed when the corresponding combined event occurs.

Defining actions and targets

Making settings for the targets and actions in the plug-in Setup is done on the 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.

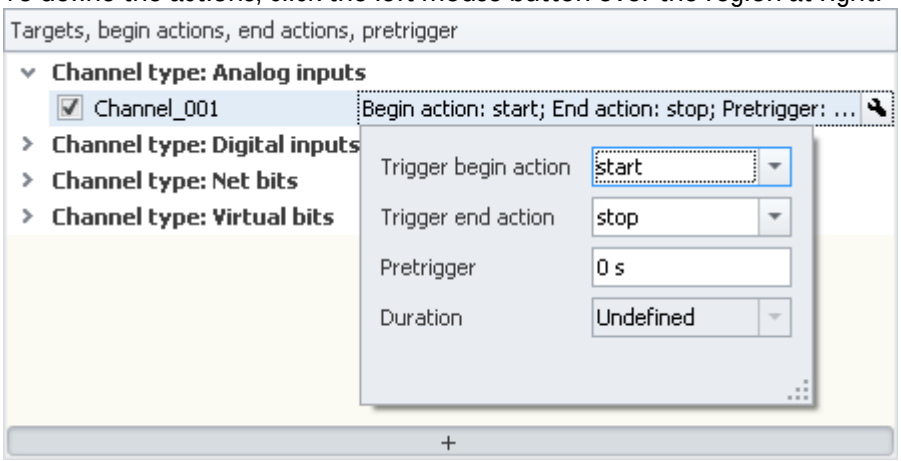


Defining targets and actions

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 197

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*
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*.

Targets	Description
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.

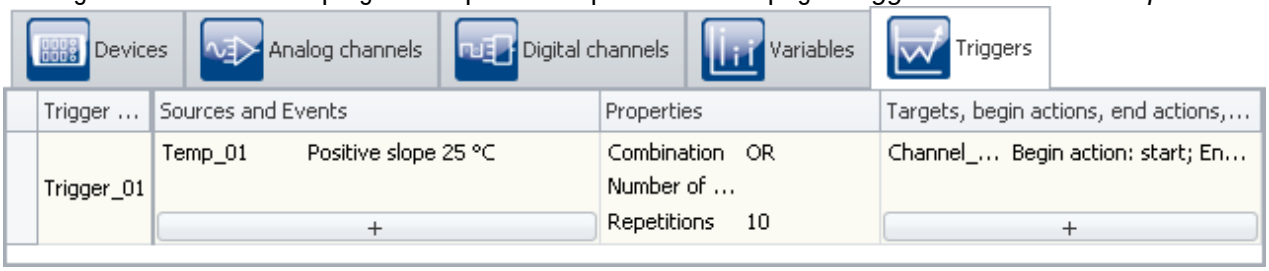
Note

- With digital bits, the actions are performed exactly one time. Afterwards, the bit remains unchanged. A virtual bit can be subsequently redefined, e.g. by means of the PC or imc Online FAMOS. The same applies to a digital output.
- Note that the actions affecting the digital bits do not depend on the arming of a trigger. This means they can be initiated independently of the measured signal's development.

5.6.5.2.7 Multiple triggering (Multi-shot operation)

Within a measurement, it is possible for individual triggers to release multiple times.

Setting the *Amount* in the plug-in Setup is accomplished on the page: *Trigger* in the column *Properties*.



Setting the amount

Using the property: *Amount*, you can specify the number of trigger releases. Normally, this amount is set to 1. You can specify any amount desired or set it to *unlimited*.

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*.

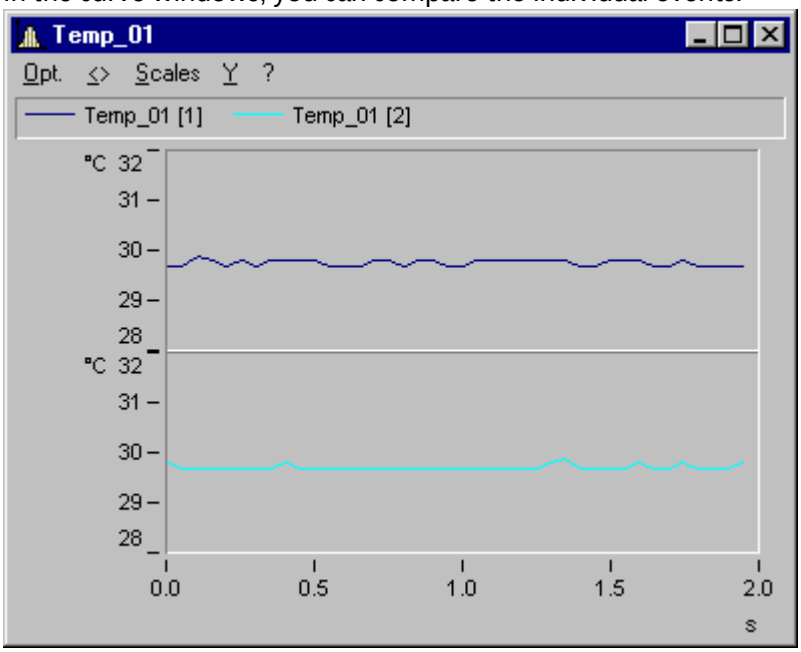
Note

- 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 *Amount* can be set to up to 32767 triggers. Any larger number is automatically altered to "unlimited".
- The action: *Set bits* is not dependent on the *Amount* set or on whether a trigger is armed. The action is always performed when the corresponding combined event occurs.
- Please note the remarks for pretriggers 197.

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 150).

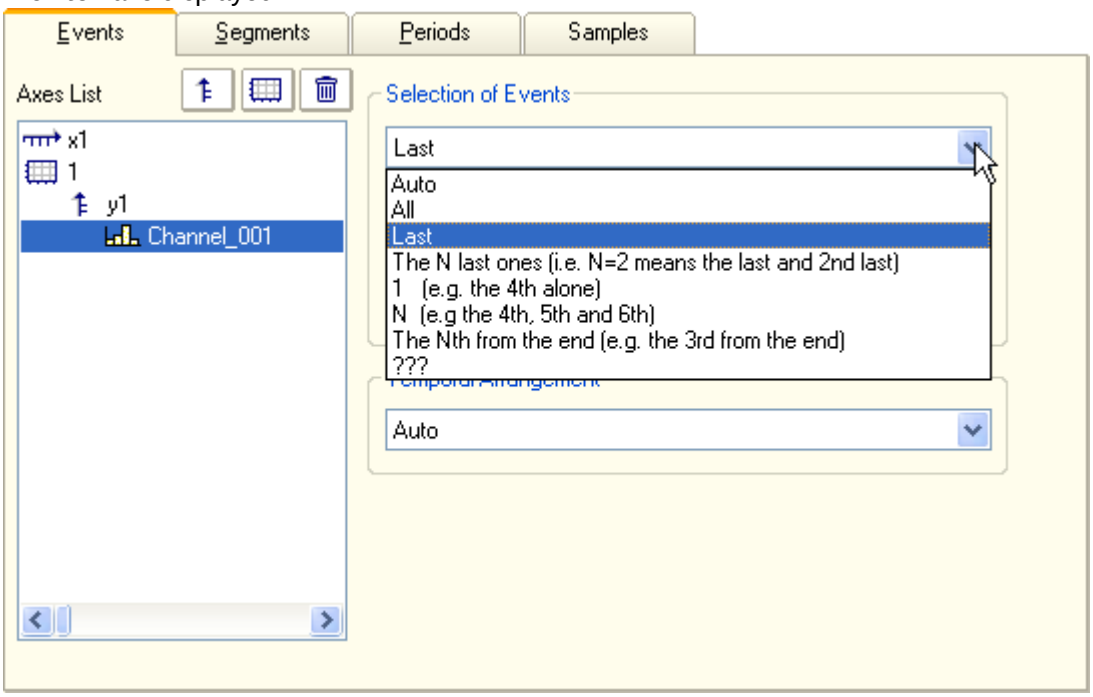
Among other things, multiple triggering enables easy, visual comparison of data from different trigger releases.

In the curve windows, you can compare the individual events:



Curve window with multiple events of the same channel

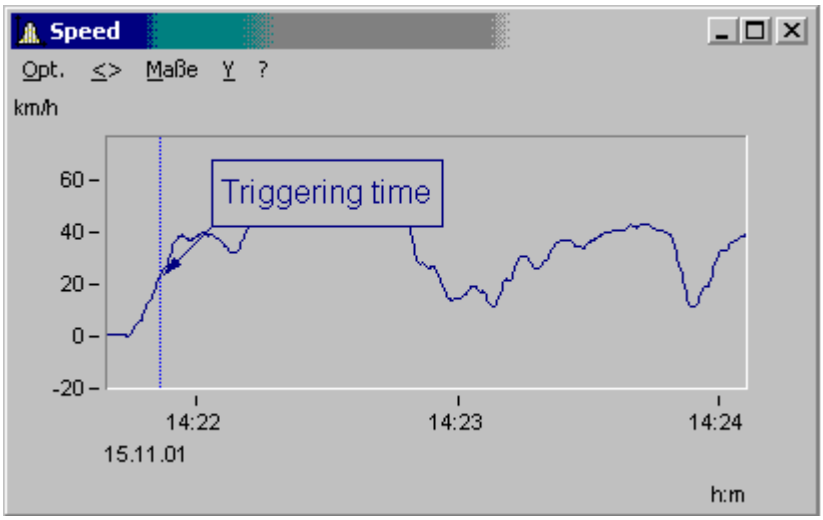
Toward this end, the curve window must be configured appropriately. Select the menu item *Configuration > Events, segments, periods...* On the page *Events*, you can select how many events you wish to have displayed.



5.6.5.2.8 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 in the plug-in Setup is performed on the 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. Channels without such a trigger assignment, in other words, which are started directly when the measurement is started, are assigned to the symbolic special trigger "Trigger_48". If one deleted a trigger assignment for a channel, which is the same as assigning the channel to "Trigger_48", 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.

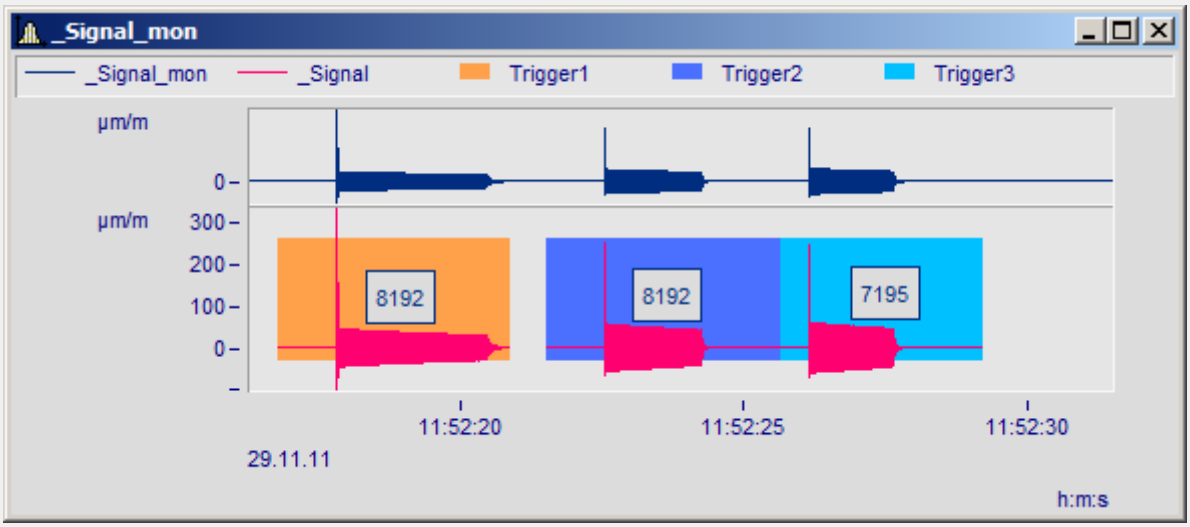
Example

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.

5.6.5.3 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.

Operations

```
RMS_01= rms(Channel_01, 1000, 1000); RMS of voltage channel_01
Virt_Bit01= Greater( RMS_01, 230)  ; Virtual Bit= 1, if RMS_01 > 230V
```

imc Online FAMOS: Exceeding the RMS value of 230V sets Virtual Bit 01

Devices | Analog channels | Digital channels | Variables | Triggers

Trigger ...	Sources and Events	Properties	Targets, begin actions, end actions, pretrigger
Trigger_01	Virt_Bit01 Signal change: 0 -> 1	Combination OR	Channel_02 Begin action: start; End acti...
	+	Number of ...	
		Repetitions Unlimited	+

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 744 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.

5.6.5.4 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.

There are no monitor channels for most field-bus channels. But they can be created with a trick 427 (CAN Monitor Channels from CAN Assistant).

5.6.5.5 The measurement process

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 1-Trigger, Trigger 48, 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.

All measured data recorded and saved to the hard drive are located in a common folder. This is where all data are found which belong to the same measurement. For a new measurement, a new folder is created.

5.6.5.6 Examples

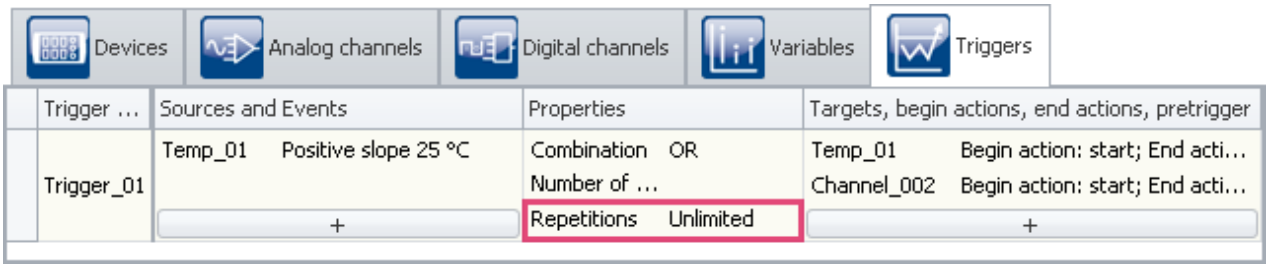
5.6.5.6.1 Background Measurements

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 in the plug-in Setup on the page Trigger accordingly.

Also set *Amount* to *Unlimited*.



Trigger settings for the measurement in the foreground

In the plug-in Setup, on the 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.

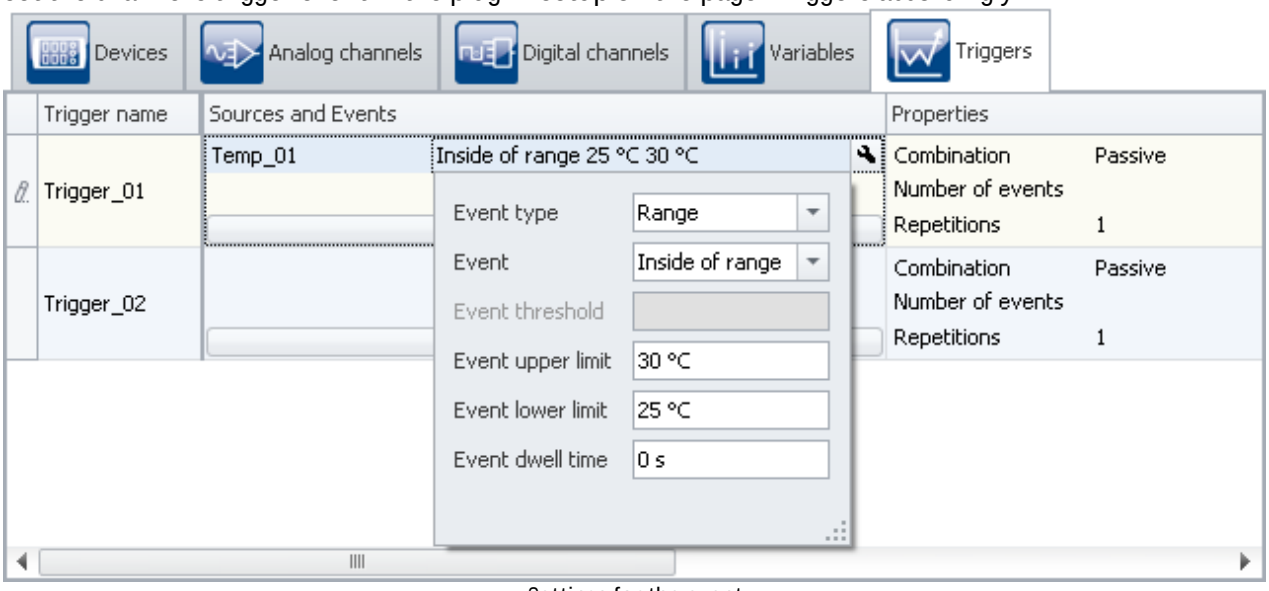
With these settings, it is possible for the trigger to release an infinite amount of times in succession during a measurement. After each concluded measurement, the trigger rearms automatically.

The measurement itself last until the background measurement is stopped by the Stop-button.

5.6.5.6.2 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.

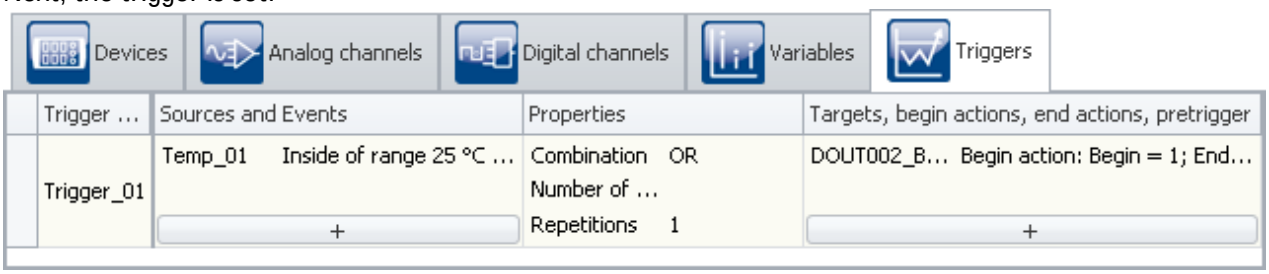
Set the channel's trigger event in the plug-in Setup on the page: Triggers accordingly:



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 ouptu bit's value is *high*.

5.6.6 TEDS - Sensors

On the Setup page TEDS, you can use sensor information from TEDS to configure channels, and you can write such information to the TEDS.

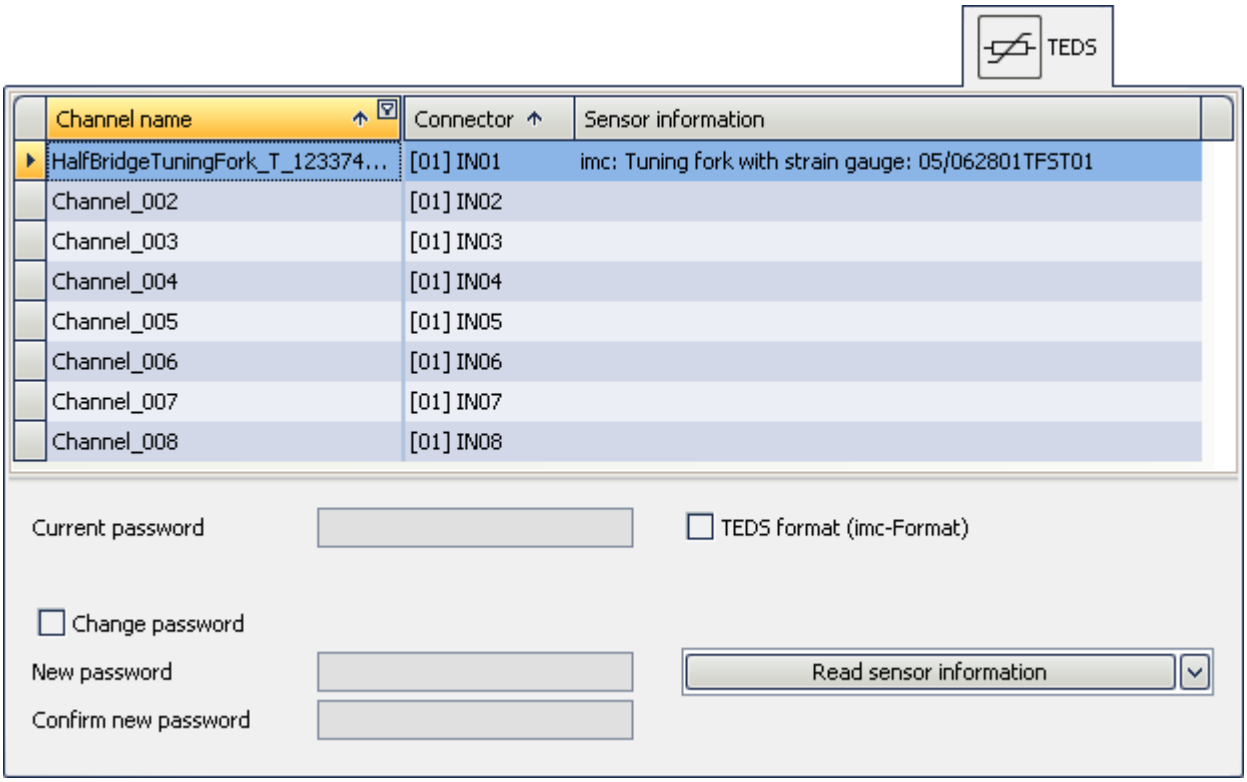
Note: The page is not available

By default, this page is not displayed. To display it, please follow the instructions in the chapter: *Additional Pages* 205.

Note on the views: Compact and Standard

Some functions (e.g. reading and discarding) are also available on the Setup page "Analog channels". In the dialog: "Channel definition".

The following descriptions always have the page "TEDS" as the point of departure.



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 129 and Dialogs 137)

What are TEDS

The device software supports readout of sensor spec-sheet data from a sensor TEDS and the application of this information in configuring channels.

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.

imc SENSORS and the Sensors tool window

There are two ways to configure the sensor information:

- by means of the product imc SENSORS 128
- using the imc STUDIO tool window Sensors (with limitations)

For more on this topic, see: The *Sensors* 122 tool window.

Reference

For a detailed description of the database itself, see the user's manual for imc SENSORS.

Below, the import of already recorded sensor data to imc STUDIO is described.

5.6.6.1 Read Sensor Information and Reset information

Read sensor information

Prerequisites:

- imc STUDIO must be connected with the device
- The amplifier and imc STUDIO must support the TEDS connected

Open the Setup page: *TEDS*.

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* 122 tool window.

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!

Reset channel's sensor information

- Open the Setup page: *TEDS*.
- 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.

5.6.6.2 Linking sensors with channels

Under Connected sensors in the tool window *Sensors* 122 presents a detailed list of the selected sensor's technical specs. However, these sensor data cannot be edited here!

5.6.6.3 List of sensor supported (TEDS/imc SENSORS)

Implemented sensor information (TEDS or imc SENSORS)	
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
Amplifier	Conditioner, amplifier with voltage output
Potentiometric	Potentiometric sensor, operated in half-bridge configuration, for example

NOT supported sensor information (TEDS or imc SENSORS)	
Encoder	Incremental encoders
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
Current	Sensors with amperage output, e.g. 0..20mA or 4..20mA
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)

5.6.6.4 Writing sensor information to the sensor

The sensor information linked to a channel can be written to the sensor TEDS connected to the channel.

Note

It is only possible to write to a sensor from within the page *TEDS*! This underlines the fact that only sensor information can be written, in other words, such info which is displayed in the database imc SENSORS. However, 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 ±5V using the sensor database or Sensor-TEDS. Later, the input range was changed to ±2V. This setting is not adopted in the Sensors tool window. The previously imported value ±5V remains intact in the sensor 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 *sensor TEDS* is password protected, the password must be entered.

Here, it is also possible to edit the password.

Notes on the password protection

- In 1-Wire technology, sensor-Eproms 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.

5.6.7 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 201
- HiL + Application module 205
- ...

* The availability of this function depends on your particular product license. See the technical spec 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) > chapter Views 80.

5.6.7.1 HiL + Application module

imc HiL is a Hardware-in-the-Loop expansion for imc measurement systems, which is based on MATLAB / Simulink (c).

Reference

For a detailed description, see the imc HiL manual.

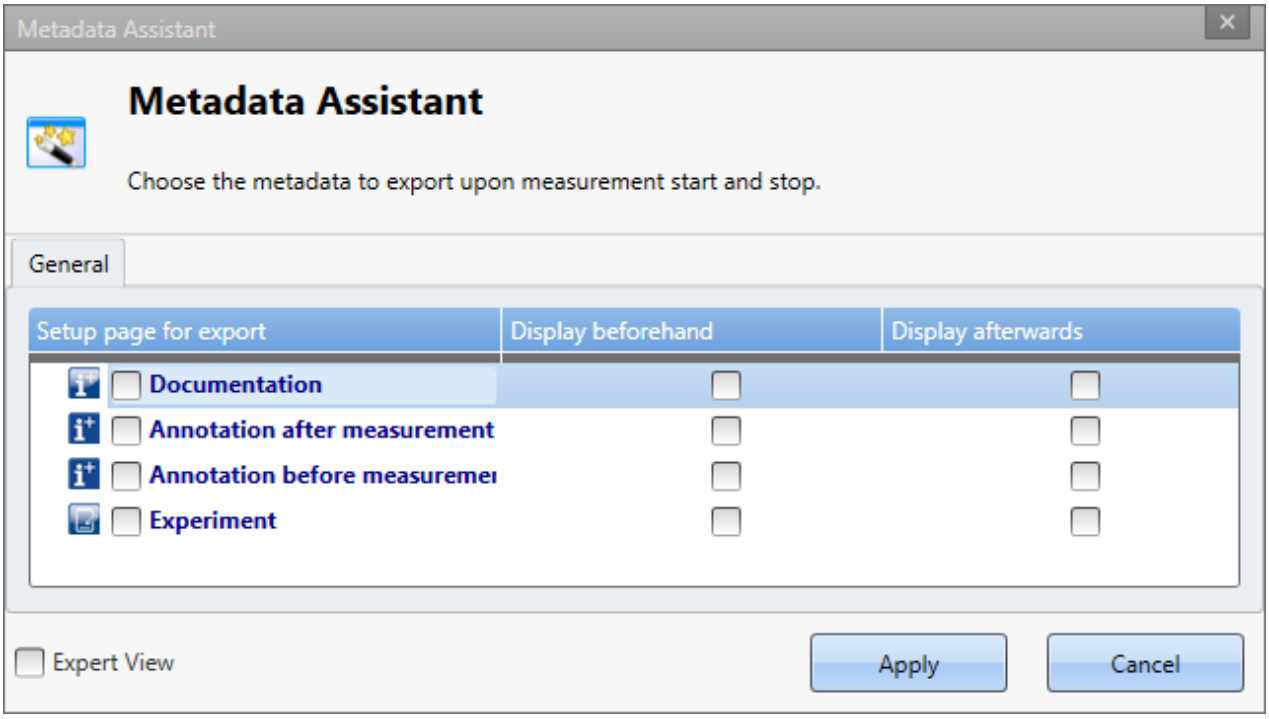
5.7 Metadata and Additional Columns

In this chapter the usage of metadata and additional columns is described.

5.7.1 Metadata Assistant

Using the Metadata Assistant, you can save metadata from the Setup along with a saved measurement or also save them directly to the channel file.

Ribbon	View
Extra > Metadata Assistant	all



Metadata Assistant in the Standard view

In order to have the data on the measurement saved, select the desired Setup page in the column *Setup page for export*. Additionally, you can have the pages displayed before the start of or after the end of the measurement. To do this, check the box for the desired Setup page in the column *Display beforehand / Display afterwards*.

FAQ

Question: How and where are the metadata saved?

Answer: The metadata are saved as a *.csv-file. For this purpose, a folder *Meta* is created in the experiment folder parallel to the measurements. In the respective measurement folder, a link to the associated csv-file is created.

When exporting an experiment with the measured data included, the metadata are also included.

Question: Can metadata also be saved on the device?

Answer: Metadata saved along with the measurement are only saved on the PC. Metadata which are saved in the channel file are also saved int he files which are stored on the device.

Note: Measurement storage area

If you are using the user-defined measurement storage area in combination with metadata, please note the FAQ in chapter Measurement storage area 612.

Use of the metadata in the Setup

If you wish to display metadata in the Panel, which were saved along with the measurement, you can get them using the *placeholder MEASUREMENT.SQL* 1661.

Example: Querying "Comment (Start)", which was saved with the measurement

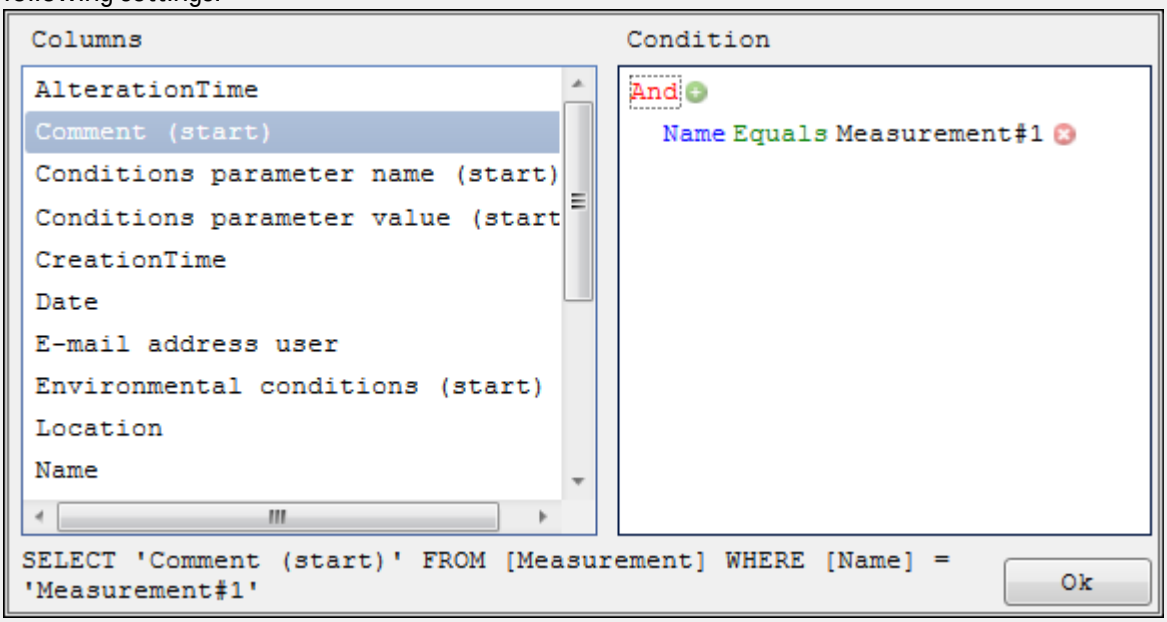
Activate one input channel of your measurement device and save it on the PC. Next, open the Metadata Assistant and for *Annotation before measurement*, set the page to be exported and displayed before starting the measurement.

Next, run a measurement. Upon its start, the page is displayed as a dialog. Enter a text in the box *Comment (Start)* and exit the dialog using *Close*. Let the measurement run a few seconds and then stop it.

Go to the Data Browser and select the saved measurement, so that it receives the measurement number #1.

You can now query the saved comment as follows:

- In the Widget: Create a text box of the group *Automotive, Industrial or Designer.* Open the text box's properties. Open the property *Text* and press CTRL+ Space to open the placeholder selection. Select the placeholder *MEASUREMENT*. In the Assistant which then opens, make the following settings:



Querying the "Comment (Start)" of the measurement having the symbolic measurement number#1

- In the Data Browser: Open the context menu in the Data Browser and select *Add metadata-columns*. From the list, select *Comment (Start)*. The column is now displayed in the Data Browser. Its respective content is in the line for the measurement.

Available Setup pages

By default, the Setup pages *Documentation*, *Experiment*, *Annotation before measurement* and *Annotation after measurement* are available in the Metadata Assistant.

FAQ

Question: Can I make additional Setup pages available?

Answer: Yes (imc STUDIO Developer only).

Whether or not Setup pages can be displayed in the Assistant is controlled via a property of the respective Setup page.

Open the Setup. In the tool window Layout Repository, select the desired Setup page and open the properties (via the context menu or the key F4). Here you will find the property *Metadata template*. Check the box and close the Properties window in order to make the Setup page available in the Metadata Assistant.

If the tool window *Layout Repository* is not visible, you can have it shown by selecting the menu item *View* > *Tool windows* or *Extras* > *Tool windows*.

Question: Can I modify the available Setup pages?

Answer: Yes (in imc STUDIO Professional and higher).

To do this, open the desired page in the Setup. If this page (e.g. *Annotation before measurement*) is not present , you can add it to the other pages using Drag&Drop from the tool window Layout Repository. If the tool window *Layout Repository* is not visible, you can have it shown by selecting the menu item *View* > *Tool windows* or *Extras* > *Tool windows*.

Next, modify the page as desired. For information on this topic, see the section *Operation* 129.

When you are finished modifying the page, open the context menu over the title of the Setup page and select *Save as complete layout*. To overwrite the existing page, confirm the suggested name. Be aware that this change will affect the entire project. For imc STUDIO Developer only: If you wish to use the page under a new name, enter the new name. In this case, after saving you still need to set the property *Metadata template* (see previous FAQ). Subsequently, save the view.

Now when you open the Metadata Assistant, the changed page is available. For technical reasons, the page must always be modified first before it is selected in the Assistant. Otherwise its old edition is used.

5.7.1.1 Expert View

To use the Expert View, check the selection box next to *Expert View* in the Assistant's lower region. Now along with the tab *General* there will be three additional tabs for configuring your metadata.

Column selection

Here you can configure which individual metadata about the measurement to save.

The tab General offers the ability to save all information belonging to a Setup page along with the measurement. On the page Column selection, you can refine this selection: All information (columns) regarding the Setup pages is listed, which can be individually selected/deselected as needed. If you require only some of the information, it is recommendable to first deselect the complete page and subsequently to select only the individual, desired items of information.

Storage is accomplished int the same way previously described for the tab *General*, with the difference, that only the selected items of information are saved and available.

Meta-data in Channels

On this tab, there are various data to select which you can save directly to the channel file. In consequence, no separate csv-file is created. When there are multiple channels, these items of information are saved in all files.

Note

Data can only be written to the file as long as it is open. This means that writing data after the end / stop of the measurement is not possible.

If you wish to export the saved data from the channel file, you can do that as follows:

- Widgets: Using the placeholder <VARS["myChannel"].PROPS["myMetadata"]>, the metadata which were saved in the channel file can be queried.
- imc FAMOS: Use the command *Run imc FAMOS sequence* 1637. Use the command *UserPropText?* (for texts) or *UserPropValue?* (for numbers).
- Data Browser: The saved metadata can be added in the Data Browser as a column. To do this, open the context menu in the Data Browser and select *Add metadata column*. In the list, select the internal name of the desired metadata. The column will be displayed along with the metadata in the respective channel's row. The columns of the Data Browser belong to the View. To retain the setting for the next start of imc STUDIO, save the view using the menu *View* or *Extras* before exiting.

Ribbon	View
View > Save View ( )	Complete
Extras > Save View ( )	Standard, Compact

Note

Metadata saved along with the measurement (tab: *General* and *Column selection*) are saved in a separate file. To access these metadata (Widget or Data Browser), the name from the user interface (e.g. *Comment (Start)*) has to be used.

To query metadata which are saved in the channel file, you need the metadata's internal name. On this topic, read the next FAQ box: How do I find out the internal name?

FAQ

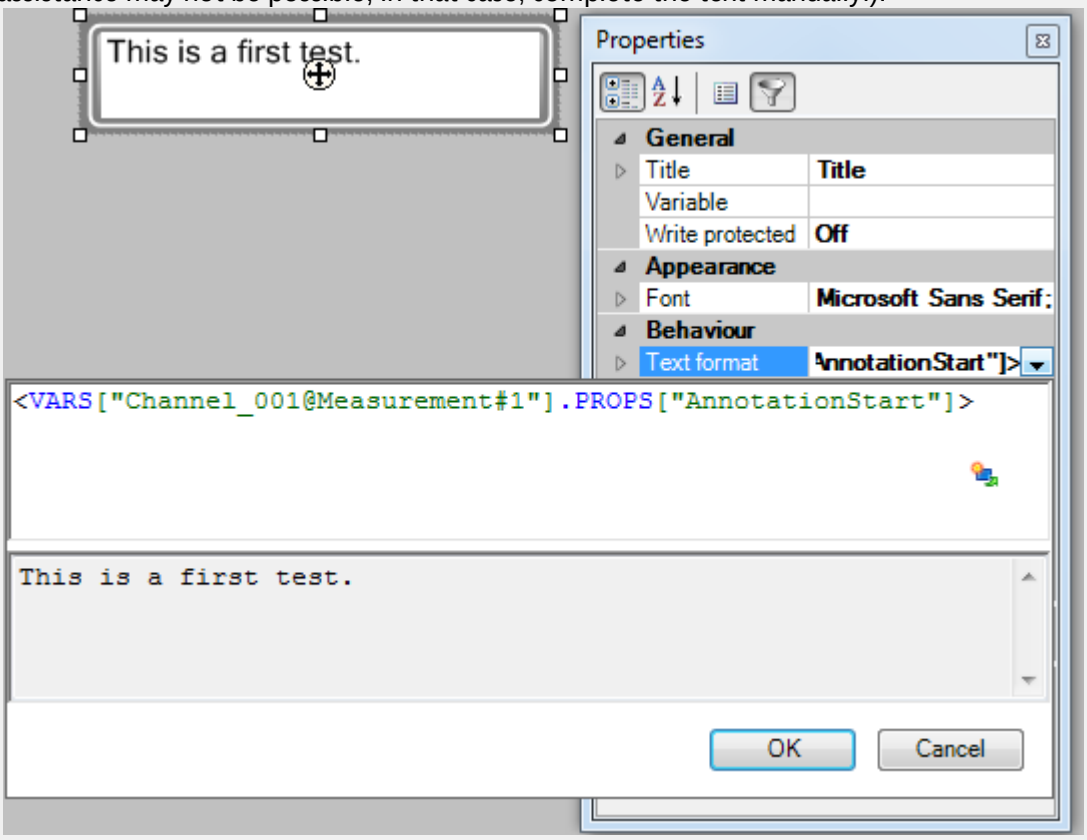
Question: How do I find out the metadata's internal name?

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 to the tab *Channel info*. Here you will find under the heading *User-defined properties* the available saved metadata with their respective internal names.

Example: Querying a channel's "Comment (Start)"

If you have set *Comment (Start)* to be saved to the channel file (selected the entry *Comment (Start)* under *Meta-data in Channels* on the tab *Other*) an already at least one measurement was performed with this configuration, you can query the saved comment in the following way:

- In the Widget: Create a text box of group *Automotive, Industrial* or *Designer.* Open the text box's properties. In the property *Text*, enter the following (making all entries by means of the input assistance may not be possible; in that case, complete the text manually.):



Querying the "Comment (Start)" of the measurement having the symbolic measurement number#1

- Via imc FAMOS: Transfer the channel using the command *Run imc FAMOS sequence* 1637. Enter as the imc FAMOS command *commentStart = UserPropText?(Channel_001, "AnnotationStart")*. The variable *commentStart* can be returned back to imc STUDIO, for example, in order to continue using it there. In order to always use the channel belonging to the measurement #1, in the transfer dialog make the setting To imc FAMOS Source=Measurement#1.
- In the Data Browser: Open the context menu in the Data Browser and select *Add metadata columns*. Select from the list *AnnotationStart*. The column will now be shown in the Data Browser. The associated content appears in the row belonging to the channel.

Configuration of the events

This tab is only visible it the current user possesses the access right *Metadata Assistant: Configuration of the events*. In the default case (so when no changes to the assess rights have been made), this function is only available for users belonging to the group *imc Developers*.

Here you can link additional commands 1592 to the starting/stopping of the devices.

5.7.2 Additional Columns

Creating/editing columns is only supported as of the product versions imc STUDIO PRO 36.

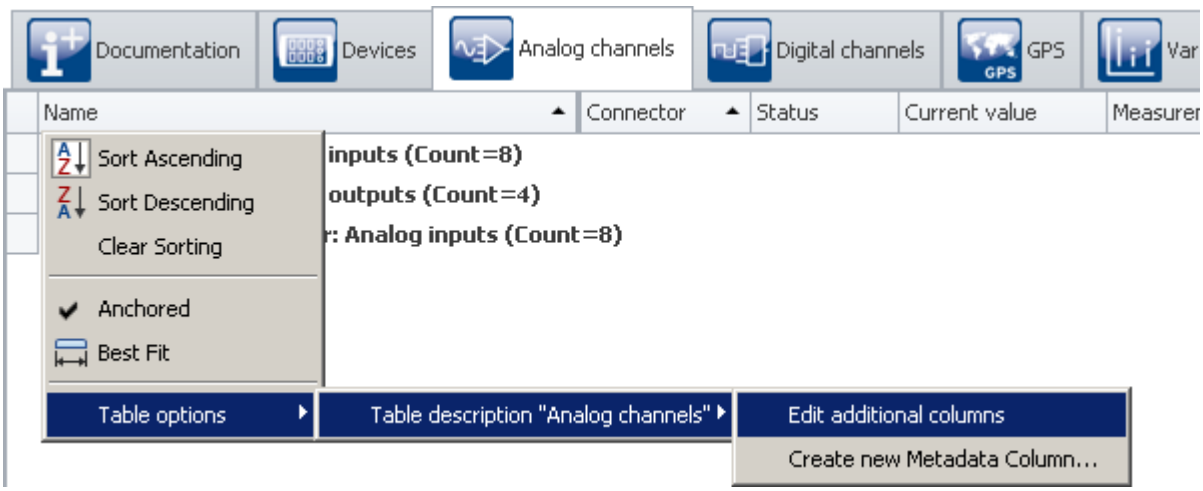
Additional columns contains controls for improving and adapting the operation and documentation of your measurement results in a wide variety of ways.

Type	Description
Combined column 213	Allows combination, operation and display of multiple parameters in a single column.
Metadata column 219	Allows display of any desired information and documents (also from third-party) applications.
Parameter set column 221	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 225	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

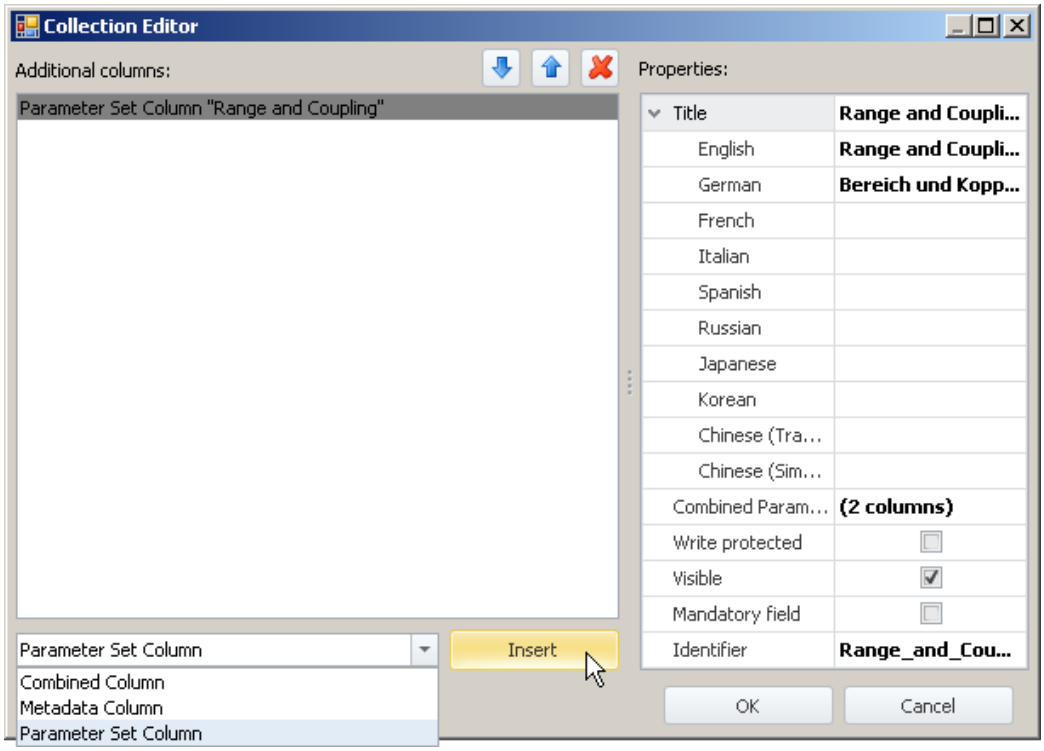
To create additional columns, open the context menu of the column headers and select *Table options > Table description "<Name>" > Edit additional columns*.



"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:



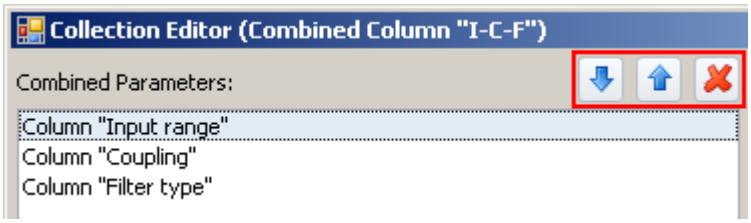
Collection Editor (example)

- In the drop-down at 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 213 or Parameter set column 221.

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 213.

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. The default value entered with the definition is used.
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.

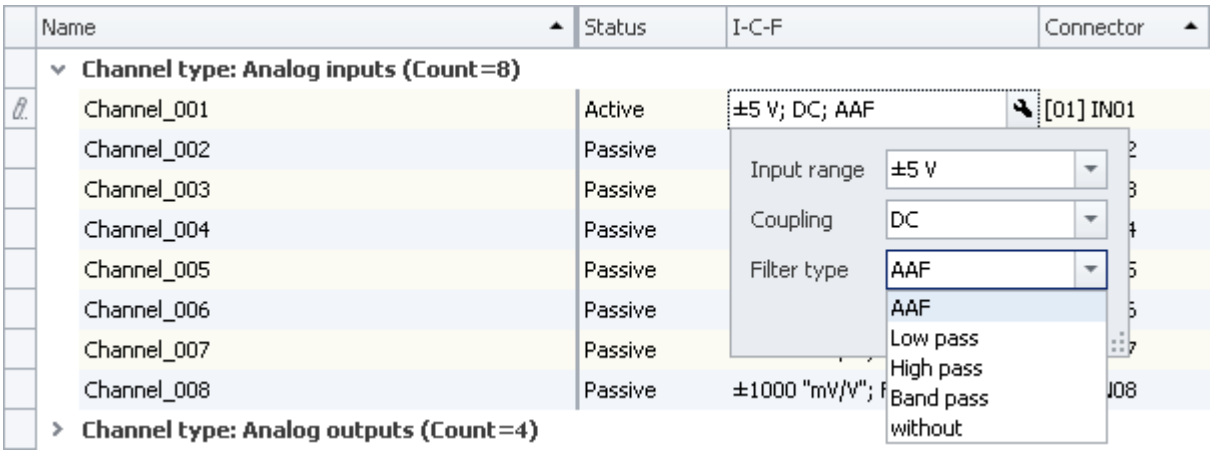
Saving the columns

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.

5.7.2.1 Combined Column

The Combined column joins multiple parameters in one column.

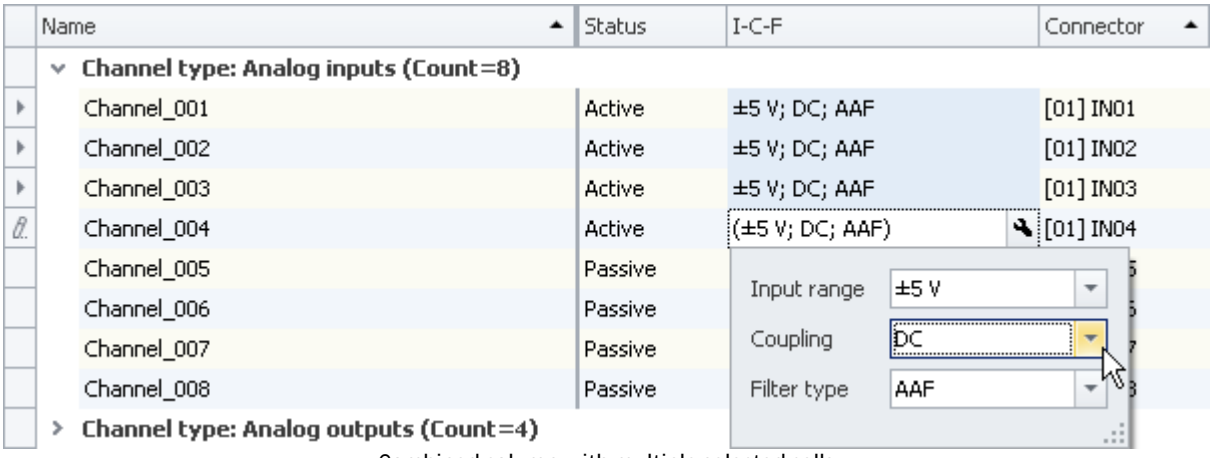
This can make sense, for instance when a measurement is often performed and a certain set of parameters must be set before every measurement. You can then set the parameters in this column and no longer need to open a series of various columns/index cards.



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* 130). 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 213.

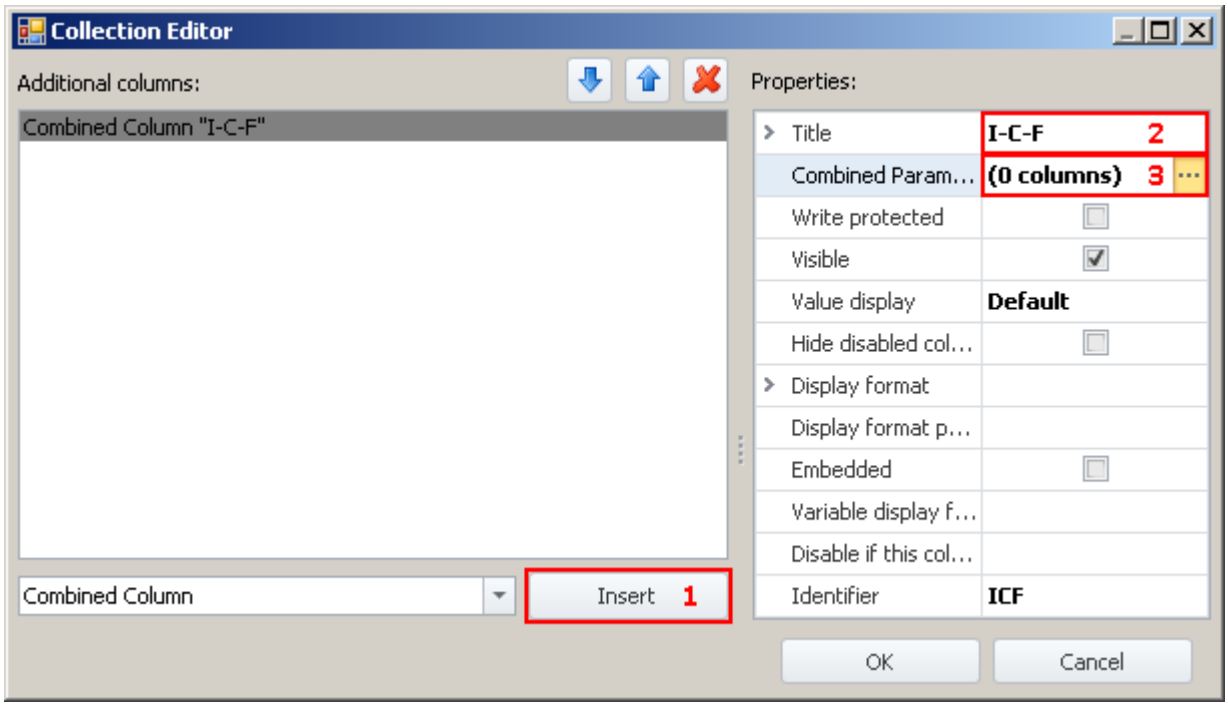
Type	Description
Combined Parameters	List of the parameters which are to be set by means of the column.
Value display	Display format in the cell. The values belonging to the combined columns are always listed in sequence in a row. By means of the following options, the format can be modified. • Default: 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

<table>
<tr><th>Type</th><th>Description</th></tr>
<tr><td>Hide disabled columns from caption</td><td>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; ;</td></tr>
<tr><td>Display format</td><td>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.</td></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>
</table>

Type	Description
Embedded	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).
Disable if this column is empty	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.

Example of a Combined column

Start the Collection editor as explained in the previous chapter Additional columns 211.



Creating a combined column
(example)

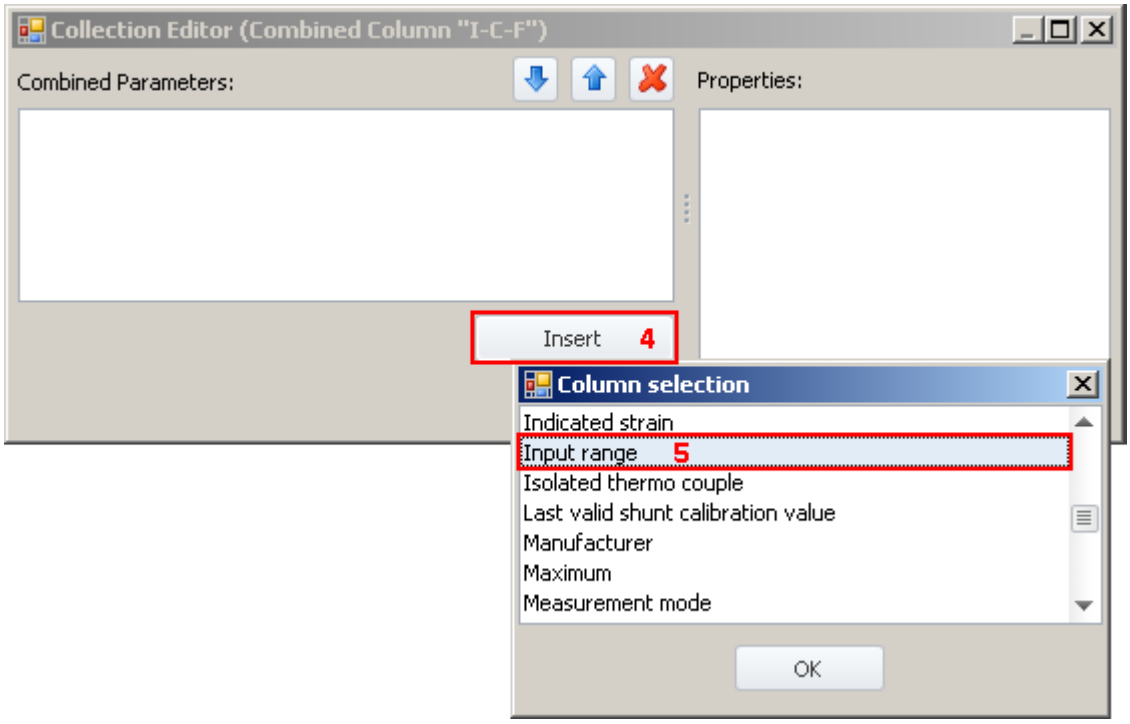
1) In the drop-down list, select the entry *Combined Column* and click on *Insert*. This creates a nameless (title) combined column.

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"

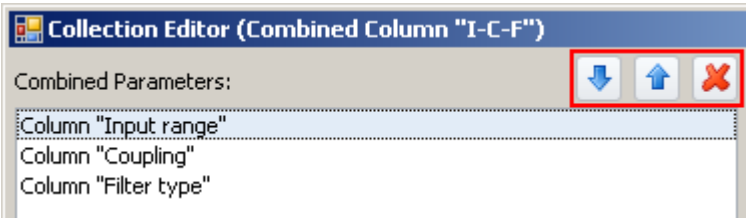
4) Click on *Insert*: An additional window with a list of all available parameters opens.

5) Select from the list the desired parameter(s) (for this example: Input range, Coupling and Filter type).

Sort

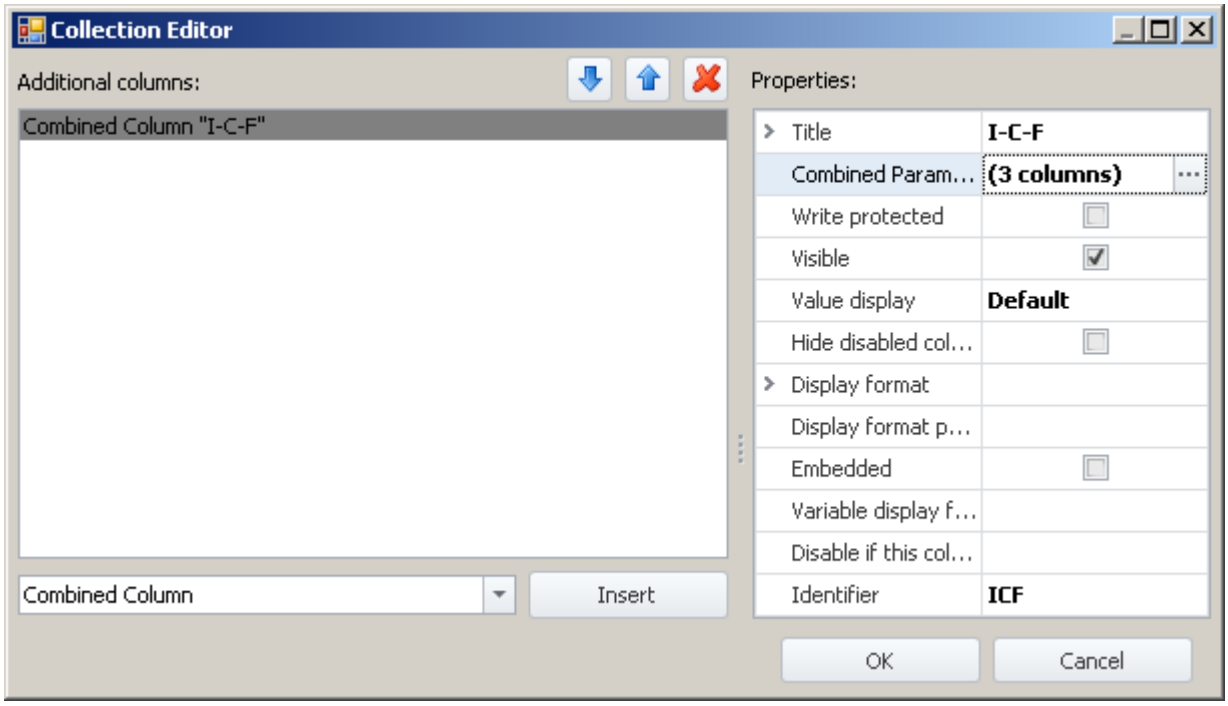
To sort entries, use the "Up/Down"-arrow icons (/).

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:



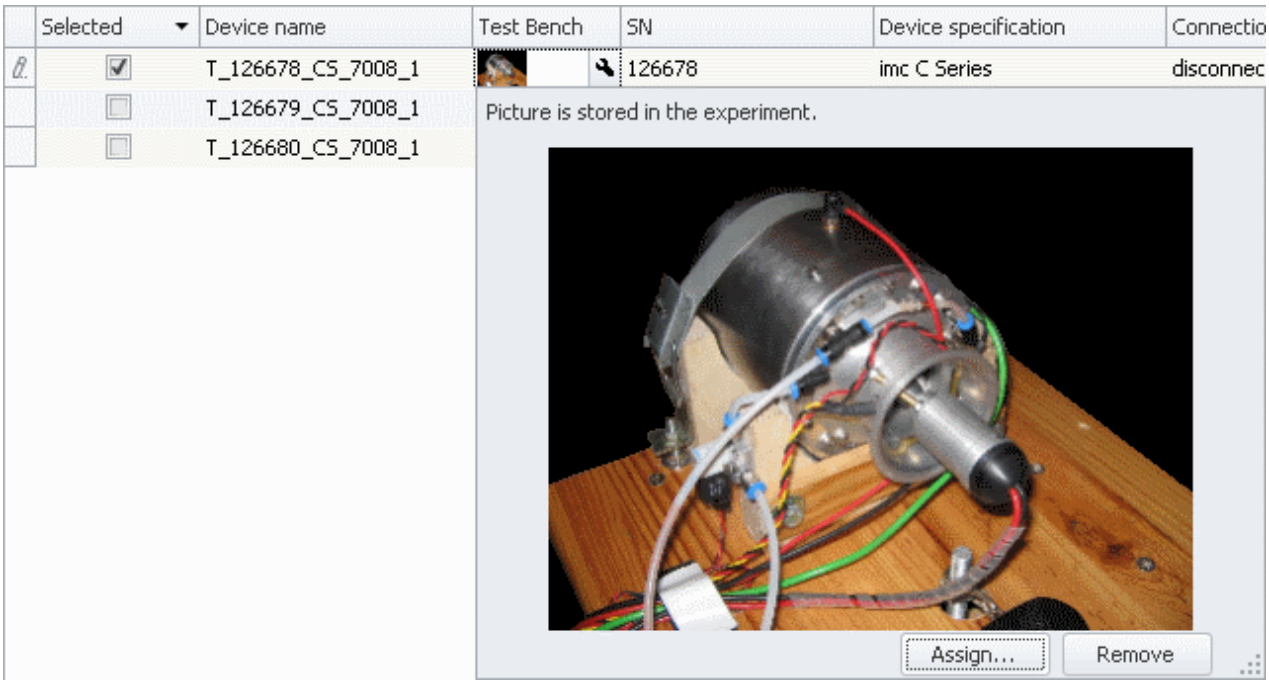
Collection editor, first window, combined column "I-C-F" (3 columns)

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.

5.7.2.2 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 213.

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 field	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 field are evaluated: • Command: "Setup page as dialog" • Command: "Panel page as dialog" • The "Metadata Assistant's" dialog An example is provided with the description of the command: Show Panel page as a dialog 1620.
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.

5.7.2.3 Parameter Set Column

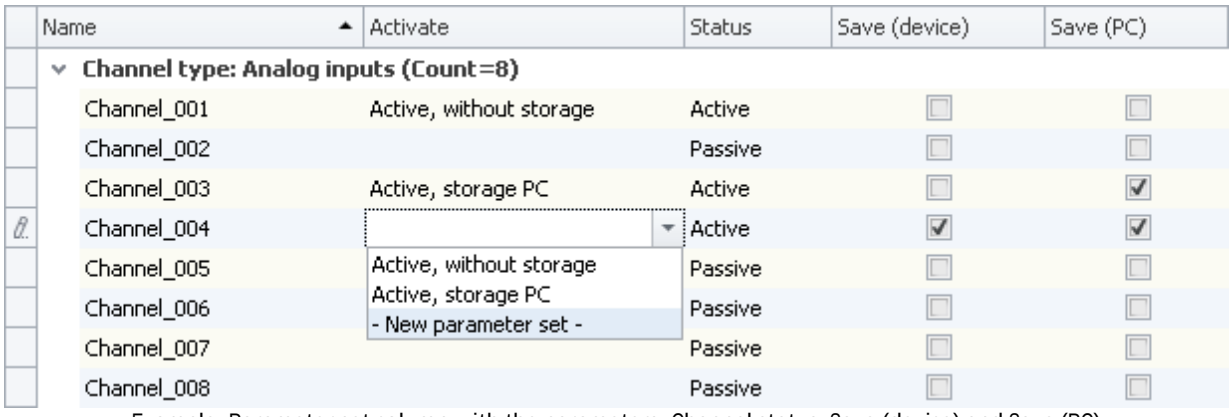
The *Parameter set column* offers the ability to join multiple settings with permanently set values to a *parameter set*. Then, if the parameter sets are pre-defined, the end user must only select the parameter set corresponding to the desired measurement. This saves the user the need to know the entire 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 -".



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.

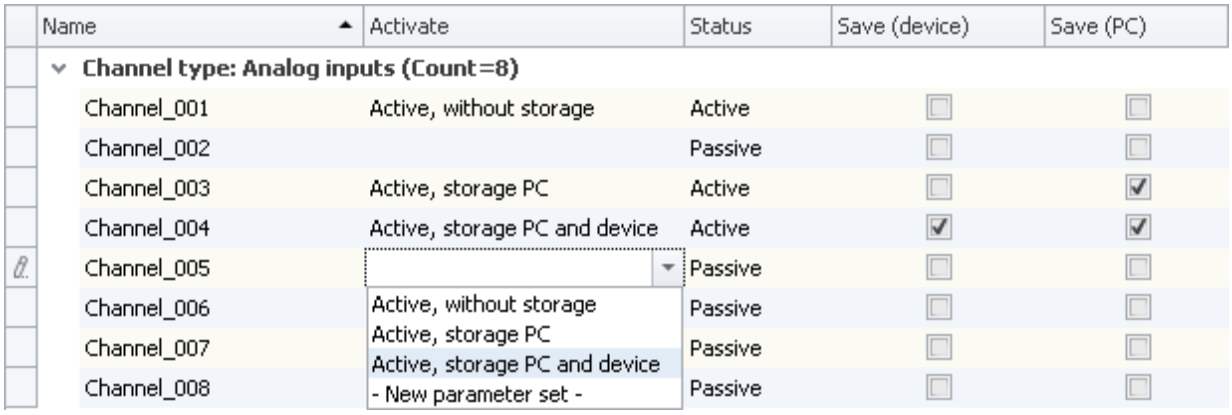
In the example above, if a new parameter set belonging to the selected column is created, the following settings are saved:

- 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*.

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.



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*.

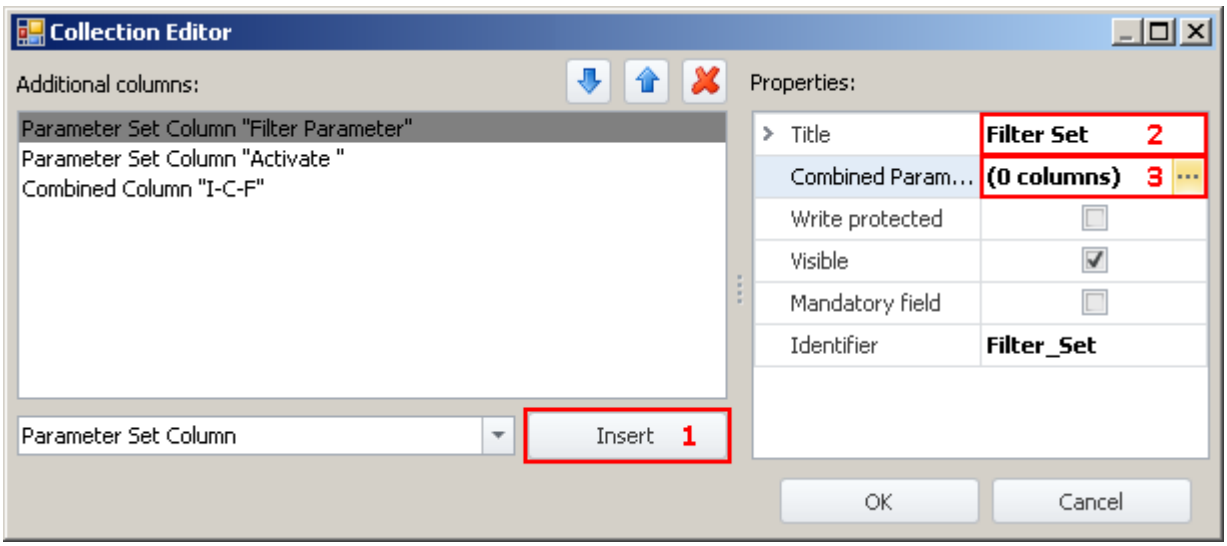
Properties

For shared properties, see: Additional columns 213.

Type	Description
Combined Parameters	List of the parameters which can be set via the column.
Mandatory field	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 1620.

Example of a Parameter set column

Start the Collection editor as explained in the chapter Additional columns 211.



Creating a parameter set column
(example)

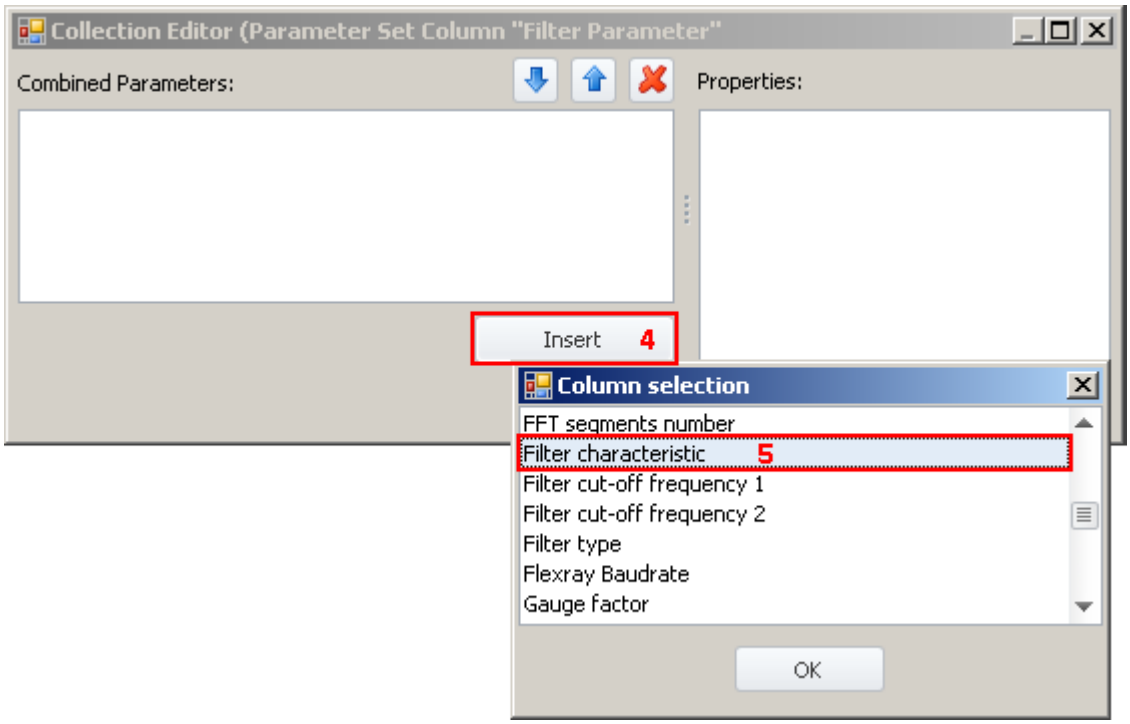
1) In the drop-down list, select the entry *Parameter Set* and click on *Insert*. This creates a nameless (title) parameter set column.

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"

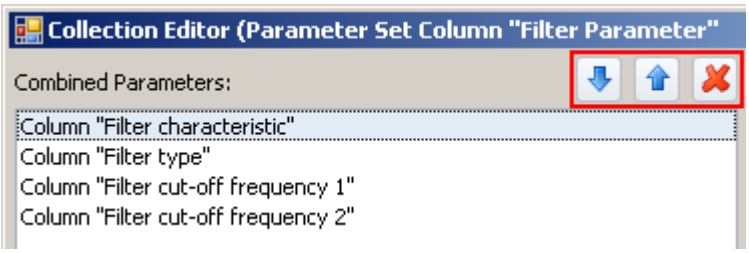
4) Click on *Insert*: An additional window with a list of all available parameters opens.

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)

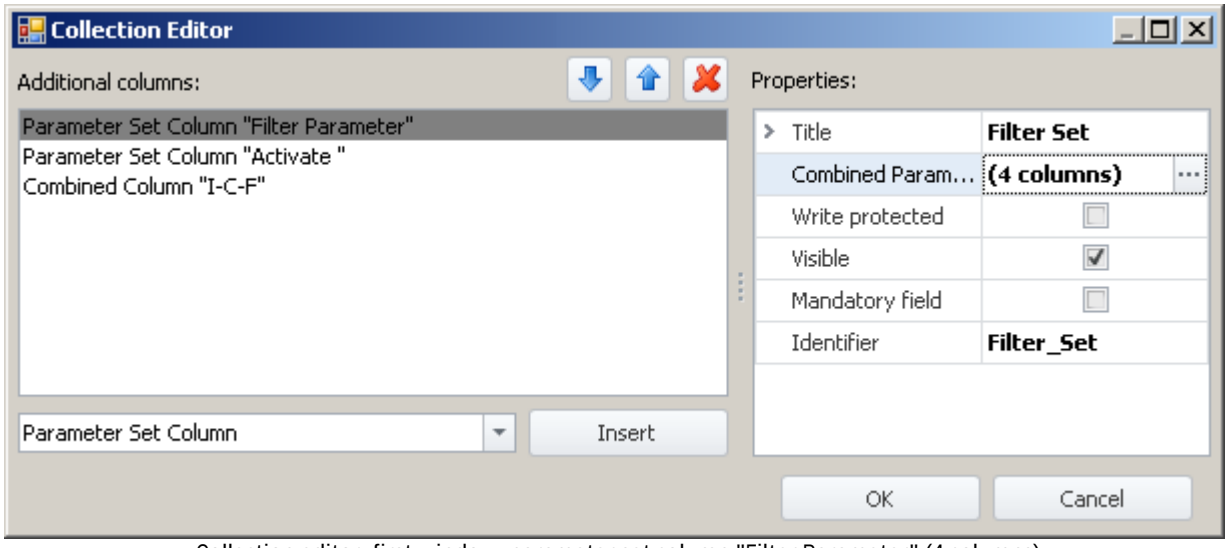
Why the sorting is important

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.

5.7.2.4 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 221 is bound to its associated columns. By contrast, the Parameter Set Import Column operates like the command: Parameter set import 1630. 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	

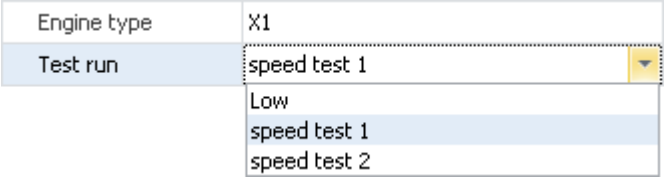
These files are saved in folders as follows:

```
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 213.

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*.

5.7.3 Table description and complete layout

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.

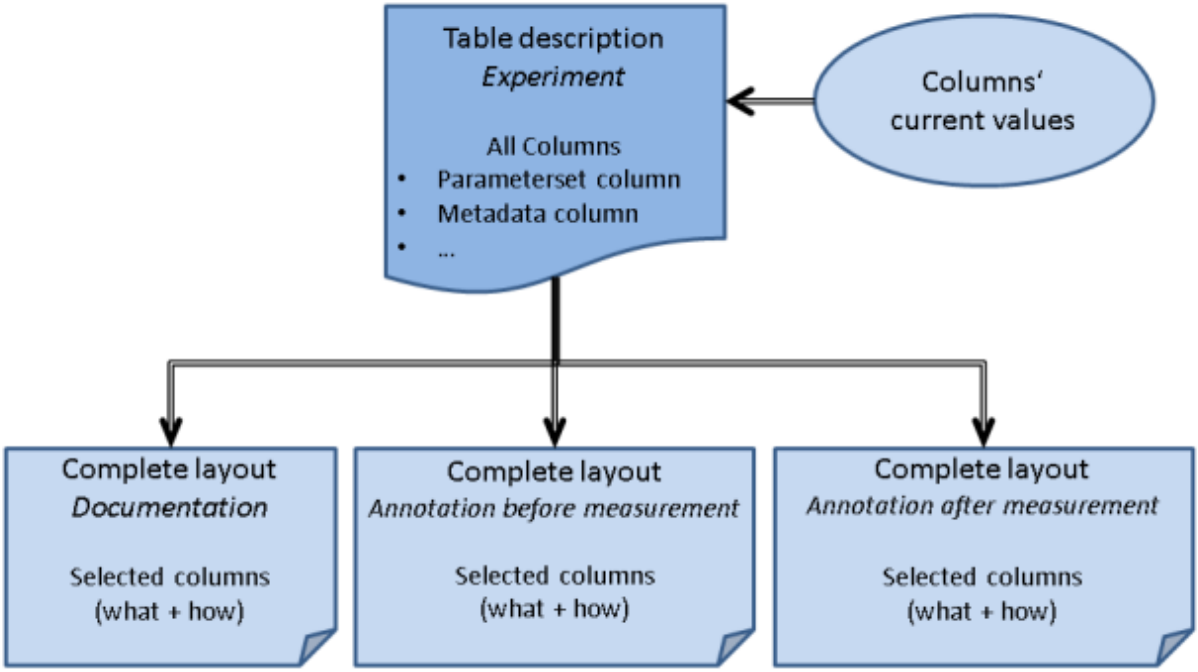


Table description and complete layout

5.8 Information and Tips

5.8.1 Incremental Encoders

The four incremental encoder 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.

5.8.1.1 Signals and Conditioning

5.8.1.1.1 Mode

The various modes comprise the following measurement types:

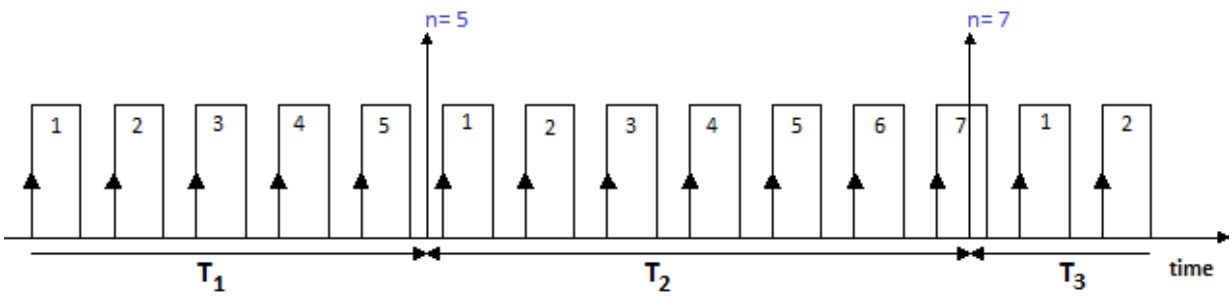
Event-counting	Time	Combined measurements
• events • distance(differential) • angle (differential) • angle (sum) • angle (abs 0-360°) • distance (sum)	• time • pulse time	• frequency • speed • RPM

Event-Counting

The following variables are derived from Event counting:

- events 234
- distance(differential) 235
- angle (differential) 235
- distance (abs.) 235
- angle (abs.) 235

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.

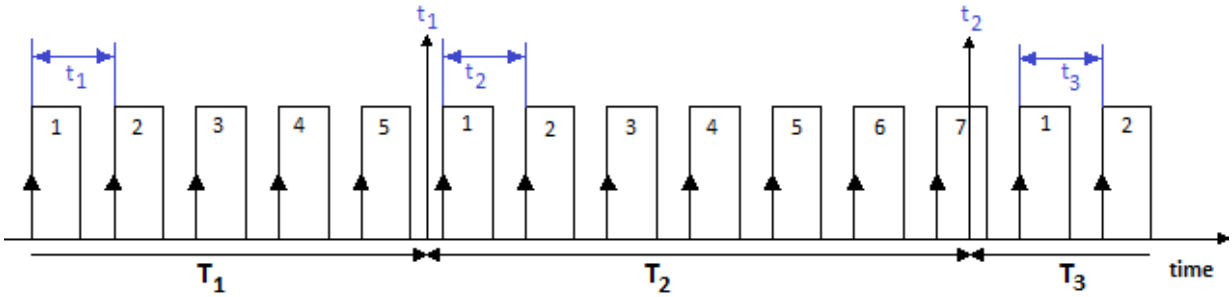


Time Measurements

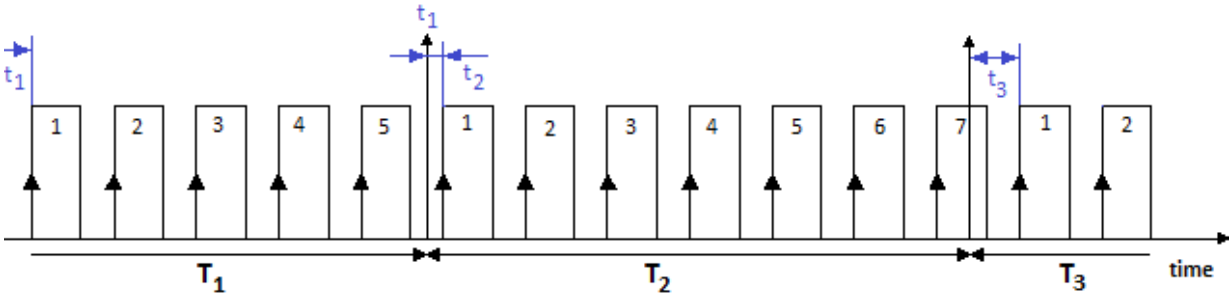
Exclusive measurement of time is performed as:

- time 235 (of two successive signal edges)
- pulse time 236 (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

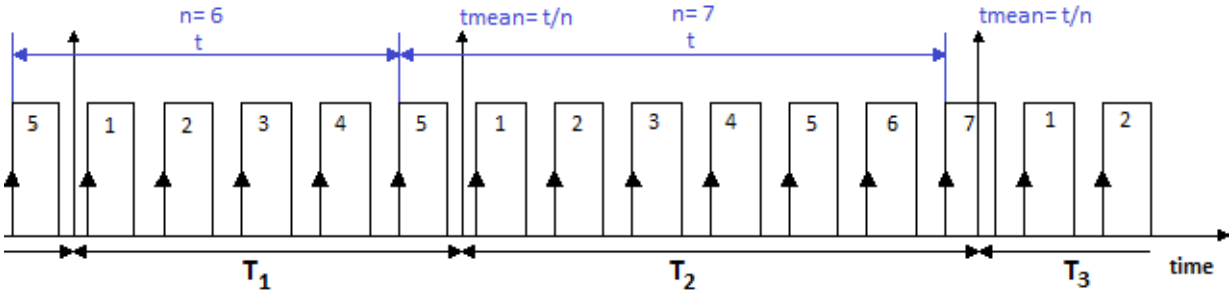


pulse time

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 237
- speed 237
- RPM 238



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 range 30 m Hz < f < 450 kHz. 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 one-signal and two-signal encoder 234
- Start of measurement with or without "Zero impulse" 234
- Number of pulses (per unit)

The frequency resolution of the measurement results depends on the input range selected.

Input ranges and the corresponding frequency resolutions					
Index	Input range	Frequency resolution	Index	Input range	Frequency resolution
1	450 kHz	15,2588 Hz	8	3 kHz	119,2 mHz
2	200 kHz	7,6294 Hz	9	1,5 kHz	59,6 mHz
3	100 kHz	3,8417 Hz	10	750 Hz	29,8 mHz
4	50 kHz	1,907 Hz	11	450 Hz	14,9 mHz
5	25 kHz	0,9537 Hz	12	200 Hz	7,45 mHz
6	12,5 kHz	0,4768 Hz	13	100 Hz	3,73 mHz
7	7 kHz	0,2384 Hz	14	50 Hz	1,86 mHz

The scaling on the CAN bus is obtained by this formula:

Resolution = Clock / 2^{20+i}

i= index in the table above

Clock = 32.000.000 Hz

The formula comes from the oscillator and the integer calculation performed.

The resolution is the frequency resolution you see in the CAN bus message.

From the resolution you calculate: True Range = 32767 * Resolution.

Watch out: with i=1 you get a range > 450 kHz. That exceeds the limits of the comparator. So you can never go safely above 450 kHz!!

Example

Let Nominal (rounded) range = 200 Hz, thus i=12

Resolution = 32000000 / 2^{32} = 0.00745 Hz

True Range = 32767 * 0.00745 = 244.13 Hz

It will be OK to use true ranges instead of rounded ranges, if you figure out the range required for you measurement task.

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 using the above table according to:

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.

5.8.1.1.2 Measurement procedures

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 Online FAMOS functions.

Cumulative measurements

The quantities derived from *event-counting*, Distance and Angle appearing with the annotation (abs.) are "cumulative" measurements. In cumulative measurement, the return value is the sum of all displacement or angle changes, or of all event which occurred.

5.8.1.1.3 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:

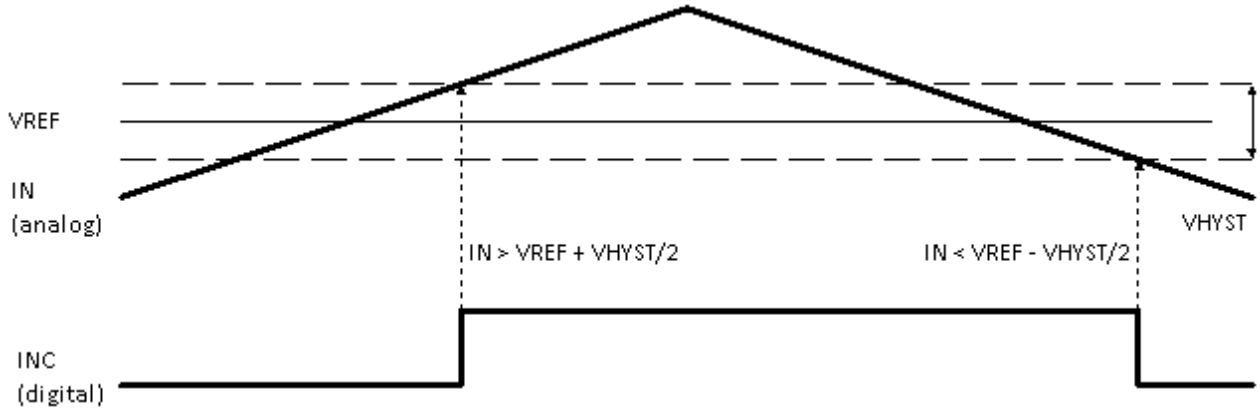
Measurement quantity	(Sensor-) scaling	Range	Maximum
Linear motion	Pulse / m	m	m / s
Angle	Pulse / U	U	U / min
Velocity	Pulse / m	m / s	m / s
RPM	Pulse / U	U / min	U / min
Event	Pulse / Pulse	1 Pulse	Hz
Frequency	Hz / Hz	Hz	Hz
Time	s / s	s	s
Pulse time	Hz/Code	Hz	Hz

5.8.1.1.4 Comparator conditioning

The incremental encoders' 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 encoder 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.



If the analog signal exceeds the threshold VREF + VHYST/2, the digital signal changes its state (: 0 -> 1) and simultaneously reduces the threshold which the signal must fall below in order for the state to return to 0 by the amount VHYST. Thus, the threshold for the next state transition from 1 to 0 is VREF – VHYST/2. 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

5.8.1.1.5 Single-signal/ Two-signal

The single signal counter 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 two signal 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. A measurement with two-signal counters is selected in the combobox "Measurement mode" together with the desired operation type.

5.8.1.1.6 Zero pulse (index)

The zero pulse starts the encoder 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 preparing the measurement.

Note

- The system only takes the zero pulse into account following preparing the measurement. Restarting the measurement does not cause a reset.
- If the zero pulse fails to appear, the INC4 does not start measurement at all. In that case, the channels only return zero.
- Depending on the particular CANSAS module, the index channel can pertain to multiple channels. With the INC4, there is one index channel for all four channels; with the ENC6, there are two index channels for a channel group having up to three channels.

5.8.1.2 Mode (events-counting)

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 one-signal and two-signal encoder 234
- Start of measurement with or without "Zero impulse" 234
- Number of pulses (per unit)

Distance

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.

Angle

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.

Angle (sum)

Cumulative angle. 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.

5.8.1.3 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**

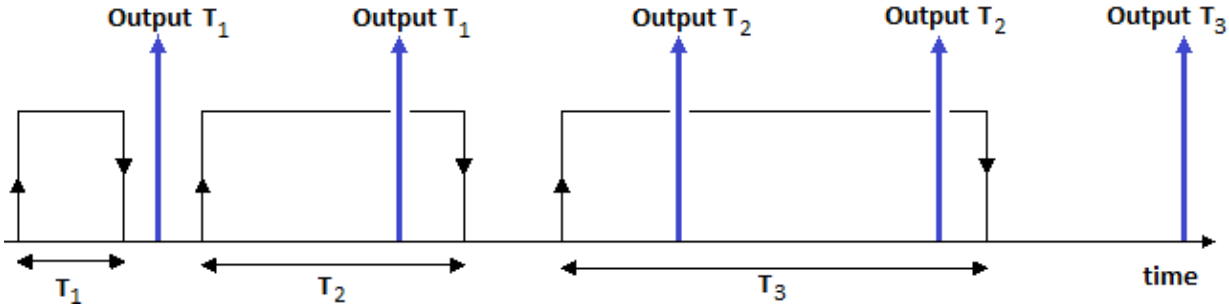
To ensure a high time resolution for the measurement results, suitable scaling must be set for the measurement. An input range (INC4) or Max. time (s) (ENC-6) specifies the maximum time interval which can be measured between the selected starting and stopping edge. The time between the signal edges may 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.

$$PWM = t_{pulse}/t_{Period\ duration} * 100\% \quad \text{or} \quad t_{pulse} * f * 100\%$$

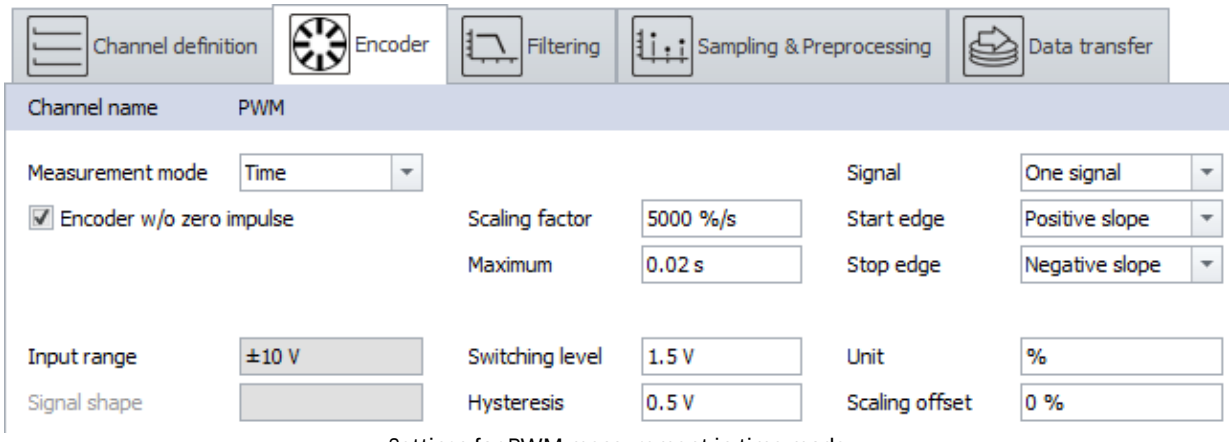
Example:

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

5.8.1.4 Mode (combined measurement)

Frequency

Frequency is determined by means of a combination measurement 230. 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 two-signal encoder for it.

Speed

The sequence of pulses is converted to m/s by means of a combination measurement 230. 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 230. Toward this end, the number of pulses per revolution must be entered.

5.8.2 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* 245.

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. 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.

What contents may a parameter set have?

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.

Parameter set files (*.csv) and 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

XML Files (*.xml)

Contains extended information on the import, so that the target table, the language and the data pool are always correctly recognized.

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 228. 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 the visible, the selected, or even the invisible parameters.

Selection	Description
All columns of tables on current page	All parameters defined in the table description are exported.
Visible columns on current page	All parameters displayed in the table are exported.
Selected columns on current page	All parameters which are selected for all channels/variables are exported.
Selected columns and rows on current page	All parameters which are selected for only the selected channels/variables are exported.
Balance settings	The balance settings are exported, regardless of the Setup page selected. Note: All necessary balance values are recorded in the file in xml-notation, even if the export is not in xml format. The balance values are associated with certain channel settings are only apply if these settings are previously restored when importing. E.g., when changing the measurement range, a new balancing procedure must be performed. For this reason, you can, for example, export the balancing to the same file for each measurement range. Example: You perform balancing for Channel_001 with the measurement range 10 V and export the result to the file: Balance.en.csv. Next, you perform another balancing for the 5 V measurement range and export the result also to the file Balance.en.csv; the file has not been overwritten, but contains both balancing values. Note: Overwriting only happens if the measurement range is already present in the export file. Upon importing, the appropriate balancing settings are imported in accordance with the measurement range set.
Choose columns 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.

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 parameter set", 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 Supplemental columns 211 > Parameter set import-column 225.

Note: Importing variables

For importing variables' values, instead of the command: "*Import parameter*", the command "*Import variables*" 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 (as long as the source language is installed). If the abbreviation is not present, the language with the most matching titles is used as the source language.
- 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 244)
- If only one row exists in a target table (e.g. experiment description), then one row is always imported from the parameter set table.

If XML is selected as the file format, then additional parameters are included in the export, so that the target table, the language and the datapool are always recognized correctly.

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.
Import of channels according to connector and device serial number	For all channels/variables (parameters of the table description: Channels), the settings are imported. The assignment is made according to the connector designation and the device serial number. Note: When using multiple devices, this selection is preferable to "*Import of channels according to connector*", 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.
Import of channels according to connector	For all channels/variables (parameters of the table description: Channels), the settings are imported. The assignment is made according to the connector designation. Note: 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.
...	Select a user-defined mapping. See *Mapping - Modifications of the default behavior* 246.

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* 246.

Editing the parameter set

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.

5.8.2.1 Mapping - Modifications of the default behavior

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:

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.

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: Import of channels according to connector

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>
            Import of channels according to connector
        </caption>
        <description>
            The settings for each of a device's channels are imported. The assignments
are made according to the connector designation.
        </description>

        <mapping>
            <source>
                SELECT *
            </source>
            <target>
                SELECT * WHERE Connector = source.Connector
            </target>
        </mapping>
    </languageblock>
</import>
```

Example: Import 4: Import of channels according to connector and device serial number

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>
            Import of channels according to connector and device serial number
        </caption>
        <description>
            The settings for each of a device's channels are imported. The assignments
are made according to the connector designation and the device serial number.
        </description>

        <mapping>
            <source>
                SELECT *
            </source>
            <target>
                SELECT * WHERE Connector = source.Connector AND Device SN =
source.Device SN
            </target>
        </mapping>
    </languageblock>
</import>
```

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>
```

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>
```

5.8.3 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 107 for example, in order to have the correct targets for the defaults.

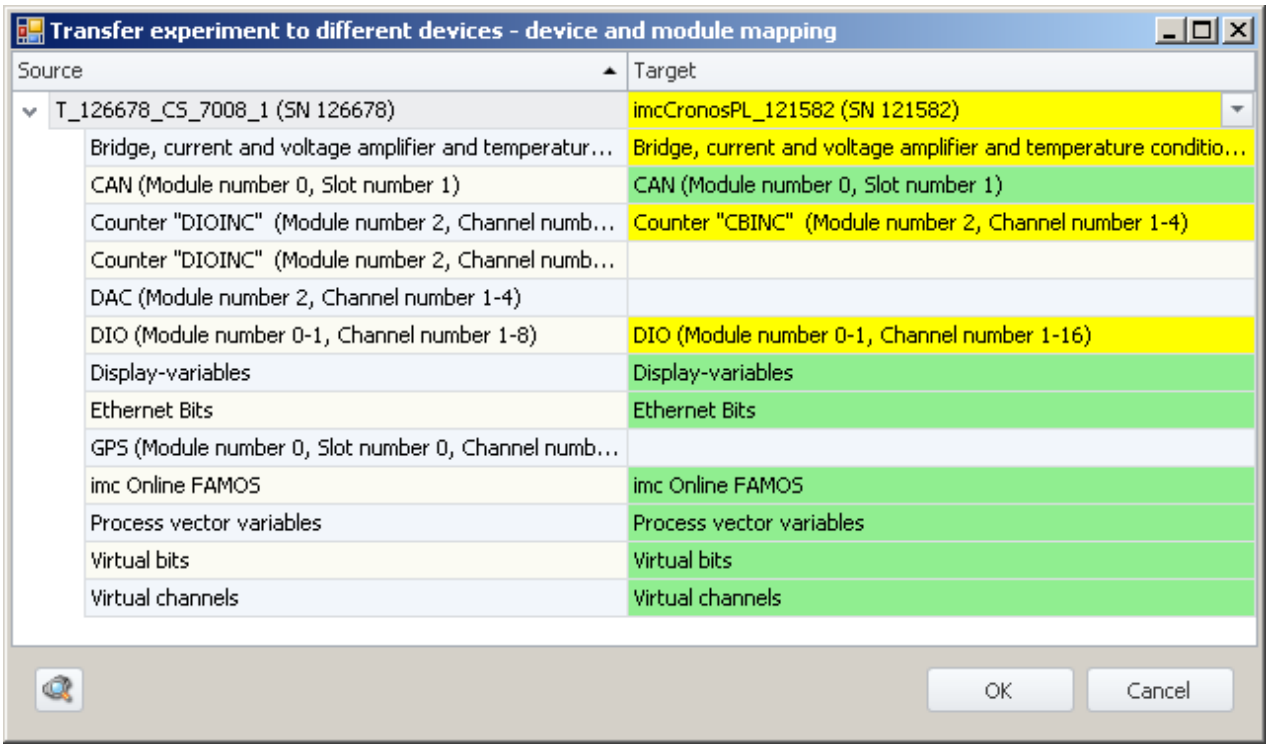
Enumerator class	Internal designation	Description
Experiment	eExperiment	Experiment metadata
Device	eDevice	Device settings
Module	eModule	Amplifier parameters
Bit	eSubChannel	DI-/DO-bits (incl. Mon-DI-Bits), virtual bits, Ethernet bits, Field-bus bits (MFBDIO), PV-variables (of DI-ports, analog-, incremental counter- or Field-bus 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)

5.8.4 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
   - Open the *Project* menu ribbon, select the command *Import / Export*
     - Select the option *Import* and next the list entry *Transfer experiment to other devices*
     - Select the desired experiment
   - The Assistant opens
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.
   - This substitution in case of failed connection can also be automated by means of the command: Transfer device settings. Automation is only recommended for devices of the same model and configuration.



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.

5.8.5 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" + "Import Variable"
  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: "Import Variable " + "Execute menu action: Start (Start Measurement)"
- Way 2:
  Use the command: "Set Variables". 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 1077 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 save the configuration of these curve windows and load them again into curve windows on the Panel pages. 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!)

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.

5.9 Tutorial

This chapter presents some examples involving the plug-in Setup. The examples use Widgets 833 from the plug-in Panel 813, which may not be included in your installation (depending on the product configuration).

- Simple measurement - First Steps 260
- Subsequent saving of measured data 265
- Pre-defined saving of measured data 268
- Simple Triggering 272
- Simple Triggering - Enhanced 279
- Artificial sine 281
- Triggered measurement 285
- Thermocouple measurement 291

5.9.1 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, the channel used must be set accordingly.

Learning goals:

- Creating an experiment
- Simple device operation

Elements used:

- Search for devices
- Device selection
- Connect
- Channel settings
- Save current Measurement Data

Additionally used plug-ins:

- Panel

Prerequisite:

- imc measurement device with an analog channel

Procedure:

Install imc STUDIO. More information on the installation is provided in the chapter: Setting Up 20.

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 (terminals, etc.).

Start imc STUDIO as described in the chapter Start 39.

5.9.1.1 Setup - Settings

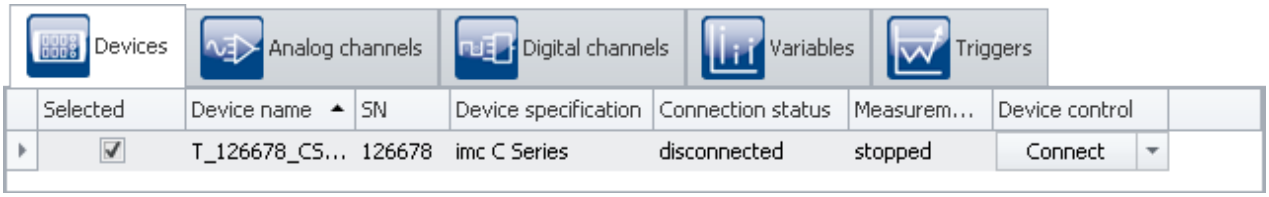
To begin, the measurement device is selected and configured.

- Open the plug-in 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, then perform a device search 139 ( ) (Ribbon *Home* (or *Setup-Control*) > *Search for Devices*). Subsequently, the list will display all devices found.
- 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

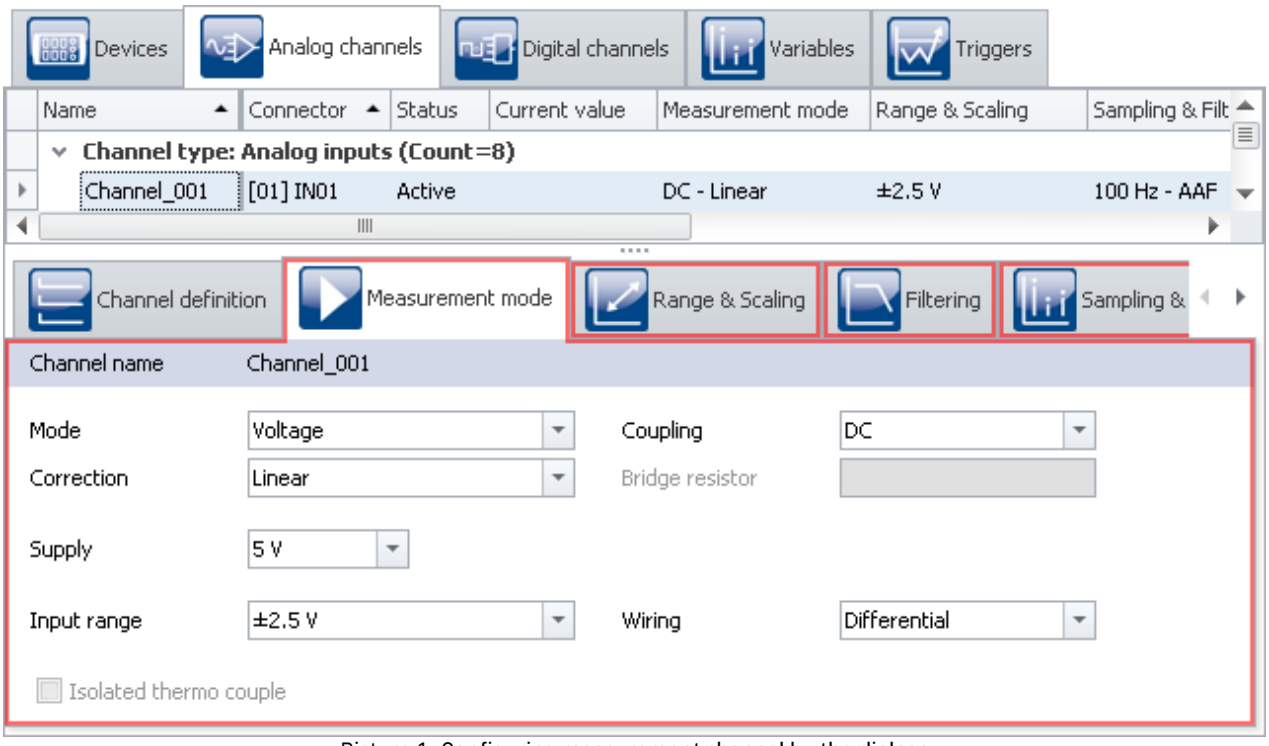
Now the channel used is configured:

- 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



Picture 1: Configuring measurement channel by the dialogs

- To apply the changes, click on the button Process configuration 139 (✔) (Ribbon *Home* (or *Setup-Control*) > *Process*).

By this means, the device settings are available to other plug-ins.

5.9.1.2 Panel - Settings / Start measurement

To display measured values, create a Panel page.

- Open the plug-in Panel.

How the plug-in Panel is structured is presented in the manual: Panel 813.

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

- Activate the Design mode (✎) (Ribbon *Panel-Design* (or for a simplified view: *Home*) > *Design Mode*).

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 139.

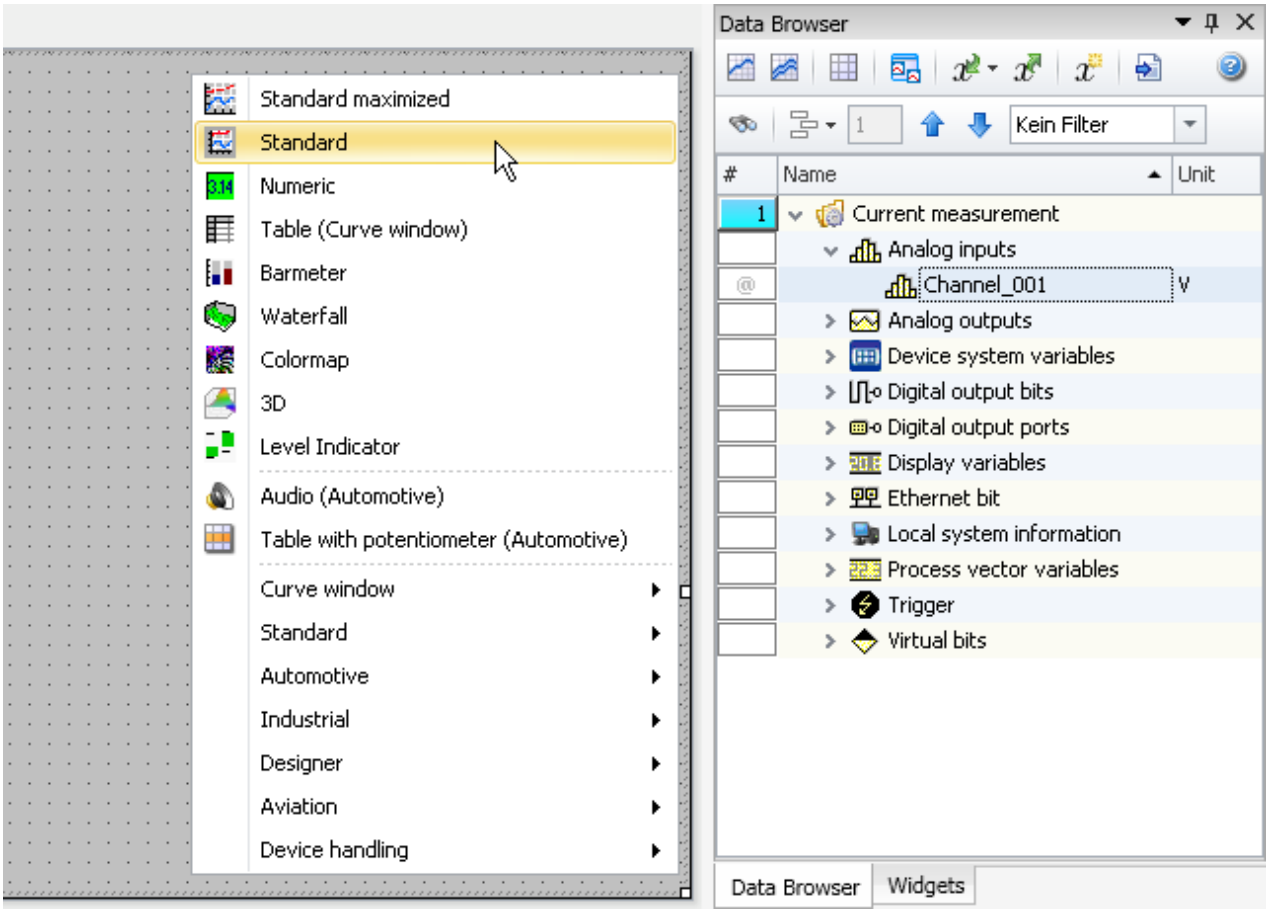
- Open the tool window Data Browser and select the analog input: *Channel_001*.
- Drag it onto the page using the Drag&Drop technique.

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.

Curve window documentation

This Widget is the familiar curve window, which is also used in other imc software packages. For more information on this topic, see curve window documentation 854.



Picture 2: Dragging an analog channel to the Panel page

There are a variety of Widgets available. The image in the example (Picture 2) may differ from your configuration.

Following selection, the selected Widget appears on the Panel page and is linked with the analog input: *Channel_001* (more information on linkage of variables is provided in the documentation under *Panel* > *Variable Linkage* 1089).

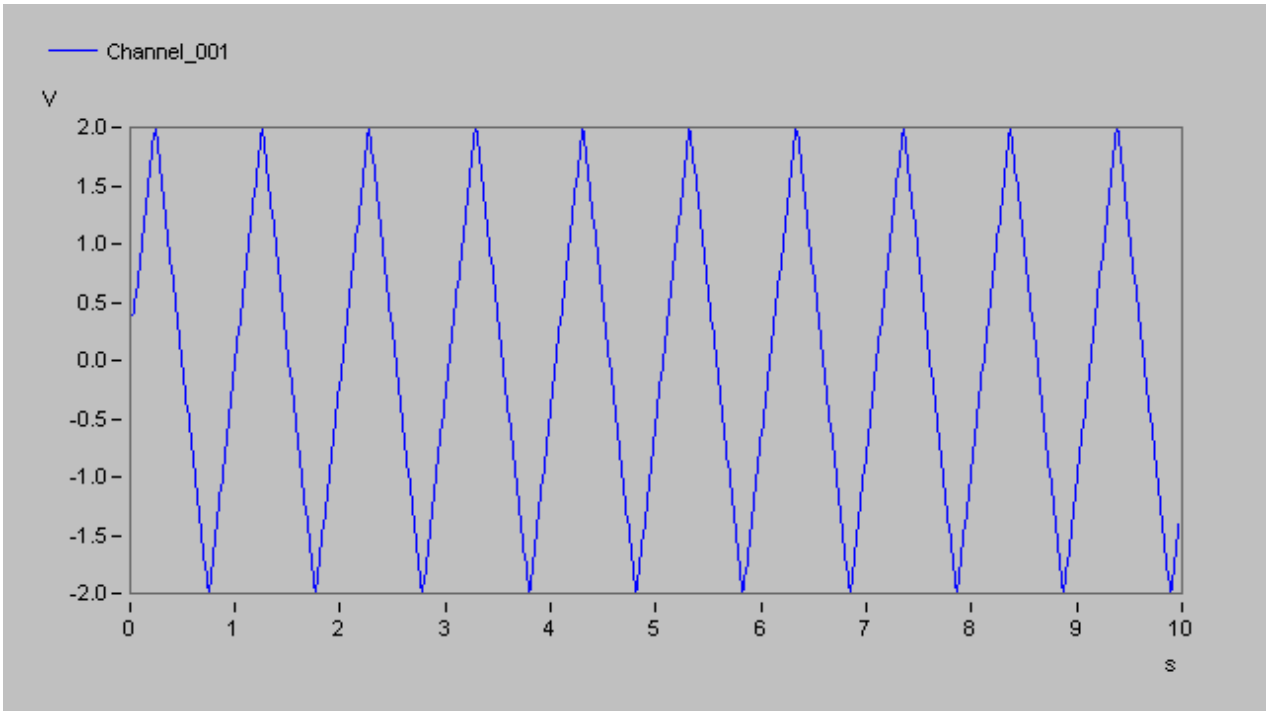
As long as the Widget is selected, a cross appears in the center (). This can be used to move the Widget or open a context menu there (by right-clicking).

- Deactivate the Design mode () (Ribbon *Panel-Design* (or for a simplified view: *Home*) > *Design Mode*).

Starting measurement

- To transfer the current settings to the device, click on the button Download 139 () (Ribbon *Home* (or *Setup/Panel-Control*) > *Download*).

- Start the measurement 139 ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Start*).



Picture 3: 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 *Home* (or *Setup/Panel-Control*) > *Stop*).

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

Save experiment

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

- Click on the button: Save as ( ) (Ribbon *Home* (or *Project*) > *Save as*).
- Save the experiment under the name "*First Steps*".

Saving the measured data

In this example, the data were not saved permanently. They are not available on the hard drive, but it is possible to save them afterward (see Subsequent saving of measured data 265). Additionally, you can activate permanent storage of the data for each channel individually before the measurement (see Pre-defined saving of measured data 265).

Note: Permanent data storage

- When data storage is activated for each channel, the measured data are saved automatically. A corresponding folder is created for each completed measurement in the Data Browser.
- 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.

5.9.2 Subsequent saving of measured data

Assignment:

Perform a measurement and save the measured data subsequently on the hard drive.

Learning goals:

- Use of subsequent data saving

Elements used:

- Save current Measurement Data

Additionally used plug-ins:

- Panel

Prerequisite:

- imc measurement device with an analog channel

Procedure:

Load the experiment *First Steps* from the tutorial Simple measurement - First Steps 260 or stop the current measurement ( ) if it is still running.

To load the experiment:

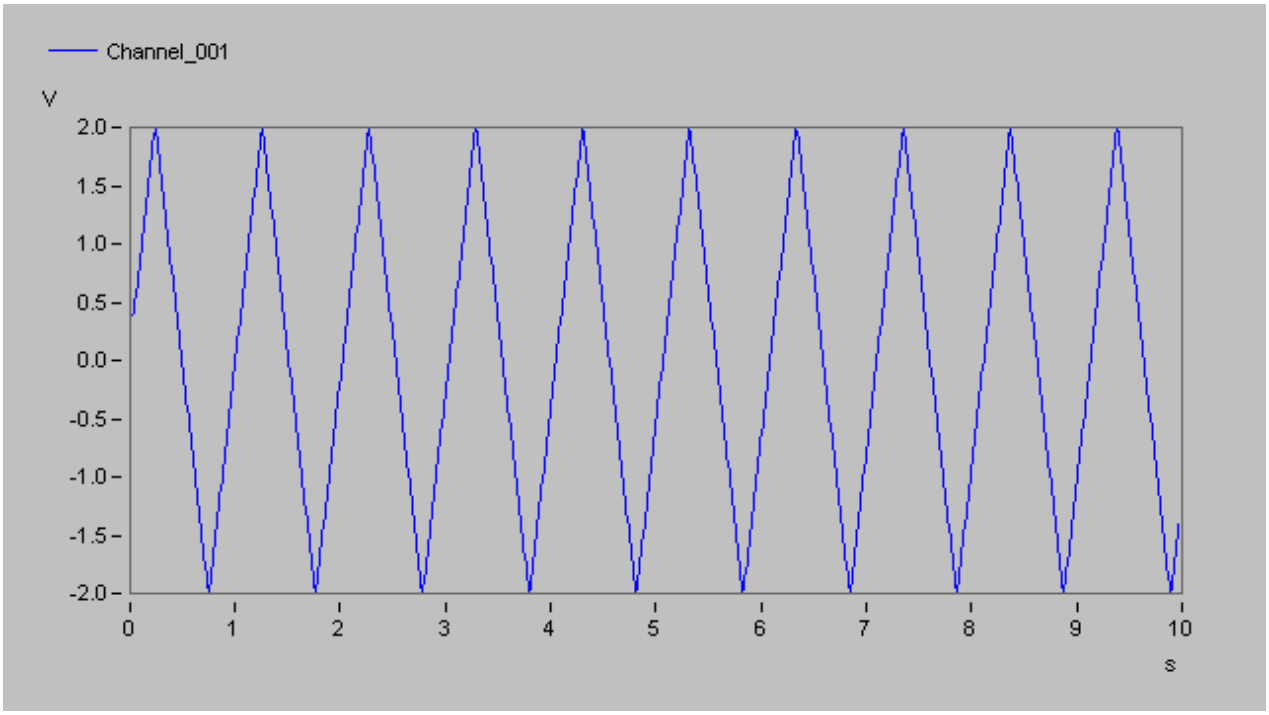
- click on the button: Open ( ) (Ribbon *Start* (or *Project*) > *Open*)
- select the experiment *First Steps* and confirm by clicking in the dialog on "Open"

5.9.2.1 Panel - Starting measurement

Open the Panel page and perform a measurement.

Starting measurement

- To transfer the current settings to the device, click on the button Download 139 ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Download*).
- Start the measurement 139 ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Start*).



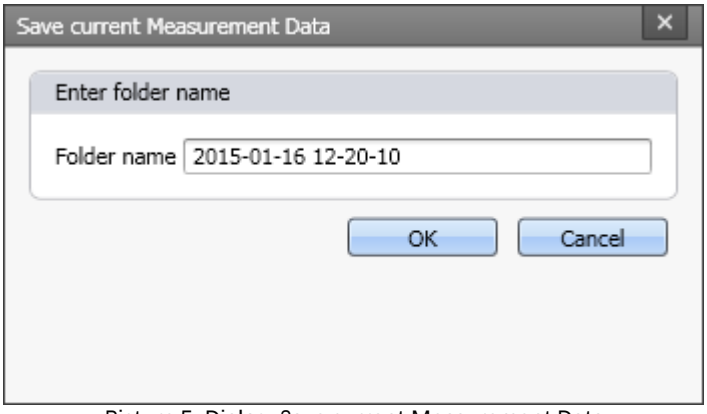
Picture 4: Curve window

After ten seconds, the measurement ends automatically.

5.9.2.2 Subsequent saving of measured data

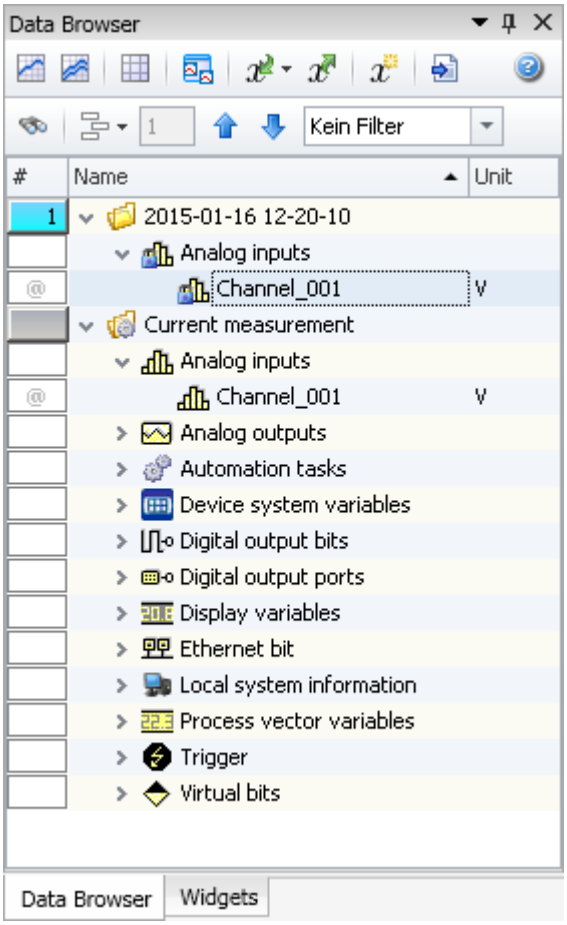
In this example, the data were not saved permanently. No data are on the hard drive, but these can be saved retroactively.

- Click on the button: Save current Measurement Data ( ) (Ribbon *Project* (or for a simplified view: *Home*) > *Save current Measurement Data*).
- 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".



Picture 5: Dialog: Save current Measurement Data

Subsequently, all channels (including analog channels, virtual channels, ...) belonging to the experiment are saved to the hard drive. 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.



Picture 6: Saved measurement data (example)

The measured data saved by means of *Save current Measurement Data* are always stored in the experiment folder. If you wish to save them to a freely selectable folder, use the button: Save current Measurement Data as (*Project* ribbon). Be aware that measured data saved in this way are not recorded in the <TERM_DATAPOOL%> and can thus not appear there.

Constraints

- 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 174 are applicable. The storage type: *Display, calculation of measured data* is used.

Reference

More information on the function: Save current Measurement Data is presented in the document: *Setup - Advanced Device Functions* > *Storage Options and Directory Structure* > *Targeted Data Saving, or Saving Subsequent to Measurement* 610.

5.9.3 Pre-defined saving of measured data

Assignment:

Configure the measurement device in such a way that the measured data are already saved while the measurement runs. Subsequently, run a measurement.

Learning goals:

- Saving measured data during measurement

Elements used:

- Channel settings: Data transfer

Additionally used plug-ins:

- Panel

Prerequisite:

- imc measurement device with an analog channel

Procedure:

Load the experiment *First Steps* or stop the current measurement ( ) if it is still running.

Save the experiment under a new name: "*First Steps with data saving*"

- To do this, click on the button: Save as ( ) (Ribbon *Home (or Project)* > *Save as*).

5.9.3.1 Setup - Settings

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

- Open the plug-in Setup.
- Go to the tab: Analog channels .

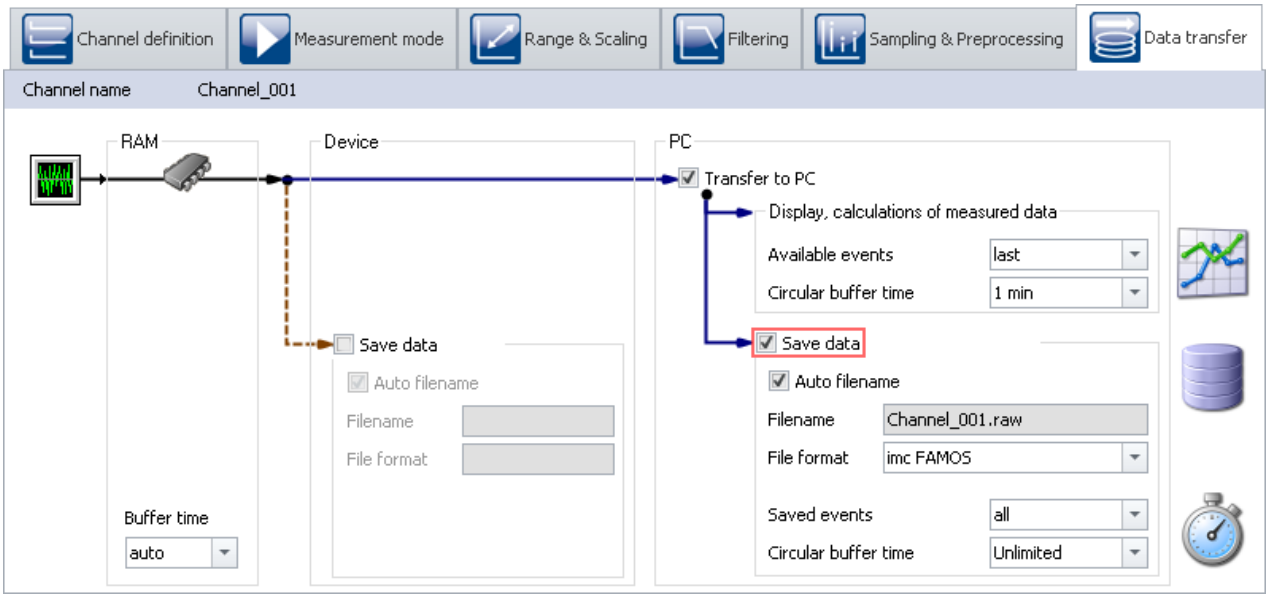
Under Channel type: Analog inputs, select *Channel_001*.

Activate 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 activated (true)	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. Ensure that you activate the type of data storage desired.



Picture 7: Activating the storage of measured data using the dialogs

- To apply the changes, click on the button Process configuration 139 (✔) (Ribbon *Home* (or *Setup-Control*) > *Process*).

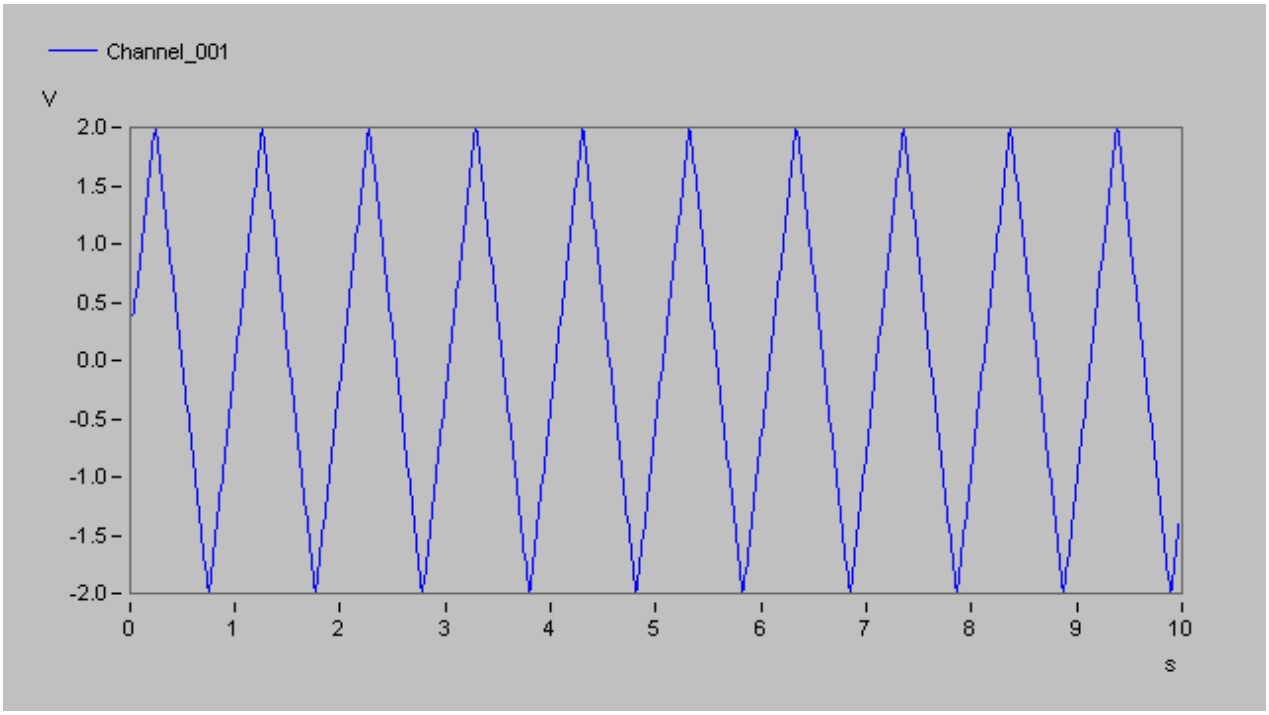
By this means, the device settings are available to other plug-ins.

5.9.3.2 Panel - Starting measurement

Open the Panel page and perform a measurement.

Starting measurement

- To transfer the current settings to the device, click on the button Download 139 ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Download*).
- Start the measurement 139 ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Start*).

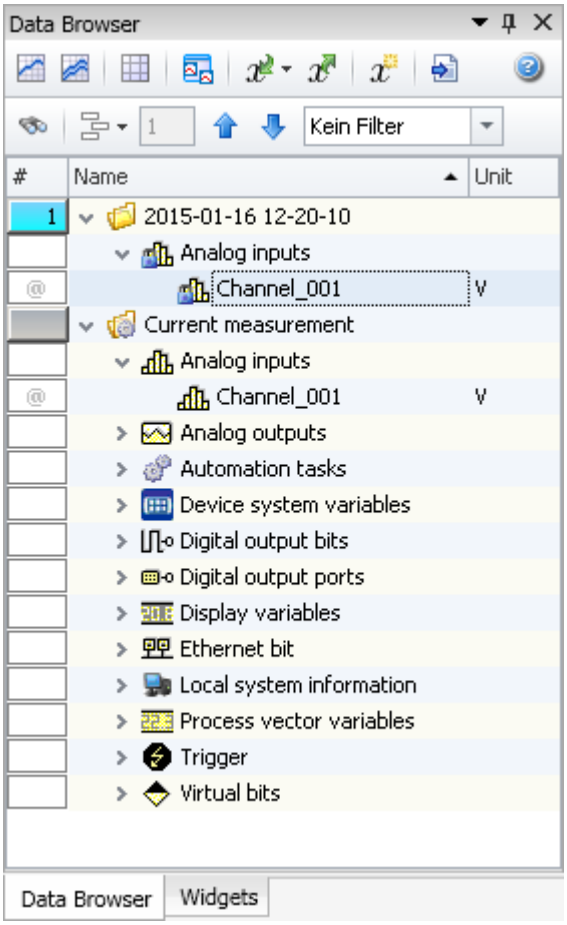


Picture 8: Curve window

After ten seconds, the measurement ends automatically.

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:



Picture 9: Saved measurement data (example)

Note: Permanent data storage

- When data storage is activated for each channel, the measured data are saved automatically. A corresponding folder is created for each completed measurement in the Data Browser.
- 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.

Reference

More information on activating the data storage is presented in the chapter: *Data transfer* 174.

5.9.4 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:

- Connection to the device
- Channel settings
- Trigger

Additionally used plug-ins:

- Panel

Prerequisite:

- imc measurement device with an analog channel

Procedure:

Begin by starting imc STUDIO.

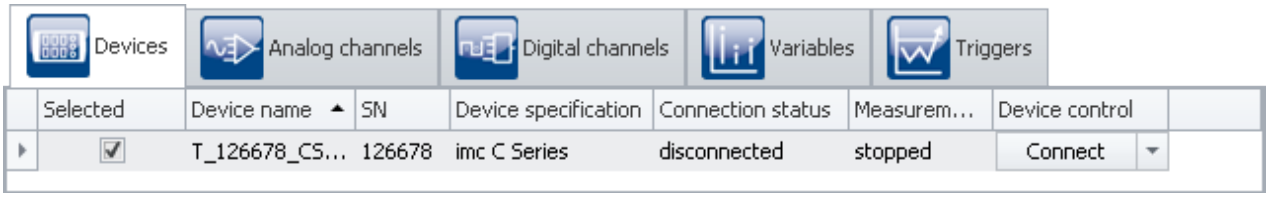
5.9.4.1 Setup - Settings

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

- Open the plug-in Setup.
- Go to the tab: Devices .

Select your device:

- If the device desired does not appear in the list, then perform a device search 139 ( ) (Ribbon *Home* (or *Setup-Control*) > *Search for Devices*). Subsequently, the list will display all devices found.
- Click on the Checkbox-symbol ( ) in order to select the device.



Selecting a device

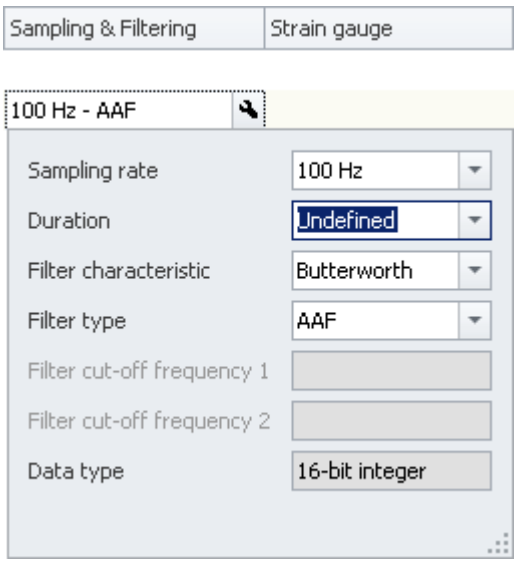
The device is now known and selected, and is available each time imc STUDIO is started.

Configuring the channels

- Go to the tab: Analog channels .

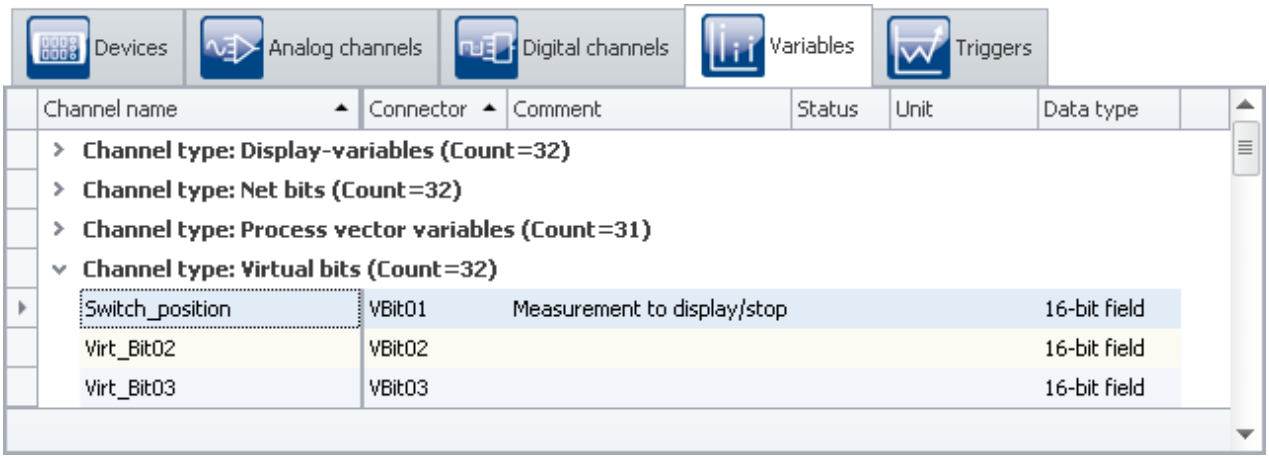
Under Channel type: Analog inputs, select *Channel_001*. By default, this is set to *active*.

- 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 ( ).



Picture 10: Unlimited measurement duration

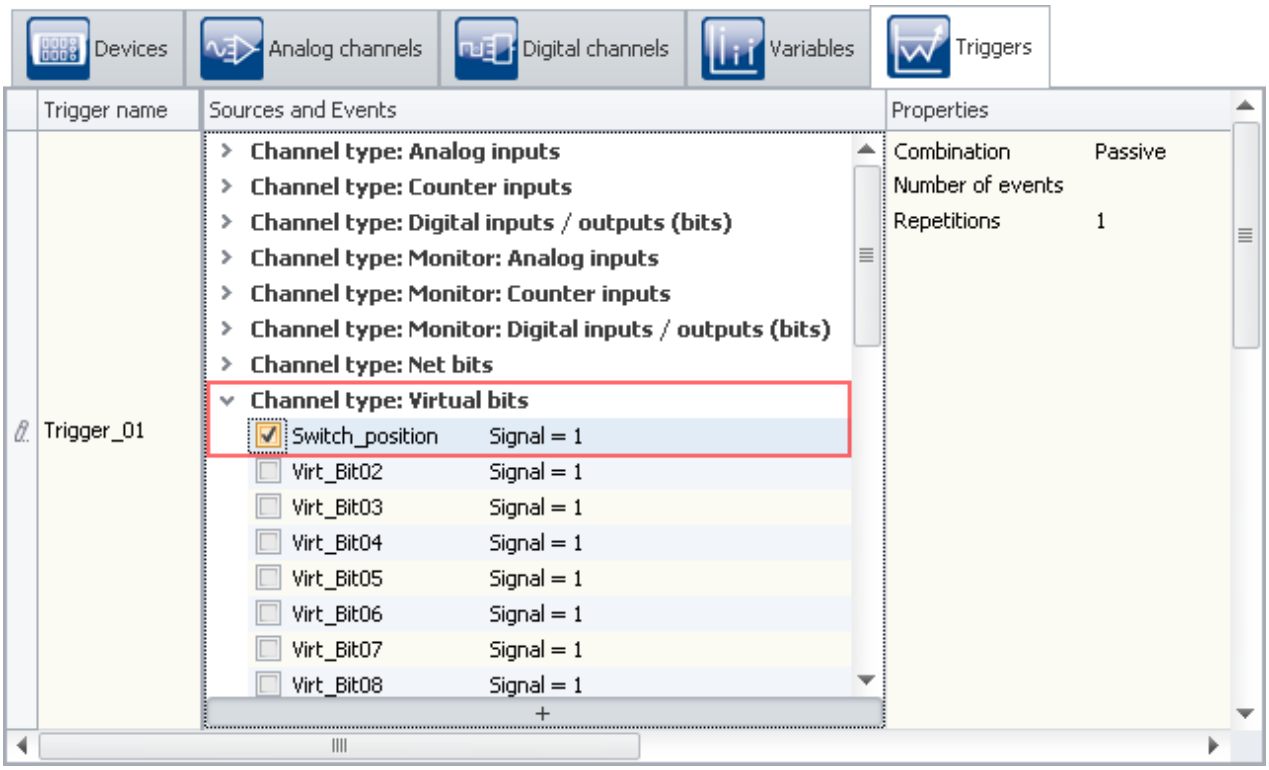
- Go to the tab: Variables .
- Under Channel type: Virtual bits, click on *Virt_Bit01* and rename it to *Switch_position*. If desired, you can enter a comment (*Measurement to display/stop*), as shown in the Picture 11:



Picture 11: 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* (see Picture: 12)

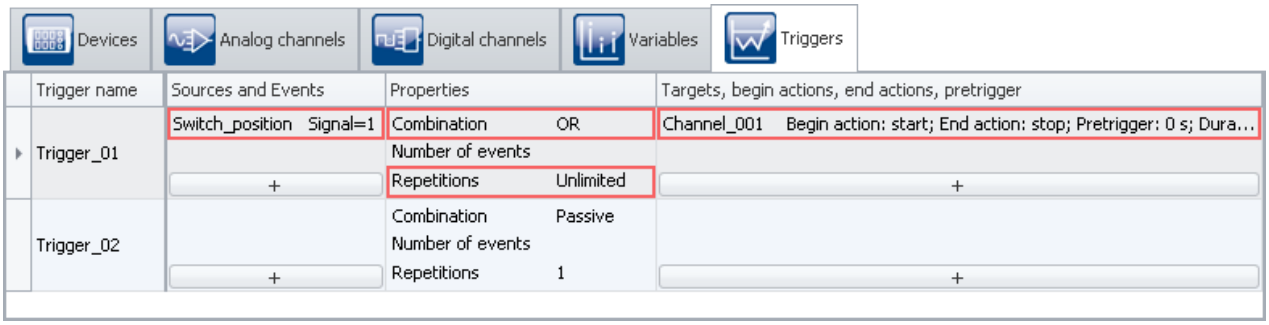


Picture 12: Trigger source

- In order to be able to release the trigger multiple times, in the column Properties set the Repetitions at *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 Picture 13.

Parameter	Value
Trigger begin action	start
Trigger end action	stop



Picture 13: Settings for the trigger

- To transfer the current settings to the device, click on the button Download 139 ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Download*).

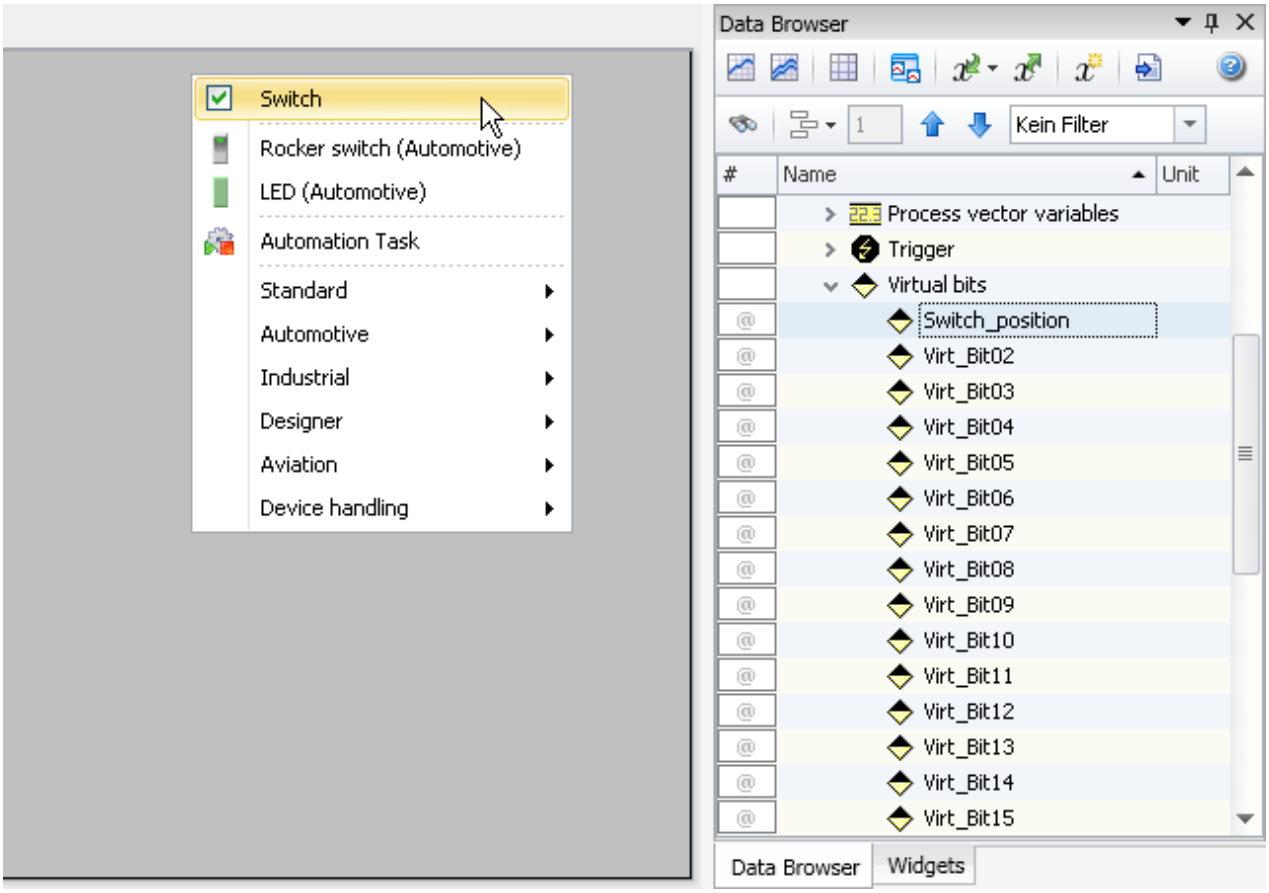
5.9.4.2 Panel - Settings / Start measurement

Now you can create the Panel page.

- Open the plug-in Panel.
- Activate the Design mode ( ) (Ribbon *Panel-Design* (or for a simplified view: *Home*) > *Design Mode*).

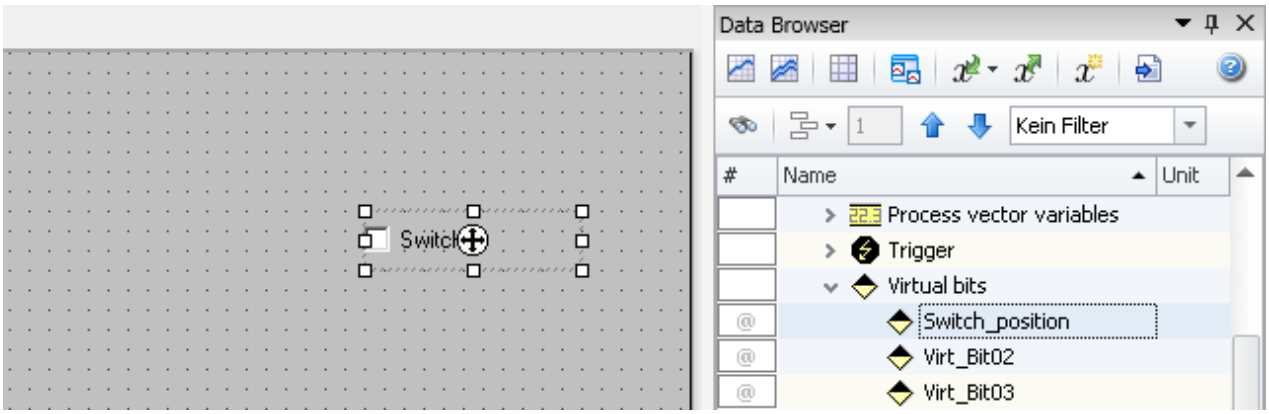
Switch

- 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.



Picture 14: 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.

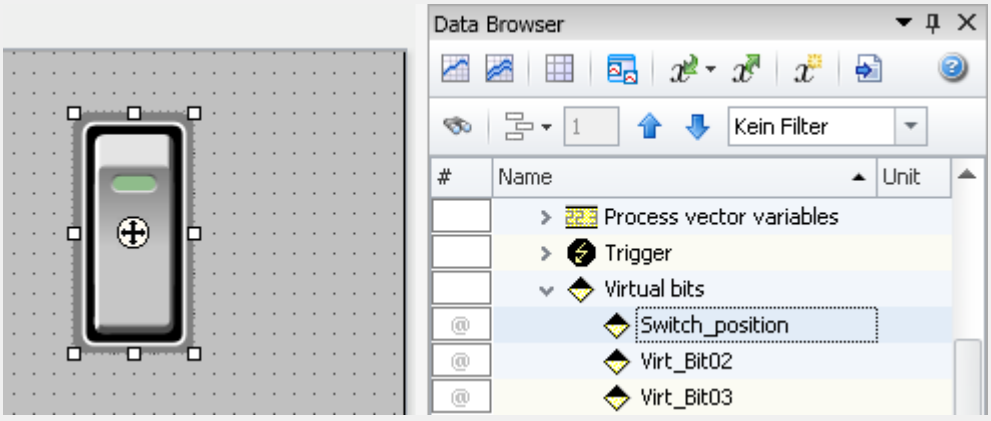


Now the switch on the Panel-page is linked with the virtual bit in the device named *Switch_position*. Using the switch, you can interactively switch the virtual bit on and off.

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*".

Additional Widgets

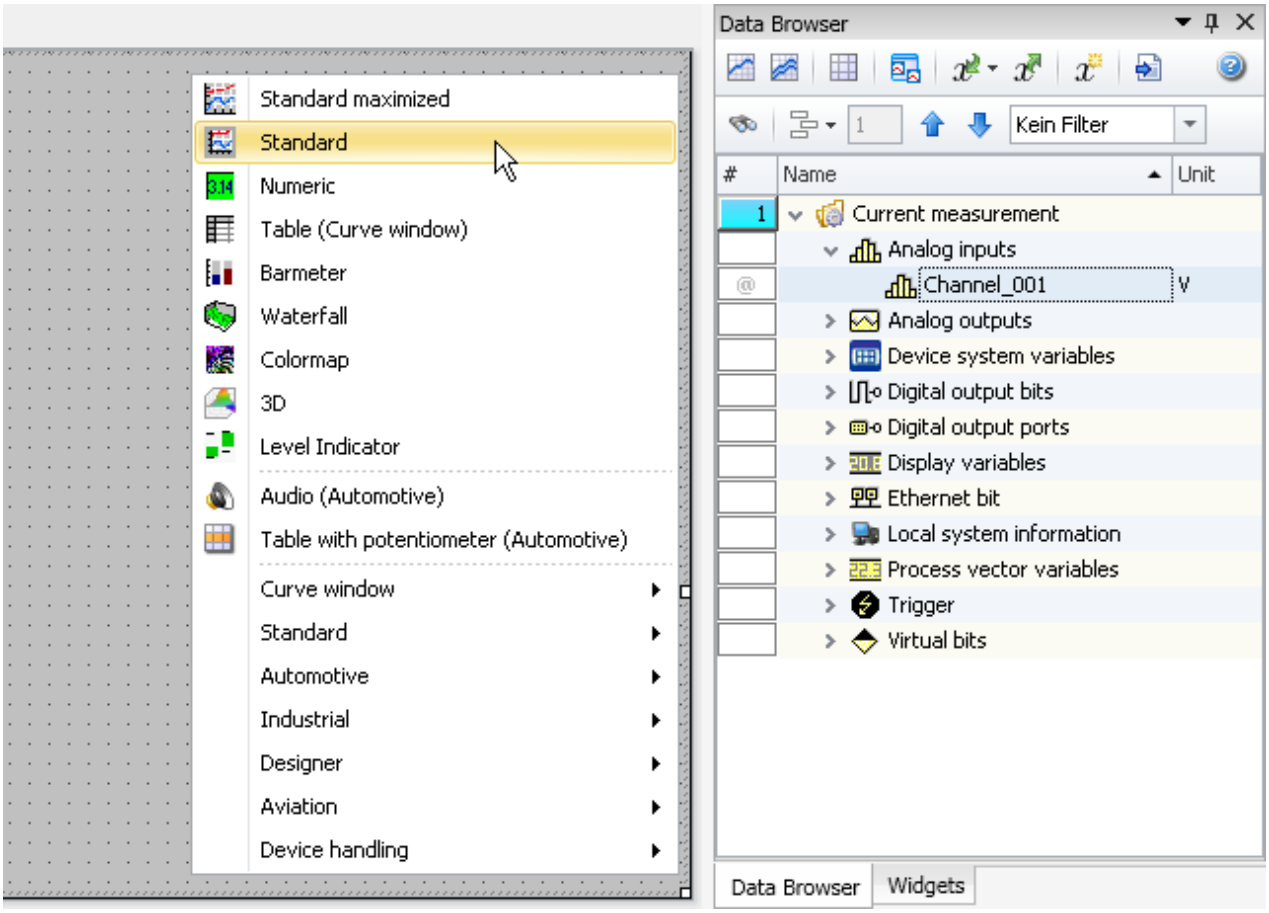
Depending on the product configuration, additional Widgets are available. Instead of a simple switch, you could select a rocker switch, for example, under the heading "Automotive":



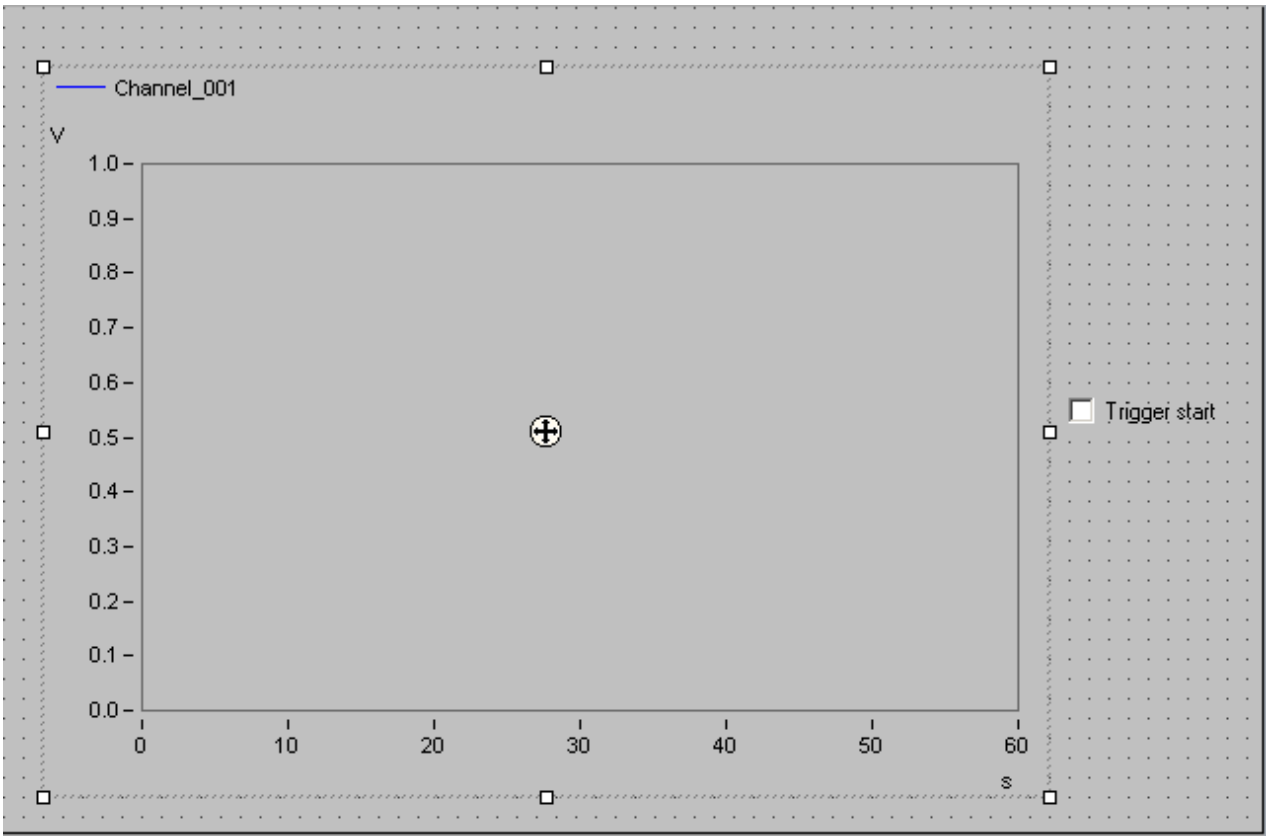
Curve window

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.



Picture 15: Dragging an analog channel to the Panel page



Picture 16: 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 () (Ribbon *Panel-Design* (or for a simplified view: *Home*) > *Design Mode*).

Starting measurement, clicking buttons, stopping

- Start the measurement 139 () (Ribbon *Home* (or *Setup/Panel-Control*) > *Start*).

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 () (Ribbon *Home* (or *Setup/Panel-Control*) > *Stop*).
- Click on the button: Save as () (Ribbon *Home* (or *Project*) > *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: Simple Triggering - Enhanced 279.

5.9.5 Simple Triggering - Enhanced

Assignment:

A text box on the Panel page displays the measurement's state (corresponding with *Switch_position* 273).
The states are (*running*, *stopped*).

Learning goals:

- Observation of the measurement's status on the Panel page

Elements used:

- Zones

Additionally used plug-ins:

- imc STUDIO Panel

Prerequisite:

- This is an enhancement of the experiment: *Simple Triggering* 272, which must already exist.
- The same conditions apply as for the experiment: *Simple Triggering* 272

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 ( ) (Ribbon *Home (or Project)* > *Save as*).

5.9.5.1 Panel - Settings / Start measurement

- Open the plug-in Panel.
- Activate the Design mode ( ) (Ribbon *Panel-Design* (or for a simplified view: *Home*) > *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*".

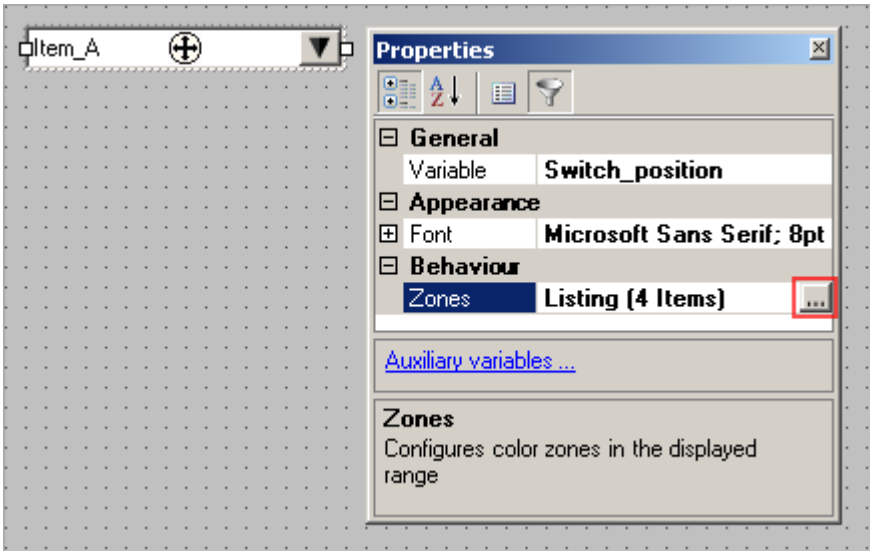
- Under the property: "Variable", select the virtual bit "*Switch_position* 273".

You have now established a link between the switch and the listbox. When you click on the switch, the listbox indicates the applicable status.

Setting up and adapting states (zones)

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

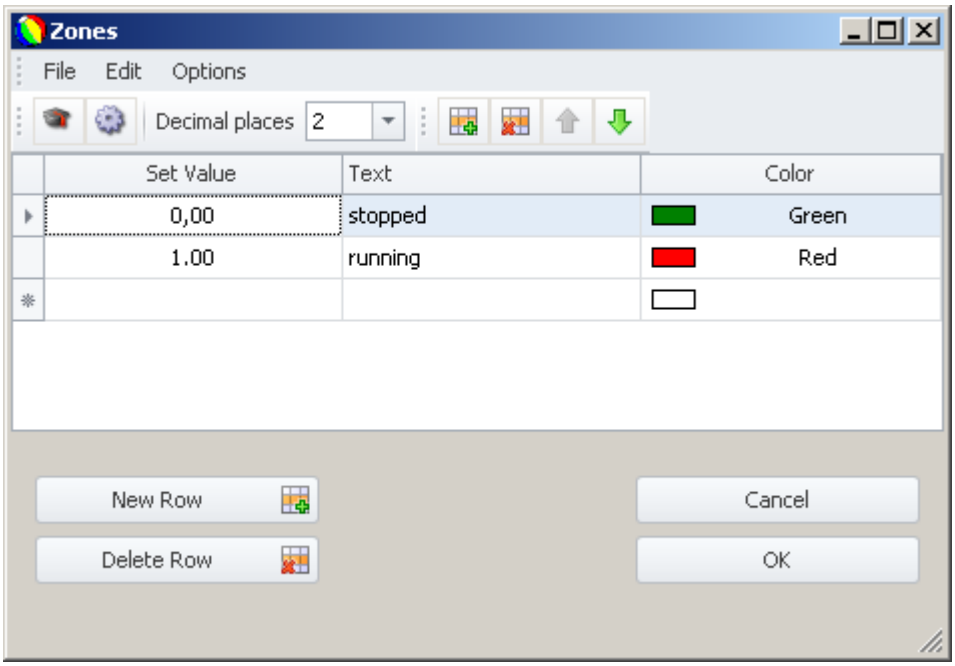
- Click on the symbol ... as shown in Picture 17:



Picture 17: 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*".

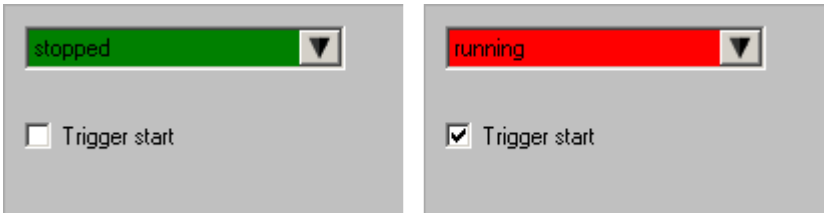
- Only two zones are required; begin by deleting all but two zones
- Enter the zones as shown in Picture 18 (the color is not relevant for the example):



Picture 18: Zones

- Deactivate the Design mode ( ) (Ribbon *Panel-Design* (or for a simplified view: *Home*) > *Design Mode*).
- Start the measurement 139 ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Start*).
- then click on the switch on the Panel page

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



With this, the goal of this enhancement has been met.

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 ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Stop*).
- Click on the button: Save ( ) (Ribbon *Home* (or *Project*) > *Save*).

5.9.6 Artificial sine

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

Additionally used plug-ins:

- Panel

Prerequisite:

- imc measurement device with an analog channel

Procedure:

Begin by start imc STUDIO.

5.9.6.1 Setup - Settings

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

- Open the plug-in 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

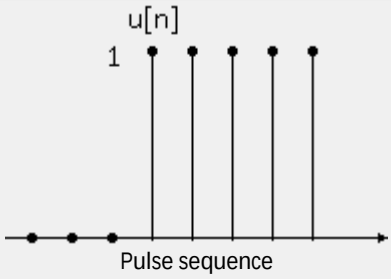
- Open imc Online FAMOS ( ) (Ribbon *Home* (or *Setup-Configuration*) > *Online FAMOS*)
- Enter the following source code in the imc Online FAMOS Editor:

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

Explanation

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 .

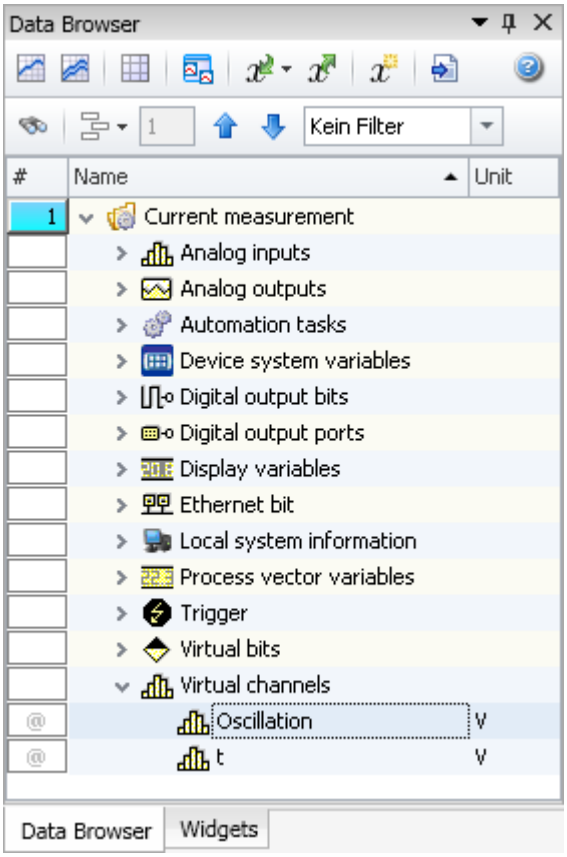
- To transfer the current settings to the device, click on the button Download 139 ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Download*).

5.9.6.2 Panel - Settings / Start measurement

Now you can create the Panel page.

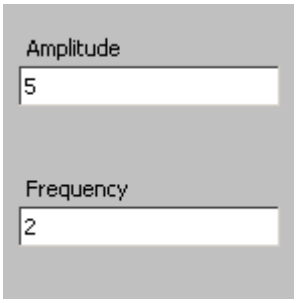
- Open the plug-in Panel.
- Activate the Design mode (🖊) (Ribbon *Panel-Design* (or for a simplified view: *Home*) > *Design Mode*).

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*.

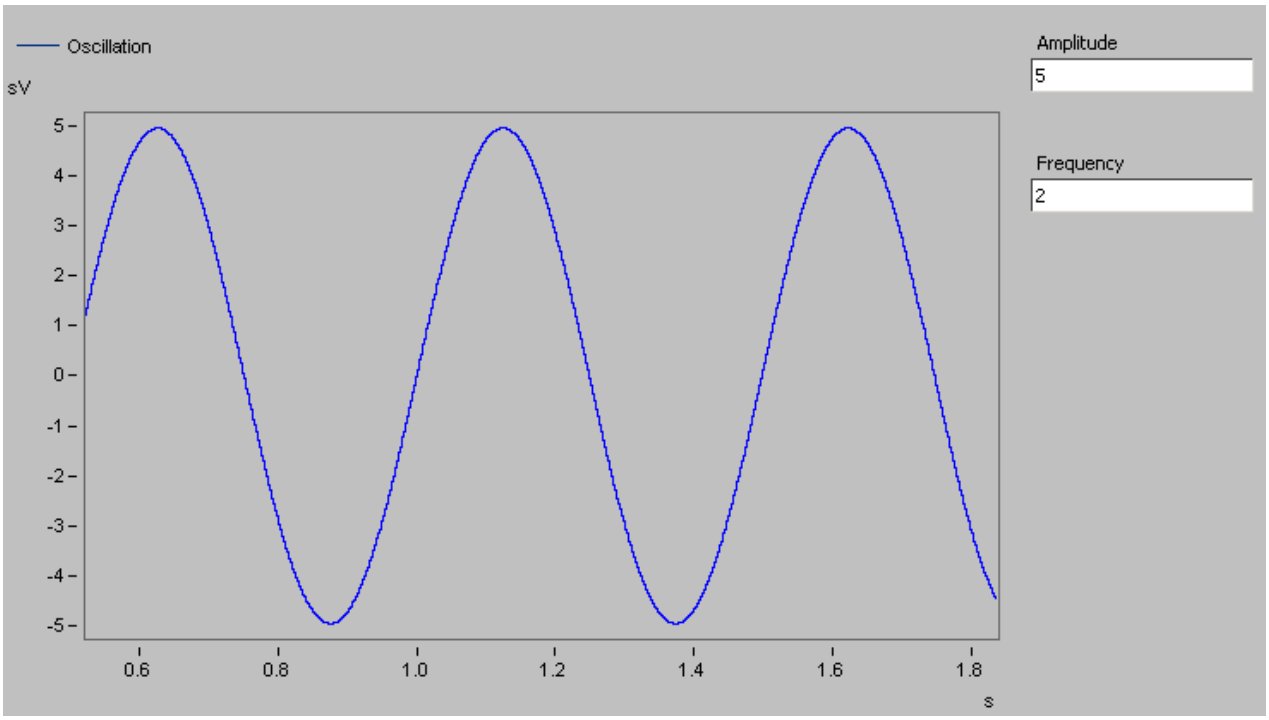
To adapt the Text, open the label's Properties.

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

Display the variable *Frequency* in the same way.

Start measurement, observe results

- Deactivate the Design mode ( ) (Ribbon *Panel-Design* (or for a simplified view: *Home*) > *Design Mode*).
- Start the measurement 139 ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Start*).
- change the *Amplitude* and the *Frequency* and observe how the curve window changes (see Picture 19)



Picture 19: Sinus

Saving

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

- Stop the measurement ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Stop*).
- Save ( ) the experiment under the name "*Artificial Sine*".

5.9.7 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:

- Connection with a device
- Channel settings
- Monitor channel

Additionally used plug-ins:

- Panel

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.

5.9.7.1 Setup - Settings

To begin, the measurement device is configured.

- Open the plug-in 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:

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.

Defining triggers

- Go to the tab: Triggers .

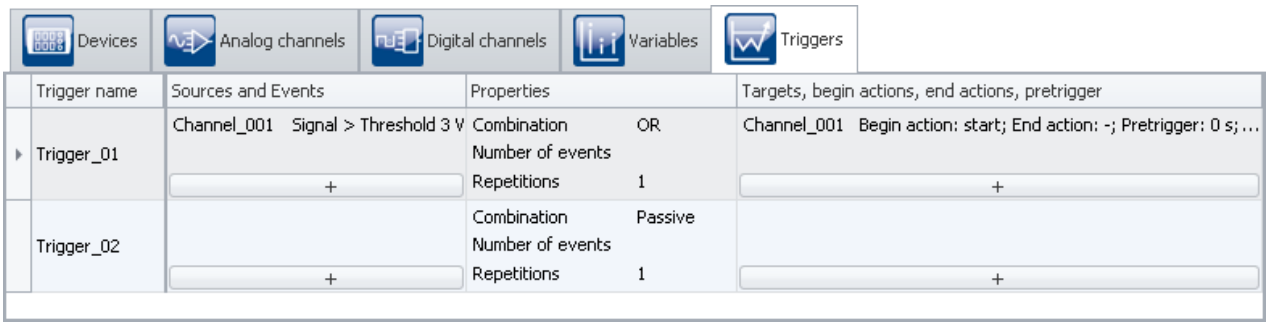
In the column Sources and Events set *Channel_001* as the source, with the following event:

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

In the column Properties set the Combination to *OR*, since only a single event is used.

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



Picture 20: Settings for the trigger

- To transfer the current settings to the device, click on the button Download 139 ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Download*).

5.9.7.2 Panel - Settings / Start measurement

Now you can create the Panel page.

- Open the plug-in Panel.
- Activate the Design mode ( ) (Ribbon *Panel-Design* (or for a simplified view: *Home*) > *Design Mode*).

Create a curve window containing *Channel_001*:

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

Curve window properties

More Waveforms

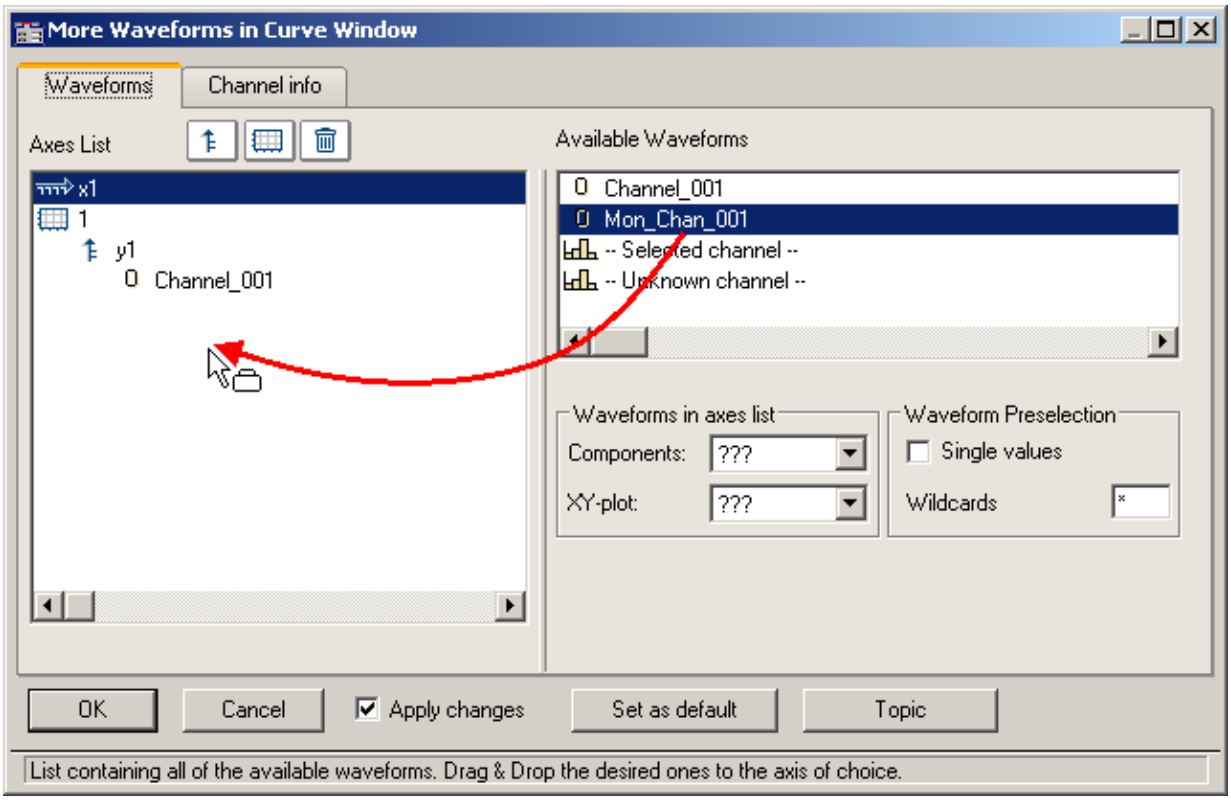
In a curve window it is possible to display more than one channel.

- 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 Waveforms...

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

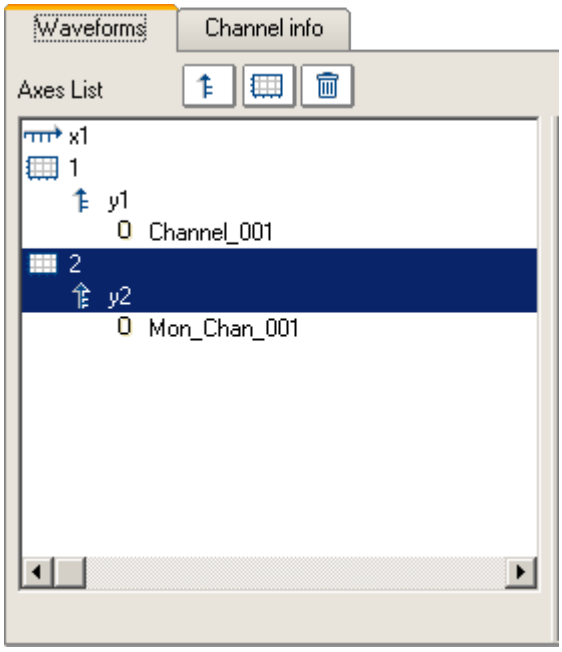
All data sets available for selection are displayed in the list at right (Available Waveforms). The data sets already displayed in the curve window are indicated in the list at left (Axes List).

- Drag *Mon_Chan_001* from Available Waveforms (see Picture 21) to the box Axes List.



Picture 21: Waveforms 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*".

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.

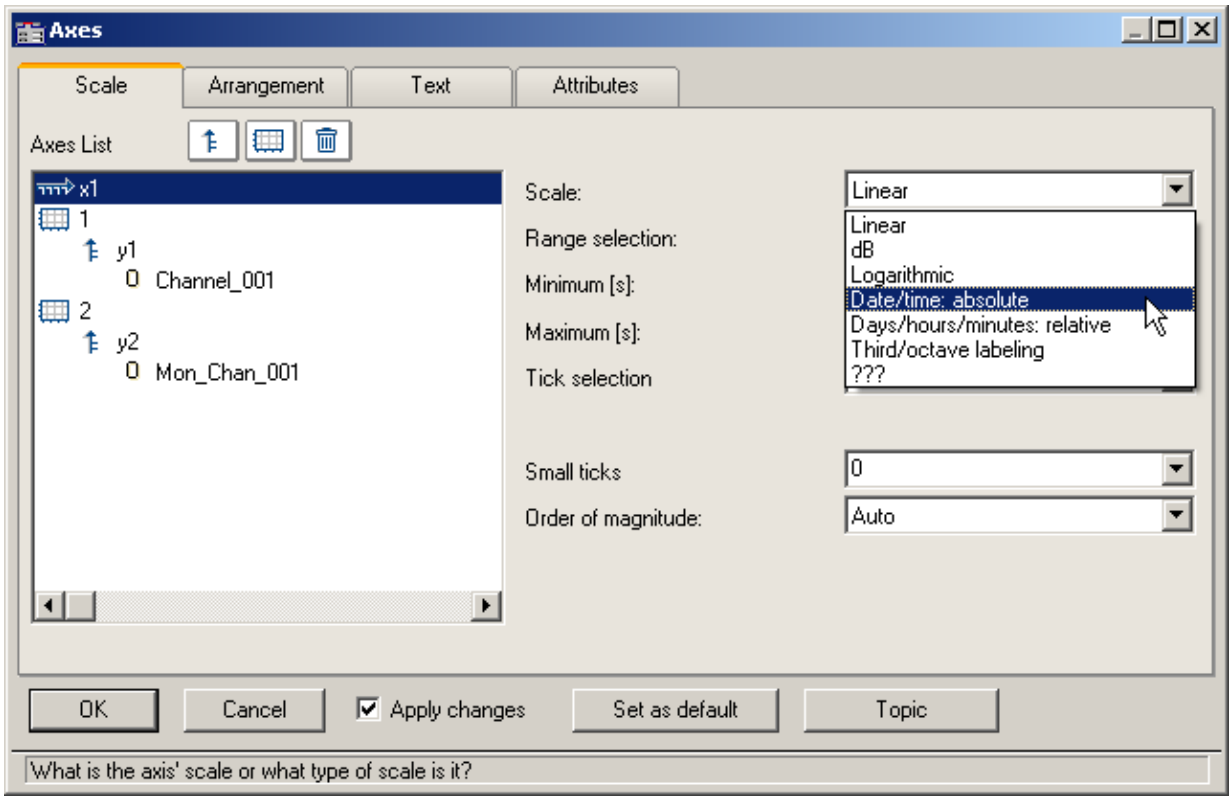
- Right-click he mouse over the curve window (not over the cross ( ) in the center).
- In the context menu which then appears, select Configuration and subsequently the command Axes...

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



Picture 22: Setting the axis for the curve window

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

Reference

- In the curve window's context menu, extensive help for using the curve window is available.
- The documentation can also be accessed via the imc STUDIO help:
  Panel > *Special Widgets* > *Curve Window* 854

Starting measurement, observing results

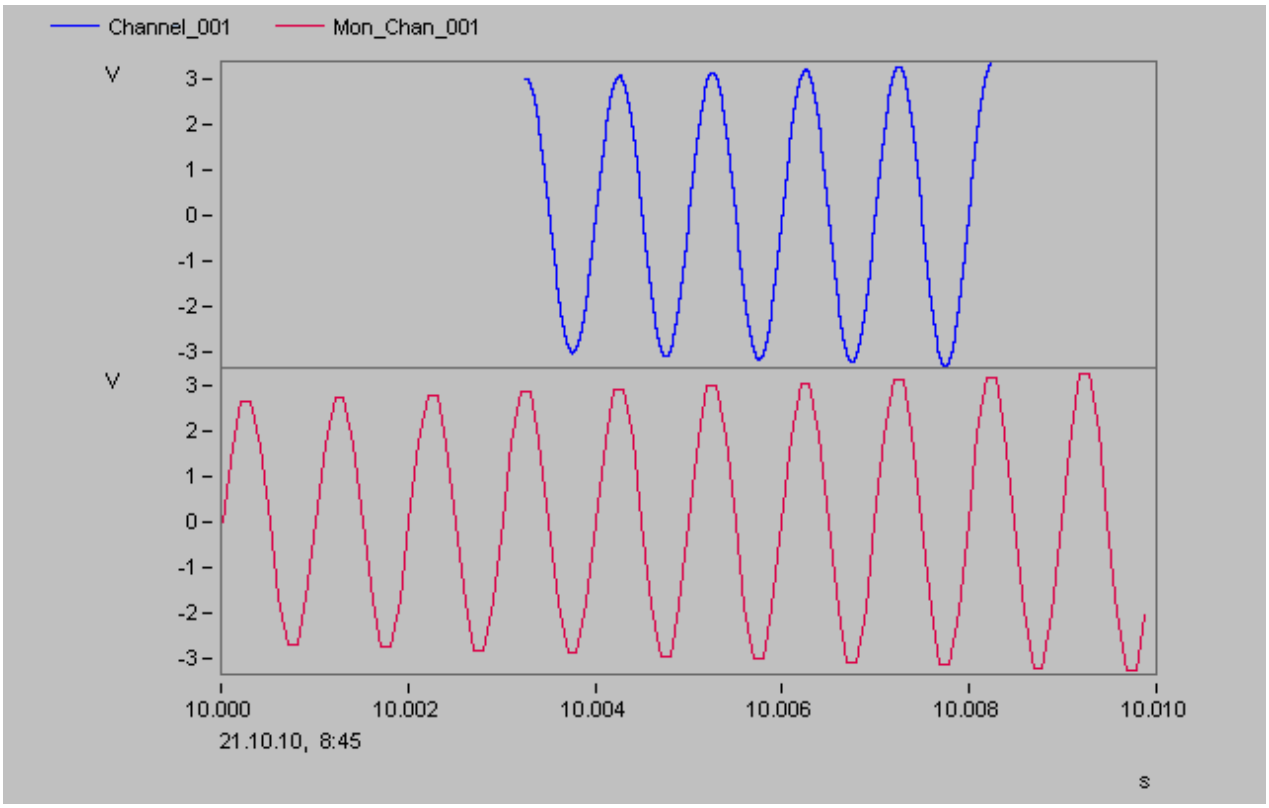
- Deactivate the Design mode ( ) (Ribbon *Panel-Design* (or for a simplified view: *Home*) > *Design Mode*).
- At the input: *Channel_001*, for example, apply a sinusoidal signal with an amplitude of < 3 V an
- Start the measurement 139 ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Start*).
- after 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 ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Stop*).

To investigate the data, you need to zoom in on the correct location in the curve window.

- Use the context menu item Edit > Zoom



Picture 23: Measurement of the channel's signal only starts once the trigger event occurs

Saving

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

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

5.9.8 Thermocouple measurement

Assignment:

A temperature is to be measured by thermocouple.

Learning goals:

- Advanced channel settings

Element used:

- Channel settings
- Sensor & scaling
- Pre-processing of data

Additionally used plug-ins:

- Panel

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.

5.9.8.1 Setup - Settings

To begin, the measurement device is configured.

- Open the plug-in 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.

- To transfer the current settings to the device, click on the button Download 139 ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Download*).

5.9.8.2 Panel - Settings / Start measurement

Now you can create the Panel page.

- Open the plug-in Panel.
- Activate the Design mode ( ) (Ribbon *Panel-Design* (or for a simplified view: *Home*) > *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:

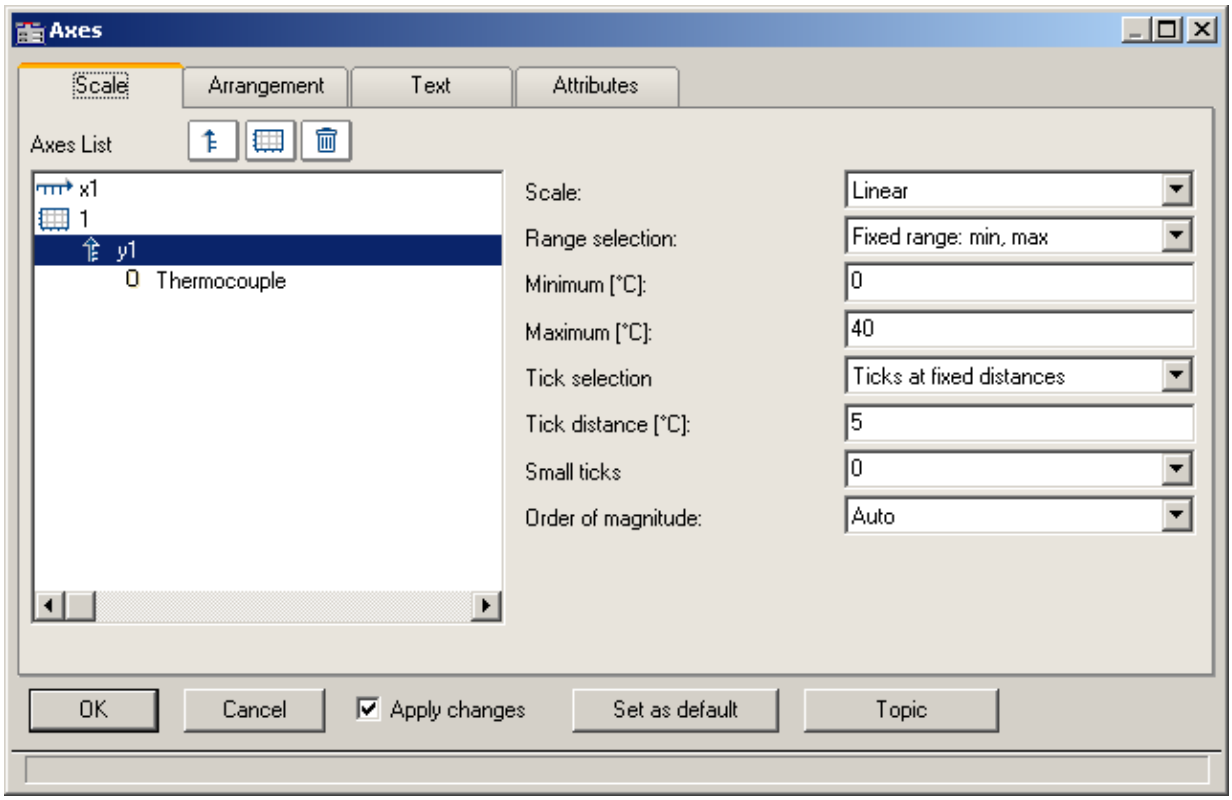
- Right-click the mouse over the curve window (not over the cross ( ) in the center).
- In the context menu then appears, select Configuration and subsequently the command Axes...

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



Picture 24: Settings for the y-axes

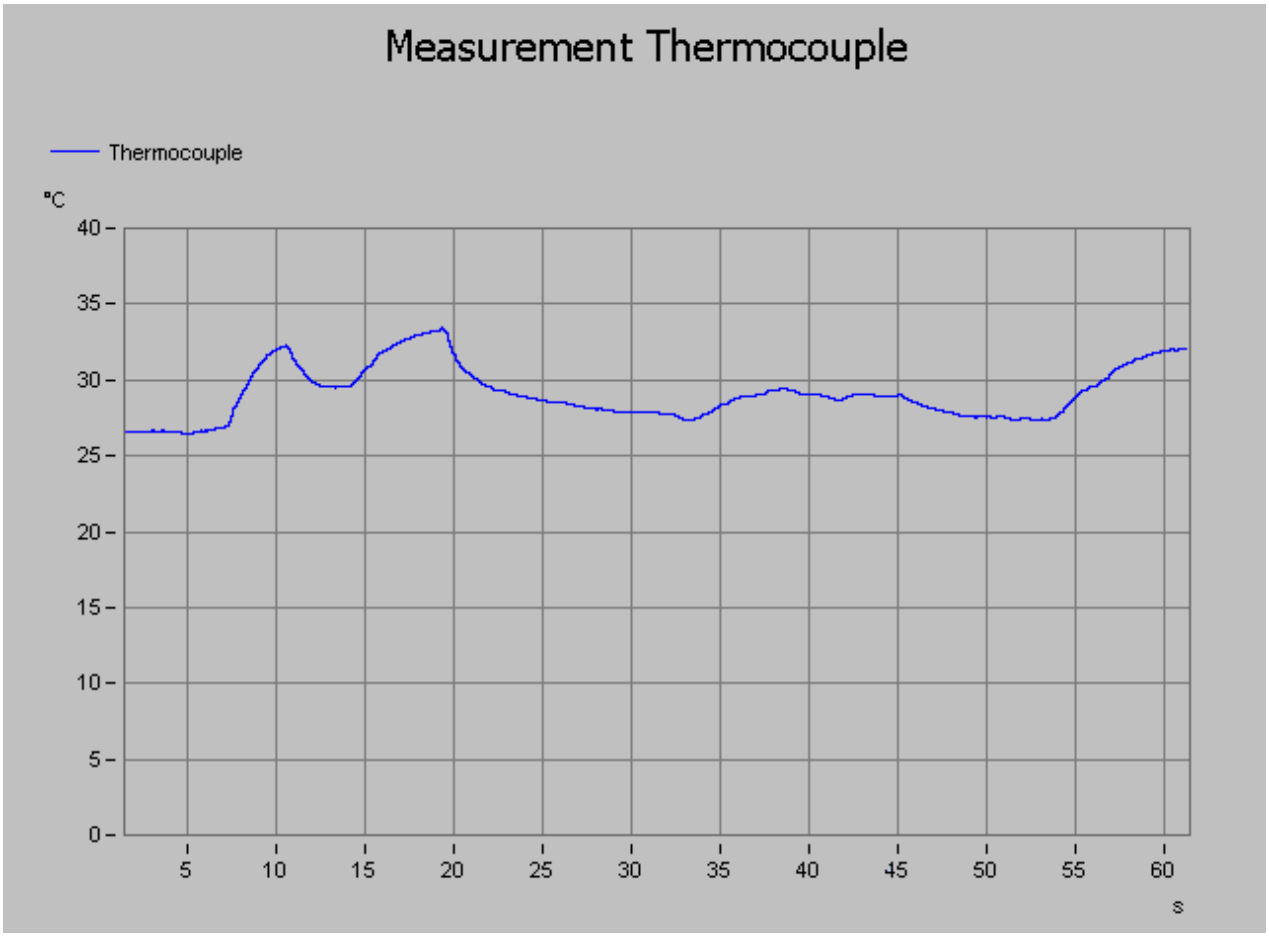
In order to apply the settings, click on the button "*OK*".

- 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 ( ) (Ribbon *Panel-Design* (or for a simplified view: *Home*) > *Design Mode*).

Starting measurement, observing results

- Start the measurement 139 ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Start*).

Picture 25 illustrates one possible display of the course of the temperature signal.



Picture 25: Curve window with plot of temperature

Saving

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

- Stop the measurement ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Stop*).
- Save ( ) the experiment under the name "*Thermocouple measurement*".

6 Setup - Connect the device

There are multiple ways to connect the imc measurement devices with the PC. In most cases, the connection via LAN 295 (local area network, Ethernet) is implemented. See section Connecting to LAN in four steps 296 for the quickest way to connect PC and measurement device.

But there are also other connection types:

- Modem 321 (TCP/IP with PPP)
- WLAN 313
- Radio modem 332 (GSM, mobile phone network)
- Null modem 330 (serial interface)

These are described in a separate chapter: Special options for connecting to the device 301.

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, Wireless LAN and modem, observe the remarks in the chapter Computers with multiple TCP/IP connections 306.

6.1 Connection via LAN

To connect via LAN there are two options:

1. The measurement device is connected to an existing network, e.g. via network switch or hub.
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. For both variants, different cables may be required! Modern PCs and network switches are usually equipped with Auto-MDI(X) automatic crossover recognition, so that having multiple various cables is no longer necessary.

1. Connecting to an existing network:
   - An uncrossed UTP cable, e.g. CB-UTP-3 is necessary
   - Connect the measurement device's LAN jack to a switch. Only with a hub or switch is it possible to run multiple devices.
2. Direct connection to the PC (point-to-point):
   - If you are using older PCs with AUTO-MDI(X) Crossover Detection, you need a special "crossed" UTP-cable (Category 5, RJ45 connector, e.g. CB-UTPX-3)
   - Connect the measurement device's LAN connector directly with the PC's LAN jack

Recommended network configuration:

If possible, always use up-to-date network equipment in order to achieve the maximum transfer bandwidth. The current standard (2014) is: 100BASE-T (Fast Ethernet 100 MBit/s) or 1000BASE-T (GBit Ethernet). GBit Ethernet network equipment (switch) is backwards compatible, so that imc devices can be operated with 100MBit Fast Ethernet.

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 100m 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. If this is not the case, you should use network bridging.

6.2 Connecting via LAN in four steps

The most common case is described below: the PC and the device are connected via cable or 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 knows the device's hardware configuration. In that case, experiment configurations can be prepared without any connection to the device.

Step 1: Determining the PC's IP-address

Before starting the configuration of your measurement device, you should determine your computer's IP-address (the following screenshots and texts refer to MS Windows 10). There are multiple ways to do this, of which some may not be possible on your PC, depending on the administrator rights you have. In such cases, you should contact your responsible administrator/IT-service.

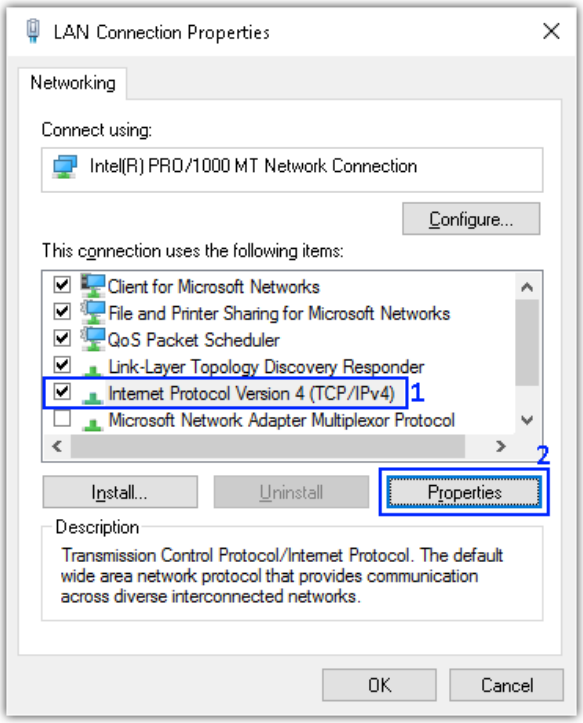
Before determining the IP-address, connect the measurement device with the PC and activate it.

To open the configuration dialog, use one of the following:

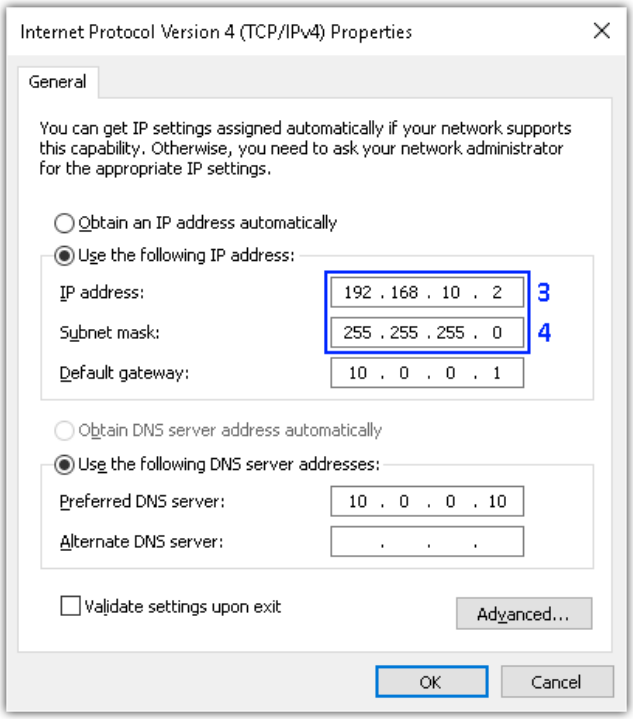
- Open the Windows Control Panel and search for "View network connections"
  - Select "View network connections"
- Open Windows' input box by using the keyboard combination of Windows-key + R `[Win-R]`.
  - Enter the following command in the edit box:
    `control netconnections`

The "Network Connections" window appears. Then right-click the mouse over the entry for your network connection and then select the item "Properties" in the context menu once again. Then the Connection Properties window appears.

Select Version 4 of the Internet Protocol Version 4 (TCP/IP4) and click on Properties. Now the current settings are visible. Please note the computer's IP address (3) and the subnet mask (4).



Select Internet Protocol (TCP/IP)



Settings for TCP/IP

Warning

Be careful in changing settings, since if the same network card is used, there may be difficulties later in the company network. Please discuss this with your responsible administrator/IT-service.

Note: Obtain IP-address automatically (DHCP)

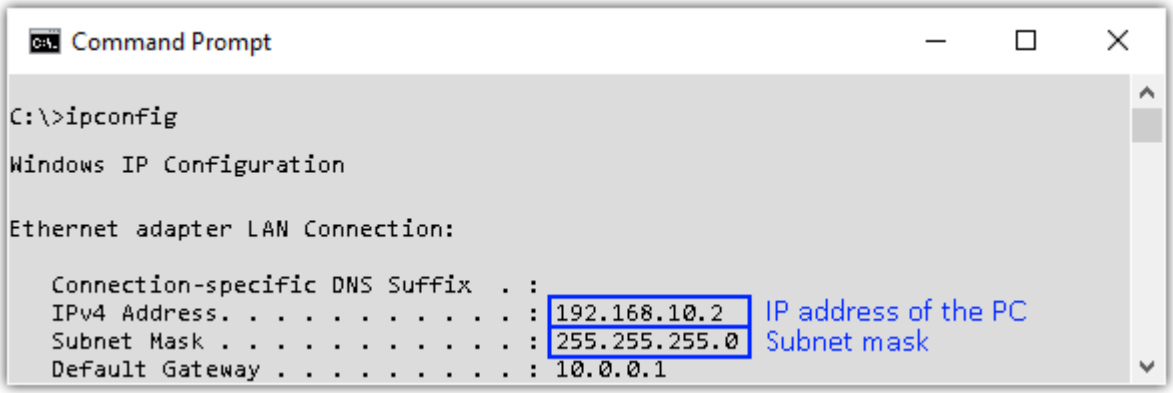
If "Obtain IP address automatically" (DHCP) is selected, no IP address is displayed. In this case, you have to determine the current IP-address using the Command Prompt/Command line.

Note, however, that automatically obtained IP-addresses might change the next time the operating system is started!

Determine the IP address using the Command line

Open Windows' input box by using the keyboard combination of Windows-key + R `[Win-R]`. Type `cmd` in the text box and confirm by pressing Return. In the command line window which then appears, type in `ipconfig`.

Now you can read the IP-address of the desired network connection:



Result of the command "ipconfig"

For the present case, the PC uses the IP 192.168.10.2 with the subnet mask 255.255.255.0. For this reason the measurement device should have a number which differs from the PC in the last block.

In the example shown, the fixed IP 192.168.10.2 with subnet mask 255.255.255.0 is selected. For measurement devices, any numbers would be suitable which begin with 192.168.10. and then do not contain 0, 2, or 255. The 0 and the 255 should not be used, if possible, due to their special significance, and 2 is the computer's number and so not suitable for the measurement device.

Step 2: Connecting the measurement device

When you connect the measurement device directly to your PC, you may need to use a "crossed" network cable in some circumstances. If the measurement device is connected to the network via a network hub or switch, or a patch box, use an uncrossed network cable. Modern PCs and network switches are able to switch electronically. Hence you can use both cable types.

Step 3: IP-configuration

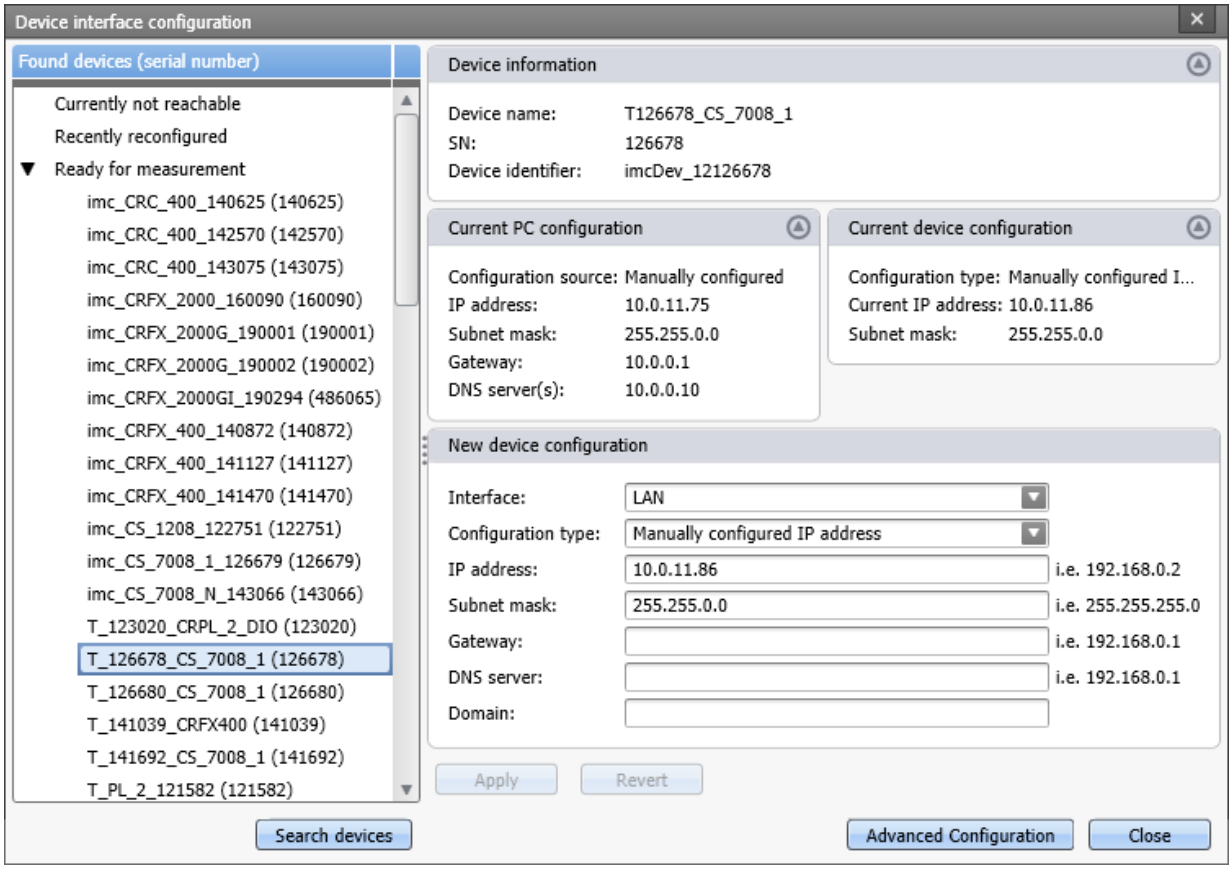
Start imc STUDIO. Open the dialog for configuring the device IP address by clicking on the button Device interfaces ( ).

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*".

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 is found in the group "Currently not reachable", you need to configure the LAN settings. If the device is found in the group "Ready for measurement", you can retain the current settings or view them.

Select the device.



Display of measurement devices found and of the IP-address

If the configuration type: *DHCP* is used, the IP-address is obtained automatically from the DHCP-server. If there is a direct connection between the device and the PC with a crossed cable, you should deactivate the DHCP. 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.).

Without DHCP, you must set the IP-address manually. Please note that the device's and PC's IP-address must fit together, in other words that in the network mask only the portion representing the device is different (see example).

Example for IP settings	PC	Device
IP-address	10. 0. 0. 34	10. 0. 0. 45
Network mask	255.255.255. 0	255.255.255. 0

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

Note: Connection via modem or PCMCIA-card

If the connection to the device is established via a modem or a PCMCIA-card, start the program imc DEVICES Interface Configuration by clicking on the button: Advanced Configuration.

An exact description is found in the chapter: *Setup - Connect the device* > *Special options for connecting to the device* 301.

Step 4: 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 96. Use the menu button Search for devices ( ).

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* 140).

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 plug-in Setup 87.

6.3 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.

6.3.1 Devices and TCP/IP

It is absolutely necessary to carry out the instructions below in the sequence given. But first read through the whole section in order to get a preview of the needed measures.

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 302.

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.

6.3.1.1 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! The chapter Devices and TCP/IP with PPP 321, contains hints on preparing your PC for running PPP-devices.

Firewall and Ports

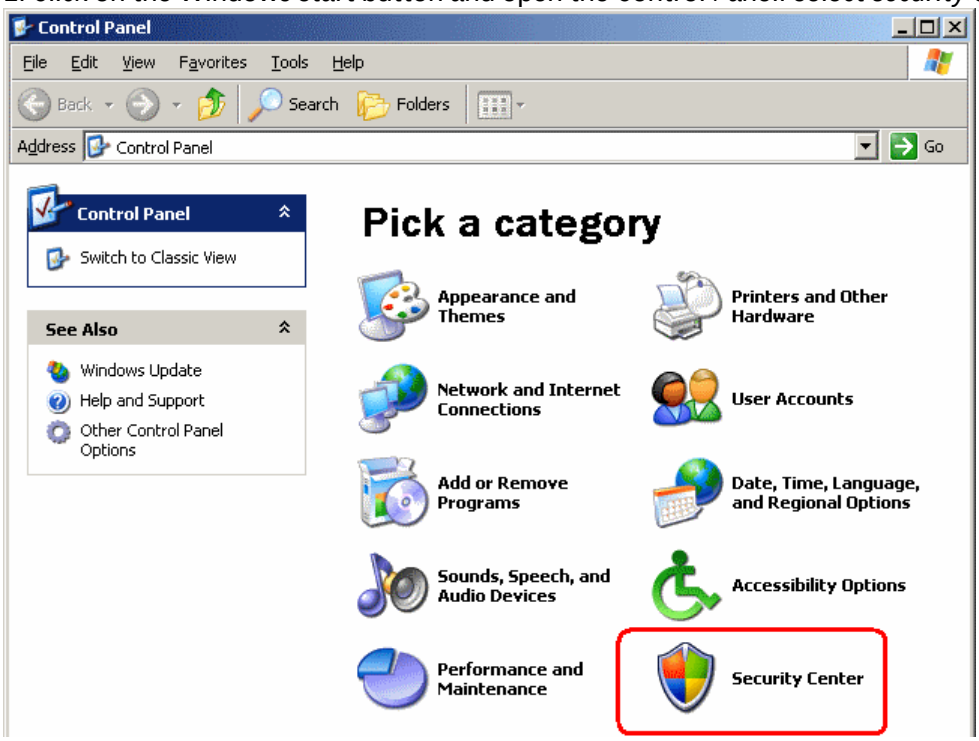
If a firewall is set up on your PC, you must ensure that the software has access to the ports used. The easiest way to do this is by enabling the program from within imc STUDIO.

The following ports are used:

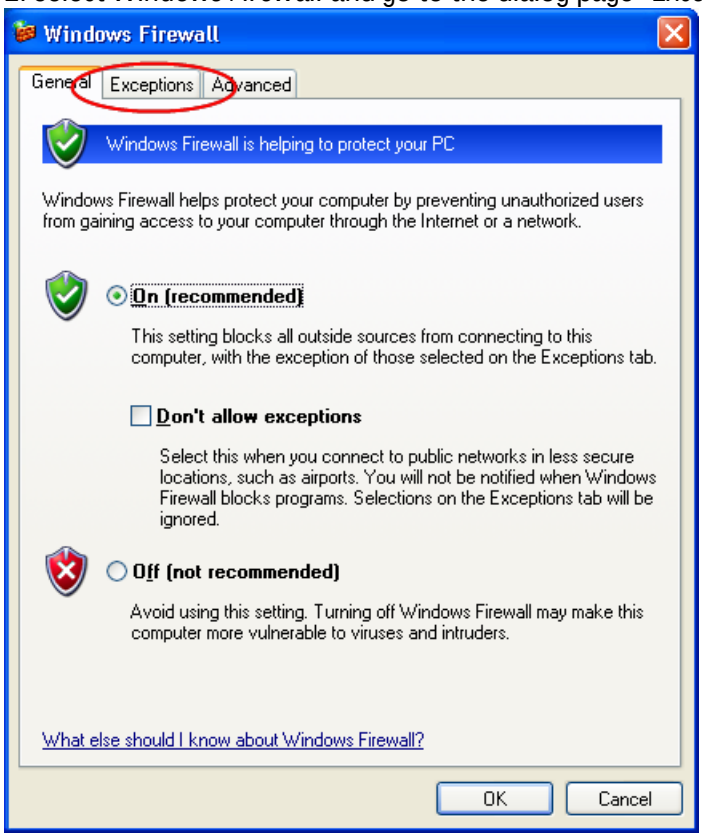
- UDP 1200
- TCP 1200 through 1202

Example for Windows XP:

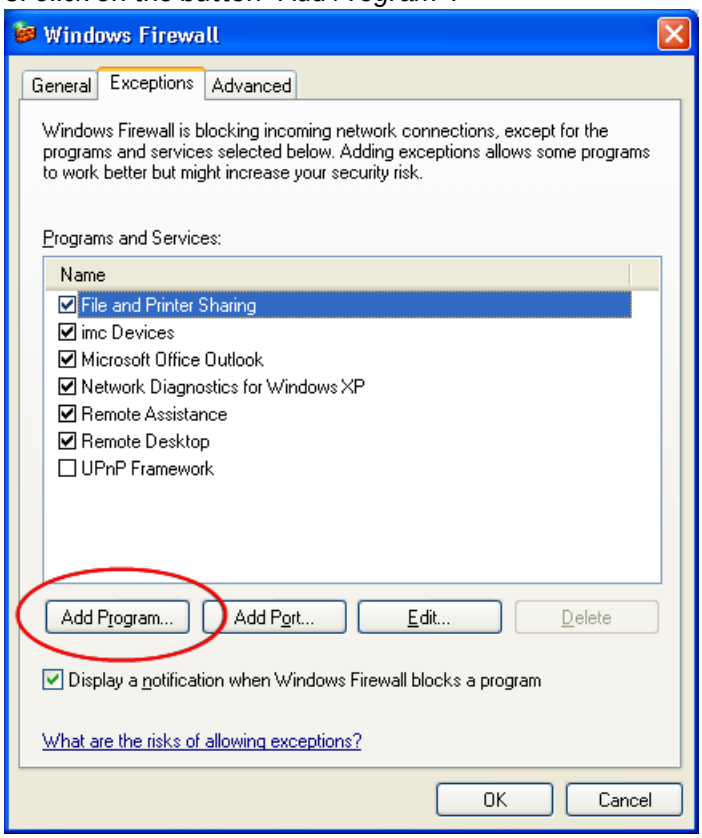
1. Click on the Windows Start button and open the Control Panel. Select *Security Center*.



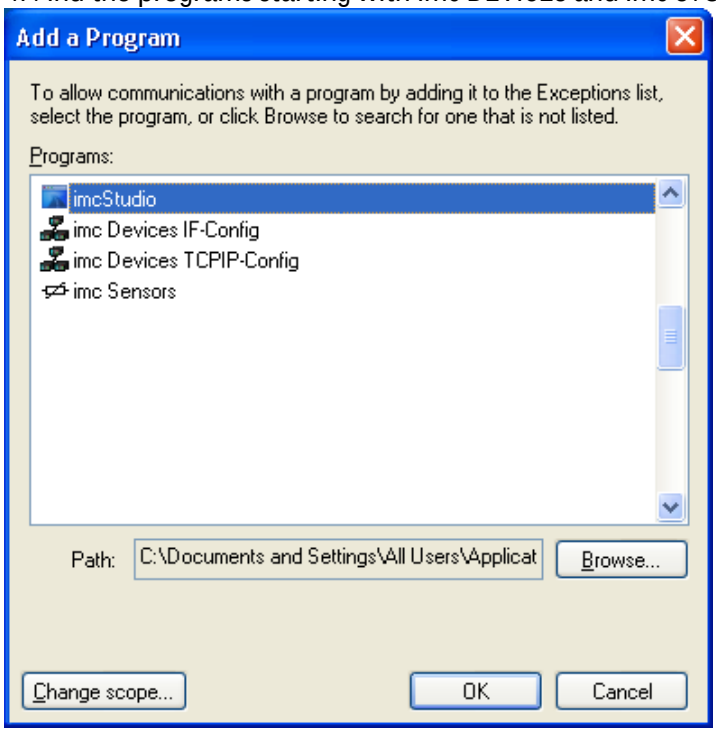
2. Select *Windows Firewall* and go to the dialog page "*Exceptions*".



3. Click on the button "*Add Program*".



4. Find the programs starting with imc DEVICES and imc STUDIO and click *OK*.



Close the dialog with "*OK*".

Note

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

6.3.1.2 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 (e.g. Windows XP: *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 306.

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":

  Windows 2000/XP/Vista/7 `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 DCHP-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!

6.3.1.2.1 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.

6.3.1.2.2 DHCP and APIPA

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

- *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 333 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).

6.3.2 Device connection via LAN

imc STUDIO comes with a variety of interface-configuration options:

Programs	Description	
Device Interfaces 307	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 310. 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 310.
Interface-Configuration 310	Interface-Configuration is for configuring imc - devices having a new motherboard (2 LEDs at the Ethernet interface). Modem (PPP)- or PCMCIA-settings can be made for new devices only with *Interface-Configuration*. Devices manufactured until July 2003 do not support *Interface-Configuration*.	
TCP/IP-Config	TCP/IP-Config is for configuring former devices, manufactured until July 2003. These devices are not supported by imc STUDIO. Instead, use imc DEVICES. The documentation for TCP/IP-Config is included in the documentation for imc DEVICES.	

6.3.2.1 LAN-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.

Preparation:

Connect the devices to the LAN (Connection via LAN 295) 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 (Connection via LAN 295). You should also check whether the Firewall and port 302 setting is preventing a network connection to the device.

Note: 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 306).

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

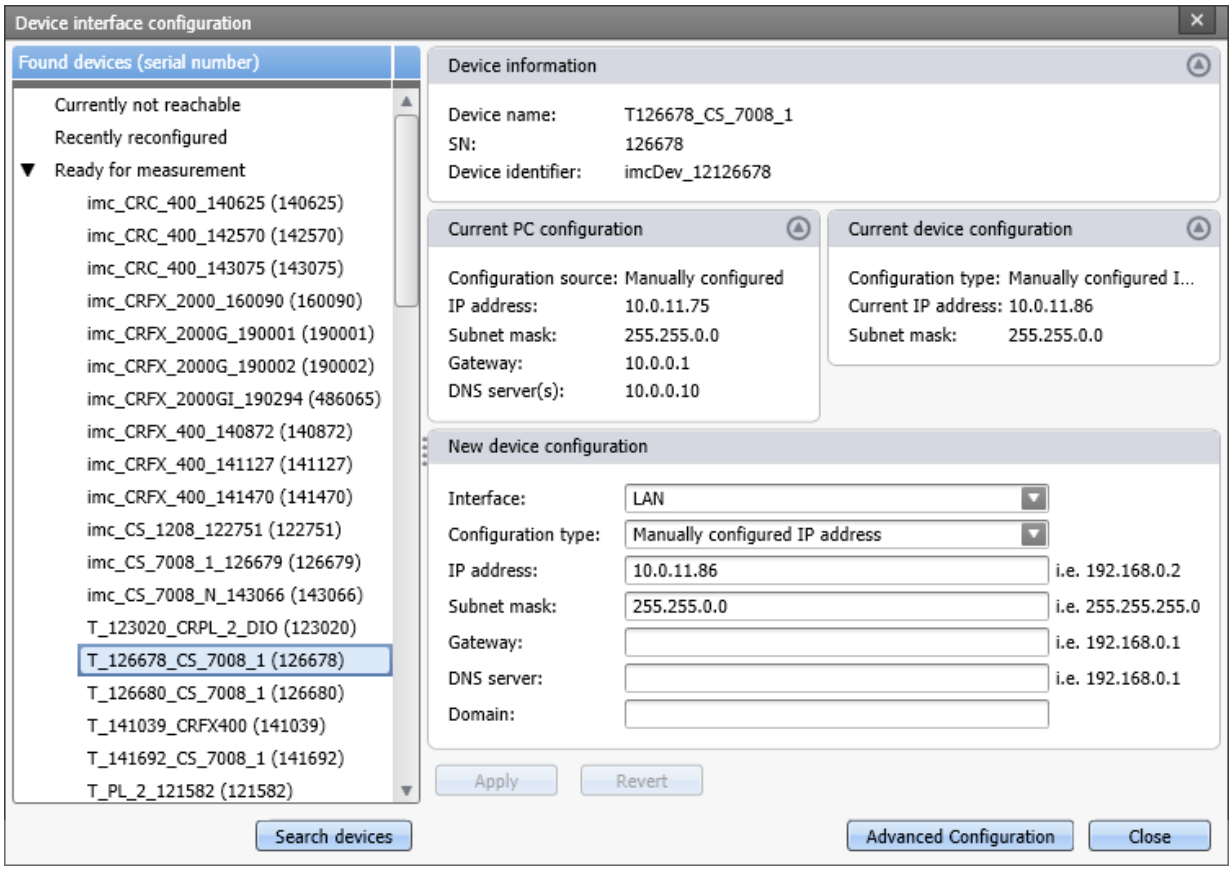
Start imc STUDIO. Open the dialog for configuring the device IP address by clicking on the button Device interfaces ( ).

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*".

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 is found in the group "Currently not reachable", you need to configure the LAN settings. If the device is found in the group "Ready for measurement", you can retain the current settings or view them.

Select the device.



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.
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.

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" opens. Follow the instructions for configuring the device interface.

LAN-Configuration with Interface-Configuration

Procedure

- Start the program imc DEVICES Interface-Configuration.
  The program can be started either via imc STUDIO or the Start menu.

Via imc STUDIO:

- Start imc STUDIO. Open the dialog for configuring the device IP-address by means of the button Device interfaces ( ) (Ribbon *Setup-Configuration* > *Device interfaces*).
- There, click on the button: "*Advanced Configuration*".

Via the Start menu:

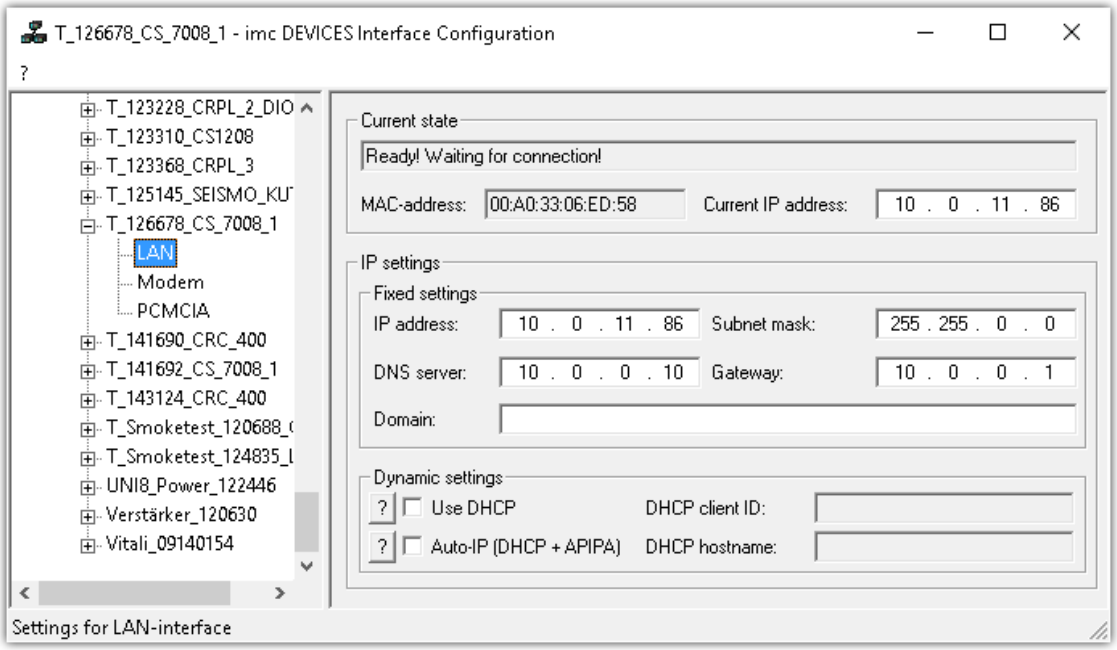
- Start the program: imc DEVICES Interface Configuration, from the program group: "imc" .

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.

- 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.

- Double-click on the device's name and then click on the entry *LAN*. You can now see the measurement device's current configuration, as shown below:



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.

TCP/IP configuration

- Click in the tree diagram on *LAN* in order to make the network settings.
- Enter the *IP settings* in the right-side region.

  Current state: Lists the device-adapter's current configuration.

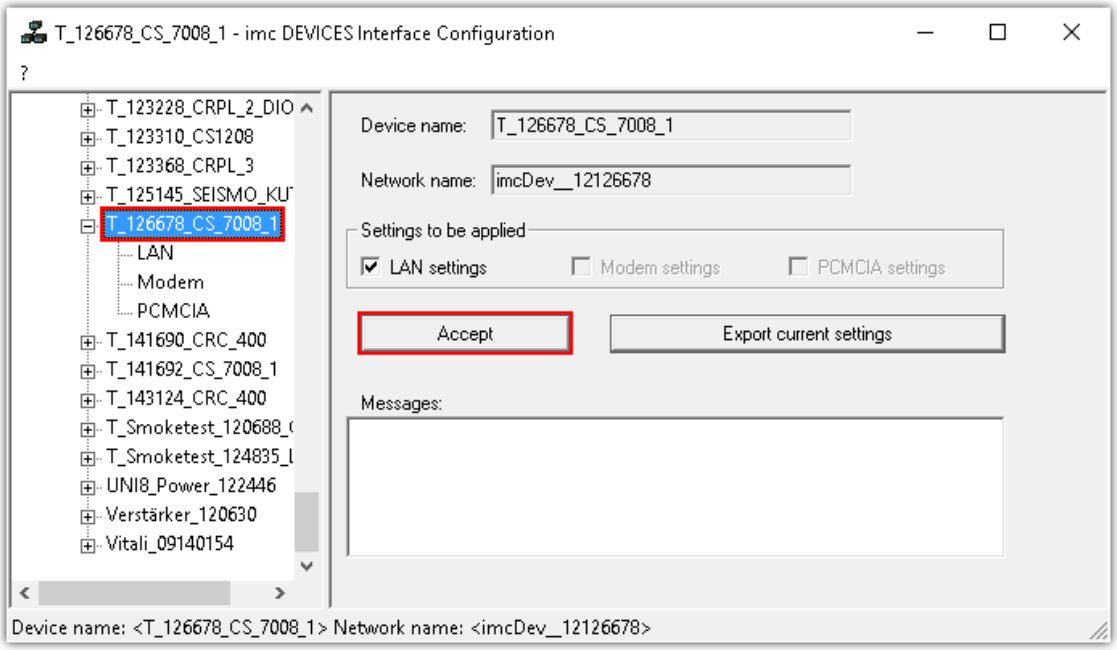
  IP settings: Here, you can modify the device interface configuration.

In this way, you can modify the settings for all measurement devices.

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.

6.3.2.2 TCP/IP, PPP via a router

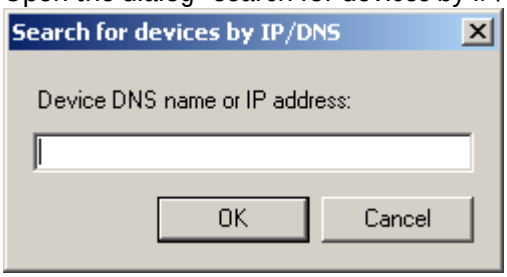
In a structured network (network with routers, Internet, ...), imc devices could not be integrated by means of a network search. With the knowledge of the IP-address or of the domain name (DNS name), it is now possible to integrate a device into the Device Table 149 and to establish a connection.

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".



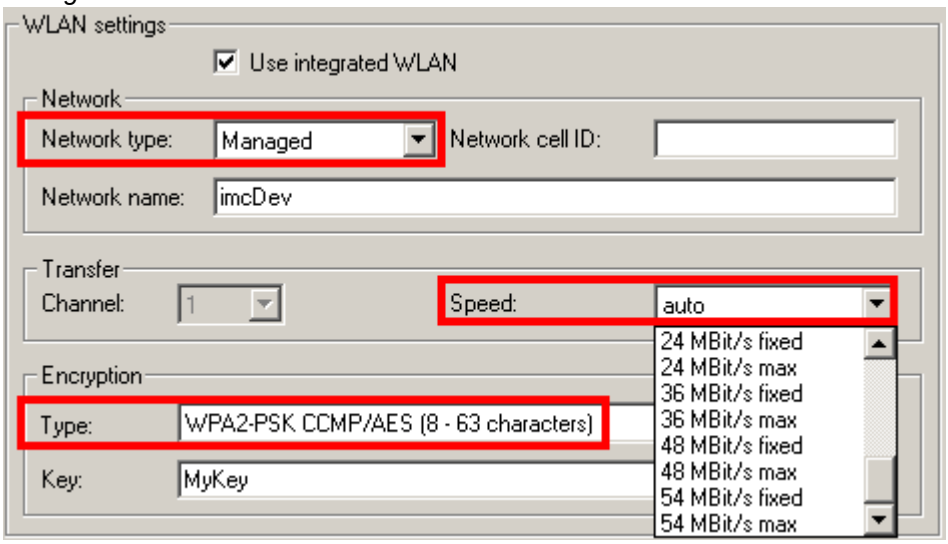
2. Enter the IP-address or the DNS name.
3. Clicking on the button *OK* initiates an attempt to establish a connection. If this succeeds, the device is integrated into the system.

Note

If you wish to define a device which is connected to the PC via the Dial-up network, you must first set up the Dial-up connection

6.3.3 Device connection via WLAN

This option is only for devices as of group 2 88 and higher and can only be set using *Interface-Configuration*.



Parameter	Description
Network type	Should be set to *Ad-Hoc* if no access point is used, otherwise to *Managed*.
Network cell ID	This is needed to select an access point (4 hexadecimal characters).
Network name	The name of the network, into which to integrate the device. This name must be the same for all devices and PCs which are to communicate together (max. 32 characters).
Channel	The available selections are channels 1-13 and channels > 36 for dual-band WLAN. 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. The exact number of characters set in the combobox must be entered. The setting 104 Bit is called 128 Bit on the PC by many drivers.

Note

- 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 4 and higher, up to 54 Mbit/s can be transmitted. The necessary prerequisite is that the system was commissioned since Summer, 2012, and that the version of the operating software used is imc Devices 2.7 R3 SP13 or respectively imc STUDIO 4.0.
  Attention: For these devices, the WLAN function can no longer be set correctly using older versions of the software.
- For devices as of group 4 and higher (SN19xxxx) 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 Zeichen)*". If you select another encryption, then the speed will be reduced to 54 Mbit/s conforming to the 802.11n Standard.

6.3.3.1 WLAN connection setup possibilities

PC via integrated WLAN - Device via integrated WLAN

Directe connection via integrated WLAN

A working WLAN connection can simplify your work significantly. But a bad WLAN connection can drastically impede your work and in many cases can lead to data overflow. Therefore, you should only use this technology if a stabile connection can be assured. Along with direct integration of WLAN with the devices, there are other options for achieving efficient data transfer with WLAN.

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.

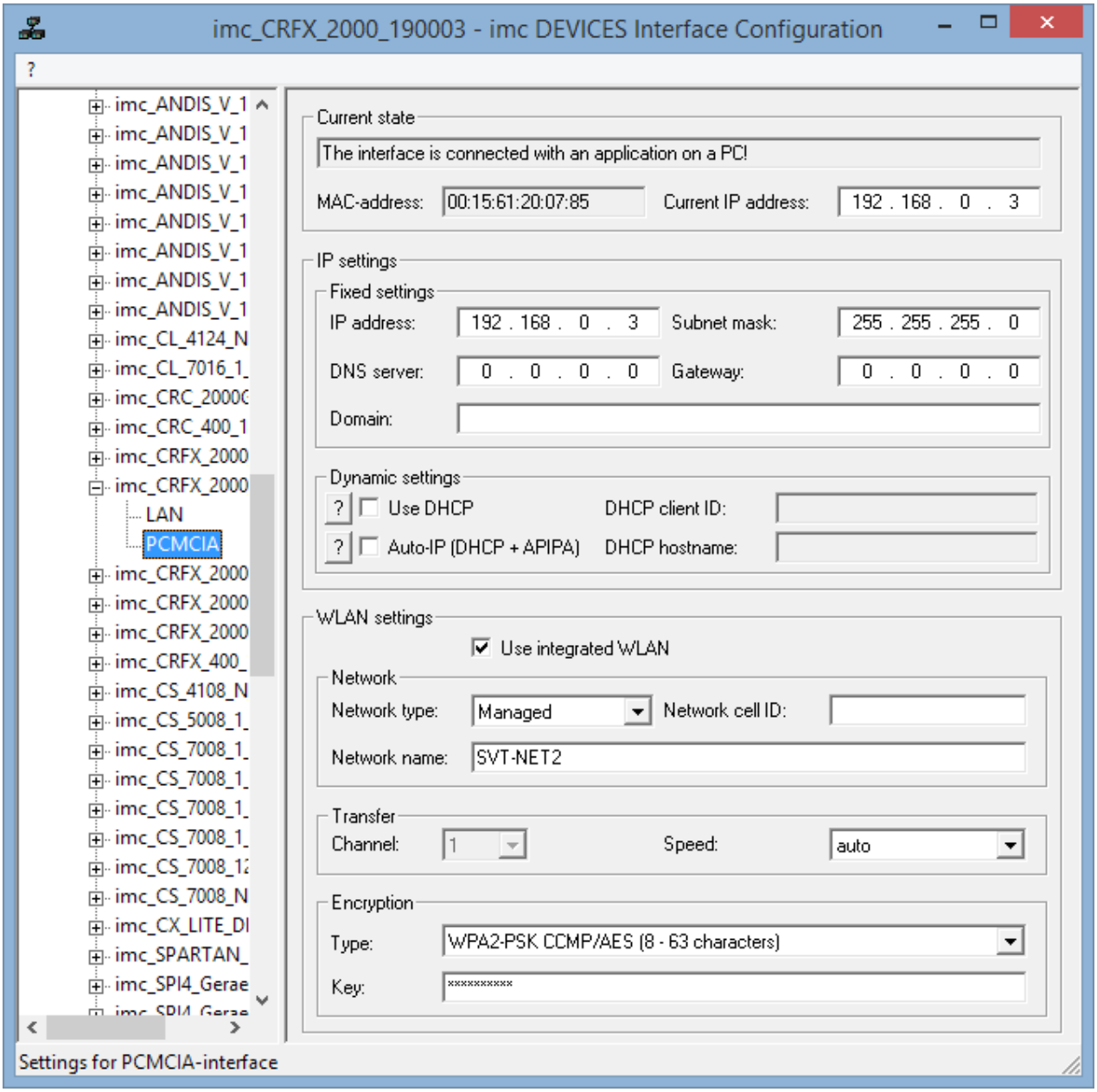
In this case, the interface from the PC and the measurement device is configured like a normal LAN connection 307. The cable is replaced by the wireless connection and administered by the Access Points. These must be configured in accordance with the instructions provided for them.

6.3.3.2 adhoc WLAN and Windows 8 / 8.1

Problem: Under Windows 8 /8.1, there is no longer any ad hoc connection.

Solution: An access point is set up using cmd, via which the device can be reliably connected.

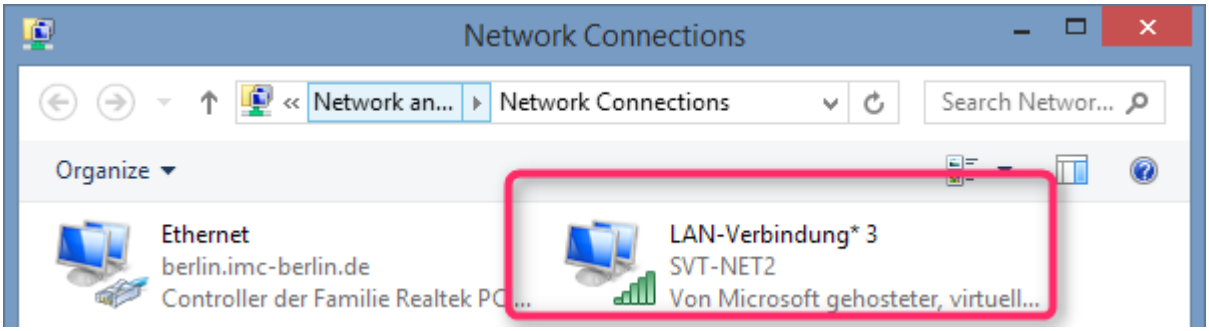
- Under PCMCIA in the Interface Configuration, make these WLAN settings:



- Call cmd with admin rights. Set the network and password, e.g.

```
netsh wlan set hostednetwork mode=allow ssid="SVT-NET2" key="0123456789"
netsh wlan start hostednetwork
```

Now an additional connection is available.



Now it should be possible to access the device. In the IP-settings, set an appropriate address for the device, here for example 192.168.0.111.

The device can now be found in a device search and connected in the normal way.

Note

After restarting the computer, the connection is no longer available. The two lines can also be written in a batch file. These command lines must then be run with admin rights. The connection settings are then restored. See also the description below of the procedure as a Task.

Creating the batch file and running it via the Task Scheduler

Creating a batch file

Create a text file with these two lines

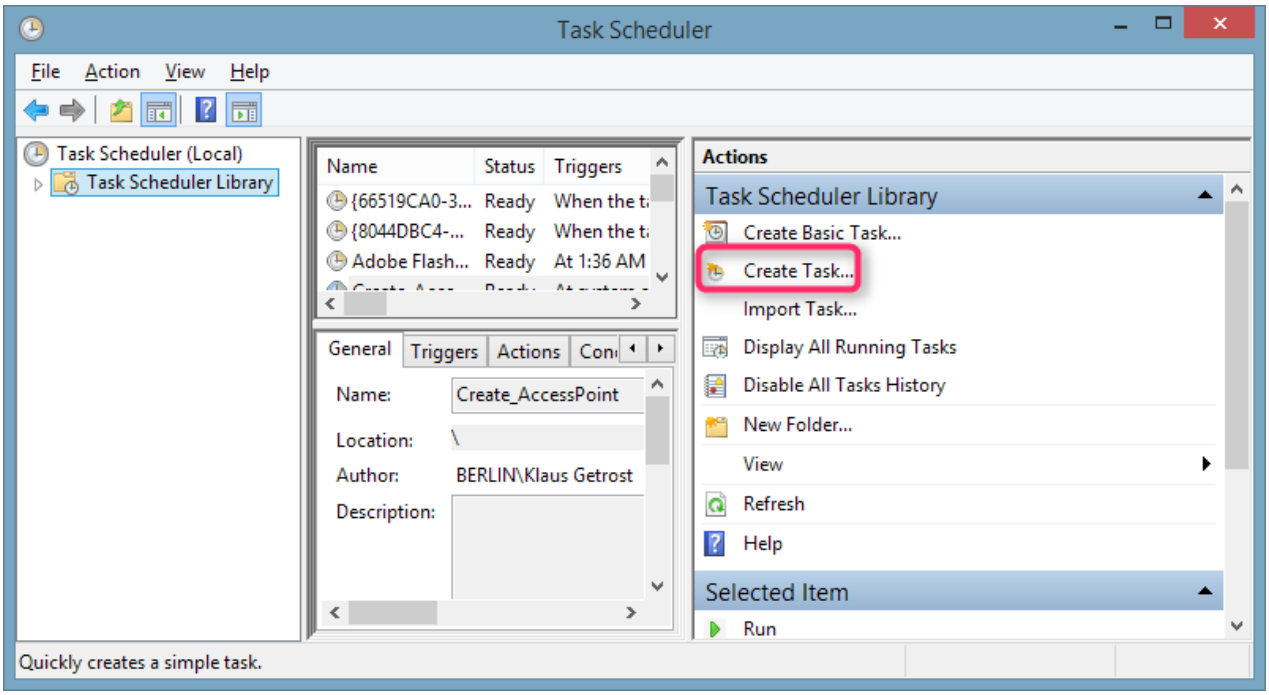
netsh wlan set hostednetwork mode=allow ssid="AP_name " key="Password"
netsh wlan start hostednetwork

e.g.
netsh wlan set hostednetwork mode=allow ssid="SVT-NET2" key="0123456789"
netsh wlan start hostednetwork

Name this file e.g. "*MyAP*" and give it the extension "*.bat*"

Calling the file via the Task Scheduler

Start the Task Scheduler:
WINDOWS-key + R and *then Taskschd.msc*

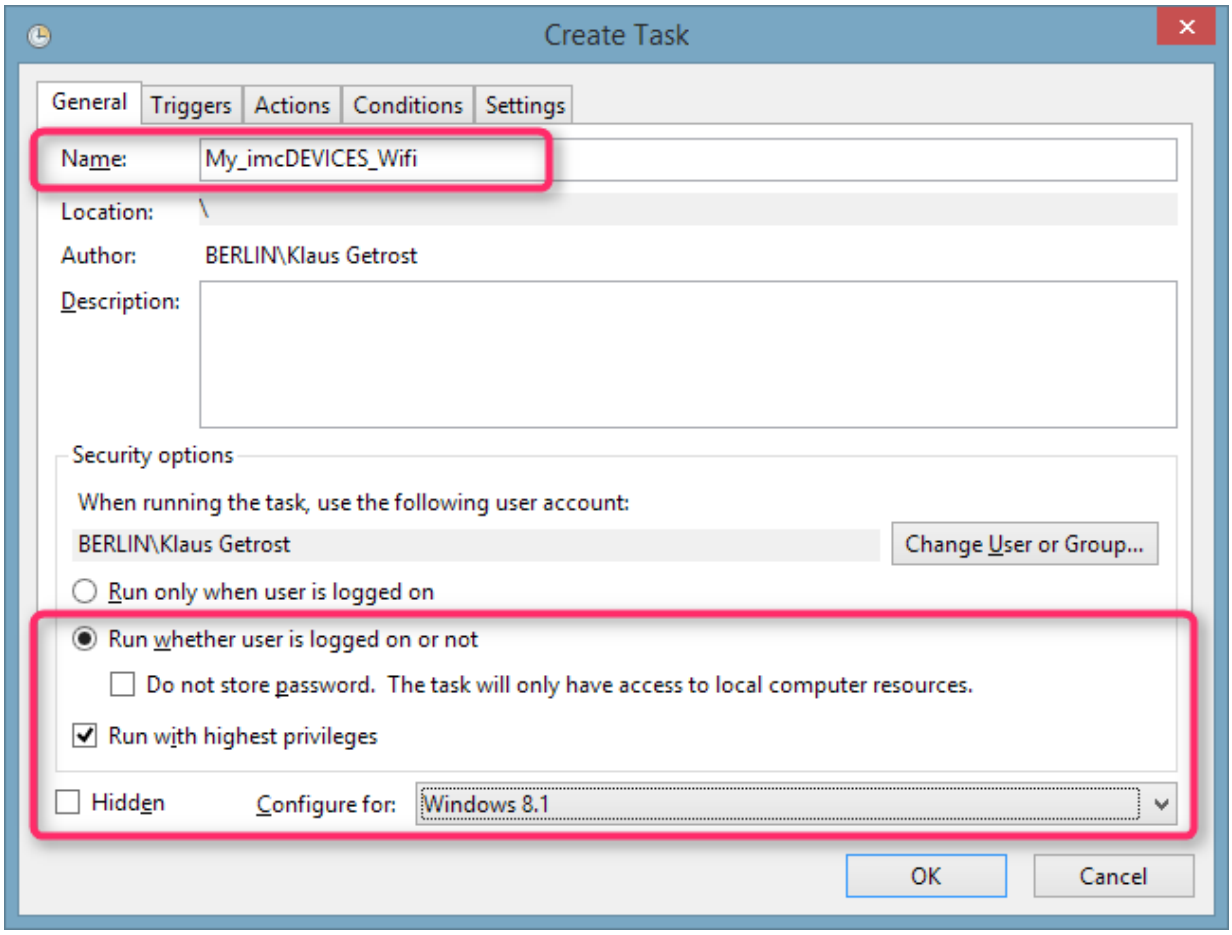


Use Create Task...

General

Name e.g. "*My_imcDEVICES_Wifi*"

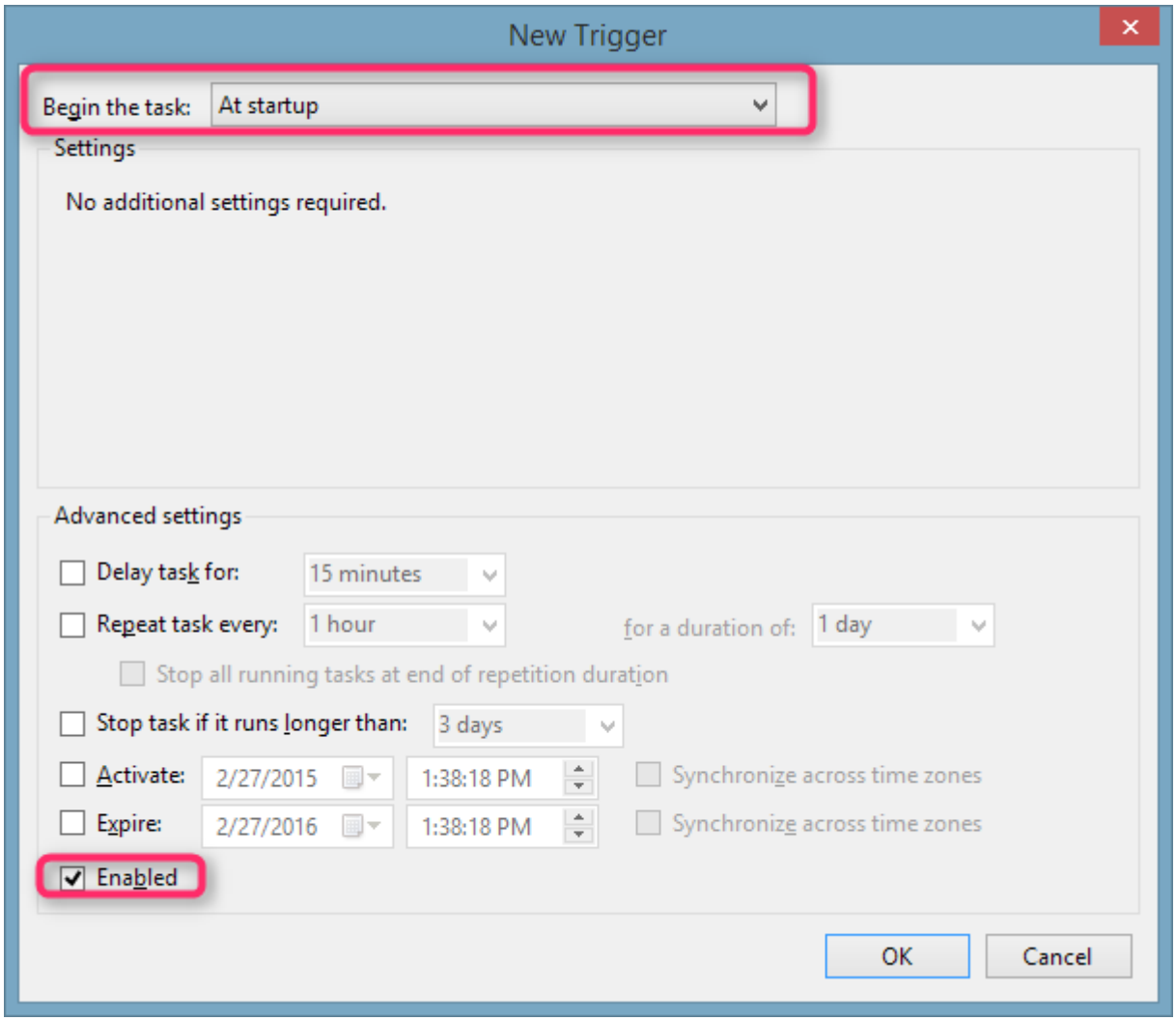
Activate *"Run whether user is logged on or not"* and *"Run with highest privileges"*. *"Configure for Windows 8.1"*



Triggers

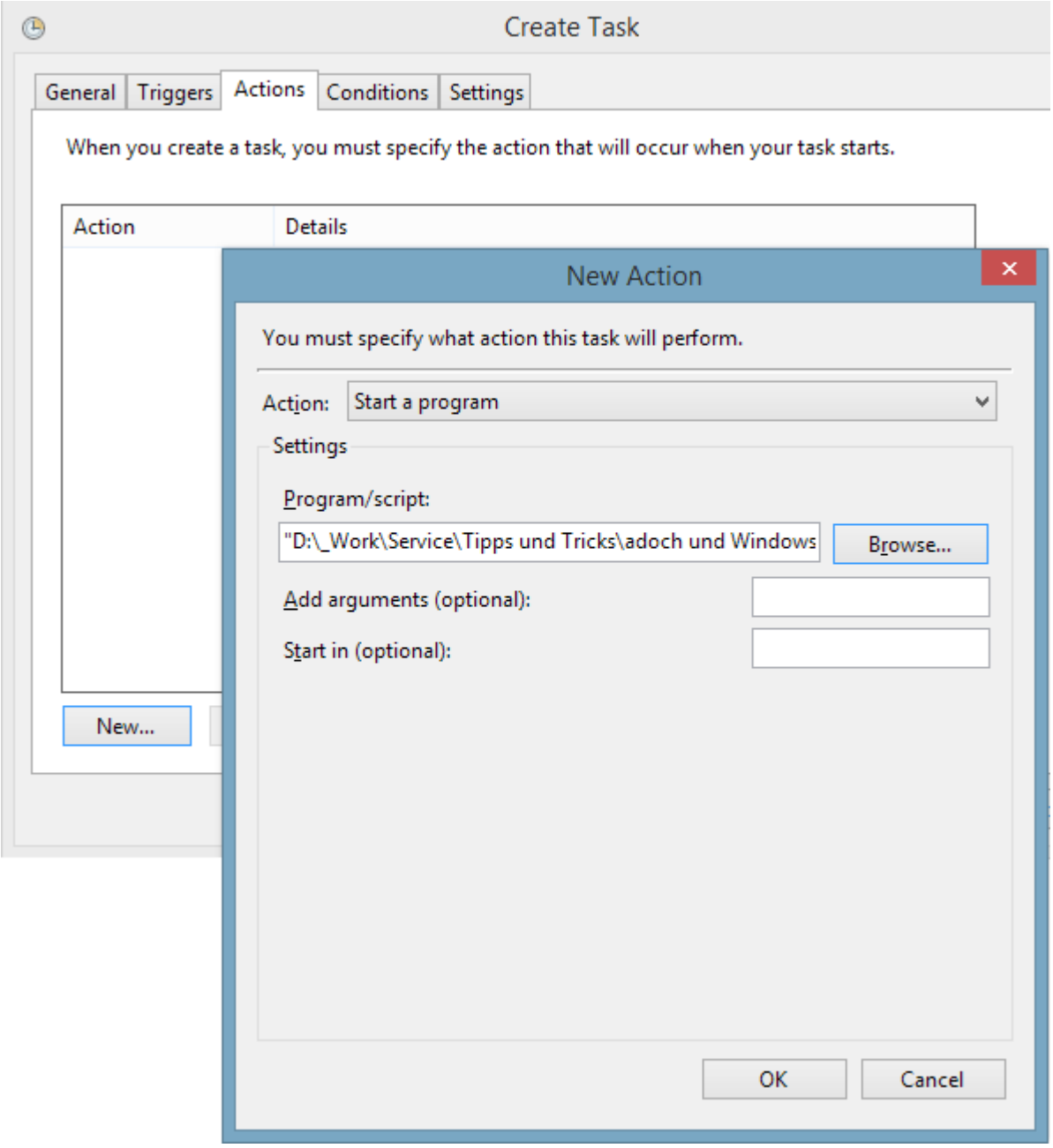
New...

Begin the task: At startup" and *"Enabled*



Actions

New... , "*Action: Start a program"* and add the batch file



Conditions and *Settings*

All settings deactivated.

6.3.4 Device connection via Modem (TCP/IP with PPP)

What is a PPP connection?

The abbreviation PPP means Point–to–Point Protocol. As the name indicates, this protocol concerns single point to single point connection (i.e., one client and one server) as opposed to Ethernet, where any number of subscribers are connected together via the Ethernet-bus. The client is always the one to set up the connection and to register with the server. The server checks the client's authorization and transfers to the client all information necessary for participating in network communication. After this, both parties configure their network interface; a connection then exists between the two. The PPP-protocol manages this preliminary procedure; it does not handle the actual communication! Instead, the PPP-protocol can be viewed as building a sort of "tunnel" between the parties and packaging the actual network communication protocol, for instance TCP/IP.

In a TCP/IP-based network, the individual subscribers are distinguished by their personal IP address. Strictly speaking, the address is not assigned to the PC or device, but to the network interface. After all, if a PC has two Ethernet connections, it can have two IP-addresses.

IP addresses, then, can be assigned not only to Ethernet interfaces but to any network interface which is able to transfer data according to the IP protocol. In this sense, even the WINDOWS Dial-Up Network (with the PPP-option activated) represents an IP interface.

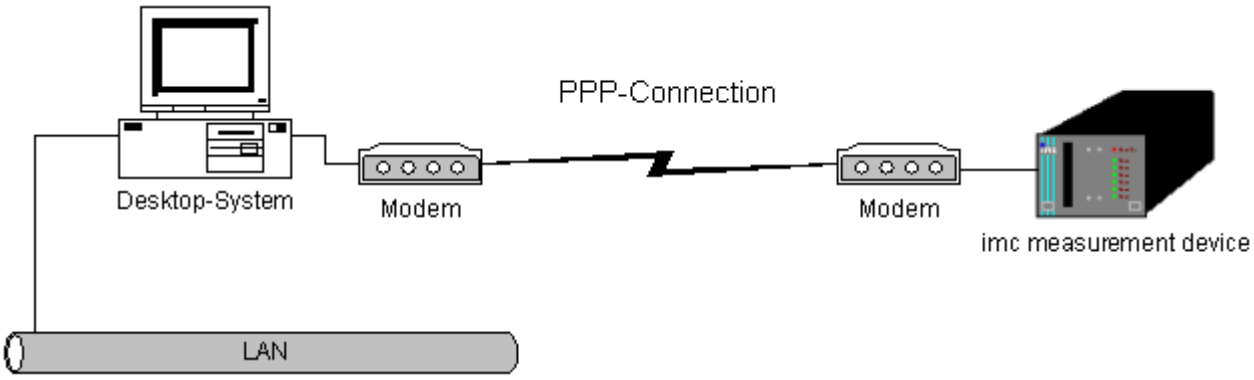
Physically, such point-to-point connections are serial connections. In the simplest case, the two PPP-subscribers are connected via a zero modem cable, but in most cases it is via modems.

Note

In summary, PPP in conjunction with TCP/IP serves to connect two PC/devices or networks via a serial modem line. The PPP-connection remains transparent for the network subscribers. The PPP connection is a property of the *Dial-up Network*.

6.3.4.1 Connection via PPP

For connection via PPP, the device needs a serial interface in addition to the Ethernet connection. This serial interface can be connected to a modem. The measurement device is in this case the PPP server, the control-PC the PPP client. That means that the PC calls the device to establish a connection. TCP/IP is used as the network data transfer protocol.



6.3.4.2 Configuring devices as a PPP-Server

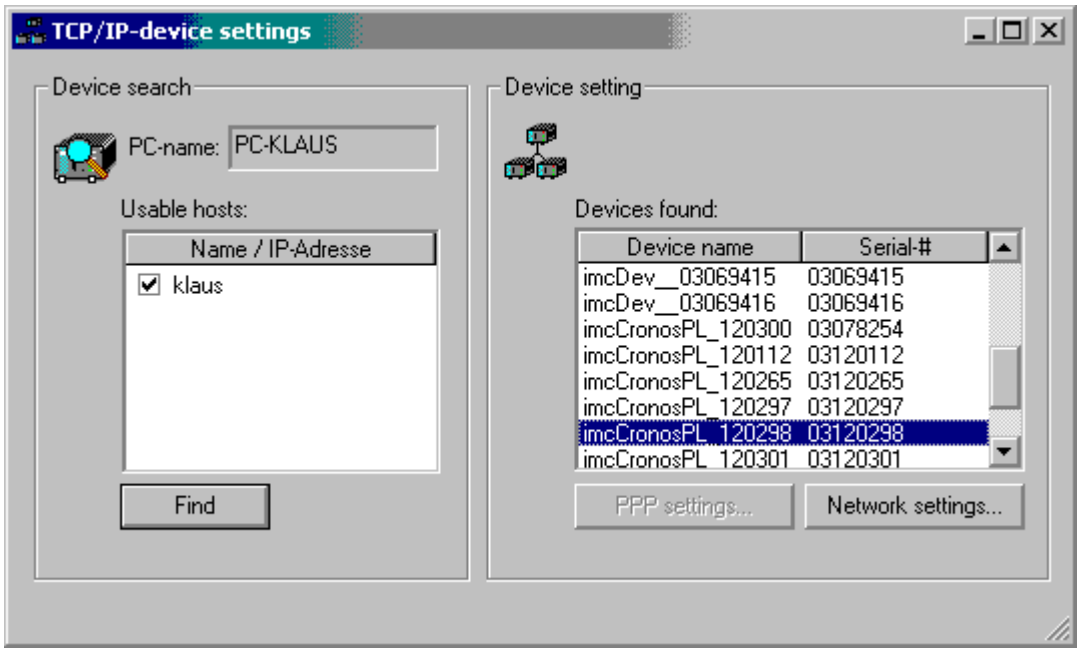
The measurement devices can be configured as a PPP-server.

- For configuring devices of group 1 88, the auxiliary program *TCP/IP Config* is provided.
- For configuring devices as of group 2 88 and higher (manufactured since August 2003), the auxiliary program *Interface-Configuration* is provided.

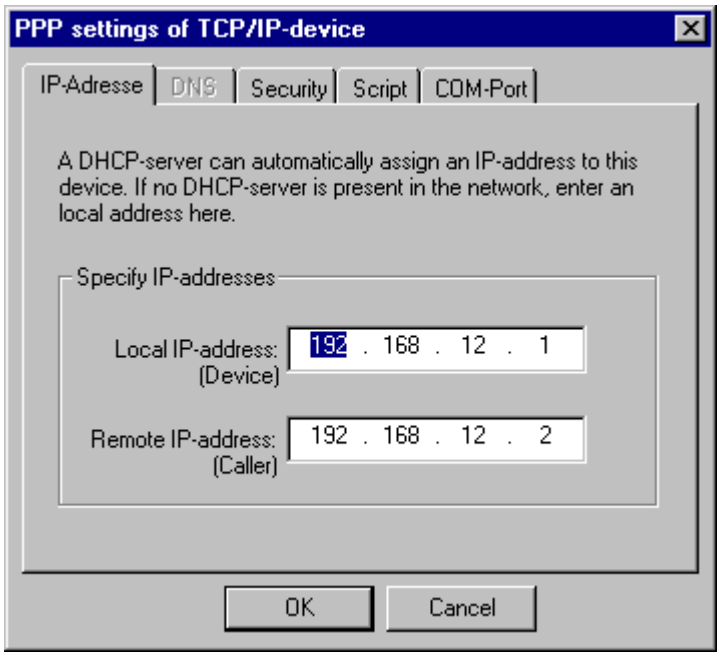
Configuring devices with TCP/IP-Config

For the PPP configuration, the device must be connected with the PC via Ethernet. Start the program TCPIP-Config. Perform a network search.

On the right side of the dialog, all devices located are listed. Once the device having the PPP option is selected, the softkey *PPP-settings* is accessible. The dialog thus called presents all PPP settings which are required for running the imc unit as a PPP server.



To set the PPP-parameters, click on the button *PPP-settings* in the main dialog.



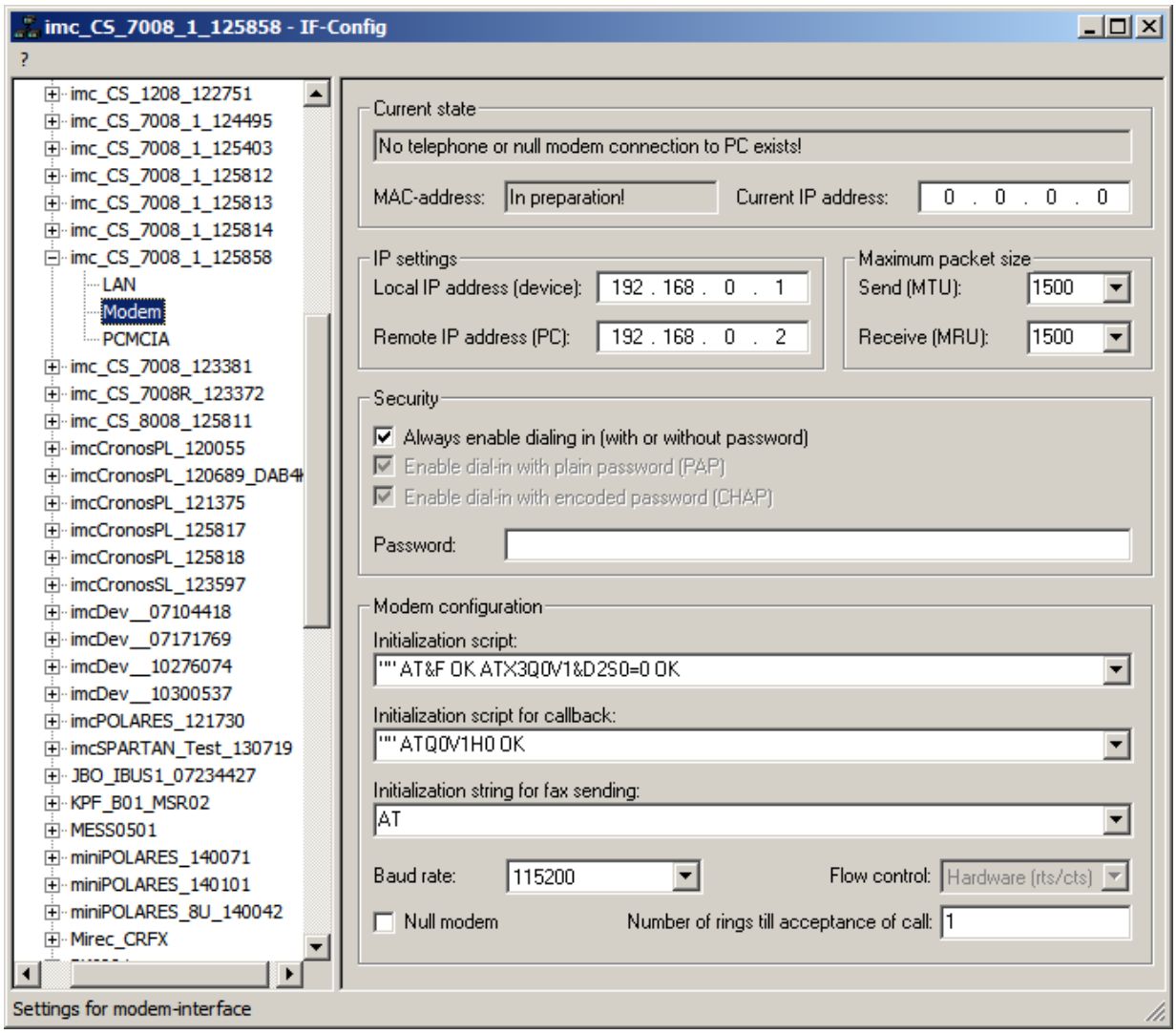
For more related info, read the chapter *Modem settings for TCP/IP devices* 325!

When you click the button *OK*, the PPP-settings are written to the device. The settings are only adopted by the device once you comply with a prompt to confirm them. The process lasts a few seconds. Once the message *Now applying network settings* appearing next to the button *Properties* fades away, configuration of your device has been completed.

Configuring devices with Interface-Configuration

For the PPP configuration, the device must be connected with the PC via Ethernet. Start the program *Interface-Configuration*. In the tree diagram, select your device on the left side. Expand the entry for your device and click on the sub-entry *Modem* there.

All of the PPP-settings necessary for operating the device as a PPP-server can be made on the control panel which covers the right side of the screen. For more related info, refer to the section *Modem settings for TCP/IP devices* 325 of this chapter!



To write the settings to the imc measurement device, simply click on the device's entry on the left side and on the right side on the button *Accept*. First check whether the settings appearing in the box above the button are to be applied, and activate or deactivate them according to your wishes.

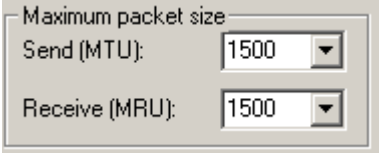
Once the configuration has been written to the unit, the device performs a restart and makes the new settings operative. This process lasts about 1-2min. When the message *Settings have been applied* appears in the *Messages* box the configuration of the device is complete, and you can continue with the next device.

6.3.4.3 Modem settings for TCP/IP devices (PPP)

IP address: Since Windows isn't able to assign an IP-address to the device, an IP-address must be entered in the box *Local IP-address (device).* The second box*, Remote IP-address (PC)* can be left on 0.0.0.0. In this case, an IP-address must be assigned to the caller (PC) by means of the Dial-Up network settings. This IP-address must be located in the same subnet as the address assigned to the device. See also the notes at Computers with multiple TCP/IP connections 306.

DNS Server: Not needed if there is a direct connection between the PC and unit.

6.3.4.3.1 Configuration with Interface-Configuration



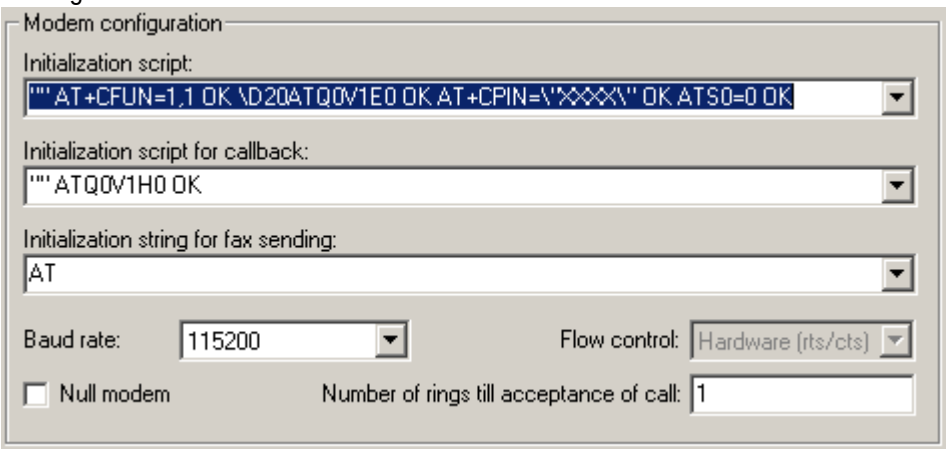
Maximum package size: This option enables the maximum package size to be reduced, in order to improve the data throughput in the face of bad conditions. The default value of *1500* should only be reduced in the case of very bad telephone connections with many CRC errors. CRC errors can be indicated either in the status of the dialup connection or in the system monitor, depending on the Windows version.

In the box ***Security***, it's possible to set a password for setting up the modem connection. By clicking on a checkbox, the corresponding option is activated, which determines in what way the password is to be received. The system later inquires the password before a modem connection can be set up.

- *Always enable dialing in* means that the password protection is deactivated! Anyone who knows the telephone number can dial in, whether they can enter the password or not.
- *Plain password* means that PAP is accepted. In this protocol, the password is transferred in plain text. Therefore, there is a possibility that it could be intercepted. Using such an intercepted password, an intruder could dial into the device.
- *Encoded password* means that CHAP is accepted. In this protocol, no information is transferred from which the password could be derived. Even if an intruder is able to discover an authorized user's response to a password prompt, it's still not possible to abuse the password. Each password prompt demands a totally different response which only can be derived if the original password is known.
- It is possible to enable both PAP and CHAP.
- The password may have up to 32 characters.

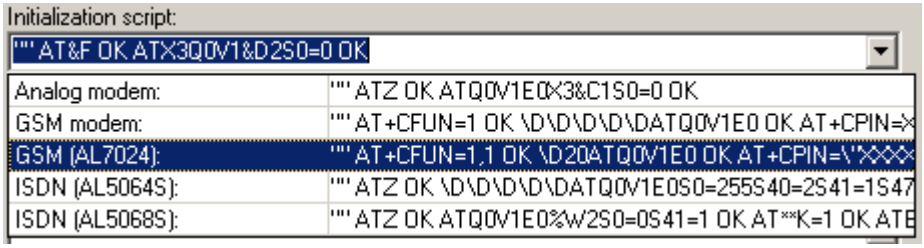
Modem configuration: Here, the *Baud rate* of the PPP interface as well as the *Scripts* are set, which govern the connection setup or disconnection process.

The PPP interface is set up via a serial connection, whose maximum possible Baud rate is 115200 Bd. The flow control of the serial interface is achieved exclusively by means of the hardware protocol. These settings must be the same for both subscribers.

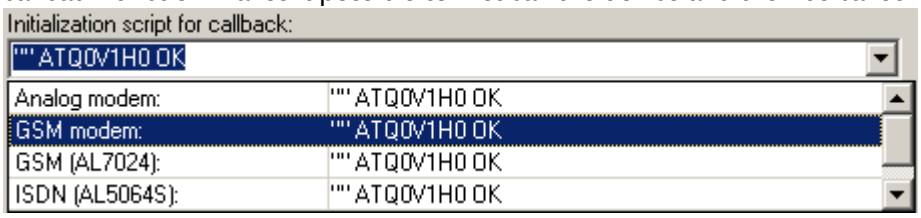


The two initialization scripts consist of a sequence of "Command-response pairs", in which, however, the first step is to await the modem's response. Since a command would normally need to be sent as the first step, the empty quotation marks indicate that there is no command to expect. Then the script proceeds with the next pair; beginning with a command such as "ATZ". And then comes the matching response, such as "OK", etc.

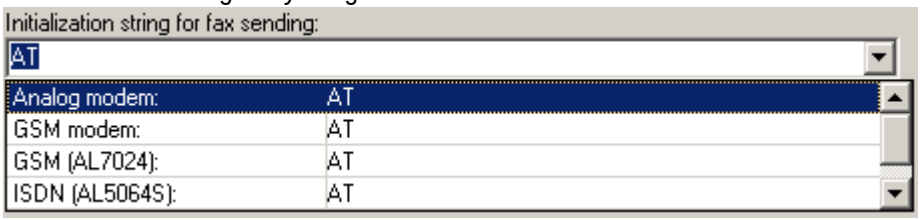
- The *initialization script* can have at most 128 characters. Its purpose is to carry out the basic initialization of the modem. If the modem requires special initialization, the appropriate commands should be entered at this point. The combobox already contains examples of the most common ones, which should be enough for most normal cases. For a GSM-modem, only a PIN must be entered.



- The *Initialization script for callback* should not be changed. Up to 64 characters can be entered. The callback function makes it possible to first call the device and then be called back by it.



- The *Initialization string for fax sensing* is a single AT-string of up to 64 characters in length. Therefore this is not a script like the one in other controls! Here you can enter special AT-commands which are needed to set the modem into fax-mode. Most of the time, there should be no reason to change anything here!



- *Number of rings till acceptance of call:* If set to a number, so a user can get the phone, the line can be used for the device and for talking.

If you use a zero modem connection, it is enough to set the check mark for the *Zero modem* control. Otherwise, you select the modem type. To do this, right-click on the little arrow at the end of the line of text, in order to expand the selection (with GSM, replace XXXX with the pin).

COM-Port*:* The PPP-interface is realized via a serial connection. For this purpose, the last dialog page, *COM-Port*, is provided:

The settings to be made are *Baud rate* and *Protocol*. The maximum Baud rate which can be set is 115200 Bd. However, for devices manufactured prior to August, 2003 (no LEDs at the Ethernet interface) it's 57600 Bd. The serial interface then functions by means of a hardware- or software protocol; devices dating from August, 2003 only by means of hardware protocol. These settings must be the same for the modem and the device.

The default values are:

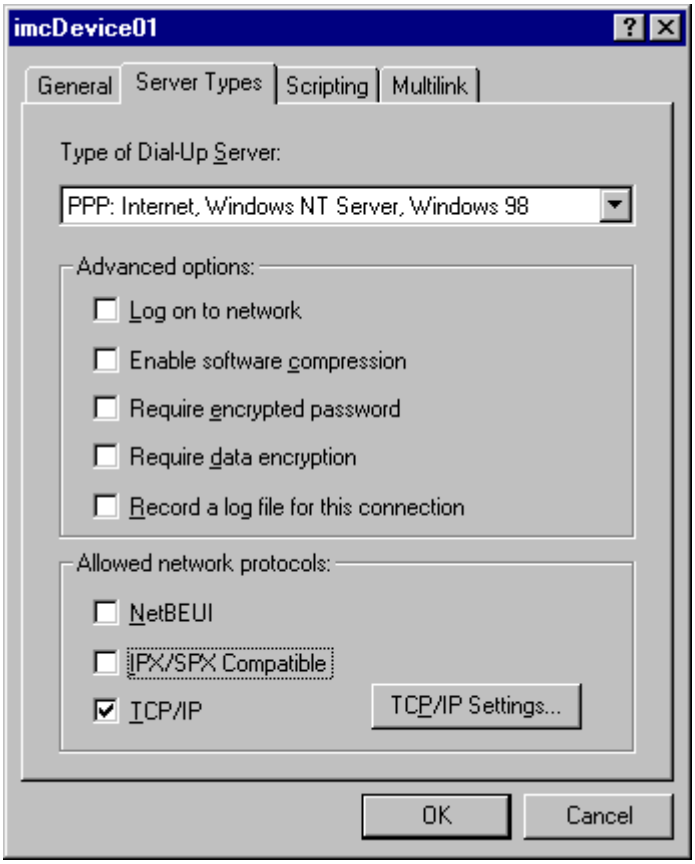
Baud rate	*max 115200 Bd./57600 Bd*
Protocol:	*Hardware*

To use a hardware flow control, a cable is needed which is connected to these lines. This isn't always the case with zero modem cables.

6.3.4.4 Configuring a PC as a PPP-client

In order to use the Windows PC as a PPP-client, the following conditions must be met:

- A modem is connected to the PC,
- It is installed under Windows,
- The Dial-Up network under Windows was installed.

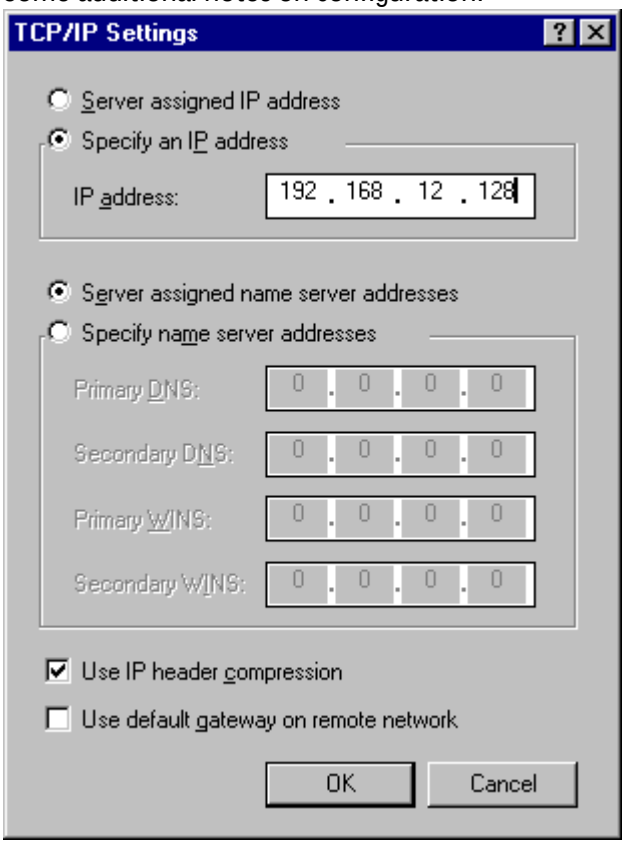


In order to be able to set up a direct connection (without a modem) between the device and the PC by way of a null modem cable, this "null modem" must also be installed. The installation to select is called, for example in WindowsXP, "Communication cable between two computers". Not all Windows operating systems are designed for such an installation as a default. With a router running under a different operating system, such as Linux, it's possible to set up a connection via a null-modem cable.

Refer to the modem manufacturer's documentation and that of Microsoft Windows for information about installing the modem and the Dial-up network. The following will restrict itself to a few explanations and points on the general procedure which is recommended.

Create a new connection. Enter the telephone number of the device's modem. Then the Dial-up Network's properties can be edited. The following illustrations show the recommended default configuration, which should always work. Only the IP-address must be entered according to the user's own needs. The dialogs shown come from Windows98. In other versions of Windows, the dialogs look different, but are used in an analogous way.

Some additional notes on configuration:



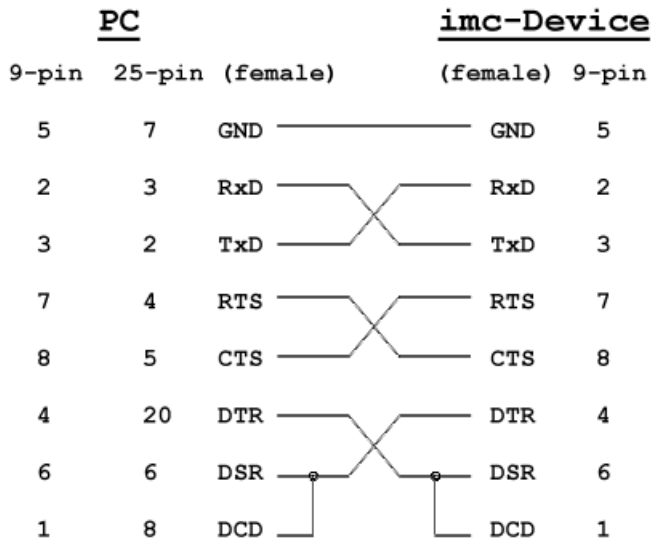
- Protocol: Select only TCP/IP! Deselect all other protocols.
- IP-address: The instructions above configure the device as a PPP server in such a way that the PPP interface on the PC has a fixed address. For this reason, an unambiguous address which identifies the Dial-up Network as a TCP/IP network interface on the local PC must be specified. If the computer has an Ethernet adapter, this IP-address may not belong to the same Subnet.
- Name-Server: Not needed if there is a direct modem connection between the PC and the unit.
- Standard-Gateway: Must be deactivated, since the device is neither an Internetprovider nor router. Otherwise there is a danger that the network will not work on the PC involved!
- Compression
- LCP Expansion: Not supported by the measurement devices.

6.3.4.5 Making a null modem cable

If you wish to operate your device via a null modem connection, you need the right kind of cable. Below, the required pin configuration is described. There is a distinction between cases where the unit's counterpart supports Hardware-Handshaking (RTS, CTS) and where it doesn't.

Null modem with Hardware-Handshaking

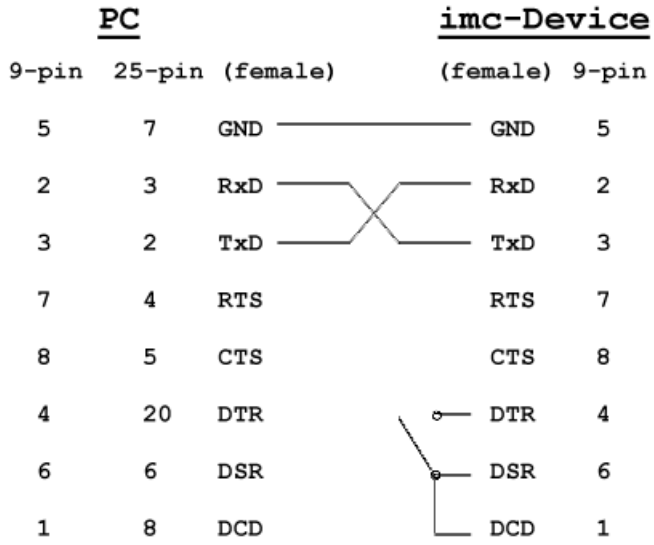
For a null modem cable with Hardware-Handshaking, your cable must be configured like this:



The PC's control line *Data Terminal Ready* (DTR) instructs the device to wait for a PPP-connection. If this line isn't controlled properly by your PC, you can use the unit's DTR-signal as a substitute.

Null modem without Hardware-Handshaking

For a null modem connection without hardware handshaking, a three-wire connection is adequate. In order for the device to be able to enter a PPP-connection, the signals Data Set Ready (DSR) and Data Carrier Detect (DCD) must be connected at the device with the line Data Terminal Ready (DTR). This can happen either right inside the terminal plug or via a switch.



Note for devices of group 1

If you permanently connect the lines, the imc unit always expects a PPP-connection. The Ethernet interface is then deactivated! You can only access the device via Ethernet if you disconnect the connection between the lines DSR, DCD and DTR.

6.3.4.6 Establishing/Disconnecting a PPP Connection

Establishing a PPP connection

As already explained in the introductory passages, a PPP connection actually remains transparent for its subscribers. That means that the imc STUDIO operating software makes no reference to the PPP connection; as far as it is concerned, the connection is of the usual TCP/IP type. In consequence, the PPP connection must first be set up by Windows before the imc STUDIO operating software is started.

To establish the PPP connection, select the icon with the link to the device in the Dial-Up Network which was set up according to the instructions in the last section 325.

After the connection has been successfully established, a TCP/IP connection is ready for the imc STUDIO operating software. Now the operating software can be started, and a network search can be carried out in the manner usual under TCP/IP. After the network search the unit should appear in the list of hardware and can be selected.

Note

In order to use your device via the PPP connection, first set up the connection via the Dial-Up Network, then start the software.

Disconnecting a PPP-connection

First disconnect from the device on the software level (Ribbon: *Home* > *Disconnect*). Only after this is done, close the modem connection by selecting *Disconnect* in the Dial-Up network. The device then returns to a standby state.

6.3.4.7 Commissioning a radio modem with TCP/IP (PPP)

Devices as of group 2 and higher are described in device overview 88.

Falcom A1, A2D-1, A2D-2

Prerequisites:

1. Modem set to 9600Baud (factory setting).
2. HyperTerminal connection settings: 9600, 8bits, no parity, 1 stop bit, no protocol.
3. HyperTerminal connection settings: 115200, 8bits, no parity, 1 stop bit, no protocol.

Preparing the modem:

1. Connect modem to PC via serial cable.
2. Start HyperTerminal connection at 9600Baud.
3. Input: "at" then ENTER. Modem must respond with "OK".
4. Change transfer speed to 115200Baud by entering "at+ipr=115200", then ENTER. Modem must respond with "OK".
5. Start HyperTerminal connection at 115200Baud.
6. Input: "at&f", then ENTER, "at&s0", ENTER, "at+ipr=115200", ENTER, "at&w", ENTER. Modem must respond each time with "OK".

Preparing the device:

The program "Interface-Configuration" is used to make the following changes in the device:

1. The IP-address is set to match the PC's Dial-up entry.
2. For the initialization script, the combobox entry for the GSM-modem is selected.
3. In the command "AT+CPIN=XXXX", XXXX is replaced by the SIM-card's PIN. If the SIM-card doesn't require a PIN, "AT+CPIN=XXXX O AT OK" is deleted.
4. The Baud rate is set to 115200.

WaveCom Fastrack

Prerequisites:

1. Modem set to 115200Baud (factory setting).
2. HyperTerminal connection settings: 115200, 8bits, no parity, 1 stop bit, no protocol.

Preparing the modem:

1. Connect modem to PC via serial cable.
2. Start HyperTerminal connection at 115200Baud.
3. Input: "at" then ENTER. Modem must respond with "OK".
4. Input: "at&f", then ENTER, "at&s0", ENTER, "at+ipr=115200", ENTER, "at&w", ENTER. Modem must respond each time with "OK".

Preparing the device:

The program "Interface-Configuration" is used to make the following changes in the device:

1. The IP-address is set to match the PC's Dial-up entry.
2. For the initialization script, the combobox entry for the GSM-modem is selected.
3. In the command "AT+CPIN=XXXX", XXXX is replaced by the SIM-card's PIN. If the SIM-card doesn't require a PIN, "AT+CPIN=XXXX O AT OK" is deleted.
4. The Baud rate is set to 115200.

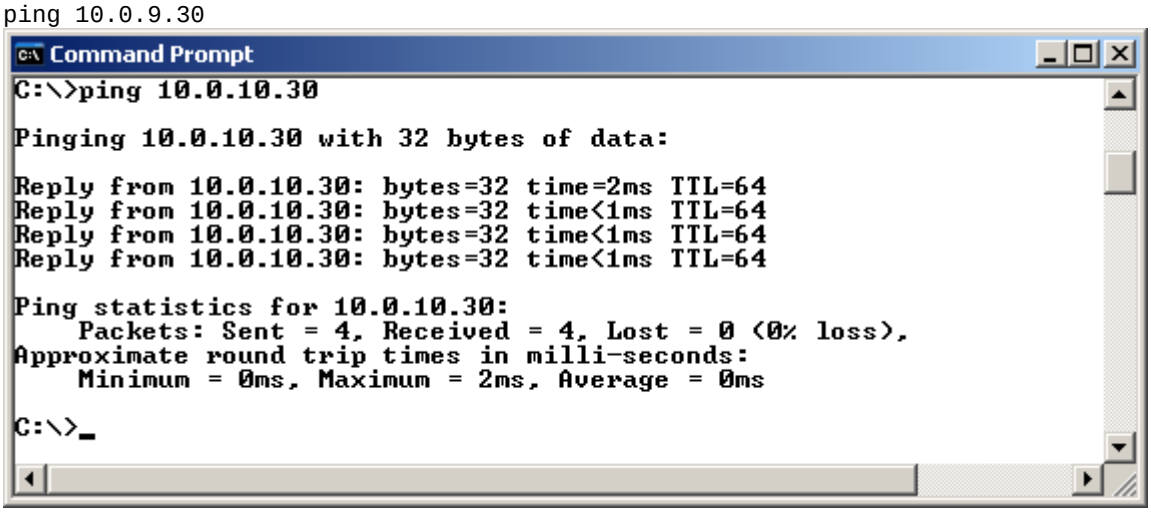
6.3.4.8 Determining the MAC-address of the measurement device

In order to find your imc measurement device's MAC-address, start the "input prompt" (e.g. Windows XP under Programs\Accessories).

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 307.

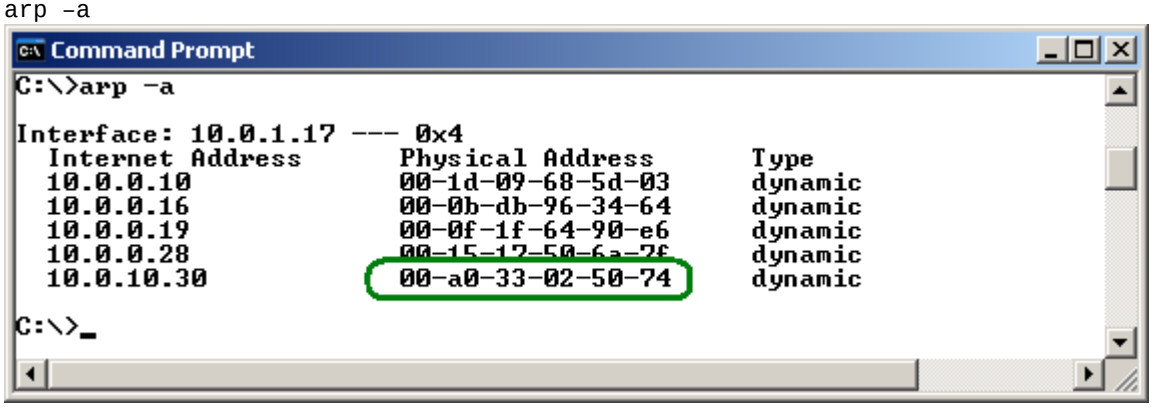
Example for the ping command:

```
ping 10.0.9.30
```



Next, enter the following command:

```
arp -a
```



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

6.3.4.9 Instructions for setting up a PPP-device

1. PC Control panel > call network
2. install TCP/IP Dial-Up adapter
3. set IP-address properties if no DNS-server is available
   a. IP 192.168.12.1 (e.g. imc)
   b. Sub 255.255.255.0
4. otherwise, configure DNS (normally, this has already been done)
   a. Host name: name; domain: imc.imc-berlin.de
   b. search order: add 192.168.11.1 to list
   c. Domain imc.imc-berlin.de
5. Creating a Dial-Up network
   a. General
      i. phone number
      ii. select modem
         1. General
            a. configuration of COM connection
            b. maximum speed 115200
         2. Settings: Standard
         3. Options: Standard
   b. Server types (win98SE)
      i. Type: PPP
      ii. Options TCP/IP everything else off
         1. TCP/IP settings
            a. permanently set 192.168.12.12
      iii. Script: Standard
      iv. Multilink: Standard

6. Call TCPIP-Config (from the imc program group)
   a. Search
   b. Choose device and call PPP settings
   c. IP-Address
      i. Set local address e.g. 192.168.12.13
      ii. Remote IP doesn't matter
      iii. Username & password if necessary
      iv. Script
         1. Default > don't change anything
         2. GSM
            Send ATZ\r
            Pause 1
            Send AT+CPIN=XXXX\r
            Pause 20
            Send ATS0=1 \r
            Pause 1
            Signal lcp open
7. Hardware: Default: 38400; Hardware
8. Write to device
9. Physically construct modem line
10. Set up Dial-Up connection
11. If the connection is established, start imc-software
    a. Choose device > edit
    b. Choose network settings
    c. Click TCP/IP
    d. Activate host address (in this example: 192.168.12.12)

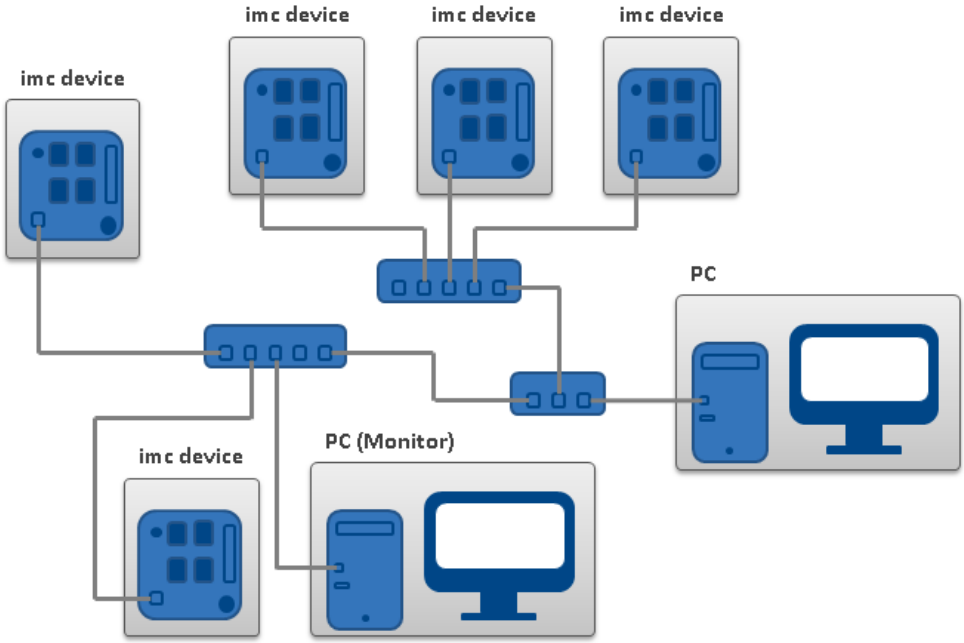
6.4 Optimize the network

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/hubs 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.
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 46! 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

6.5 Complex Networks

Theoretically, every type of network can possibly function. Until now, we have only considered networks consisting of a PC and n imc Devices units. However, it is possible for example, to have 3 PC's, two printers and n imc-Devices units connected in a network:



Monitoring

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

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.

7 Setup - Advanced Device Functions

This is where to find the documentation for the devices functions in the plug-in Setup.

Overview

- Fieldbusses 338
- Storage Options and Directory Structure 586
- Device Hard Disk, removable drive 616
- Synchronization 629
- imc Display Editor 664
- imc Messaging 682
- Configure via FTP 707
- imc REMOTE WebServer 720
- imc Online FAMOS 739
- Miscellaneous 792

7.1 Device Overview

imc STUDIO connects with the imc measurement devices e.g. via the local area network (LAN). In order to be used e.g. via the LAN, the devices must normally be prepared (see Setup - Connect the device 295).

Some of the capabilities discussed in this manual only pertain to certain device models. The associated device groups are indicated at the respective locations in the manual. The groups are shown in the following table which can be used with imc STUDIO.

— not available • standard ▯ optional

CRFX imc CRONOS*flex* CRC imc CRONOS*compact* CRPL imc CRONOS-PL

Device	Interface protocol / Bit/s		Supported data carrier			RAM		Rate [1] [kHz]	Short description
	Std.	MBit/s	CF [2]	PCMCIA	Hard drive	Data [MB]	Interface [MB]		
Group 2: SN12XXXX									
imc CRPL -2, -3, -4, -8, -13, -16 imc CRSL-2, -4	TCP/IP	100	—	•	▯	14	16 (32 from 2007	400	Modular system to identify on date of manufacture (as of summer, 2003)
Group 3: SN12XXXX									
imc C1 imc C-SERIES	TCP/IP	100	—	•	—	14	32	400	
Group 4: SN13XXXX									
imc BUSDAQ imc BUSDAQ*flex*	TCP/IP	100	•	—	▯	16	32	400	Field bus data logger
imc SPARTAN	TCP/IP	100	•	—	▯	16	32	400	Modular system
Group 5: SN14XXXX									
imc SPARTAN-R	TCP/IP	100	•	—	▯	16	32	400	Modular system
imc CRC-400 imc CRFX-400	TCP/IP	100	•	—	▯	16	32	400	Modular system
imc miniPOLARES	TCP/IP	100	•	—	—	16	32	400	
imc C1-N imc C-SERIES-N	TCP/IP	100	•	—	—	16	32	400	

Device	Interface protocol / Bit/s		Supported data carrier			RAM		Rate [1] [kHz]	Short description
	Std.	MBit/s	USB [2]	Express Card [2]	Hard drive	Data [MB]	Interface [MB]		
Group 6: SN16XXXX									
imc CRFX-2000	TCP/IP	100	•	•	0	16	512	2000 via EtherCAT else 400	Modular system

Device	Interface protocol / Bit/s		Supported data carrier			RAM		Rate [1] [kHz]	Short description
	Std.	MBit/s	USB [2]	CFast [2]	Hard drive	Data [MB]	Interface [MB]		
Group 7: SN19XXXX									
imc CRC-400GP imc CRC-2000G imc CRFX-2000G(P)	TCP/IP	1000	•	•	0	16	256	2000 via EtherCAT else 400	Modular system

1 Max. aggregate sampling rate (see data sheet)

2 We recommend storage media that are tested by imc (please consider the hotline for a current list)

Groups	Storage media
Group 4-6	For the purpose of onboard data storage, devices within those groups, are equipped with CF-Card, ExpressCard slot. imc CRONOS-devices within those groups can be equipped with an internally fixed hard drive available as an option.
Group 7	For the purpose of onboard data storage devices within this group are equipped with an CFast-Card slot. Devices within this group can be equipped with an internally fixed hard drive available as an option.

7.2 Fieldbusses

This chapter describes the different fieldbusses and their operation with imc STUDIO.

7.2.1 General Notes on Field-Busses

Field-bus channels: Operating methodology

Analog channels are configured using the setup page.

There is a dialog corresponding to each Field-bus, the Field-bus Assistant. In a sense, these dialogs allows you to configure the "digital signal conditioning" for the field-bus channels. Just like with the amplifier channels, the channels defined in the Field-bus 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 field-bus assistant. Once the field-bus channels have been created, the storage and recording settings can be specified on the Setup page for the digital channels. Field-bus channels can also be used in imc Online FAMOS and imc Inline FAMOS.

7.2.1.1 Field-bus Channels: Synchronization with Analog Channels

Parallel recording of analog data and Field-bus channels is possible.

With Field-busses, 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 100µs. 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 Field-busses, it is even two sampling periods.

If the clocks of an analog data recording channel and a Field-bus channel are to be compared, then the possible maximum time offset Δt_{max} is

$$\Delta t_{max} = \max (100\mu s, 2 * \max (t_{samp\text{-}analog}, t_{samp\text{-}FB})) + t_{FB\text{-}delay}$$

max(a,b)	function that returns the larger of two values
$t_{samp\text{-}analog}$	the sampling period used on the analog channels
$t_{samp\text{-}FB}$	the sampling period used on the field-bus channels
$t_{FB\text{-}delay}$	the maximum delay between A/D conversion in field-bus sensor and actual transmission on the field-bus

Example

A CAN channel is sampled every 2ms (sampling rate = 500Hz) and an analog channel every 1ms (sampling rate = 1000Hz). The maximum delay on the field-bus is 3 ms.

Δt_{max} = 2 * 2ms + 3ms = 7ms

7.2.1.2 General Notes on Sampling and Time Stamping

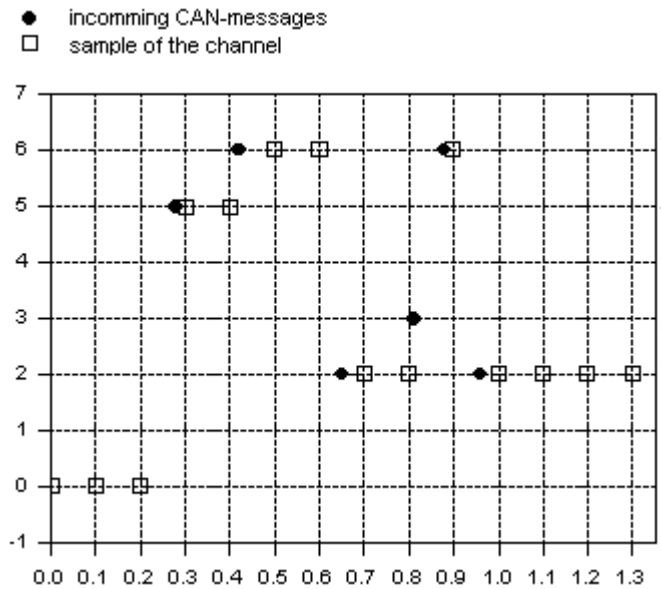
Each channel which is recorded with the field-bus, can be stored in two different ways.

- Sampled with a fixed rare
- Time stamp for each sample

7.2.1.2.1 Data Recording with Sampling

Field-bus 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 field-bus 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.

For example, many processing algorithms (e.g. digital filters, FFT's) require equidistant data.



imc STUDIO samples fiel-bus channels equidistantly using the following algorithm: if a new message has not arrived after the specified sampling interval, the previous value is copied to the current one.

This resulting data therefore consist of a series of amplitudes and are therefore more compact than data saved using time stamps. The time coordinate can be determined from the start time and sampling period stored in the data header.

This method has important consequences:

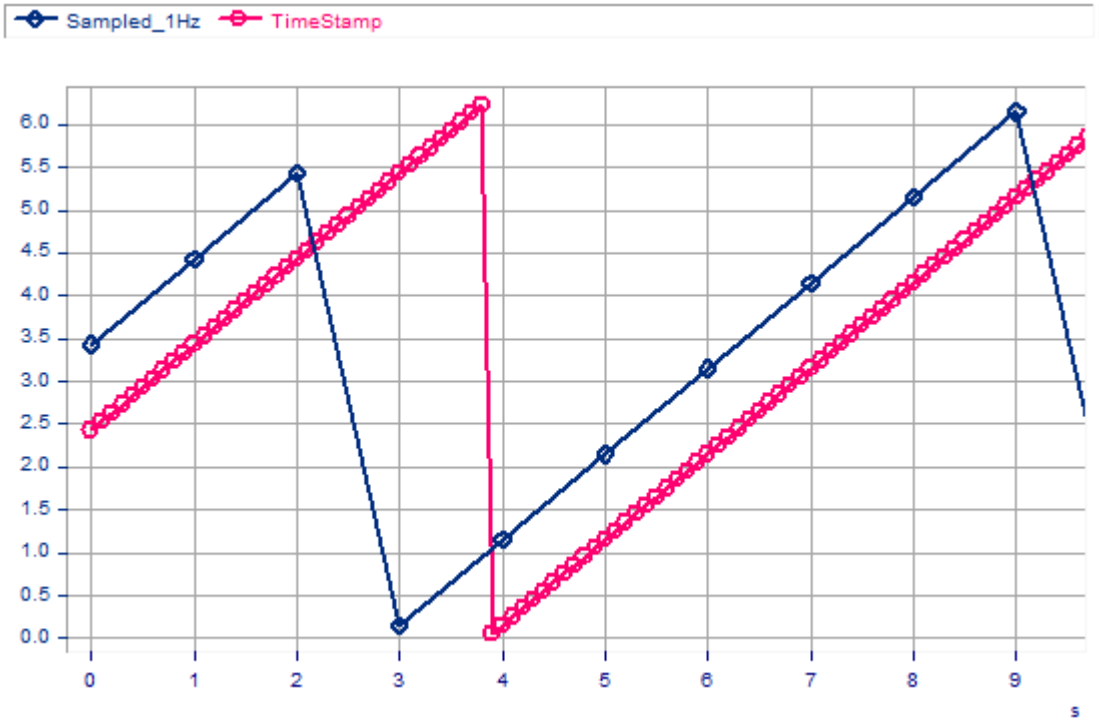
- As long as no messages have been received, the field-bus channel is filled with zeroes. This can only occur at the beginning of an untriggered measurement before the first field-bus messages have been received. When a measurement is stopped and restarted without being re-initialized, there are usually previous messages which are being used. In general, zeroes are not created when messages are being sent every 200ms 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.

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.

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 100ms. The same channel is additionally sampled at a fixed sampling rate 1Hz. As the diagram illustrates, both channels begin at 0s. The first value of the 1Hz 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.

7.2.1.2.2 Data Recording with Time Stamp

In time stamping mode, data values are only created when they are received. A time stamp is assigned to every measured value. This time stamp states the time of reception of the message containing the channel data. The time stamp requires 6 bytes. This is necessary in order to ensure high resolution of the time even for long-term measurements. A single time-stamped sample thus requires 6 bytes more memory than a sample with fixed sampling rate.

When to use time stamping

- When the exact reception times of messages need to be known.
- When the sensor transmits messages at very irregular intervals and equidistant sampling would lead to an unnecessary increase in the amount of data.

Disadvantages of time-stamp data acquisition

- Field-bus channels recorded in time stamping mode, can't be proceeded in imc Online FAMOS!
- Interval saving of time-stamped field-bus-channels to the PC is not supported! Instead, you can have the data saved at intervals to the internal hard drive
- 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.

7.2.1.2.2.1 Setting the Buffer Duration, Circular Buffer Duration and Pretrigger Duration for Time-stamped Channels

The buffer duration, circular buffer duration and pretrigger duration are specified as a time duration in the Setup. 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 Sreal, then the duration to set can be calculated according to the following equation:

$$t_{toset} = t_{real} * S_{real} / S_{assumed}$$

t_{toset}	the buffer duration, circular buffer duration or pretrigger duration [s]
t_{real}	the desired actual circular buffer duration and pretrigger duration [s]
$S_{assumed}$	the average sample rate assumed by the software $S_{assumed}$ = 100 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.

$$t_{toset} = t_{real} * S_{real} / S_{assumed}$$
$$= 60s * (1200 \text{ Samples/s}) / (100 \text{ Samples/s})$$
$$= 720s$$

7.2.1.2.2.2 Setting the Buffer Duration, Circular Buffer Duration and Pretrigger Duration for Time-stamped Text Channels (time stamped ASCII, "TSA")

Time-stamped text channels (TSA-channels) are used for, among other things, message protocols of the various Field busses. 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 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 Field-busses 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 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.

$$D_{real} = S_{real} * (L_{real} + 12 \text{ Bytes/Sample})$$

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 buffer duration, circular buffer duration or pretrigger duration [s]
t_{real}	the desired actual buffer duration, circular buffer duration or pretrigger duration [s]
$D_{assumed}$	the average data rate assumed by the program [Bytes/s] $D_{assumed} = S_{assumed} * L_{assumed}$ $= 100 \text{ Samples/s} * 20 \text{ Bytes/Sample}$ $= 2000 \text{ Bytes/s}$
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 Bytes/Sample

D_{real} = S_{real} * (L_{real} + 12 Bytes/Sample)

= 150 Samples/s * (48 Byte/Sample + 12 Byte/Sample)

= 9000 Bytes/s

The desired buffer duration t_{real} is 1min.

t_{toset} = t_{real} * D_{real} / $D_{assumed}$

= 60s * (9000 Bytes/Sample) / (2000 Bytes/s)

= 279s

7.2.2 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 AERONAUTICAL RADIO, 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

Observe the general Field bus notes regarding the operation methodology 338 of fixed sampling rates 340 versus timestamp 342.

7.2.2.1 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.

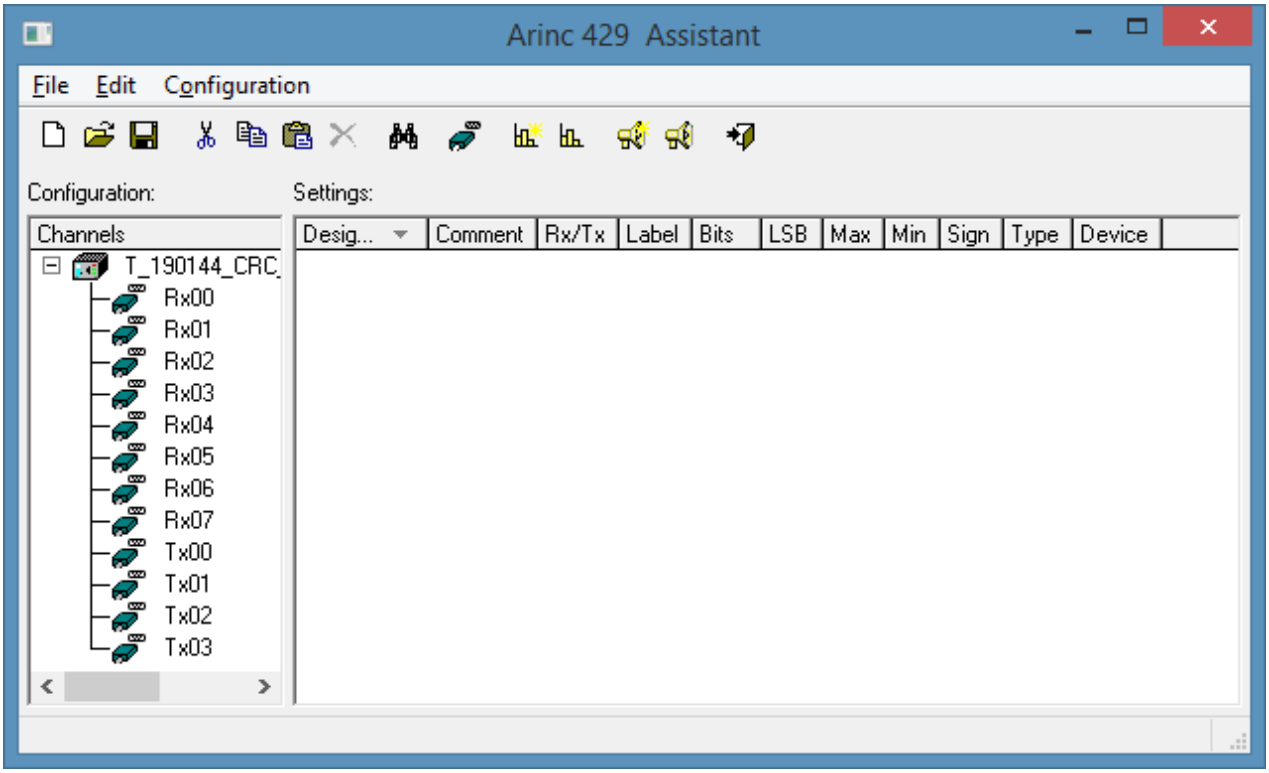
Philosophy of the ARINC-bus

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.

7.2.2.2 Starting the Assistant

Start the Assistant from within the imc STUDIO software via the Ribbon *Setup-Configuration* > *ARINC-Assistant*.

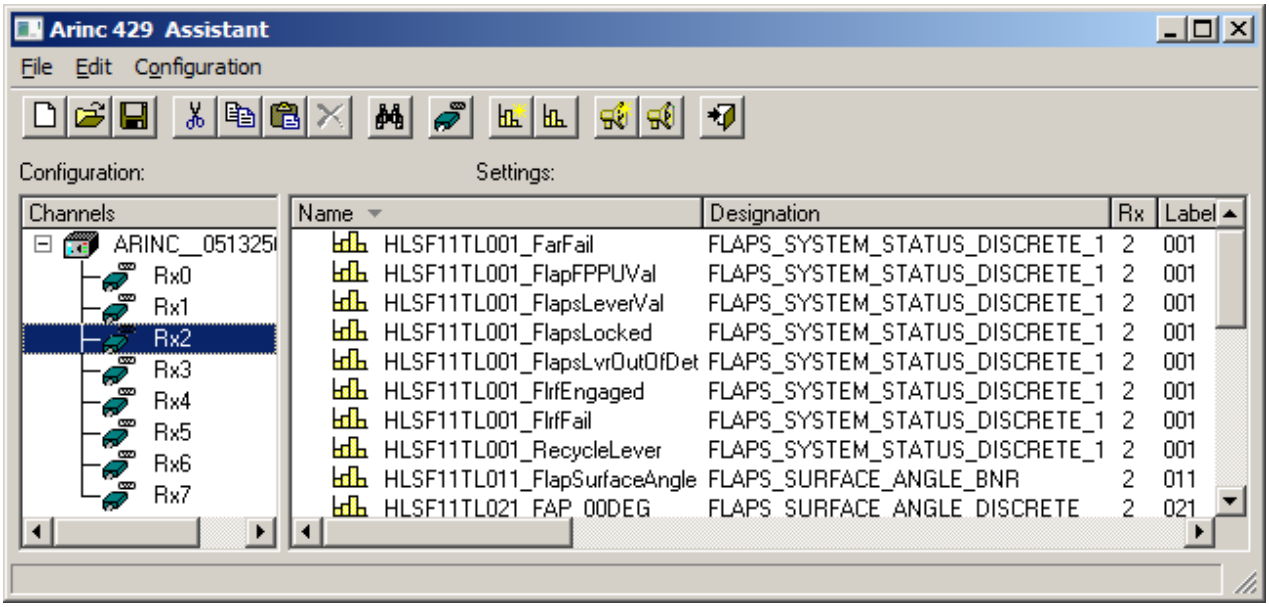
The Assistant begins by displaying the following dialog:



ARINC Assistant

The ARINC-Assistant's main window is divided in halves:

- The left side, designated Configuration 363, 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.

7.2.2.3 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.

As of imc STUDIO 4.0R1 (April 2013), the format *.idb2 is additionally available. With this format, the Parity-Bit and SSM are additionally saved.

7.2.2.3.1 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*.

7.2.2.3.2 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 protocol channels have been activated (Settings > protocol channel), those channes are listed in the channel list.
- If a new IDB file will be loaded, the protocol channels disappear from the channel list. However they are still active for the bus settings.
- After reloading a ARINC configuration, those protocol channels must be deactivated and reactivated. Settings like saving will be lost and must be redone.

7.2.2.3.3 Save Configuration File (as)

To save a configuration file, click on either the corresponding button in the toolbar or the menu item *File* > *Save*.

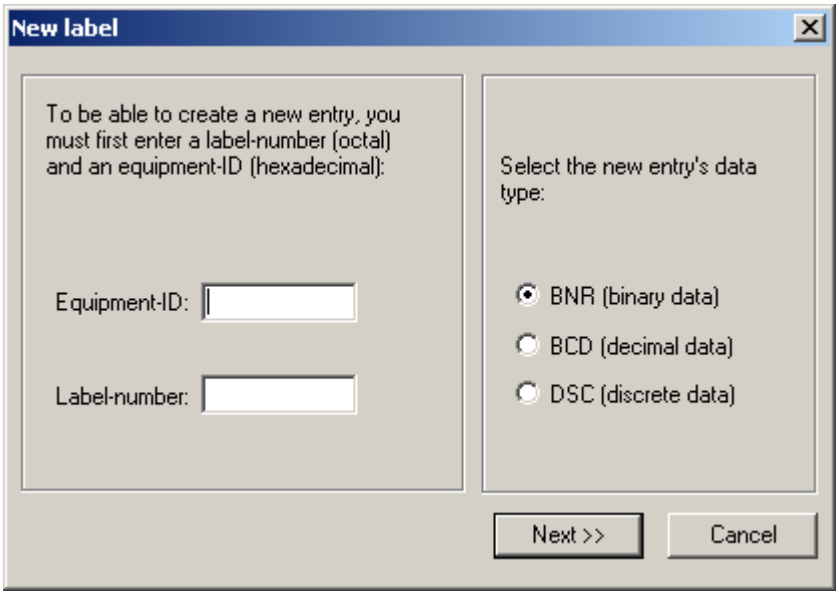
If the open configuration file is to be saved under a different name or in a different directory, menu item *File* > *Save as* is used. In the Windows file dialog which then appears, the name, without its extension, must be entered. The extension *.idb is appended automatically.

7.2.2.3.4 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.

7.2.2.3.4.1 Configuring a BNR Data Word

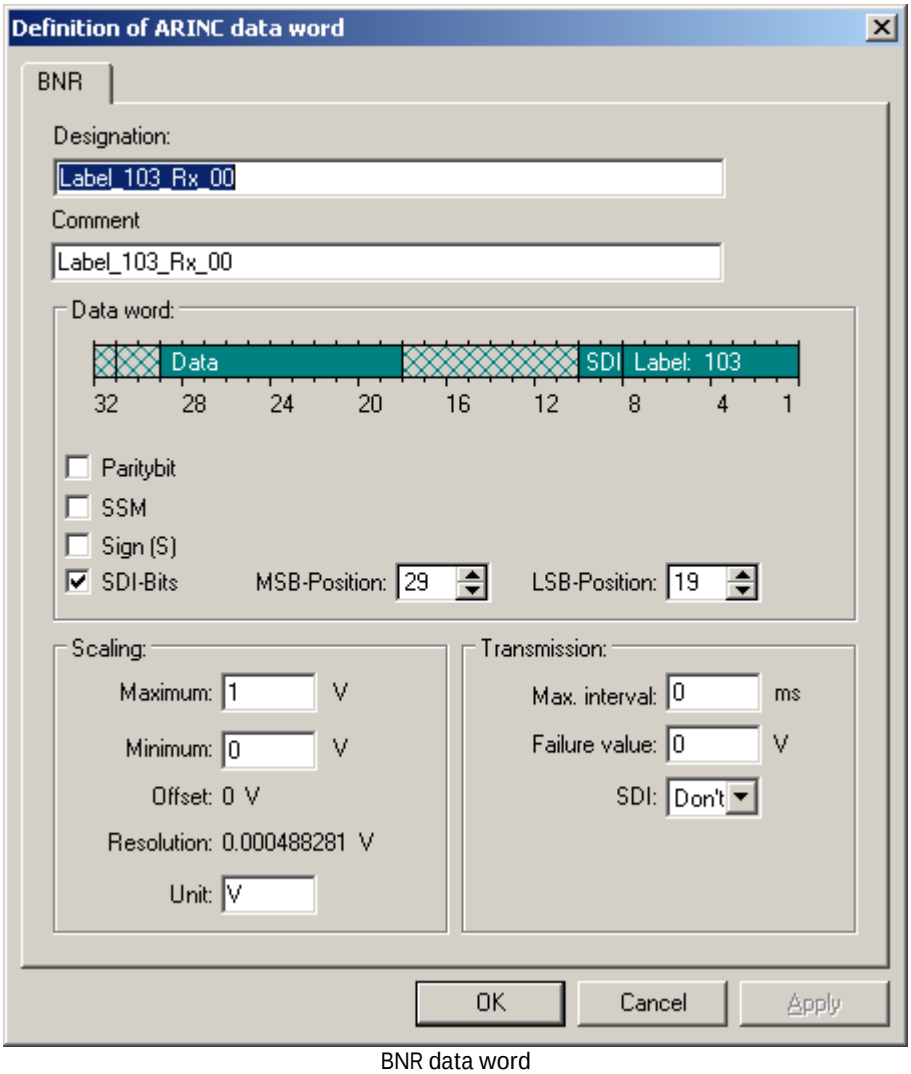
The dialog is thematically arranged in groups each denoted by a gray frame.

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

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.

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

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

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.

7.2.2.3.4.2 Configuring a BCD Data Word

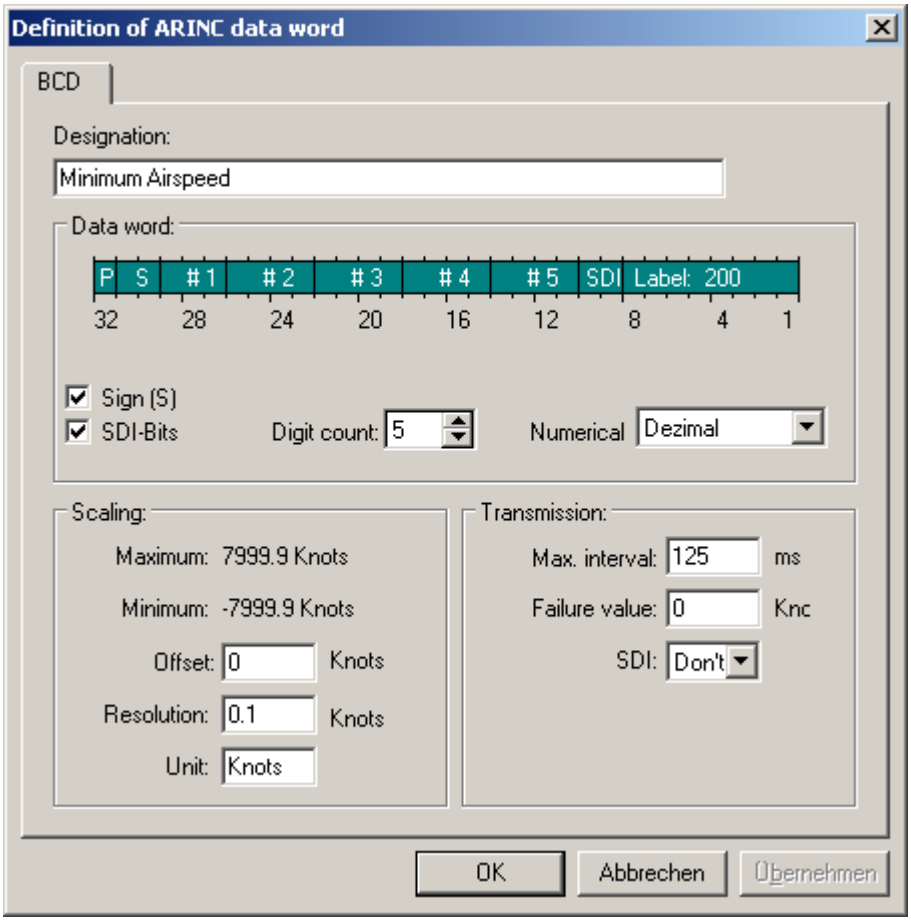
The dialog is thematically arranged in groups each denoted by a gray frame.

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

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

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.
- 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.

Number of digits:

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.

Numerical format:

Is a numerical format. There is a choice among three formats:

Decimal:

- The number is interpreted as a normal fixed-point number. This is the default case.

Angle:

- 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:

- 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.

Scaling

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

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.

7.2.2.3.4.3 Configuring discrete (DSC) Data Words

The dialog is thematically arranged in groups each denoted by a gray frame.

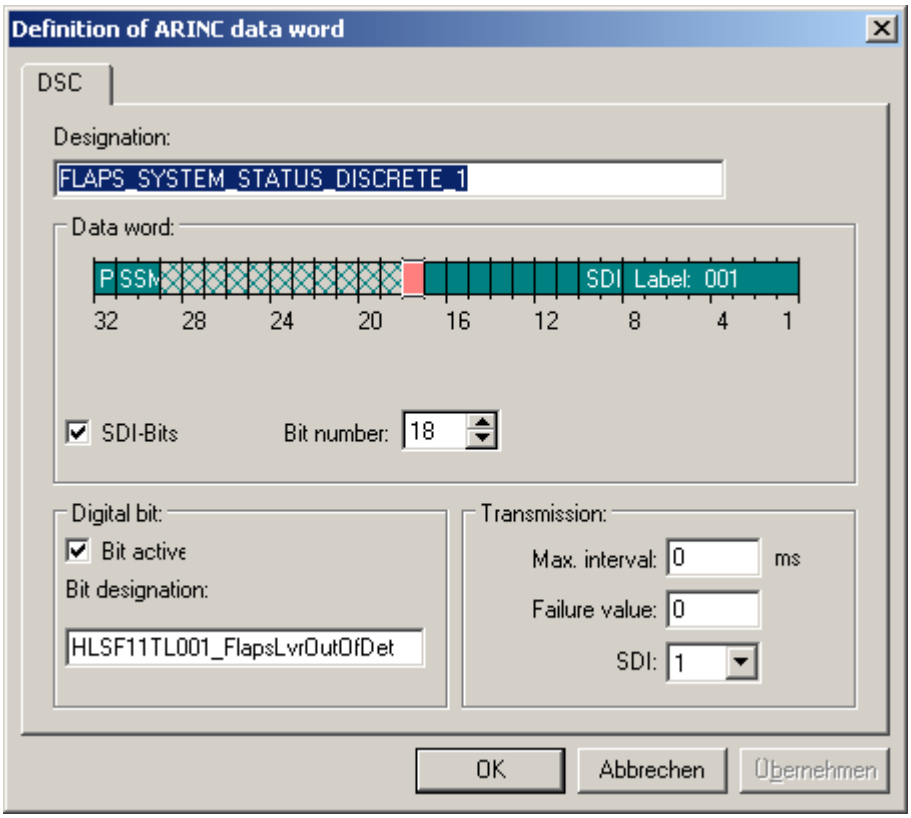
The group Data word comprises all settings concerning the structure of the ARINC data word. The currently set word structure is visually represented by a green horizontal bar.

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

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

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.

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

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.

7.2.2.3.4.4 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).

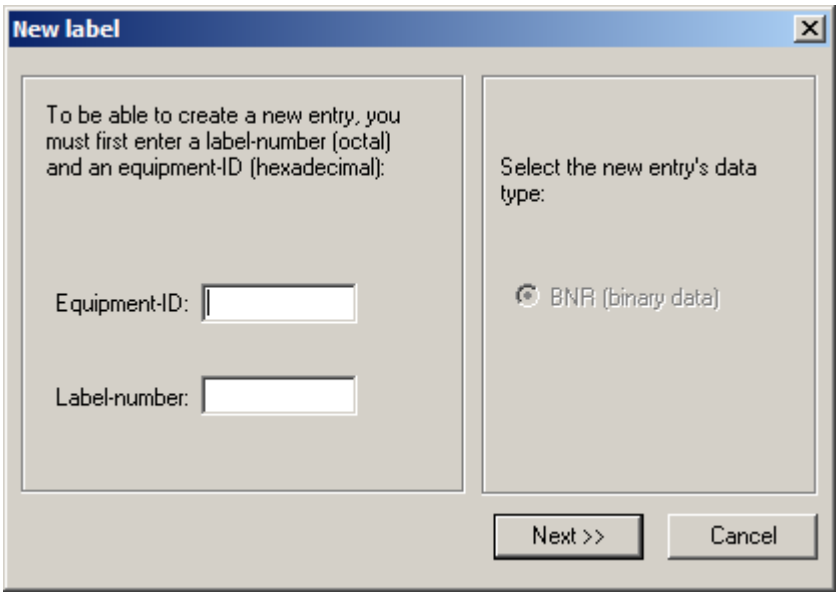
The Assistant allows any number of definitions to be set for a single label. In this context, even the data type (BNR, BCD or DSC) doesn't matter.

However, for ease of reference, it is good to specify a descriptive name in the Designation-control.

7.2.2.3.5 Transmitter Channel (Tx) - Creating a ARINC Label

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.

7.2.2.3.5.1 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.

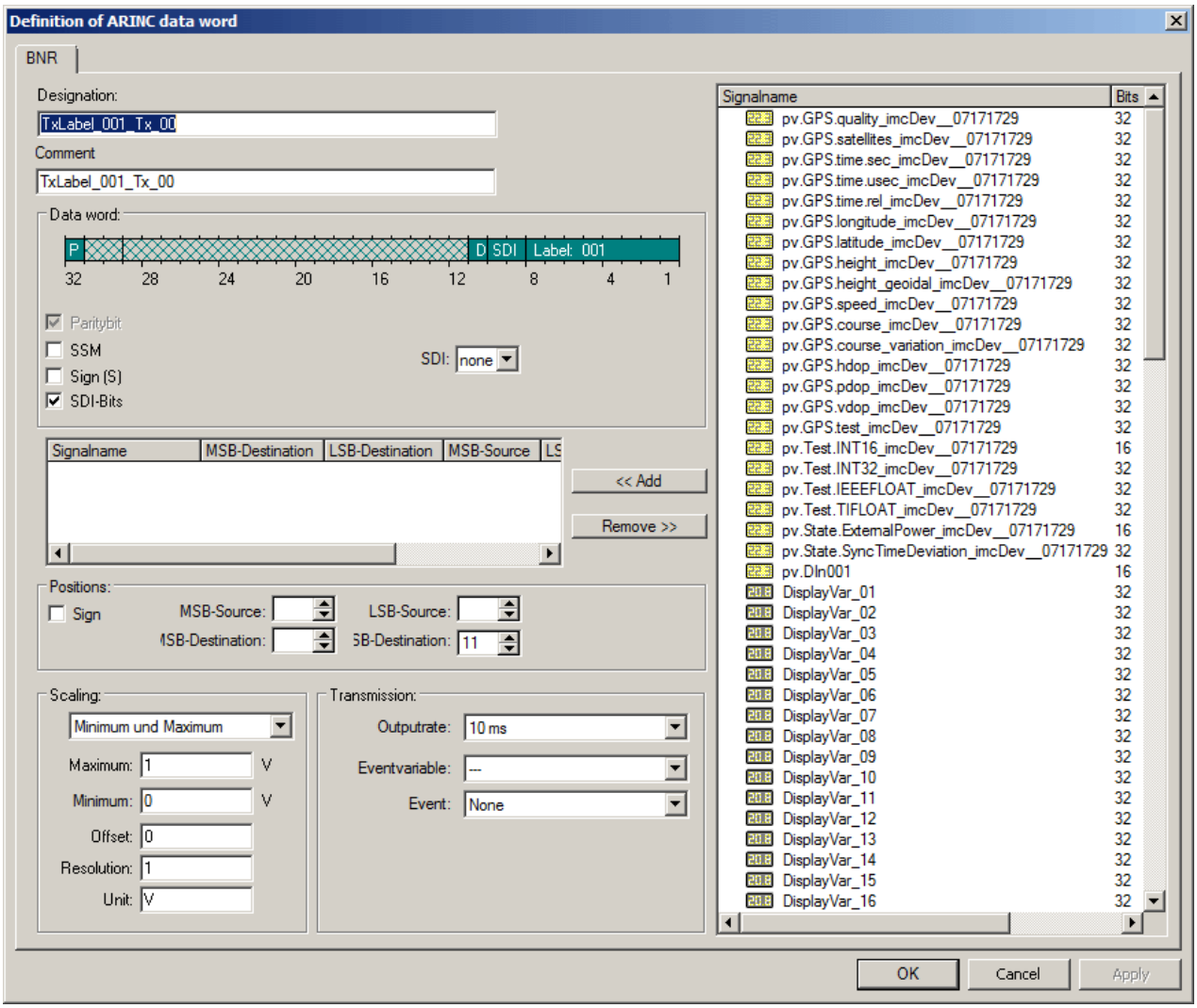
The table at left lists all data which can be added to the message to send.

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

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

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.

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.
```

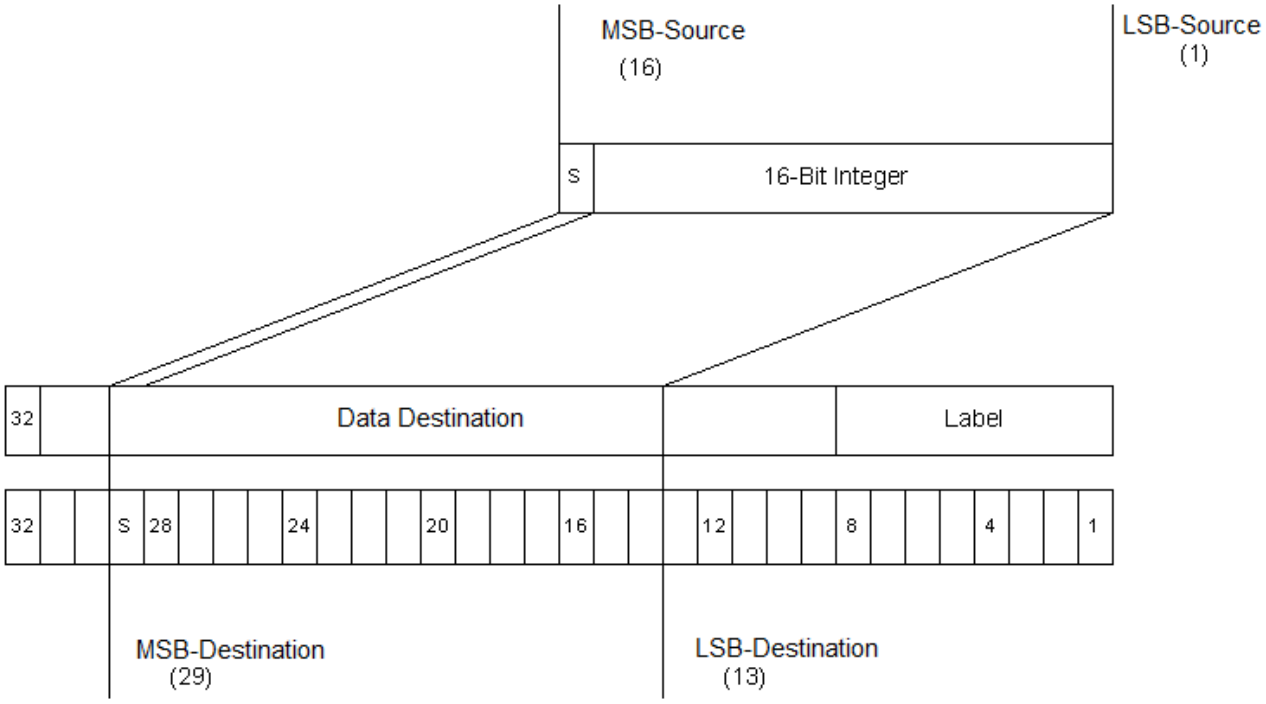
Sign:

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

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".

Output rate:

Cycle time for the output: 500µs - 1h, provided the Baud rate set is adequate.

Event variable:

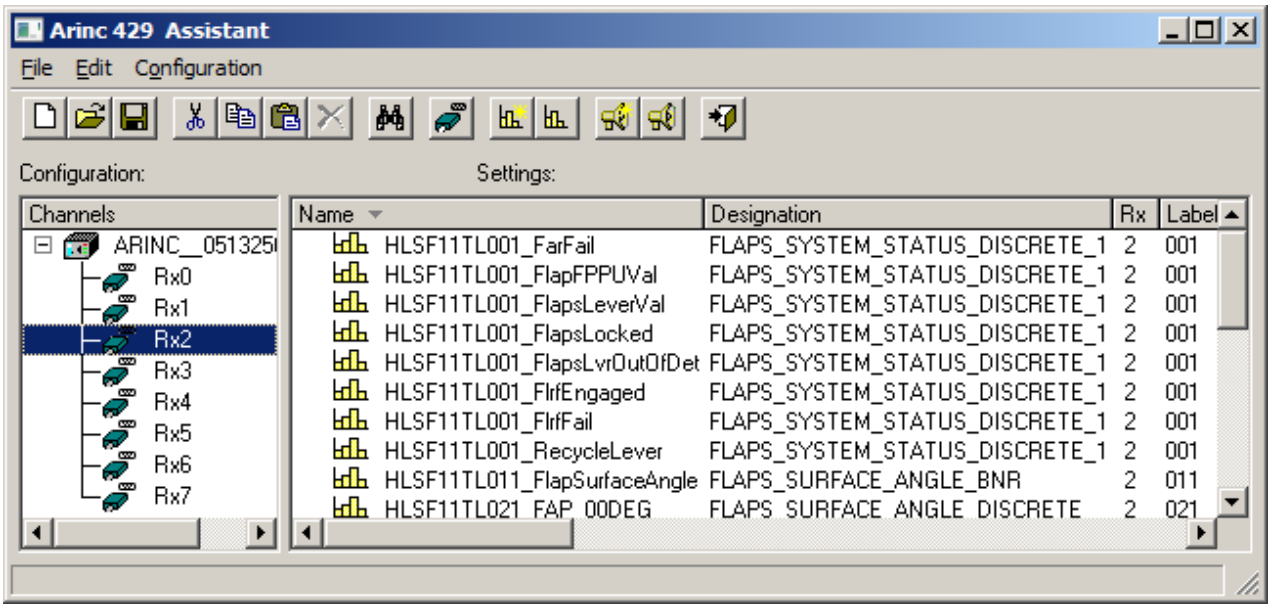
Sending only takes place once the variable entered here meets the condition of the following event.

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*.

7.2.2.3.6 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:



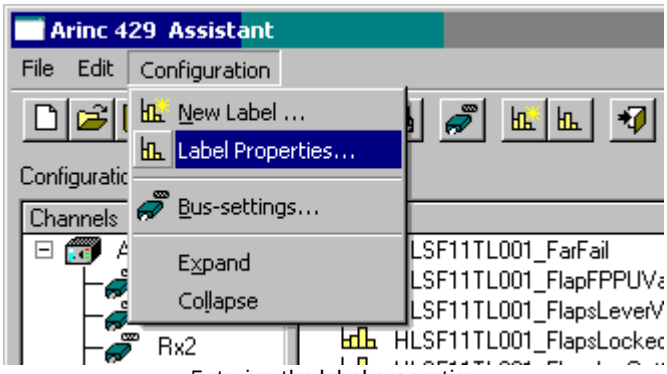
ARINC configration list

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.

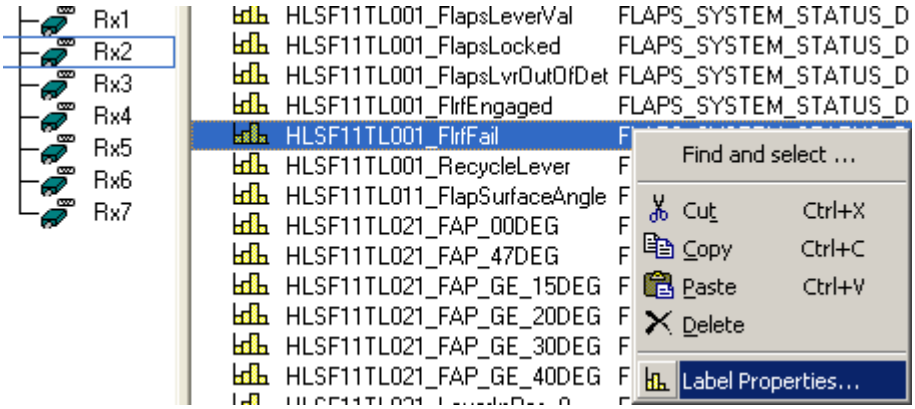
7.2.2.3.7 Editing a Label Definition

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.

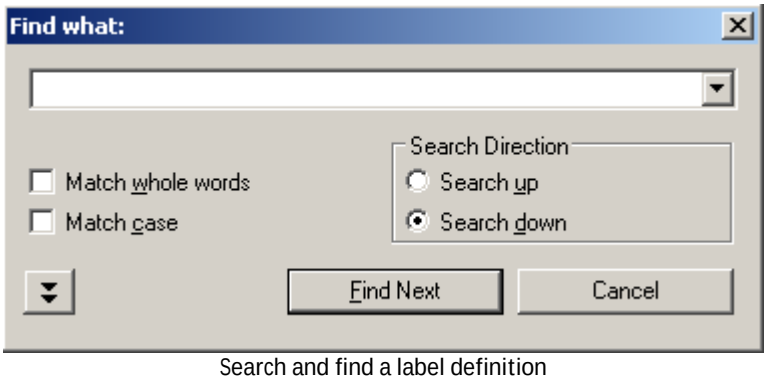
7.2.2.3.8 Cutting, Copying, Pasting or Deleting a Label Definition

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.

7.2.2.3.9 Searching and Selecting a Label Definition

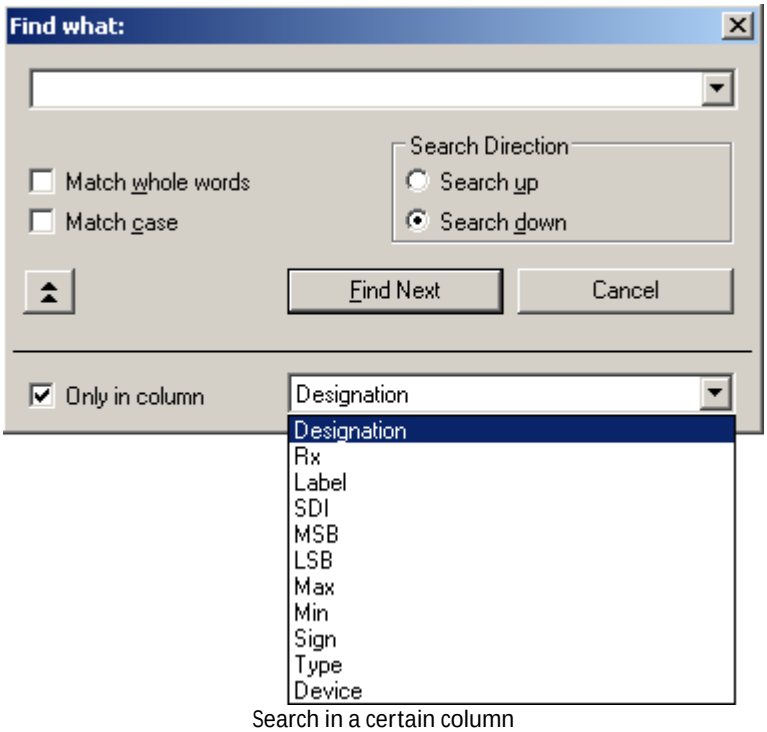
The context menu, called by right-clicking the mouse, also contains an item for the function Search and select. Using this function, you can search for a label definition directly.

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.



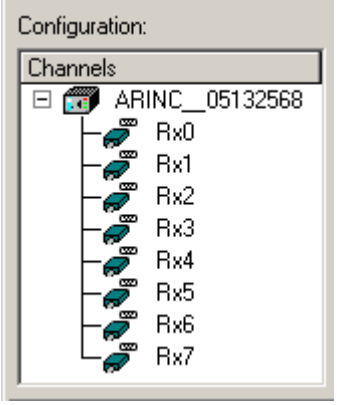
Search in a certain column

7.2.2.4 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.

7.2.2.4.1 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).

7.2.2.4.2 Changing Channel Parameters

Channel parameters like channel name, sampling rate, measurement duration etc. are changed in the same way as with an analog channel.

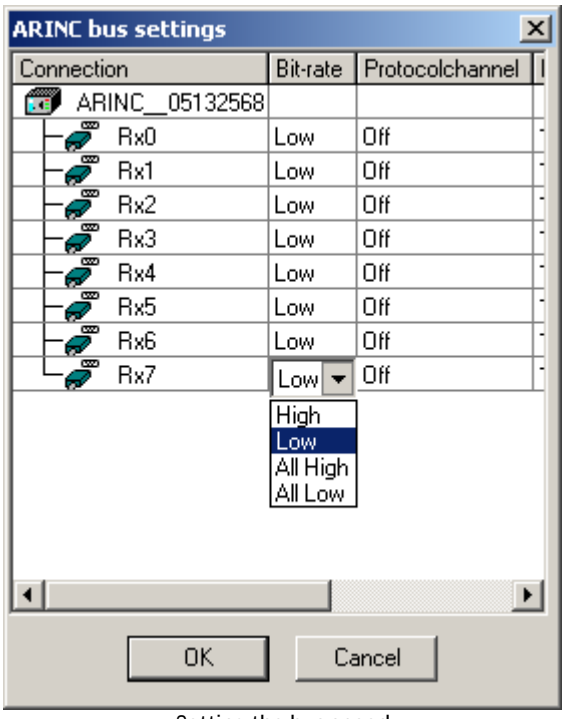
All channels configured with the Assistant are also listed as field bus channels.

7.2.2.4.3 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

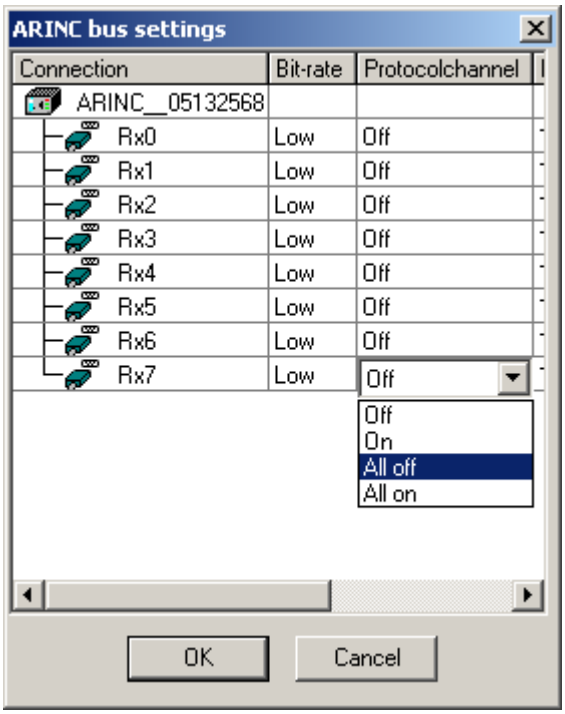
1. by clicking on the button .

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 Protocol 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 Protocol 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.

Protocol channels can be used for offline analysis of the data traffic, e.g. by means of the program imc FAMOS.

Note

When protocol channels are activated, these are indicated in the channel list.

If a new IDB channel is later imported via the IDB interface, the protocol 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.).

7.2.2.5 Specification IDB Configuration File

7.2.2.5.1 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	29
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

7.2.2.5.2 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 366

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

7.2.2.5.3 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”

7.2.3 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.

Notes

Also observe the general notes on file buses in regard to control philosophy 338 and the sampling rate either as fixed sampling rate 340 or time stamp 342.

7.2.3.1 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.

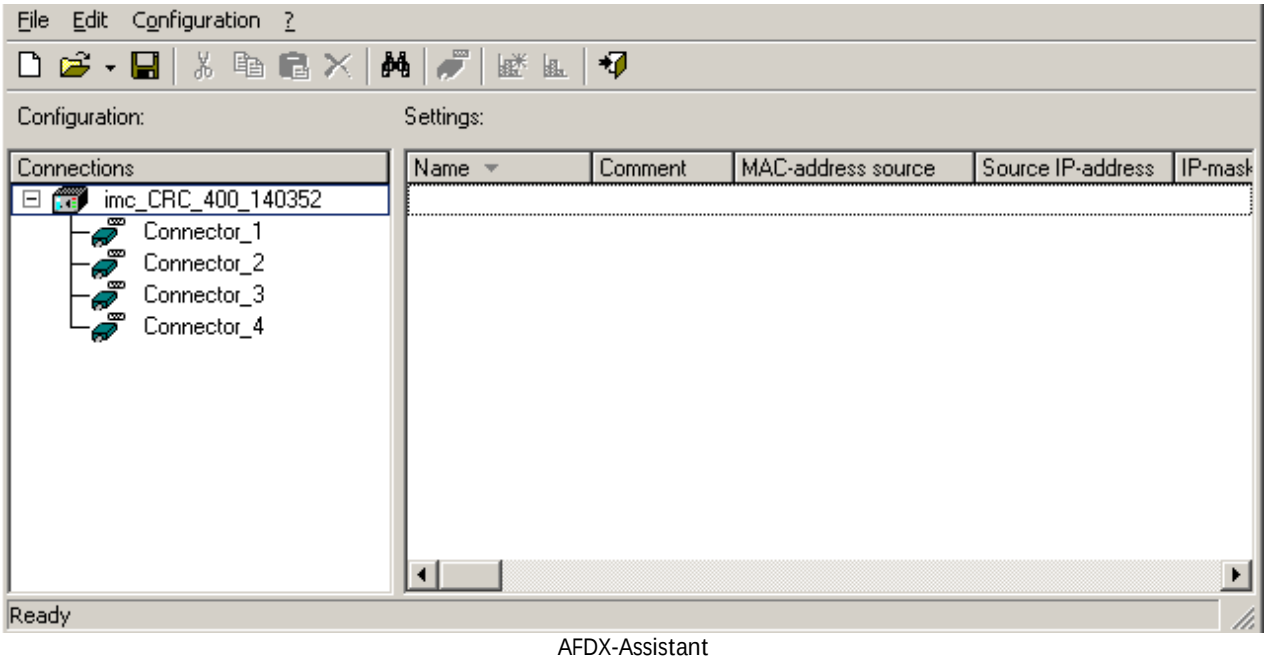
Once this step has been completed and the Assistant closed, an additional measurement channel is available in imc STUDIO, which can be used in a very similar way to any analog or digital channel. This means that the channel can be subjected to operations in imc Online FAMOS, for example, or linked together 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.

7.2.3.2 Starting the Assistant

In order to launch the Assistant from imc STUDIO, click in the menu ribbon on *Setup-Configuration* > *AFDX-Assistant*.

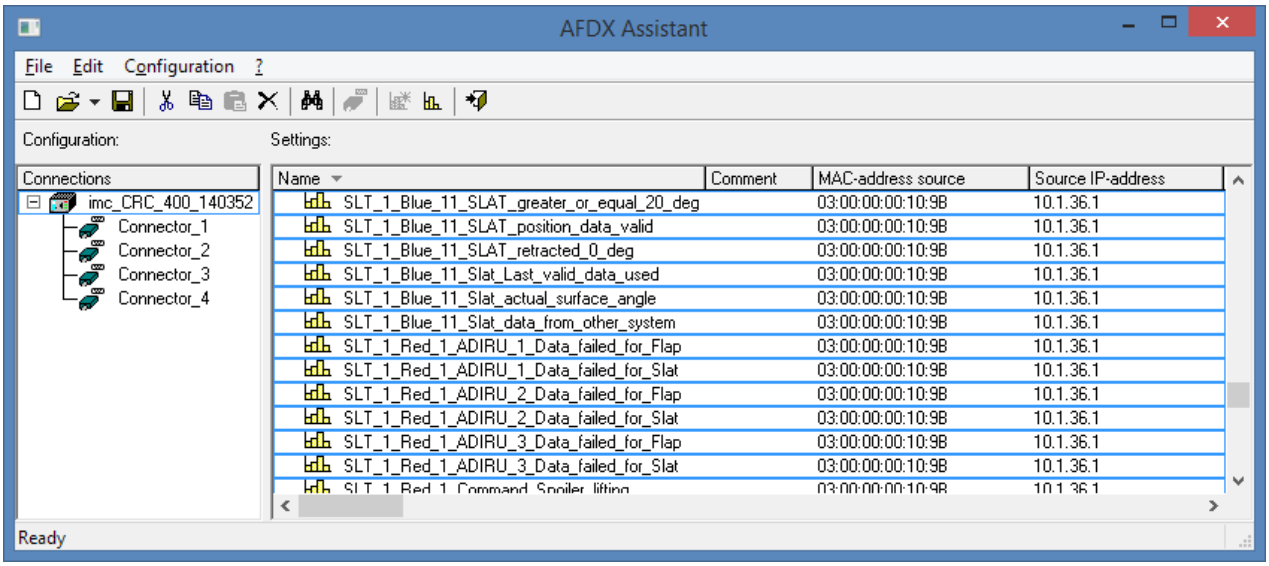
After launching, the Assistant appears with the following dialog:



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.

7.2.3.3 AFDX-configuration file

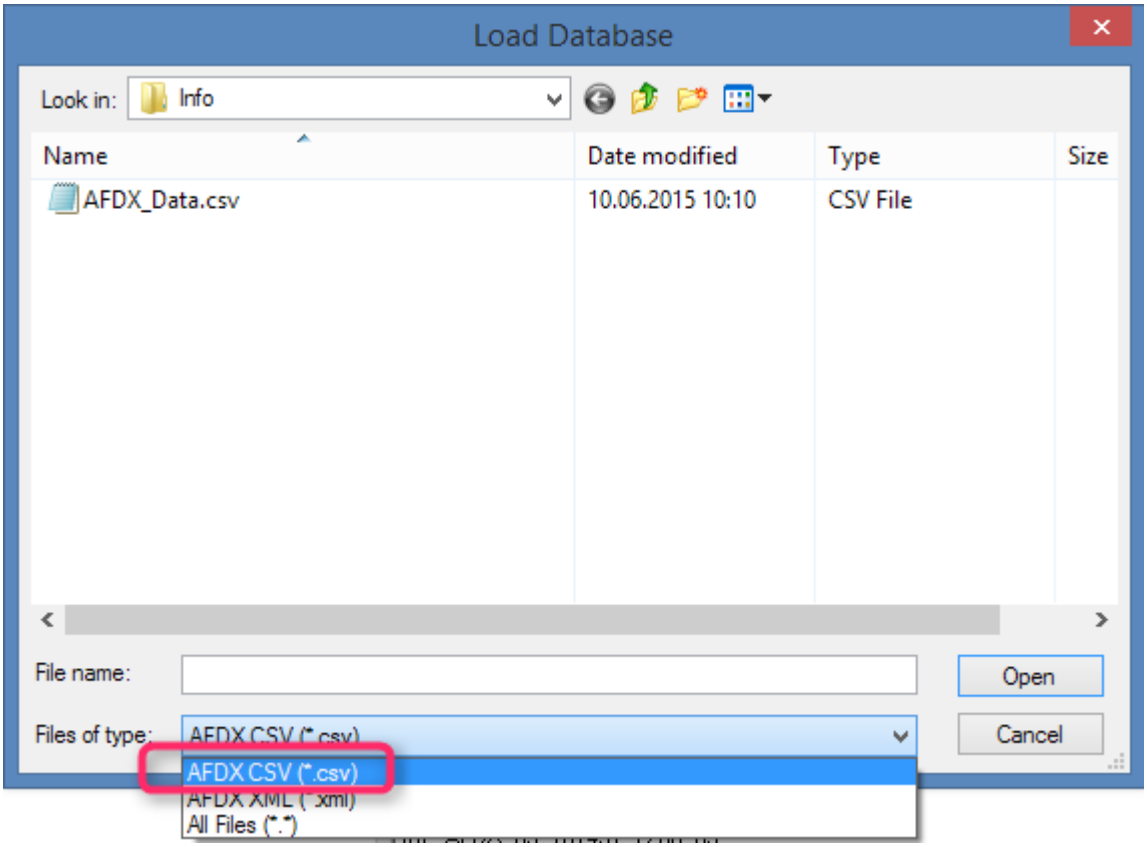
All AFDX channel definitions are saved in a configuration file. This configuration file can be used as a central database. The file can be loaded for every experiment, the AFDX channels can be deleted, copied, or moved as desired. Thus, the signal definitions are independent of the particular experiment. Based on this principle, a custom configuration for a new experiment can quickly be generated at any time from an existing configuration file.

7.2.3.3.1 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*.

7.2.3.3.2 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

7.2.3.3.3 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*.

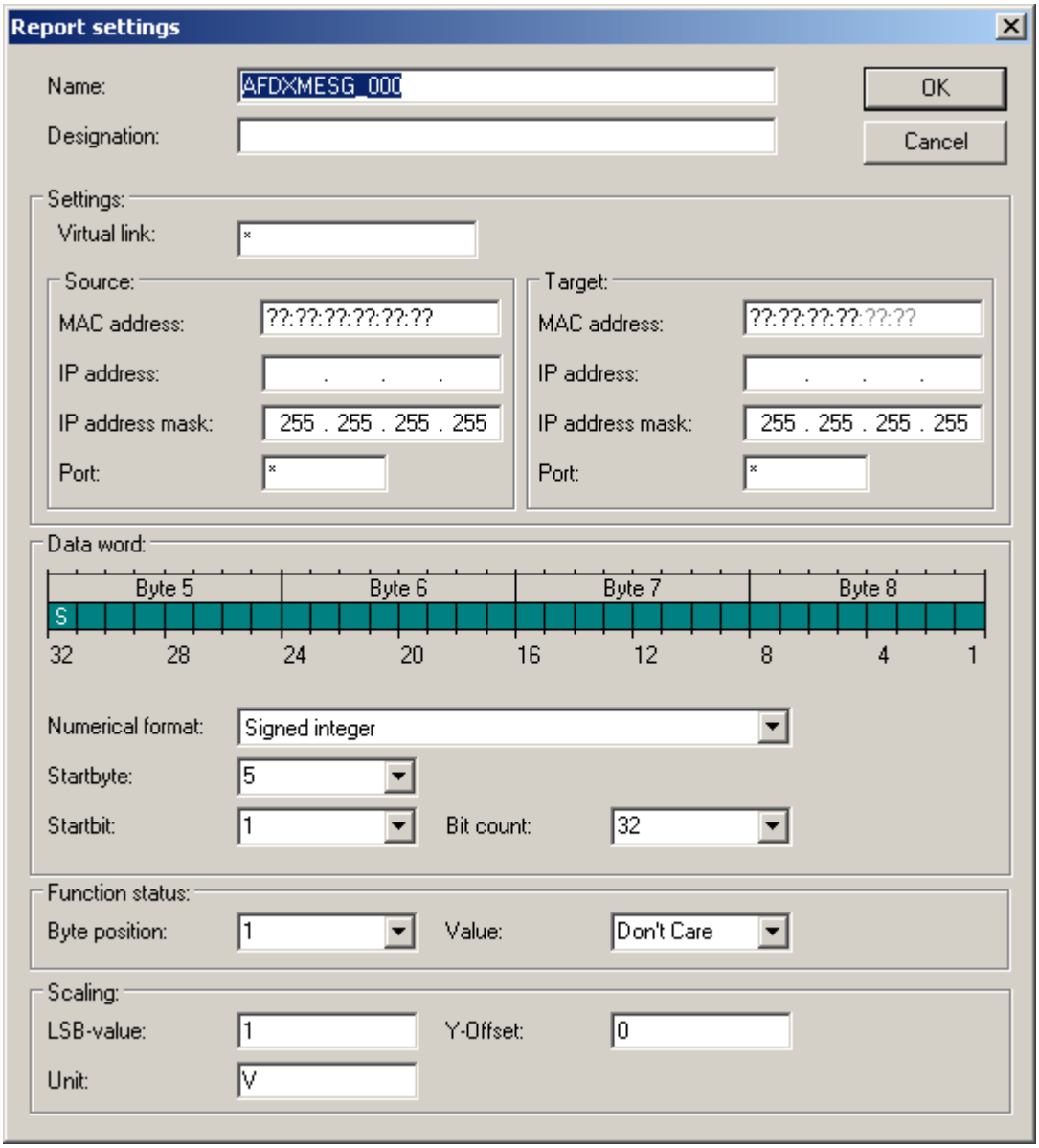
If you wish to save the loaded configuration file under a different name or path, use the menu item *File* > *Save As...*. In the Windows "File" dialog, you must ehn enter the filename without an extension. The extension *.xml is appended automatically.

7.2.3.3.4 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 374, Data Word 375, Function Status 376 and Scaling 376 are described further below.

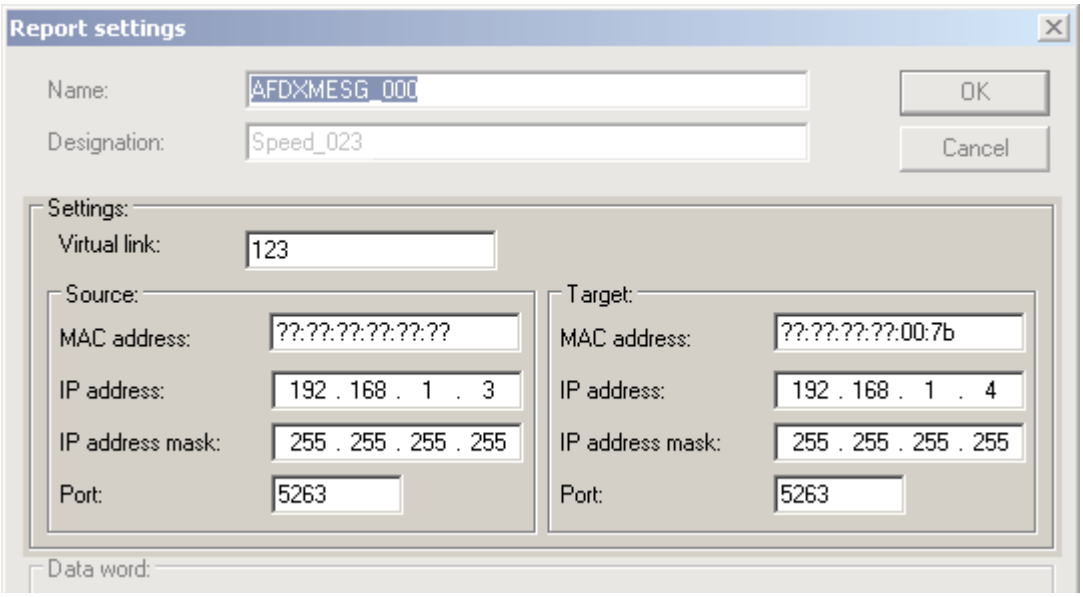
Name:

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.

7.2.3.3.4.1 Settings



Settings

Virtual Link:

Here, a virtual link can be entered, which defines the connection in one direction from a source to a destination.

Source and Target:

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:

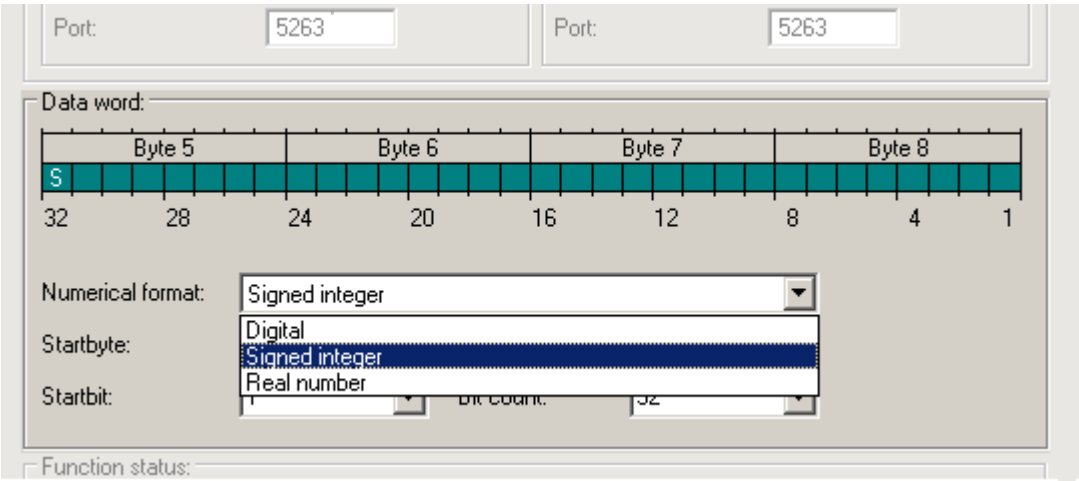
Here, the IP address mask of the source/target is entered, within which the corresponding IP address is located. The IP address mask denotes the network in which the source/target is located.

Port:

Here, the source's/target's port is entered.

7.2.3.3.4.2 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.



Data Word

Numerical format:

The available options are Digital, Signed Integer and Real. For all three numerical formats, the start byte can be selected.

Start byte:

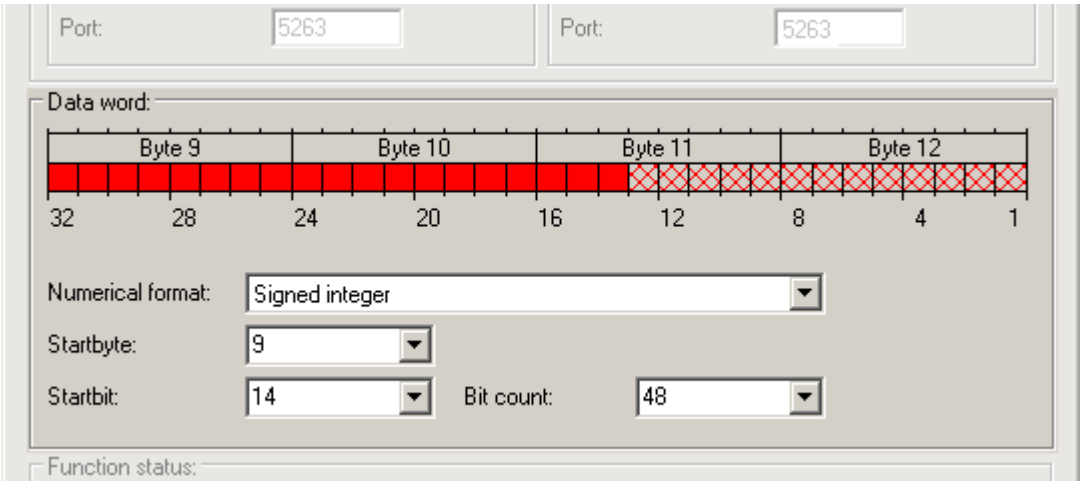
In the single-line text box, the desired start byte can be selected. Due to the limitation of the message size, only values up to 1496 are possible for the start byte.

Start bit:

The start bit can only be selected for the *numerical formats Digital* and *Signed Integer*. The start bit must be ≤ 32.

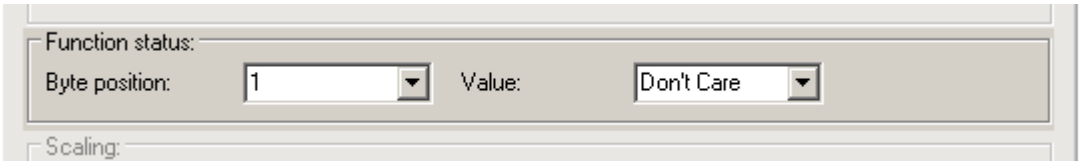
Bit count:

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

7.2.3.3.4.3 Function Status



Function Status

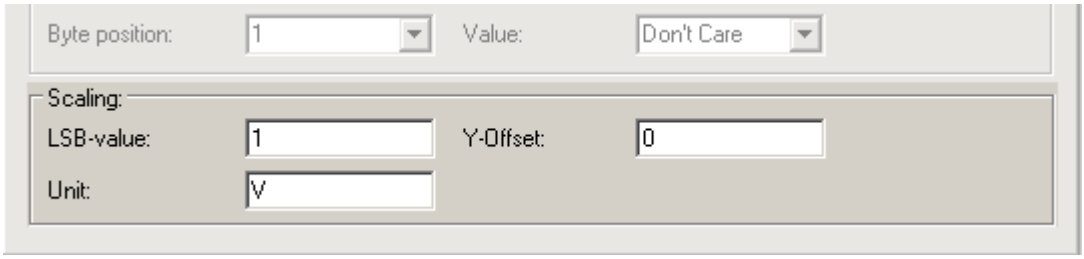
Byte position:

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.

7.2.3.3.4.4 Scaling



Scaling

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:

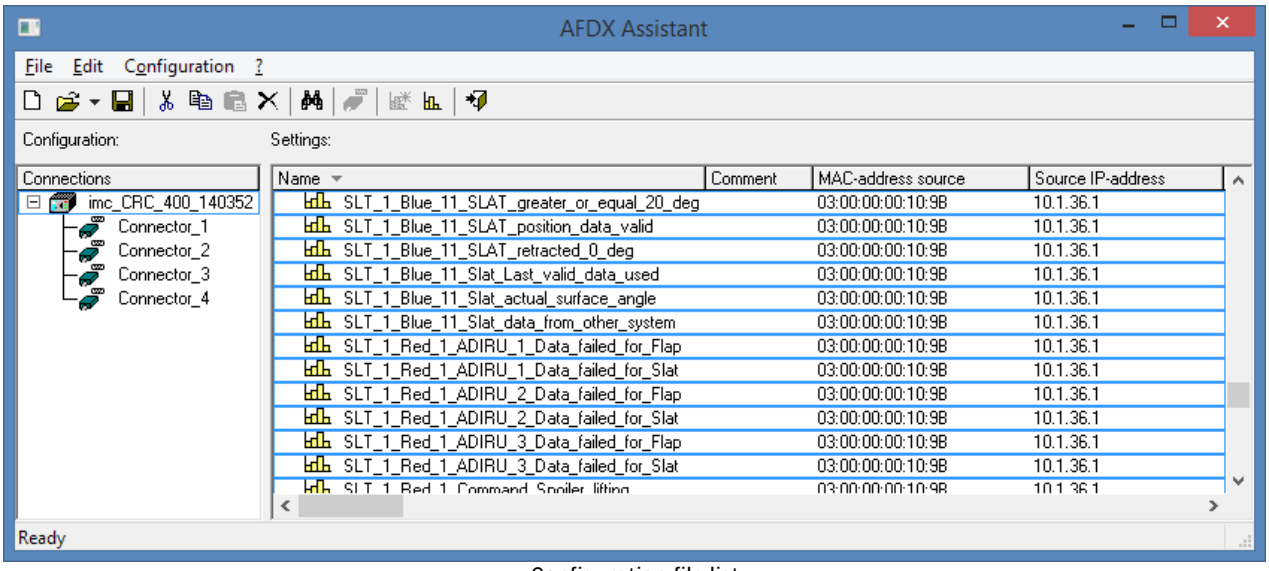
The signal's unit of measure.

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

7.2.3.3.5 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:



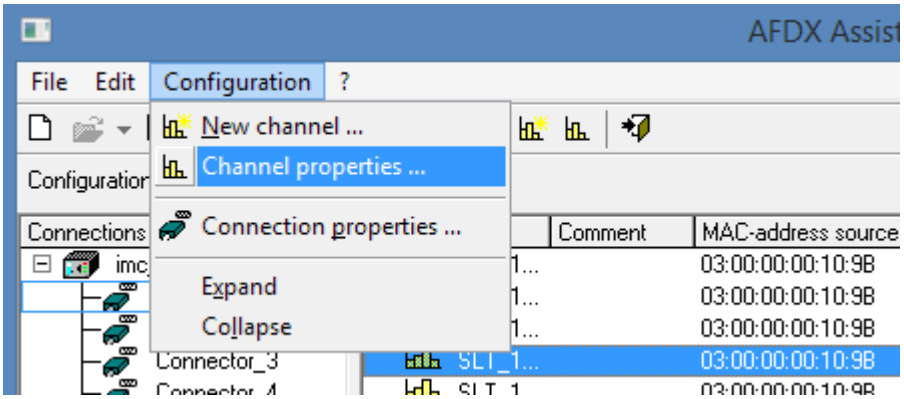
Configuration file list

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.

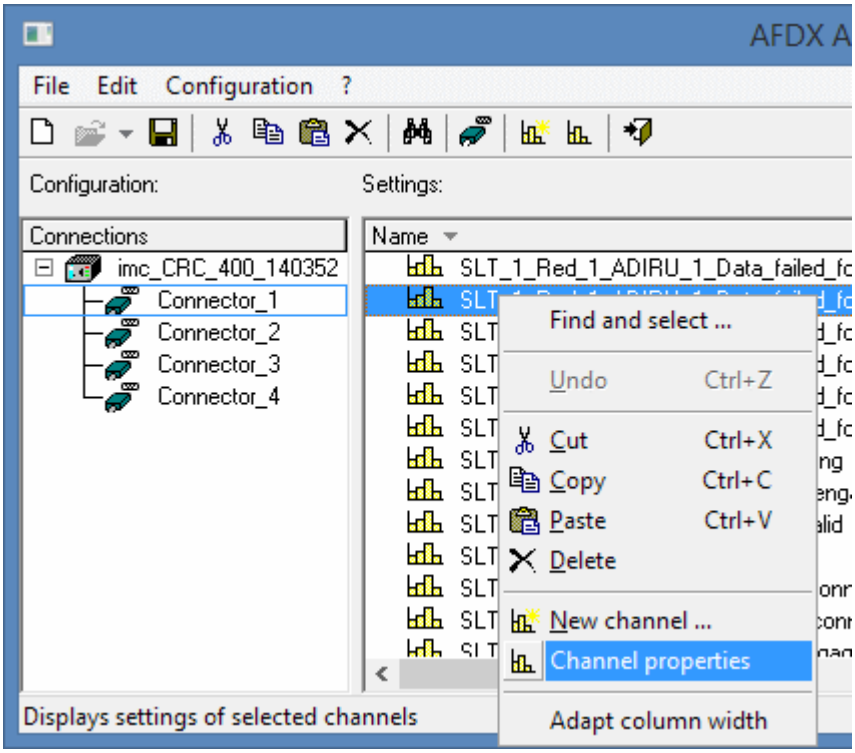
7.2.3.3.6 Editing an AFDX-channel

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.

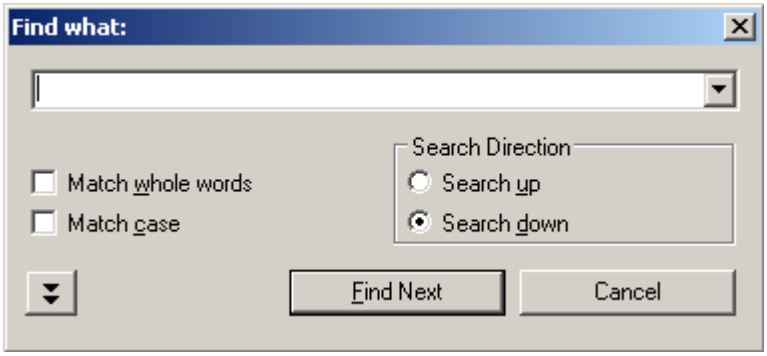
7.2.3.3.7 Cutting, copying, pasting or deleting an AFDX-channel

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.

7.2.3.3.8 Finding and selecting an AFDX-channel

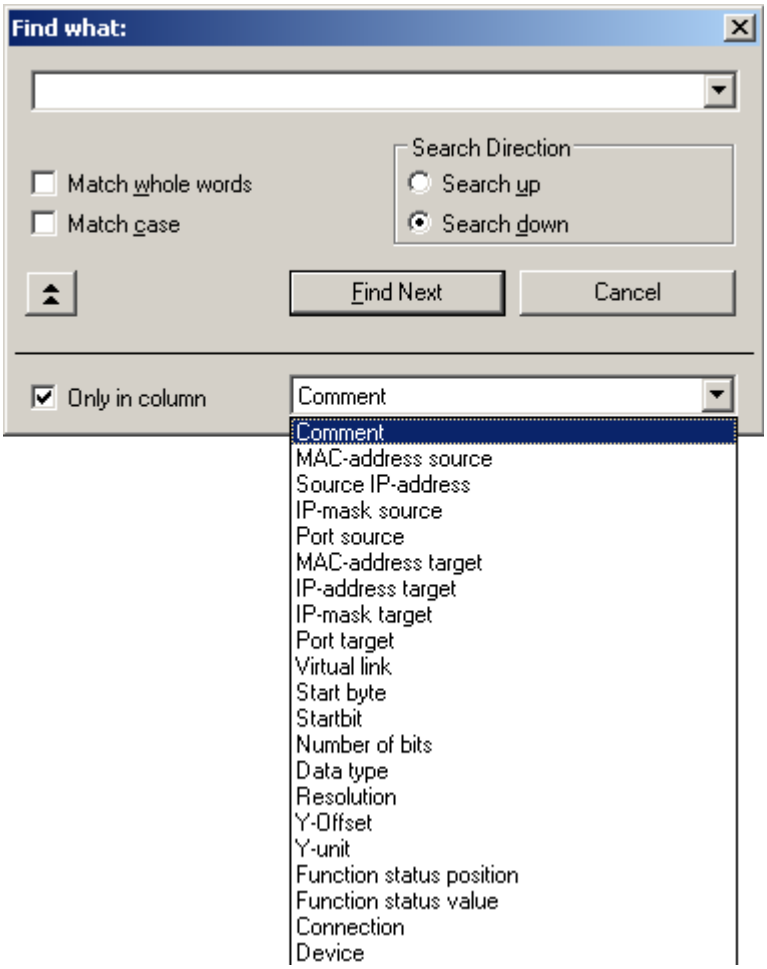
The context menu (accessed by right-clicking the mouse) also offers the function *Find and select...*. Using this, it is possible to directly search for particular AFDX-channels.

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.



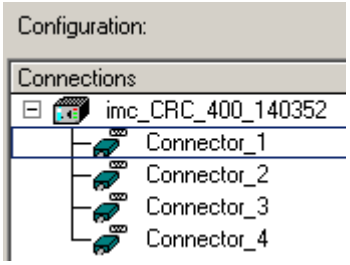
Searching in a particular column

7.2.3.4 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.

7.2.3.4.1 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.

7.2.3.4.2 Editing channel parameters

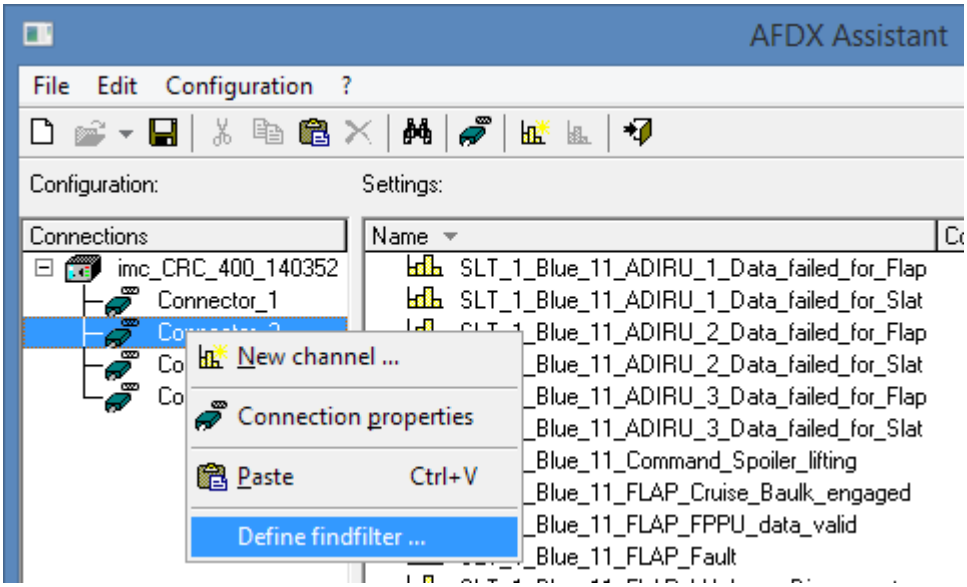
Channel parameters such as the channel name, sampling time, measurement duration and etc. are edited in the same way as for an analog channel.

All channels configured in the Assistant appear with the Fieldbus inputs.

7.2.3.4.3 Defining search filters

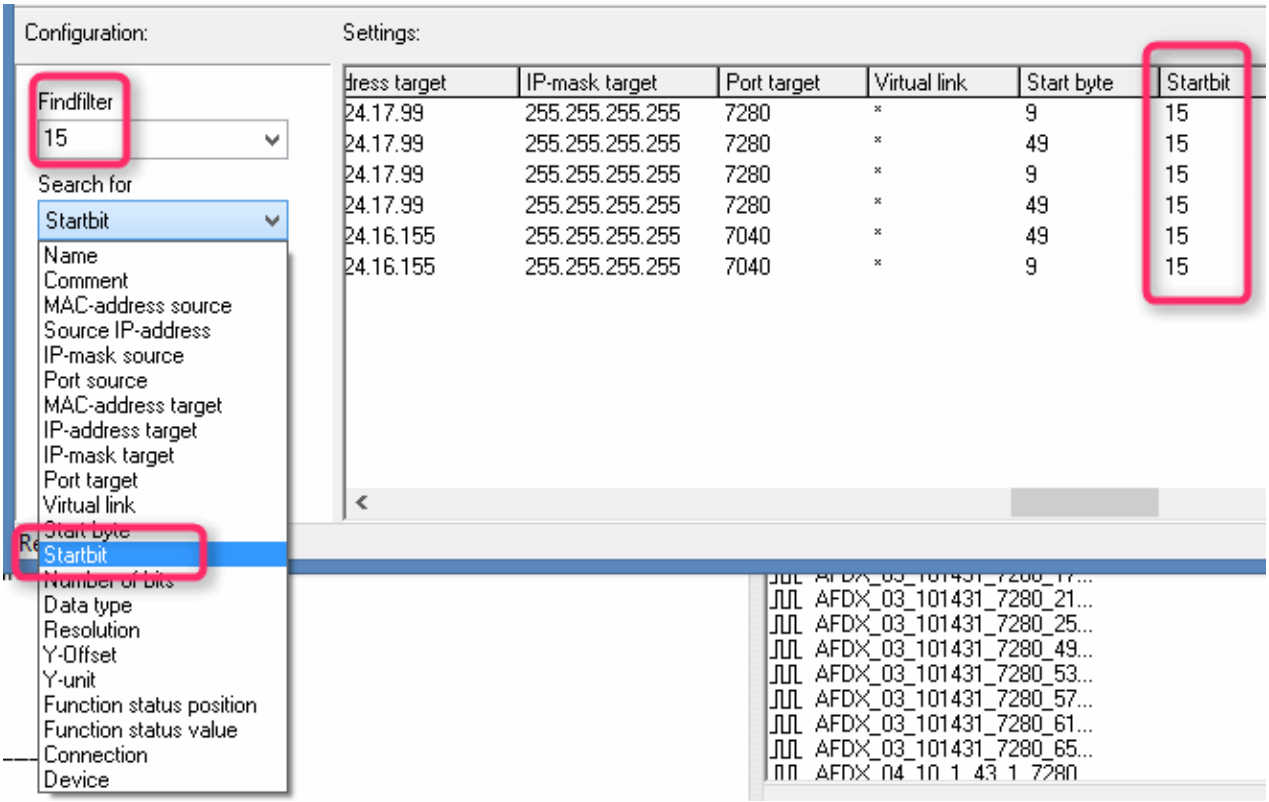
To start the search filter, open the context menu in the Connections list and select Define FindFilter.

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

7.2.4 CAN-Bus Interface

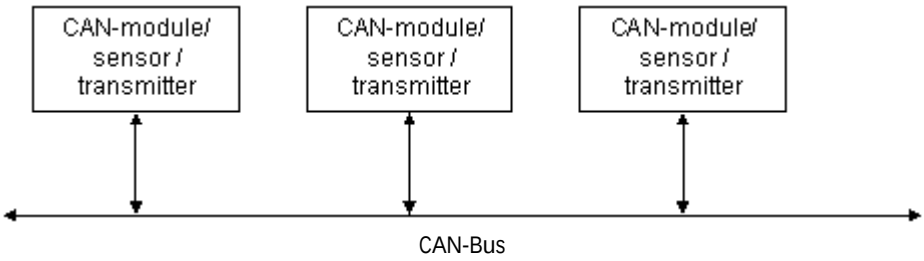
What is the CAN-Bus?

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.



7.2.4.1 Information

7.2.4.1.1 Ordering information

Our standard product package comprises two high-speed CAN-Busses (as per ISO 11898). A low-speed model (ISO 11519-2) is also available. If one or two low-speed busses are desired, please note this in your purchase order, as for instance: "CAN1 = ISO 11898, CAN2 = ISO 11519-2".

7.2.4.1.2 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)

7.2.4.1.3 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

Observe the general Field bus notes regarding the operation methodology 338 of fixed sampling rates 340 versus timestamp 342.

7.2.4.2 Connection to the device

7.2.4.2.1 CAN port pin configuration

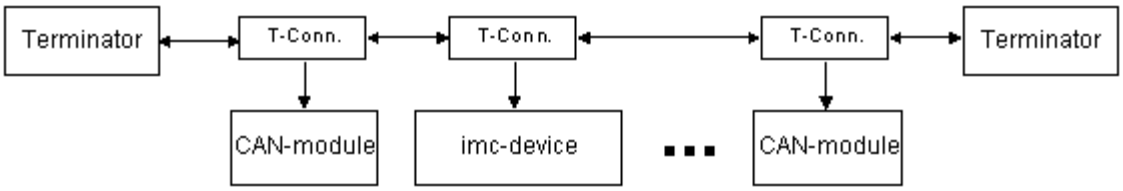
The device comes with a pair of terminals for each CAN-node, configured as follows:

DSUB-PIN	Signal	Description	Usage
1	nc	Reserved	Not used
2	CAN_L	Dominant low bus line	Connected
3	CAN_GND	CAN Ground	Connected
4	nc	Reserved	Not used
5	nc	Optional CAN Shield	Not used
6	CAN_GND	Optional CAN Ground	Connected
7	CAN_H	Dominant high bus line	Connected
8	nc	Reserved (error line)	Not used
9	nc	Optional supply 7V..13V	Not connected

7.2.4.2.2 CAN cabling (Y-adapter)

Device connection with a Y-adapter

If the module has only one connection terminal, it's a 9-pin D-SUB socket. These modules must be connected using an external Y-adapter.



Device connected with Y-adapter

To guarantee a CAN data transfer rate of 1Mbit/s, the **maximum length** for the pathway through a Y-adapter is **30cm**. Usually the wiring within the device is already 30 cm. If using an external Y-adapter it has to be attached directly at the connector.

In this case it doesn't matter whether the other sensors are connected with or without a Y-adapter. The diagram shows just one of several connection possibilities.

7.2.4.2.3 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.

With the CAN-businterface it is possible to terminate the bus within the device controlled by software, see Terminator in device 399.

7.2.4.2.4 Connecting and activating

At first, the CAN-Bus must be completely wired and connected to the measurement device. Only then should the device be activated. During operation, the CAN connector should not be disconnected.

Safety note!

In general, it is OK to pull the CAN-plug during operation. When it is re-plugged afterwards, a reset of the device's CAN-controllers will be carried out. Following that, the device it will resume to collect CAN-Bus data. However, this kind of operation is not specified. Although it normally works, it cannot be guaranteed. For example, re-connecting can lead to an electrostatic discharge where voltages higher than permitted can occur and **cause malfunction of the circuitry**. A person can receive a charge of several thousand volts

Caution!

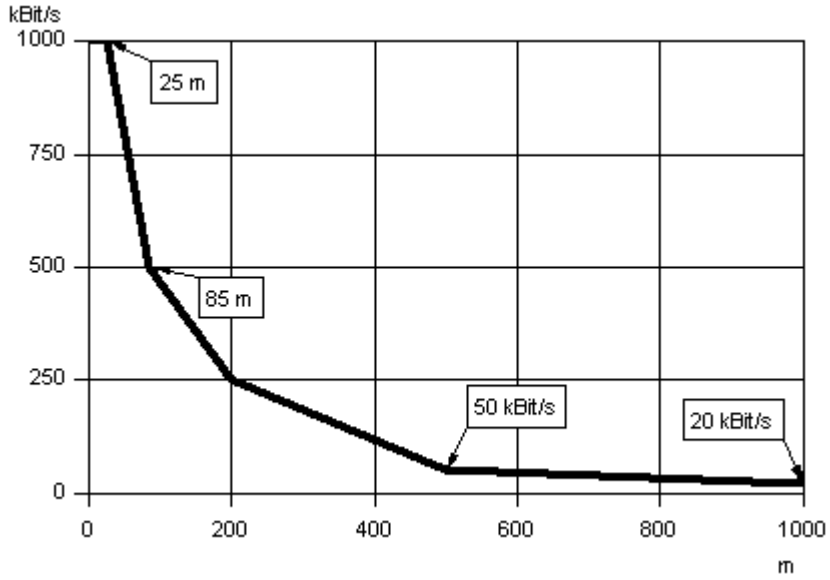
The device can suffer damage as a result of electrostatic discharge! Usually it is necessary to switch the device off and then on again to ensure proper functioning.

7.2.4.3 Technical specification

7.2.4.3.1 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. Furthermore, the possible data rate depends on the bus timing, which in turn is determined by the hardware configuration and software version.



Data rate for devices with CAN2 interface

The diagram shows the transfer rate for devices with CAN2 interface. Transfer rates for imc-USB, imc CANSAS modules and older CAN interface cards are presented with the technical specs in the device documentation.

Line length [m]	Data rate [kBit/s]	Note:
25	1000	Devices from imc are built with electrical isolation from the bus. Devices without electrical isolation achieve somewhat higher values for the maximum line length: 40 m at 1000 kBit/s or 100 m at 500 kBit/s.
85	500	
200	250	
500	50	
1000	20	

Net data rate (CAN-standard 1Mbit/s)

Conversely, the net data rate depends on the package size and on whether the Standard Frame or the Extended Frame is used. For instance, if each CANSAS message has been optimally packed in 8byte chunks, the Standard Frame is used and there are no further subscribers connected to the node, the rate reaches 576.6 kBit/s, amounting to 36 kSamples/s. The table shows the net data rates for 1 MBit/s.

Data length	Net data rate with	
	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

7.2.4.3.2 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 410 is used.

Remarks on CAN FD

- imc Online FAMOS is set up for CAN FD. It is possible to send and receive larger messages with Online FAMOS 410.
- Fundamentally, all CAN-experiments can be transferred to devices with CAN FD.
- Messages can be sent at a higher baud rate using the option " 410 *Switch Bit Rate* 410 " 410.
- 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.

7.2.4.3.3 Number of CAN nodes

The CAN-Bus permits you to use a large number of nodes, depending on the line length, the number of nodes and the cable cross section:

Line Length	Number of Nodes/ Cable Cross Section		
	32	64	100
100m	0.25mm²	0.25mm²	0.25mm²
250m	0.34mm²	0.5mm²	0.5mm²
500m	0.75mm²	0.75mm²	1.0mm²

7.2.4.3.4 Specifications imc STUDIO

- Baud rate up to 1Mbit/s; 8Mbit/s with CAN FD.
- Multiple independent CAN nodes (ports)
- A maximum of 512 channels including analog, virtual and CAN channels
- 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. 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!

7.2.4.3.5 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:	500kbit/s
ID:	455
Length:	5 bytes
Repeat rate:	every 10ms

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.
- Observe the general Field bus notes regarding the operation methodology 338 of fixed sampling rates 340 versus timestamp 342.

7.2.4.3.6 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?

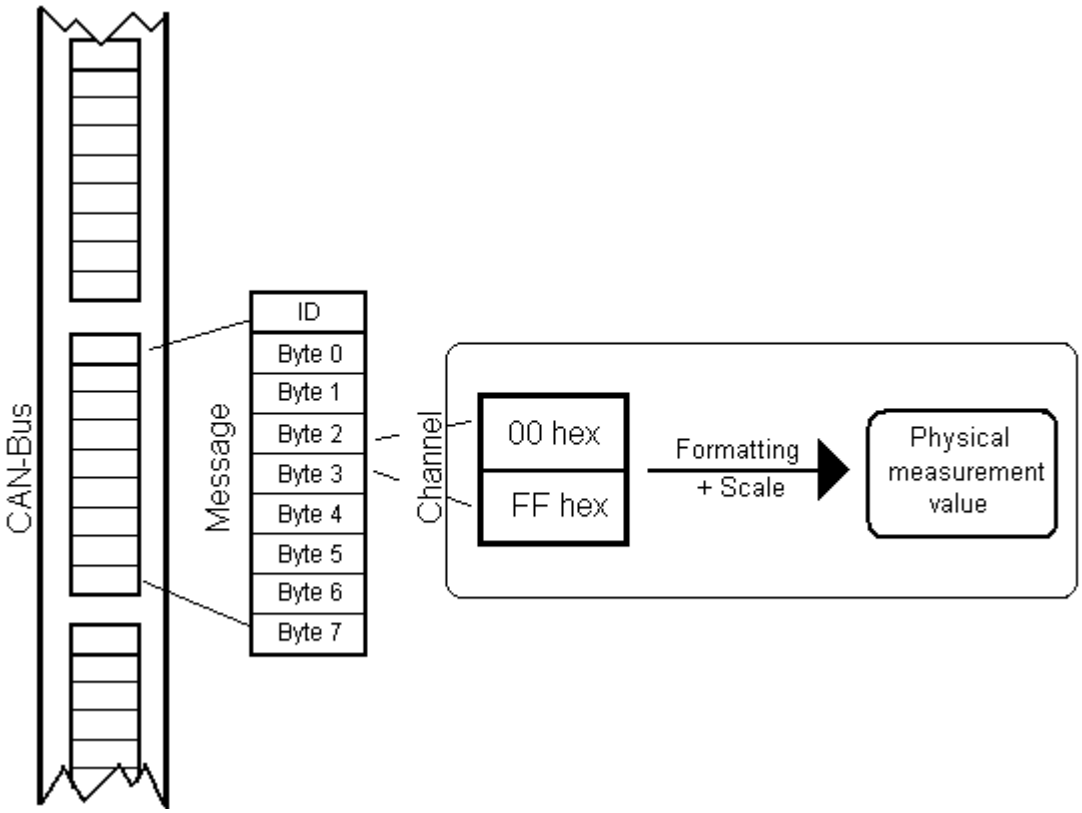
- terminal channels
- recording and analysis of 100-byte error protocols
- control of other devices on the CAN-Bus by the measurement device
- consideration of complex conditions
- compensation for CAN-Bus delay
- multiple protocols
- higher protocols

All major imc STUDIO features available for CAN channels

- Online processing using imc Online FAMOS (incl. Transitional Recording)
- 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.
- Options include: radio clock module
- hand-held display terminal
- modem module
- Binary signals from CAN-Bus possible
- Networking of multiple measurement devices 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

7.2.4.3.7 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

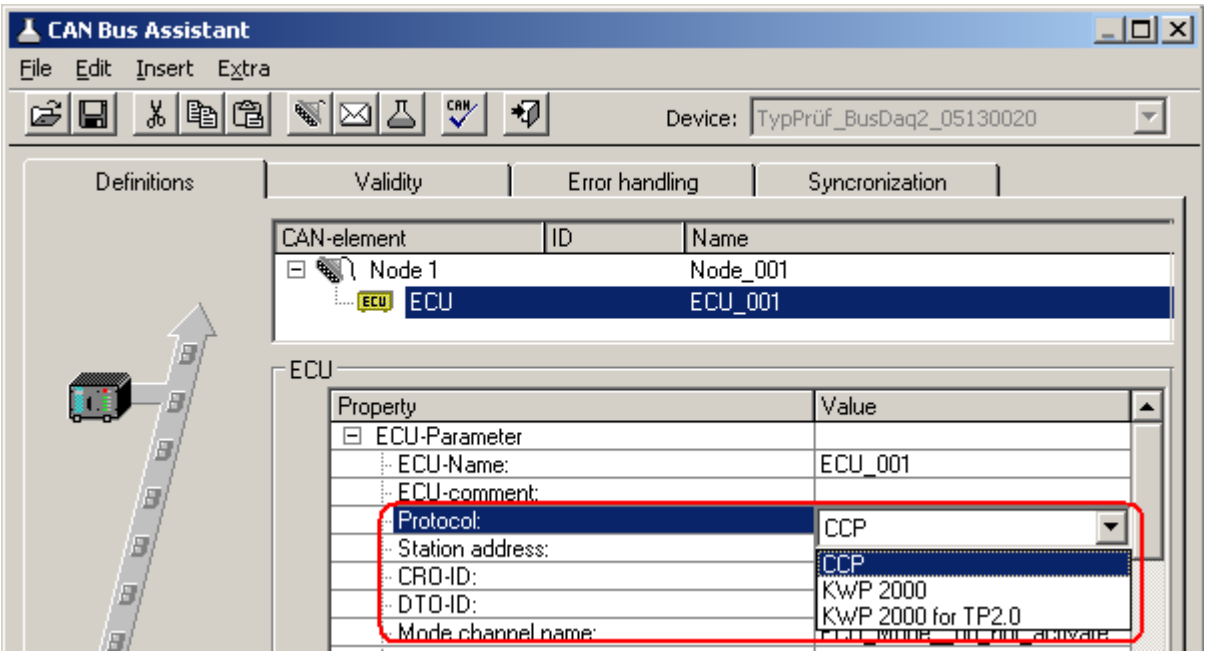
Hardware

- A device with CAN-module(CAN2) is necessary with the common device control functions and all desired protocols being enabled or
- Devices as of group 4 88 and higher

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 437.



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 437.

Basic functions

Keyword Protocol 2000 with ISO Transport Protocol

A subset of keyword protocol 2000 is used with addressing modes Normal and NormalFixed.

10h	StartDiagnosticSession
13h	ReadDiagnosticTroubleCodes
14h	ClearDiagnosticInformation
17h	ReadStatusOfDiagnosticTroubleCodes
18h	ReadDTCsByStatus
20h	StopDiagnosticSession
21h	ReadDataByLocalIdentifier
22h	ReadDataByCommonIdentifier
23h	ReadDataByAddress
27h	SecurityAccess
2Eh	WriteDataByCommonIdentifier
31h	StartRoutineByLocalIdentifier
33h	RequestRoutineResultsByLocalIdentifier
38h	StartRoutineByAddress
3Ah	RequestRoutineResultsByAddress
3Bh	WriteDataByLocalIdentifier
3Dh	WriteDataByAddress (maximum two byte data)
3Eh	TesterPresent

Keyword Protocol 2000 with VW TP 2.0 (ISO 14230-3)

13h	ReadDiagnosticTroubleCodes
14h	ClearDiagnosticInformation
17h	ReadStatusOfDiagnosticTroubleCodes
21h	ReadDataByLocalIdentifier
22h	ReadDataByCommonIdentifier
23h	ReadDataByAddress
27h	SecurityAccess
2Eh	WriteDataByCommonIdentifier
31h	StartRoutineByLocalIdentifier
33h	RequestRoutineResultsByLocalIdentifier
38h	StartRoutineByAddress
3Ah	RequestRoutineResultsByAddress
3Bh	WriteDataByLocalIdentifier
3Dh	WriteDataByAddress (maximum two byte data)

CCP

01h	CONNECT
03h	DNLOAD
02h	SET_MTA
04h	UPLOAD
07h	DISCONNECT
0Fh	SHORT_UP
12h	GET_SEED
13h	UNLOCK
14h	GET_DAQ_SIZE
15h	SET_DAQ_PTR
16h	WRITE_DAQ

XCP

D3h	ALLOC_ODT_ENTRY
D4h	ALLOC_ODT
D5h	ALLOC_DAQ
D6h	FREE_DAQ
E0h	SET_DAQ_LIST_MODE
E1h	WRITE_DAQ
E2h	SET_DAQ_PTR
E3h	CLEAR_DAQ_LIST
F0h	DNLOAD
F4h	SHORT_UPLOAD
F5h	UPLOAD
F6h	SET_MTA
F7h	UNLOCK
F8h	GET_SEED
FEh	DISCONNECT
FFh	CONNECT

Diagnostics on CAN (ISO 15765)

10h	StartDiagnosticSession
12h	ReadFreezeFrameData
14h	ClearDiagnosticInformation
17h	ReadStatusOfDiagnosticTroubleCodes
21h	ReadDataByLocalIdentifier
22h	ReadDataByIdentifier
23h	ReadMemoryByAddress
27h	SecurityAccess
2Ch	DynamicallyDefineLocalIdentifier
31h	StartRoutineByLocalIdentifier
3Bh	WriteDataByLocalIdentifier
3Eh	TesterPresent

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.

7.2.4.4 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.

7.2.4.5 Sensor initialization

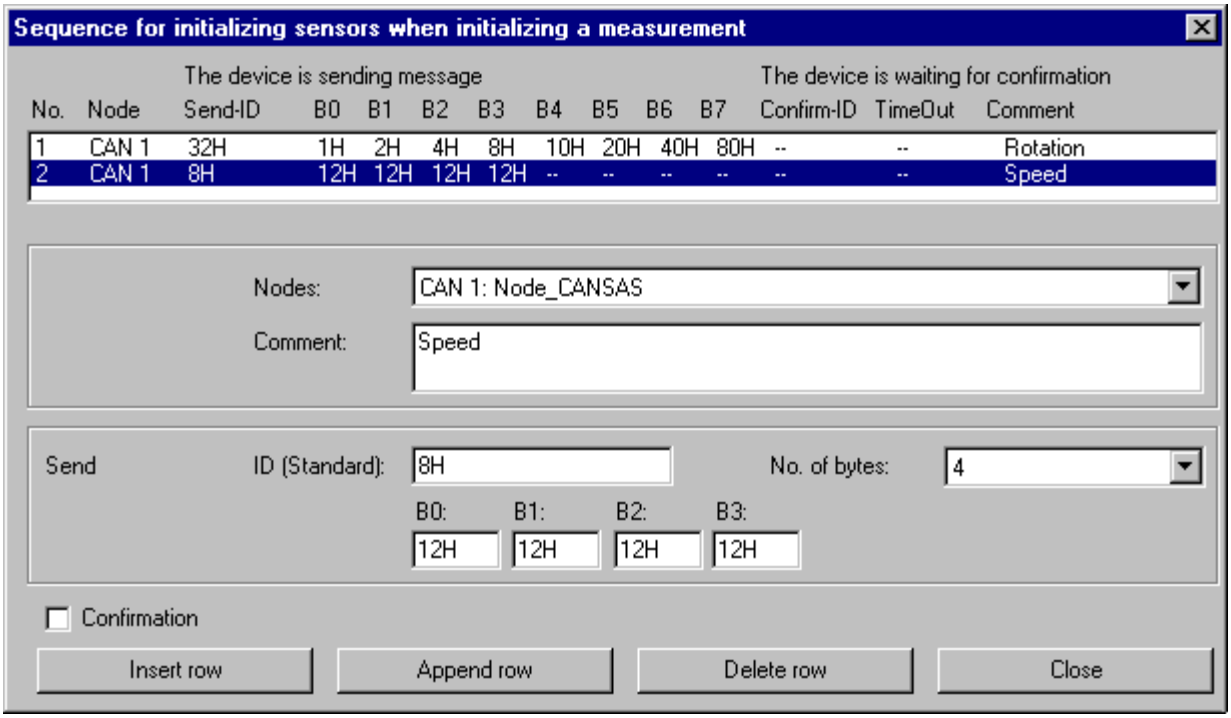
Many CAN sensors need to be initialized after being turned on. The initialization usually consists of one or more messages.

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.

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):

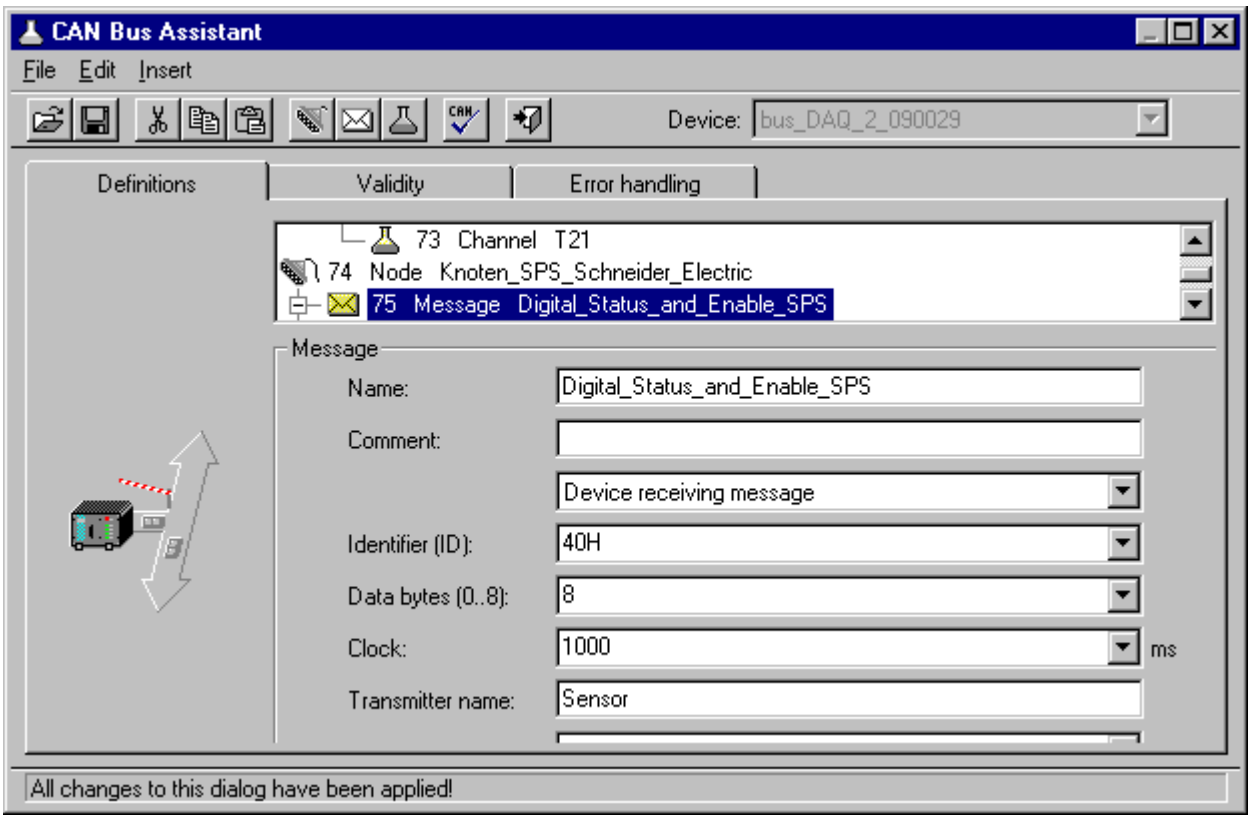
The identifier for the confirmation (hand-shaking) message which the sensor returns to the device to indicate that the initialization sequence has been carried out. The unit waits for this message before continuing with the next initialization sequence in the list.

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.

7.2.4.6 CAN-Bus Assistant

The CAN-Bus Assistant provides you for defining nodes, messages and channels. . Start the Assistant by using the Ribbon *Home* (or *Setup-Configuration*) > *CAN*.



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 423.

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.

7.2.4.6.1 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.

7.2.4.6.2 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

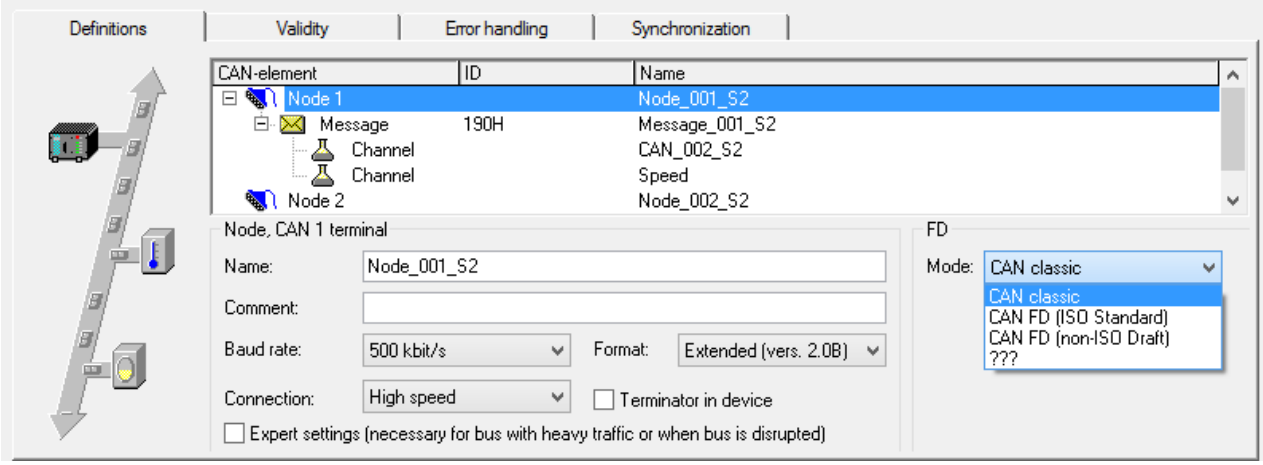
Any changes you make to the properties are applied immediately. Even incorrect settings (e.g. channel names) are applied. Any errors are shown in the status bar at the bottom of the list.

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.

7.2.4.6.3 Nodes

7.2.4.6.3.1 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:

- *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.

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.

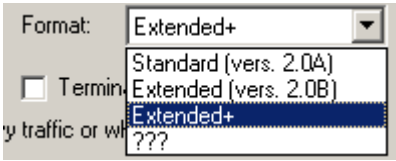
Baud rate:

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.

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.

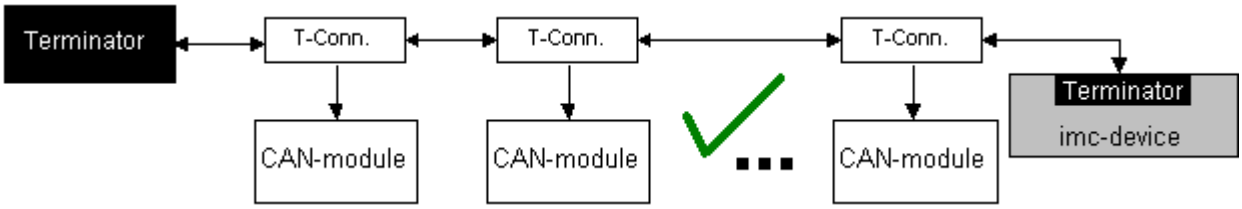
The *Extended+* format is able to receive *Standard-* and *Extended* identifier, but sends in *Extended* format only.

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 in device:

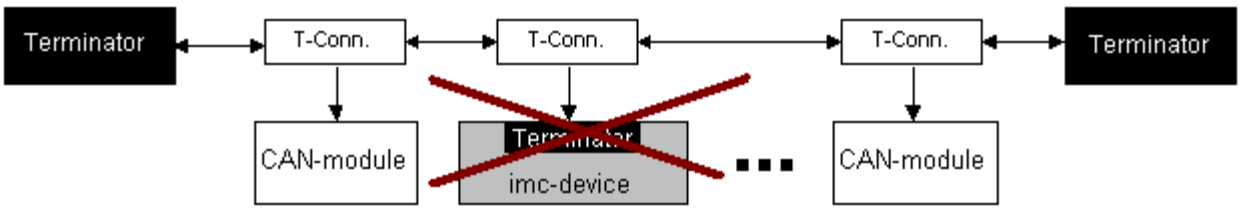
Devices equipped with CAN-bus interface CAN2, can switch a terminator resistor internal via software. If the devices is connected at one end of the CAN line, an external Y-cable with terminator is not needed.



Correct termination

Warning

If the CAN-bus is already terminated correctly, that option cannot be used!



Wrong termination

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.

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

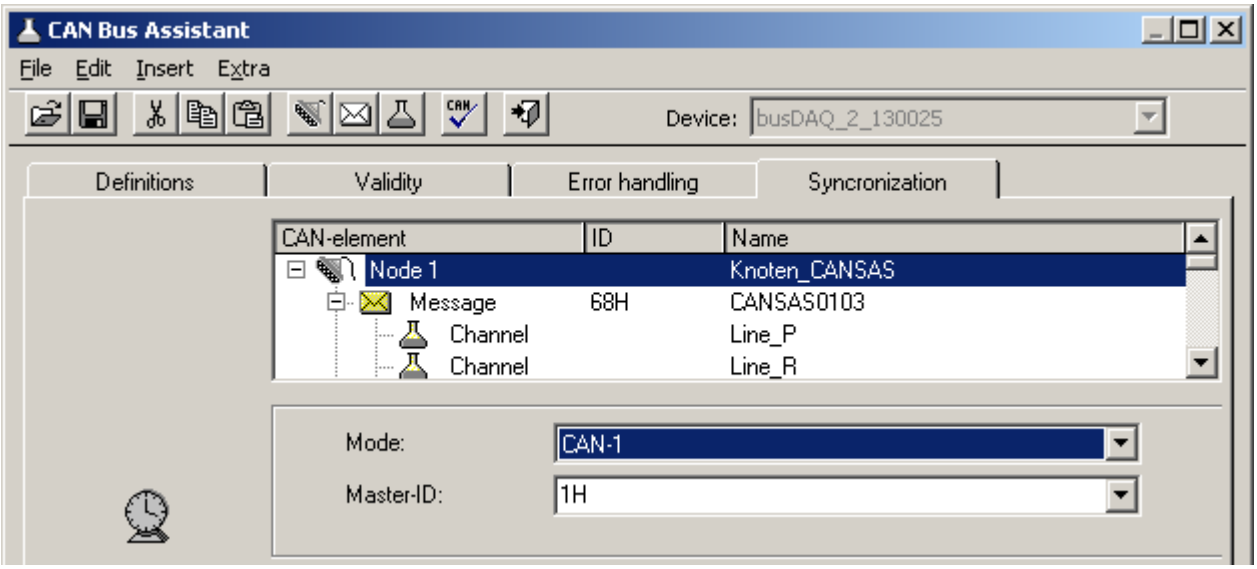
- In most cases, disturbance on the bus can be compensated by shifting the ***position of the sampling time*** backwards.
- Disturbance is usually the result of long cables. The low-pass effect of such long lines prevents voltage levels from changing rapidly. In extreme cases, this can mean that the 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.

7.2.4.6.3.2 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.



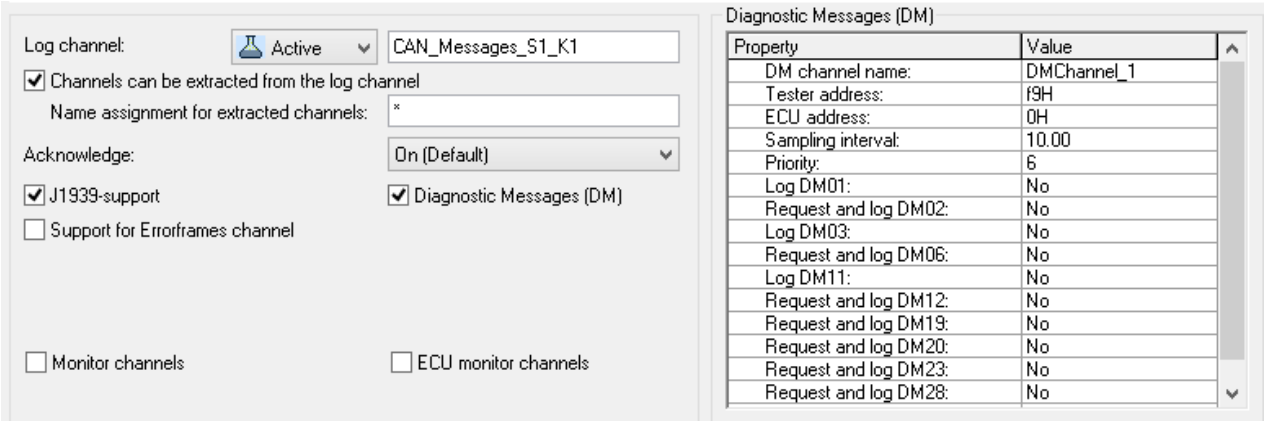
Synchronization of a node

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.

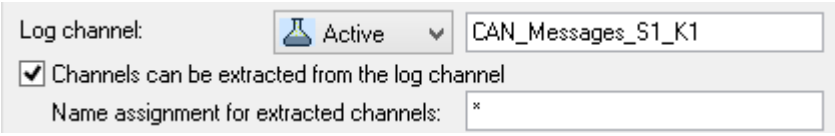
Conditions

- 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 4 88): The imc measurement device may not be operated in Sleep/Resume mode.

7.2.4.6.3.3 Validity



Validity of a node



Parameter	Description
Log channel	With Log channel "*Active*", all messages belonging to the node are logged in imc STUDIO. For this purpose, the text in the input box is used as the channel name. The messages' IDs and data bytes are recorded.
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 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 chapter Storage Options 591.

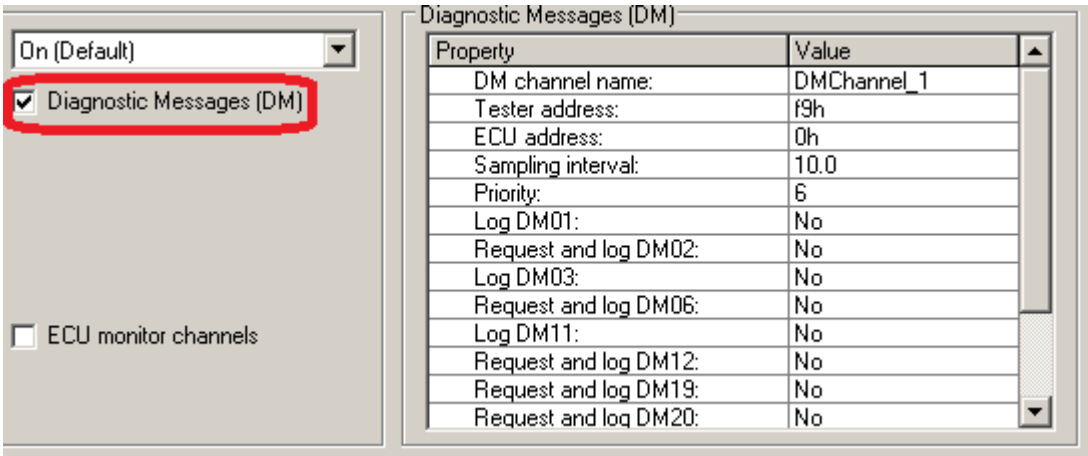
Disables the transmission of Acknowledge messages. The CAN controller usually sends this packages to confirm successful received messages.

Activated the J1939-support, see also Messages: Validity 411

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: 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: 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 1: 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: 2: higher-value byte of PGN

Byte: 3: lower-value byte of PGN

Byte: 4: source

Subsequent 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	

Otpion *IBC node addressing mode* can be used for nodes with format Extended:

If selected, the Channel bit, the Source bit, the Lifesign bits and the Telegram CRC bits are ignored (masked) when receiving and evaluating messages.

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.*

Prerequisites

This function is only available for the new imc BUSDAQ, group 4 88. The devices must be configured via the Remote connection in accordance with the description in the device manual.

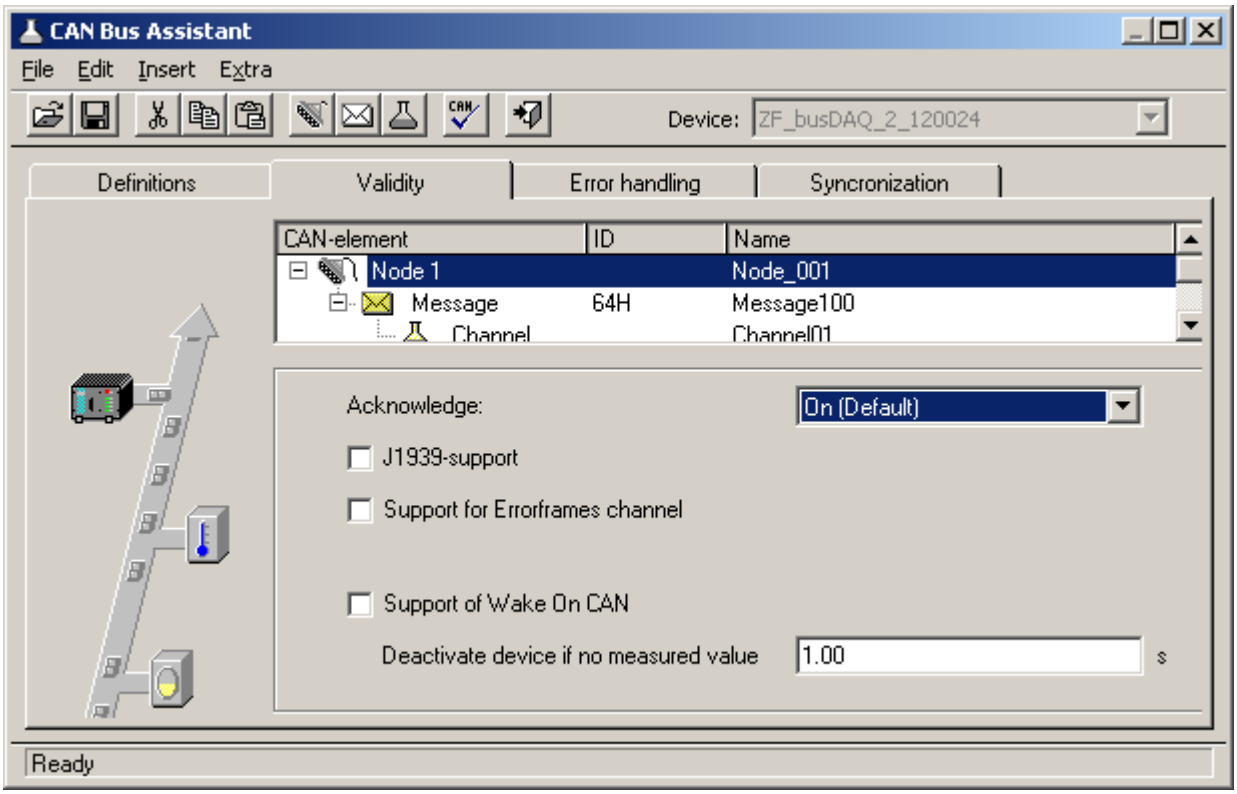
Description

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.

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-Assistant.



Activation of Wake on CAN

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:

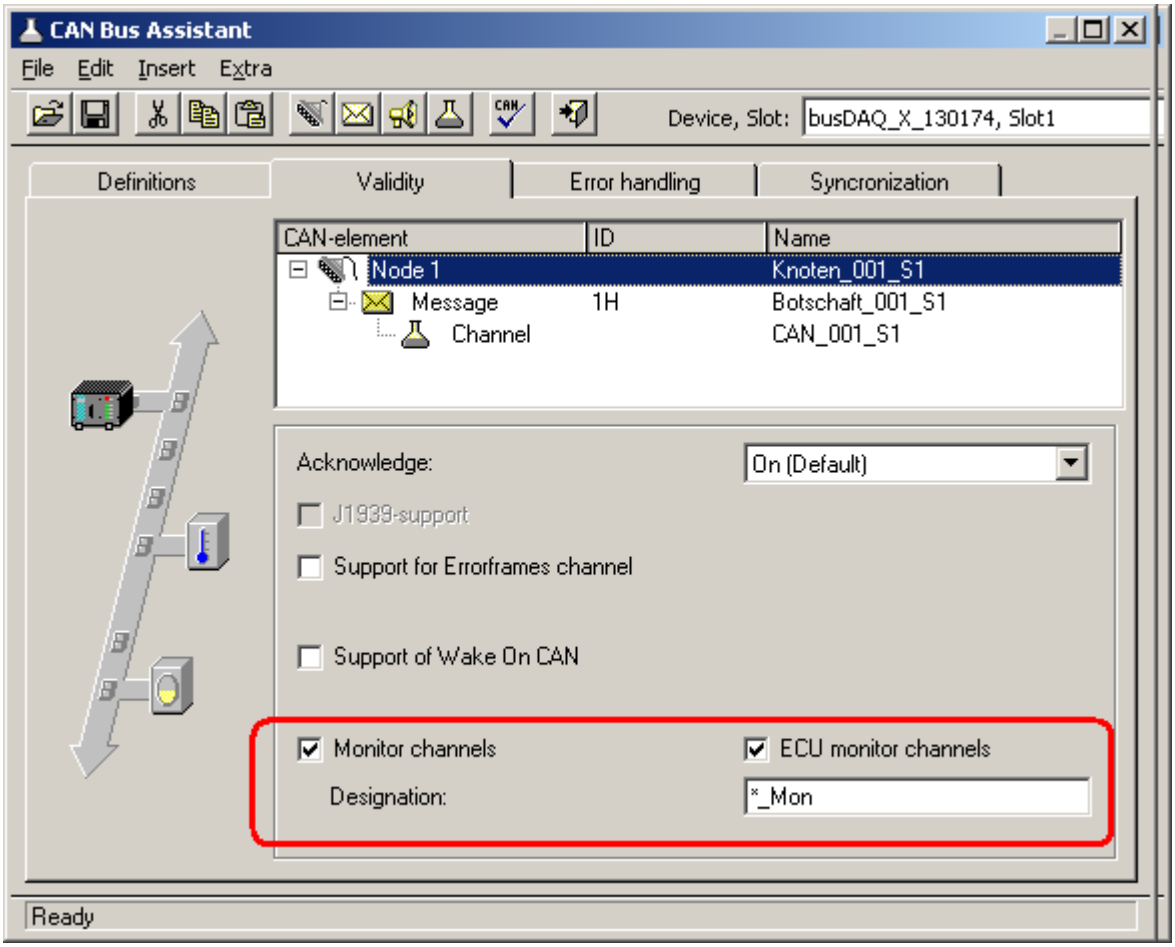
Enter the time interval whose elapse without the arrival of data activate the sleep mode. Intervals between 1ms and 14h (50400s) are possible.

Each node can be adjusted individually.

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. If you only wish to create individual CAN channels as monitor channels, proceed according to the previous method 427.

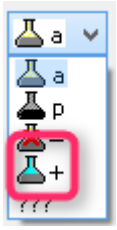
7.2.4.6.3.4 Blob - Decoding with the Bus Decoder

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.

To do this, you make the following settings in the CAN-Assistant:

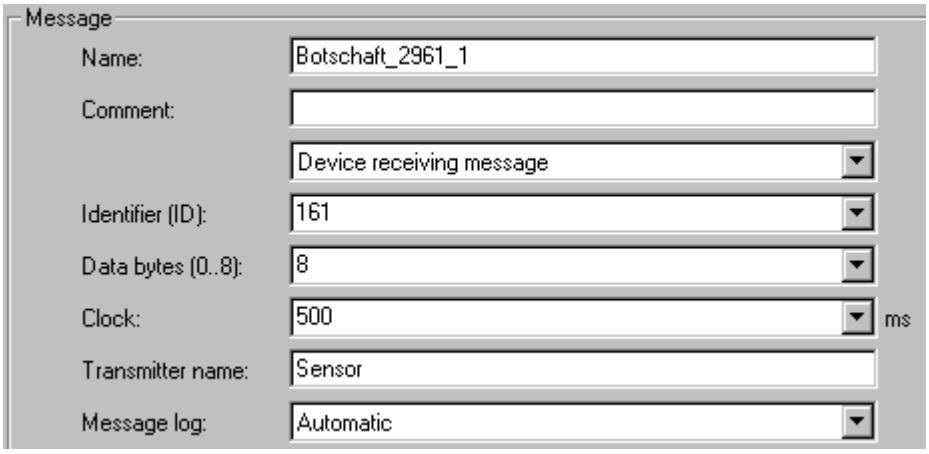
- A log channel which logs all messages 409.
- On the node's page *Validity*, the option "*Channels can be extracted from the log channel*" must be activated.



- The CAN-channels to be extracted in the Bus Decoder from the log channel must be set to "*extract in imc STUDIO with the Bus Decoder*" on the *Validity* page (light blue highlighting). To select all channels, use the menu item *Edit\Select channels\Select all receiving channels*.
- Channels which can not be extracted in the device can be found using the menu item *Edit\Select channels\Select channels (not in device extractable)*. In this way, it is possible to select and delete these channels. Alternatively these channel can already be ignored while importing 425.
- 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).

7.2.4.6.4 Message

7.2.4.6.4.1 Definition



Receiving a message

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.

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:

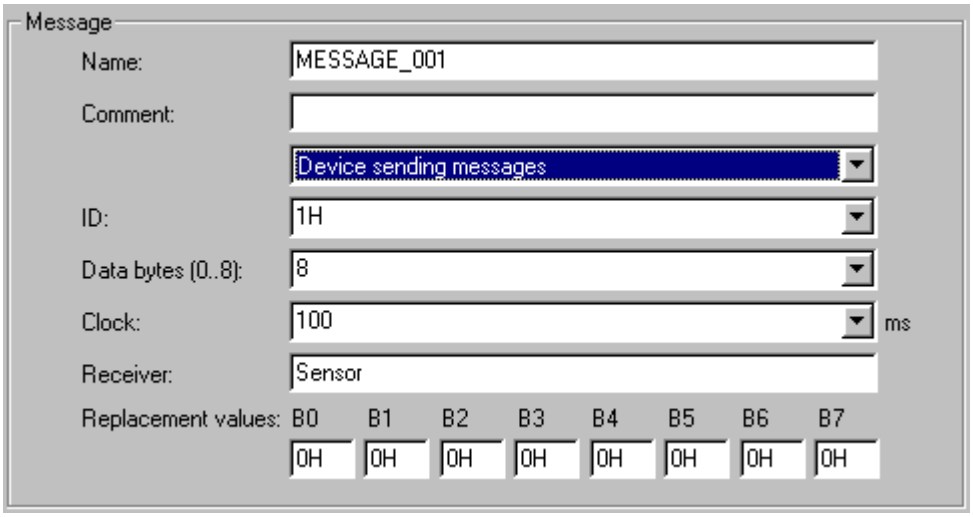
The transmitter name describes the transmitting sensor. This name is not used for any other purposes - therefore it need not be unique.

Message log:

If you wish to log all of a node's channels, activate the log channel on the page with the tab "Validity 402".

If you wish to log individual messages, select "*Log CAN Bus message"* under *Message log.* In imc STUDIO,a global channel is created automatically in which a node's messages are logged. Select the desired message by its respective identifier.

If you set the message-ID to *All messages*, you obtain a log channel like on the page with the tab "Validity 402".



Sending a message

The CAN-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 (with CAN2 only), see below.

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 399. For this purpose, the option *Bit Rate Switch* must be activated, which appears at the CAN FD node.

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.

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 399 is set to *Extended* or *Extended+* format, a message can be send as extended or as standard-format.

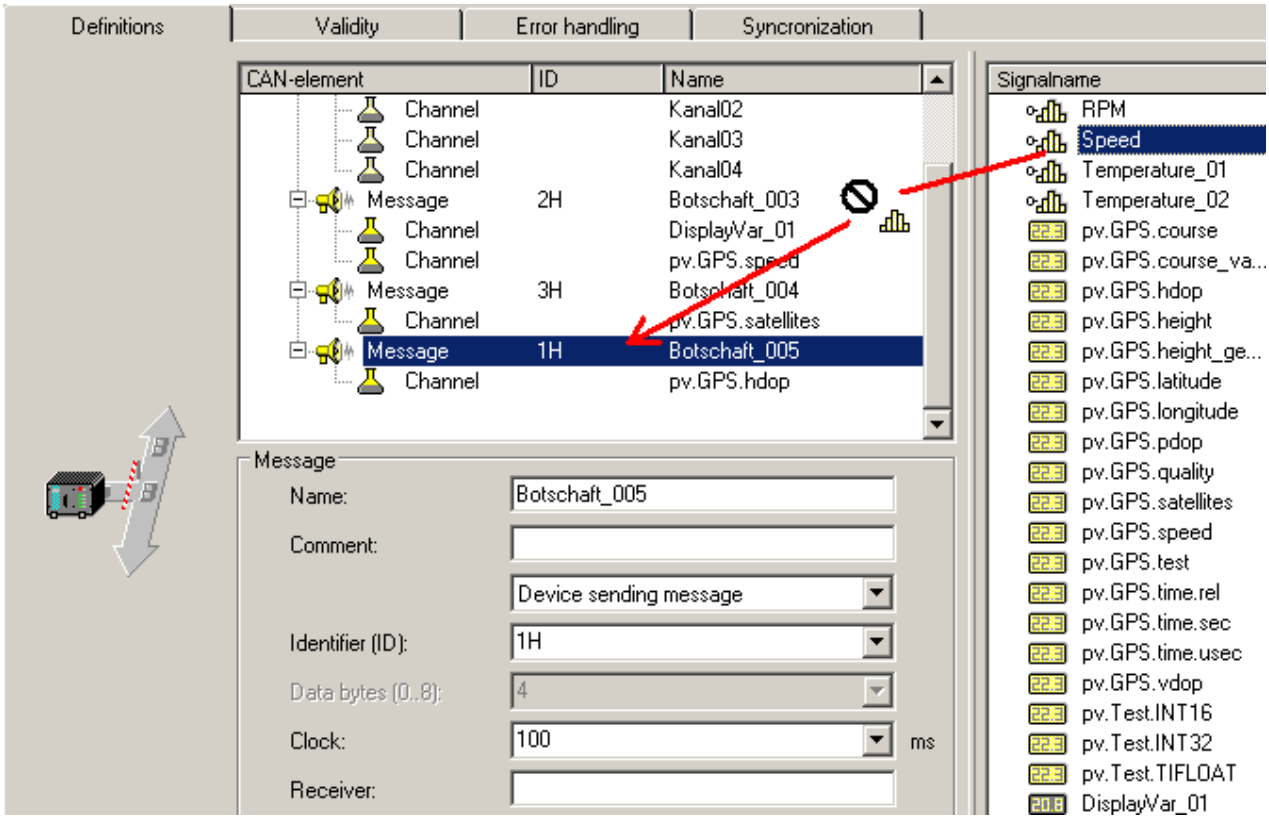
The CAN-Assistant is used to create a formal framework of messages and channels. Using imc Online FAMOS, this framework is filled with data and transmitted. The channels or single values are transmitted in the format set in the CAN-Assistant. As for the reception of CAN-messages, channels are defined 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-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.

For examples to send messages, see Sending CAN messages 428.

Channels, process vector variables and display variables can be assigned out of the CAN-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.

To use this feature the interface CAN2 is required.



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.

7.2.4.6.4.2 Validity

For work with J1939, the setting ***J1939-support*** must be activated for the control *CAN-node* on the *Validity* page. In consequence, the settings options for *Messages* become visible on the *Validity* page:

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:

$$ID_Extended = Source\text{-}Address + ProgramGroupNumber * 28 + Priority * 2^{26}$$

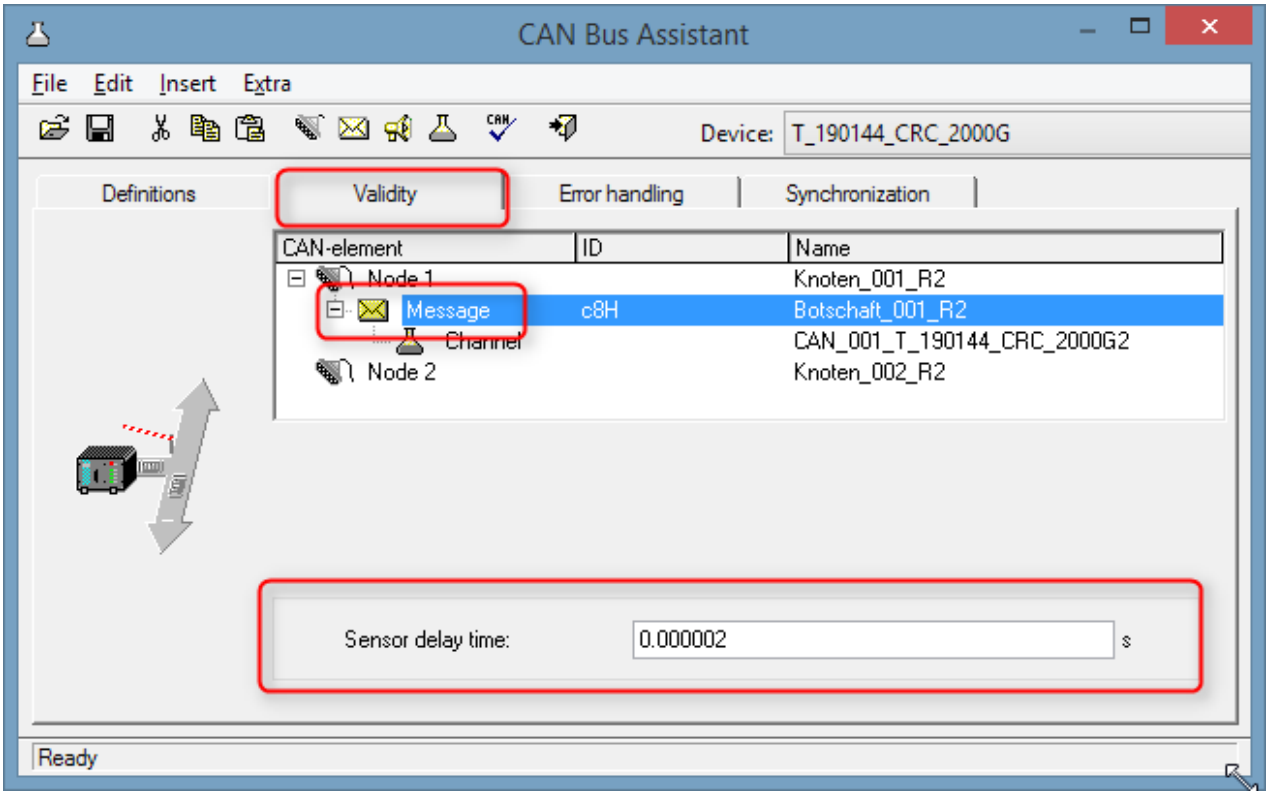
For standard IDs, the following applies:

$$ID_Standard = PGN\text{-}number + Priority * 2^{8}$$

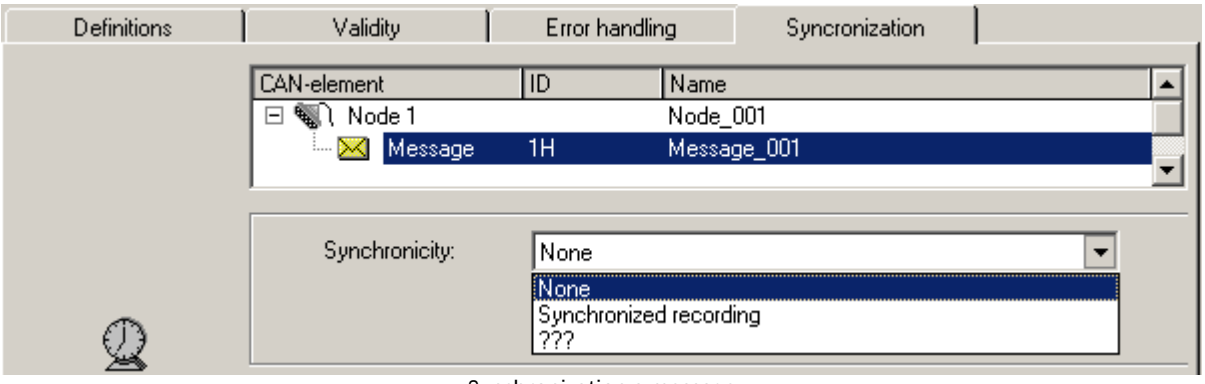
When importing from a DBC-file, J1939 messages could be greater then 8 Byte. Those messages can be received into several messages from imcDevices Version 2.7R2.

If the import does not return any explicit specification of priority and source address, they are extracted from the DBC file's message-ID. Together with the appropriate bits in the DBC file's message-ID, the message-ID is generated in the CAN-Assistant.

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.



7.2.4.6.4.3 Synchronization



Synchronization a message

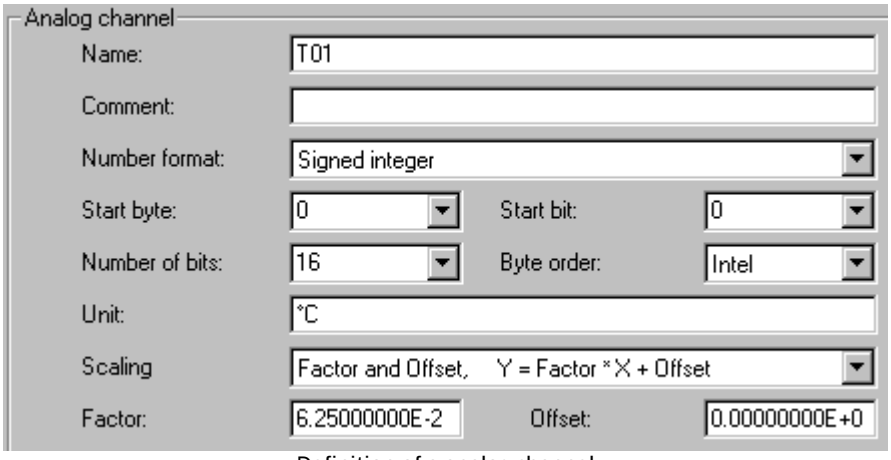
Synchronization is activated at the node. At this node, other devices which do not understand the CAN-1 protocol can be used alongside imc CANSAS. It should be possible to record these devices' messages, even without monitoring of the synchronization. If synchronized recording is selected for such messages, 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-Assistant to "Synchronized recording".

7.2.4.6.5 Channel

7.2.4.6.5.1 Definition

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

Name, Comment:

See Message 408

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

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. The imc hotline 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.

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:

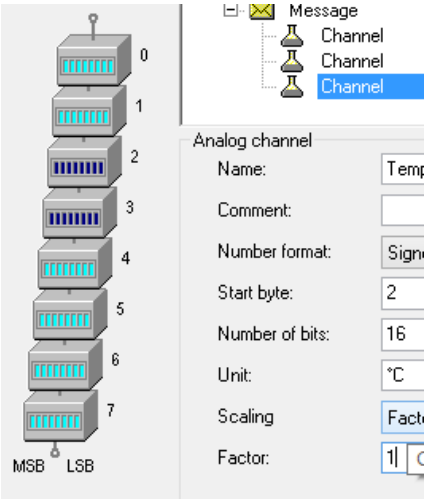
The start bit specifies where the number starts within the start byte.

Bits:

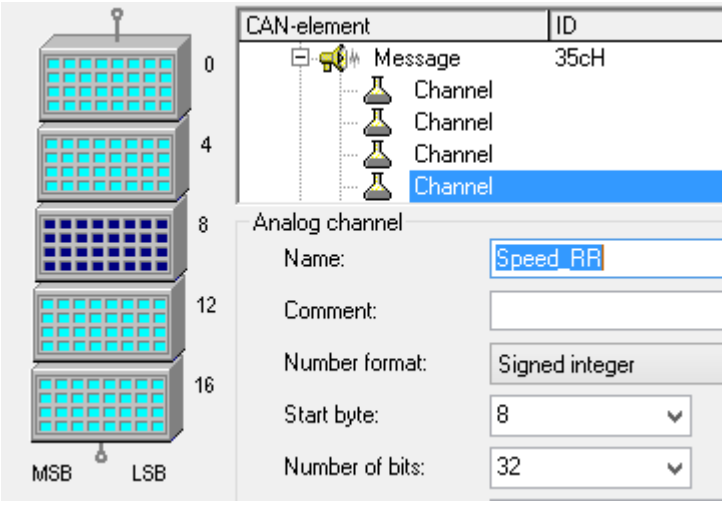
This is the number of bits used in the channel. The remaining bits continue in the direction of the increasing byte indices. For signed and unsigned integers, 1 to 32 bits are allowed. Real numbers can range from 32 (floating point) to 64 (double).

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

Byte Order:

This is the bit order used by the CAN module, either *Intel format* or *Motorola format*.

- Intel format

  The CAN Assistant interprets Intel numbers as follows: The start bit is the LSB and the following bits proceed in the direction of *increasing* bit values. In the message byte graphic, this corresponds to moving to the *left*. 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 CAN Assistant interprets Motorola numbers as follows: The start bit is the MSB and the following bits proceed in the direction of *decreasing* bit values. In the message byte graphic, this corresponds to moving to the *right*. 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

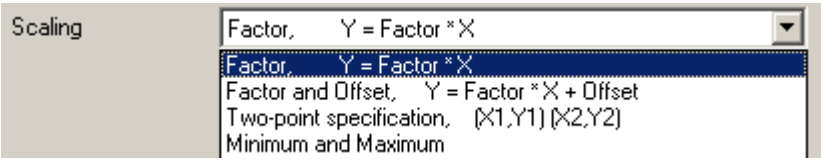
- The CAN tables from manufacturers do not always indicate whether Intel or Motorola byte order is used. It is sometimes possible to determine the byte order indirectly. Within one message typically only one of the variations will be used.
- *Example:* Two byte data packet, message Byte 3 is LOW and 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.

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 four different choices available, to determine the conversion, as shown in this picture:



The Scaling options

Y = physical measurement value X = CAN bus integer value

- 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)

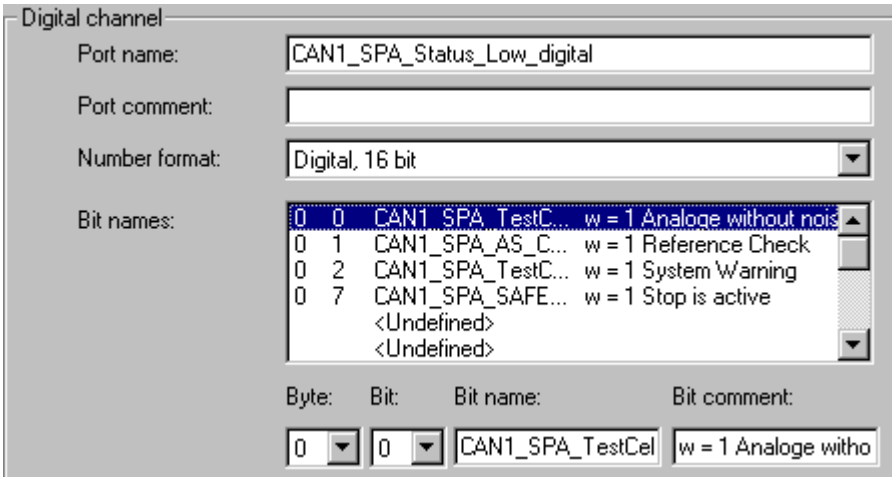
  X1: x-coordinate of 1st point Y1: y-coordinate of 1st point

  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)

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.

Name, Comment: see Message 408

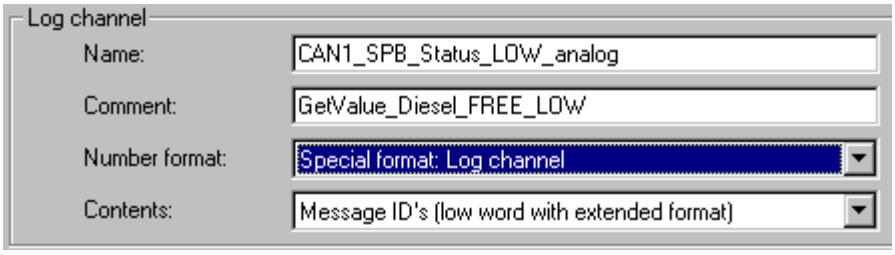
Numerical format: must be set to *16 bit*

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 408

The numerical format is set to Special format: Log-channel (only possible for reception of CAN-Bus-messages). At the same time, the message affected must be set to All messages.



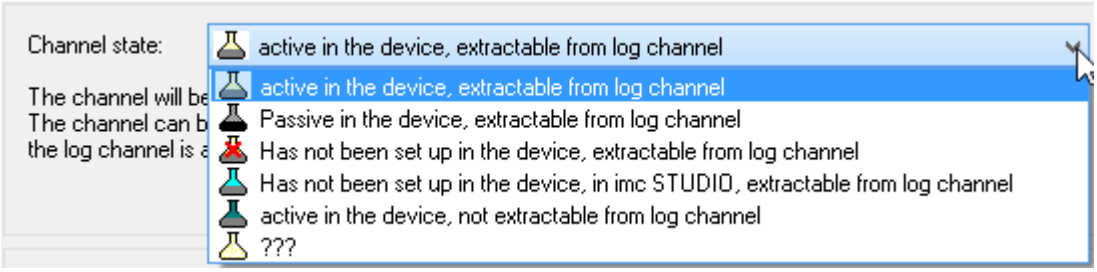
Definition of a log-channel

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.

7.2.4.6.5.2 Validity

The dialog illustrated below appears if the system is set to receive CAN-Bus messages.

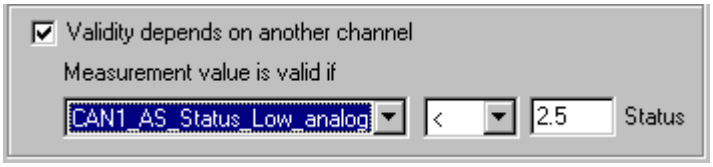


		log channel 409 extractable		
	device setup	with e.g. imc FAMOS	with imc STUDIO	Comment
	active	yes	no	Normal data acquisition on a CAN-channel. Default: active
	passive	yes	no	Normal data acquisition on a CAN-channel. Default: passive
	no	yes	no	Channel extracted offline, with imc FAMOS for instance.
	no	no	yes	Channel extracted with imc STUDIO's bus decoder.
	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 409.

With imc STUDIO, you also have the option to extract channels from the log channel during a running measurement. This is accomplished by the *imc STUDIO Bus decoder*. The channels to be extracted must be registered for the bus decoder with the channel status "*Has not been set up in the device*".



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.

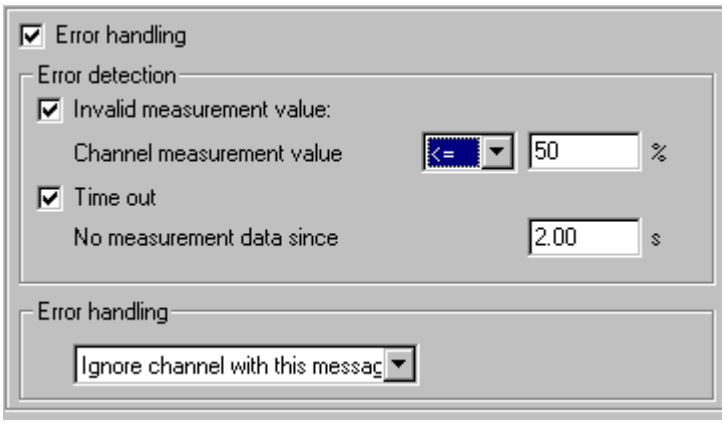
The following two conditions must also be fulfilled by a mode channel:

- 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!

7.2.4.6.5.3 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.

7.2.4.6.6 Menu

7.2.4.6.6.1 File - menu

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.

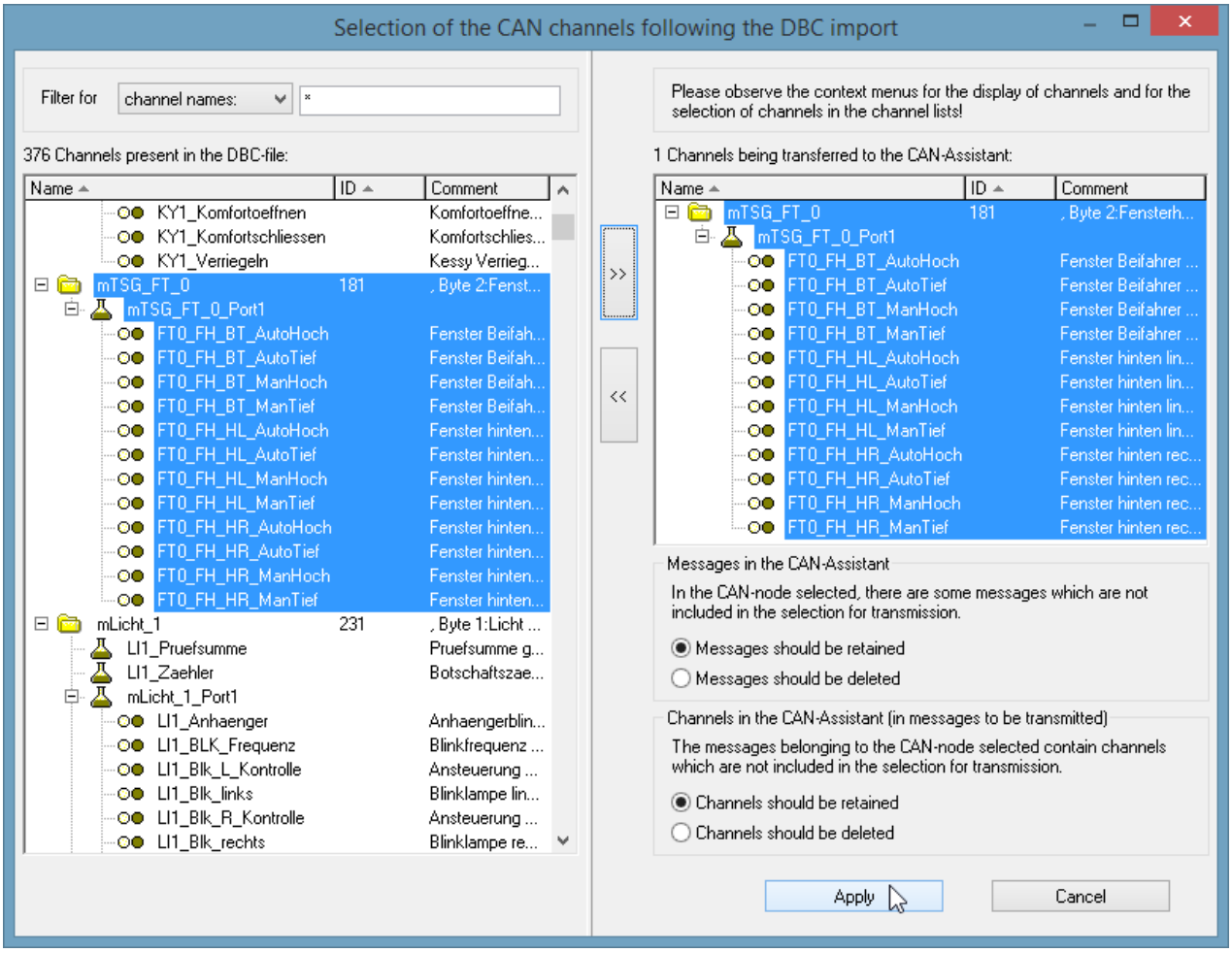
Import...

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).

- 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 [411] 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-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.
  As of imc STUDIO 5.0R8 the extended format XCPplus is also supported.

- 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.

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.

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 417 and assign to the unneeded channels the status "Do not adopt in system" –.

7.2.4.6.6.2 Edit - menu

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:

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):

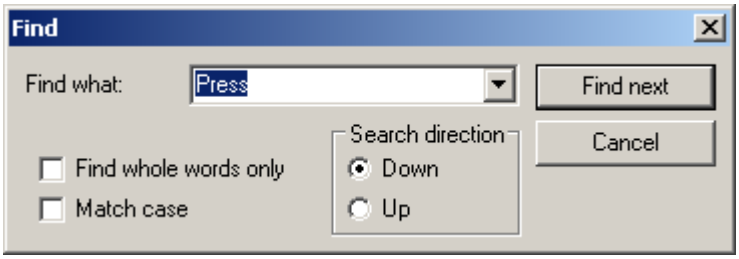
Pastes lines from the Clipboard into the current configuration.

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.



Find dialog CAN-Assistant

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.

Sort: Messages can be ordered by the following criteria:

- messages by name
- messages by IDs
- channels by name
- channels by start bits

Import a Symbol:

This dialog allows you to specify a different icon for the selected nodes, messages and/ or channels. The "Default Symbol" refers to the normal icons used by the CAN-Bus Assistant for nodes, messages and channels. To leave the dialog and use the selected icon, click on OK.

Sensor Initialization...:

See detailed description 393.

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.

7.2.4.6.6.3 Insert - menu

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 408.

Sendable message:

Inserts a new message to send channels. See also Sending messages 409.

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.

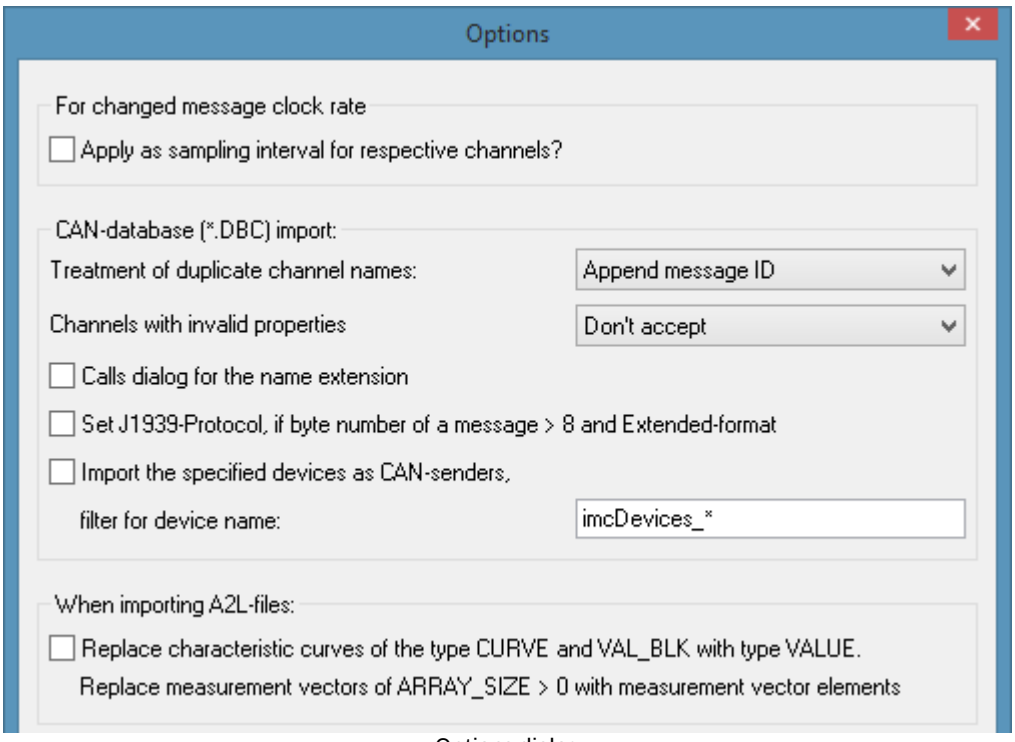
By default, the channel created is always an *analog* channel.

ECU:

Shows the control unit properties 437 and the values to acquire. That entry is only enabled for devices prepared for the support of ECU.

7.2.4.6.6.4 Extras - menu

Options



Options dialog

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 Field bus channels are made in the CAN-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-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* are not allowed in imc STUDIO. In order to load a CAN-configuration with a duplicate name anyway, there are these possibilities:
  - *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-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)* 407", and then selected and deleted.

- *Channels with invalid properties:*
  - 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-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-Assistant and are thus no longer displayed.
- *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.
- *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.
- *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 456.
  Replace measurement vectors of ARRAY_SIZE >0 with measurement vector elements.

7.2.4.6.7 Configuration / processing of field bus channels

Analog inputs

The channels created using the CAN-Bus-Assistant appear in the channel list under Analog channels > Field bus: 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 > Field bus: Digital inputs / outputs.

7.2.4.6.7.1 Settings

If field-bus channels are selected in the channel list (assuming the device currently allows this), then additional setting options are available.

X-Axis:

- Sampling interval: Specifications concerning sampling are entered, as customary, in terms of the *Sampling values* and *Sampling rate/Sampling time*.
- Time stamp: Data capture is handled by external, asynchronous sampling within the field bus. The respective points in time are transferred for subsequent processing via time stamp along with each data point. *Sampling values* and *Sampling rate/Sampling time* do not apply in this case.

Duration:

If *X-axis* is set to *Time stamp*, the size of the measurement is entered in terms of the time duration, rather than in terms of *Sampling values* and *Sampling rate/Sampling time*.

- *undefined*: The measurement continues until an explicit instruction to stop is issued.
- *Duration*: fixed duration per channel.

Assignment:

- *None:* Field-bus channel data are not assigned to any target.
- Selection of a *Display-variable*: Field-bus channel data are assigned as results to the selected target.

7.2.4.6.7.2 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.

7.2.4.6.7.3 Create monitor channels

With imc STUDIO it is possible to create monitor channels for all CAN- or ECU-channel, see here for details 407.

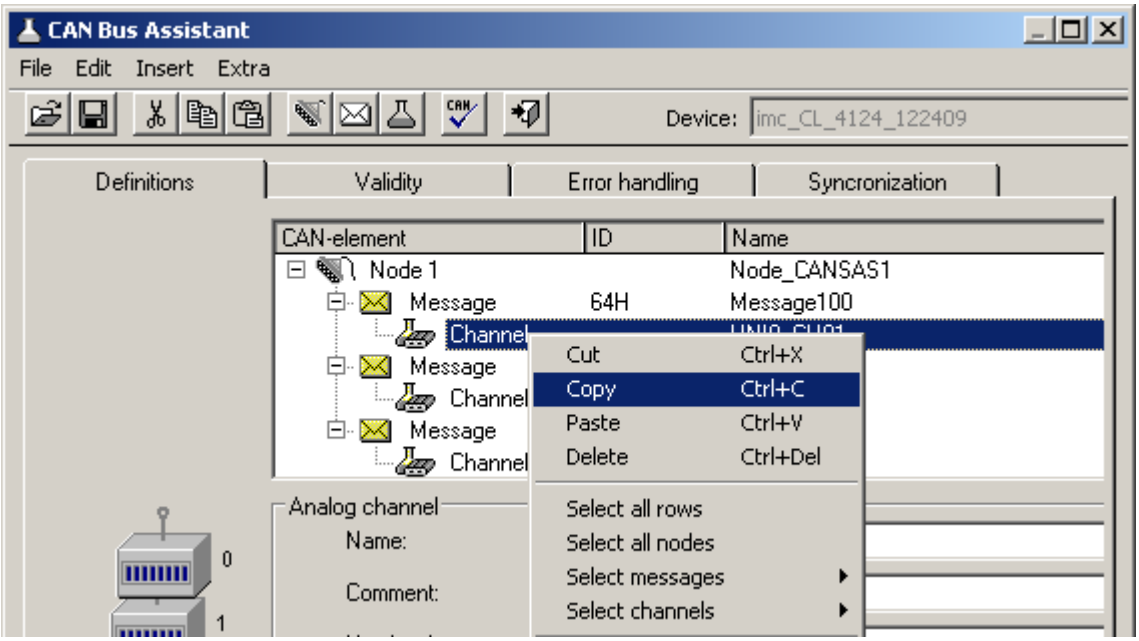
Create monitor channels for single CAN channels:

Triggered Field-bus 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. These monitor channels aren't available with the Field bus.

But there is a an indirect way to get the same channel twice:

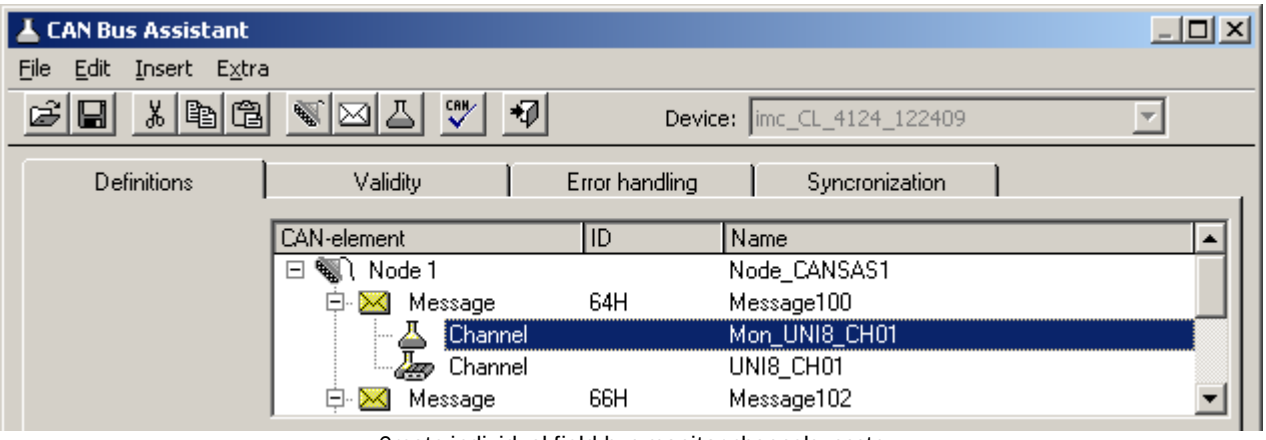
Use the Assistant to create within a channel's message an additional channel having a different name and the exact same settings. This channel can then be captured without triggering.

Copy:



Create individual field-bus monitor channels: copy

and Paste:



Create individual field-bus monitor channels: paste

Note

Monitor channels, created by modules that are synchronized by CAN-1, may not be sampled with a lower sampling rate.

7.2.4.7 Application examples

7.2.4.7.1 Sending CAN messages

For this example, an imc measurement device with a CAN-interface is necessary.

7.2.4.7.1.1 Balancing a bridge module via CAN

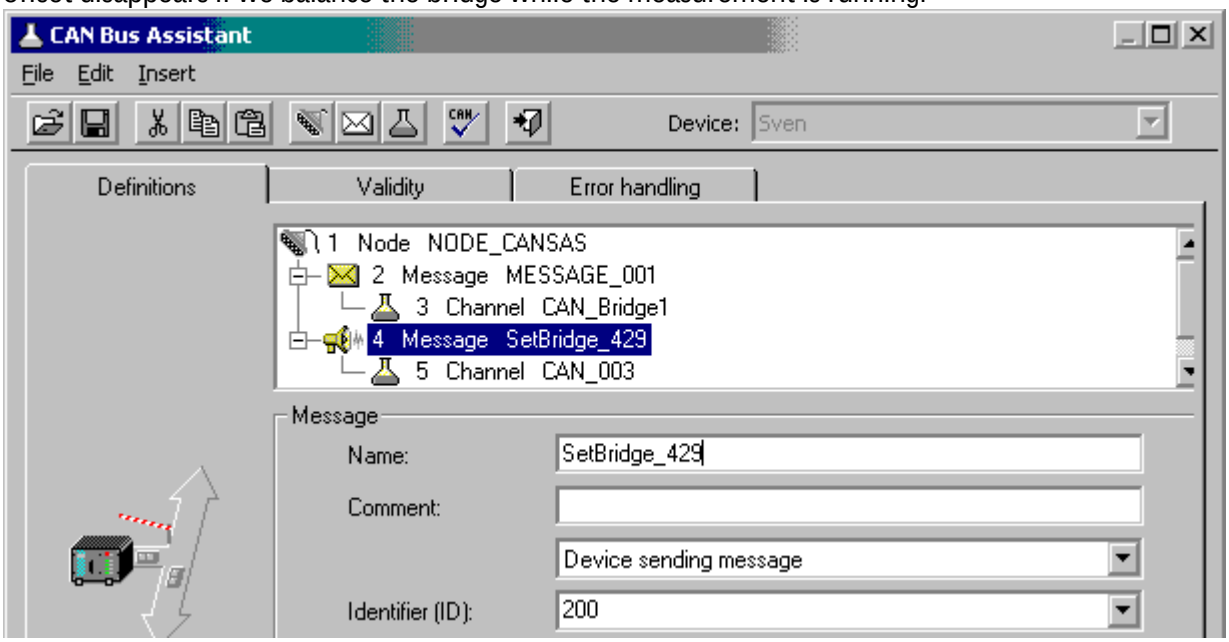
A CANSAS bridge module is used to take a strain gauge measurement. The bridge is to be remotely balanced in its rest state.

The module Bridge2 is 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:

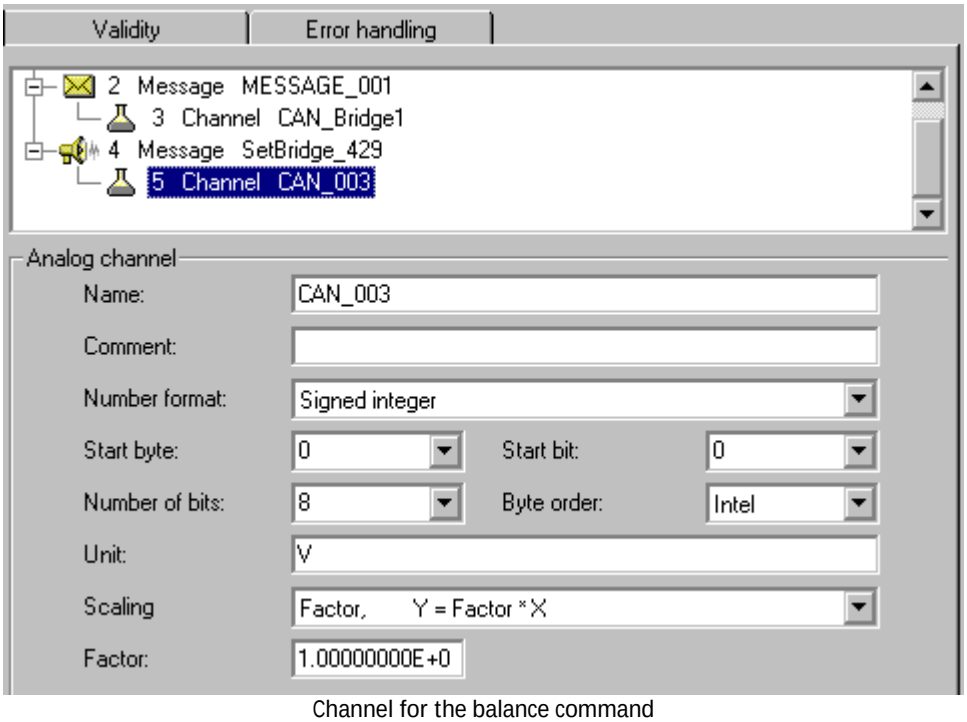
As message "SetBridge_429" is created. The name is arbitrary, the suffix _429 is an aid in locating the module by the serial number. The message must be given the ID which was previously set in Bridge2 for reception purposes, the default is 200.

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.



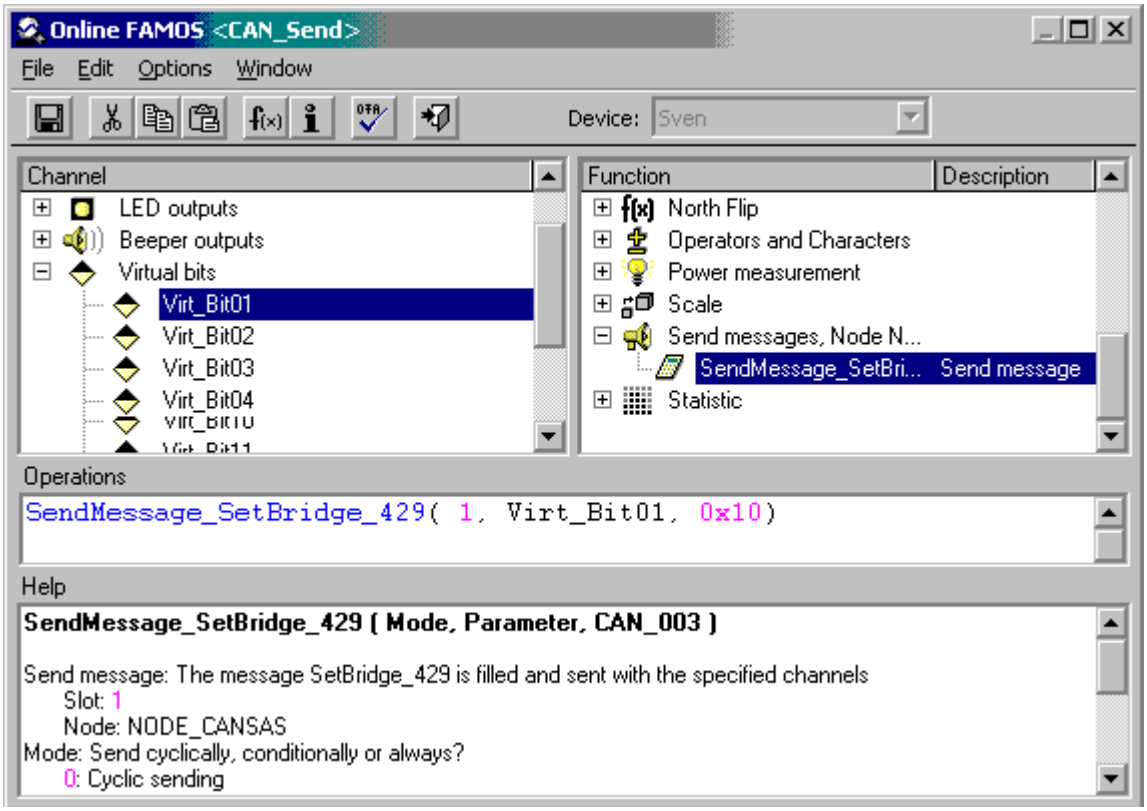
Channel for the balance command

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-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.

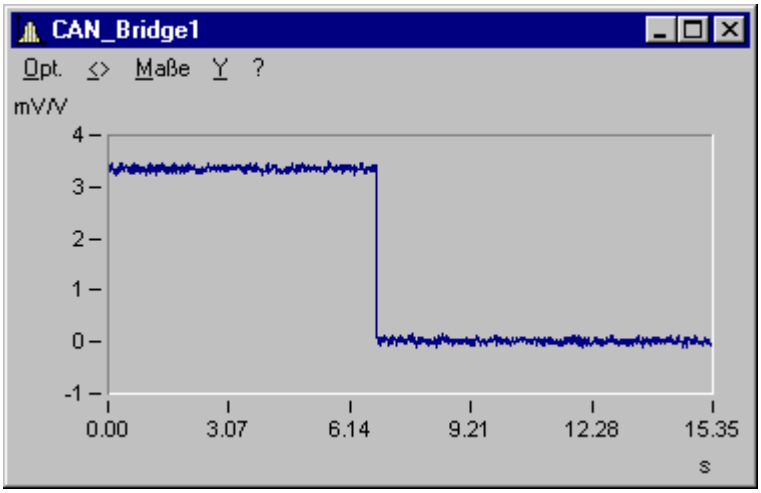
We select the operating type *With condition* and select our virtual bit *Virt_Bit01* as the condition. The number we send is 10 Hex, which causes Bridge2 to be balanced.



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

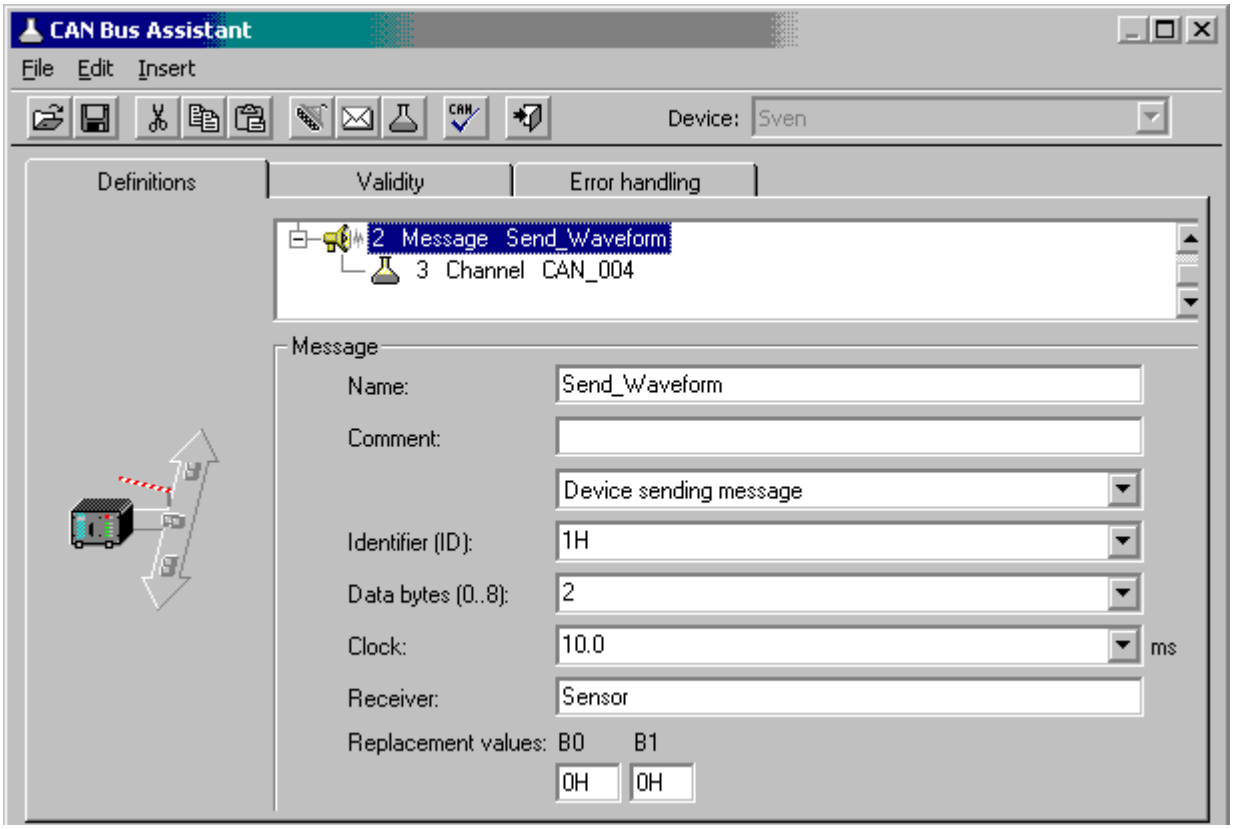
7.2.4.7.1.2 Initiating a shunt calibration for a bridge module via CAN

Proceed as for Balancing a bridge module via CAN 428. Simply substitute the HEX-code to send with 20Hex.

7.2.4.7.1.3 Outputting a waveform to a DAC module

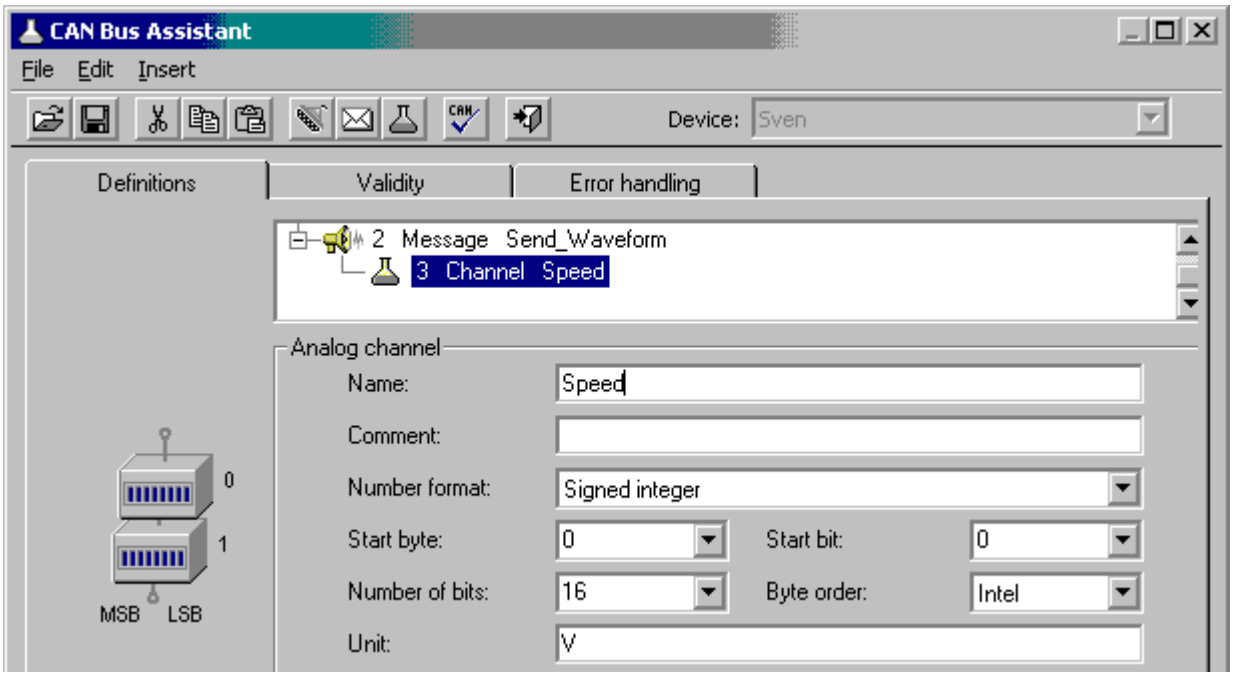
Cyclical output of values is another example. In this example, a ramp is generated. The data set to be outputted is loaded in the form of a characteristic curve. The `Charact` function is used to convert the ramp to the output values. This is sent as a message to a DAC module.

CAN-Assistant settings: Message



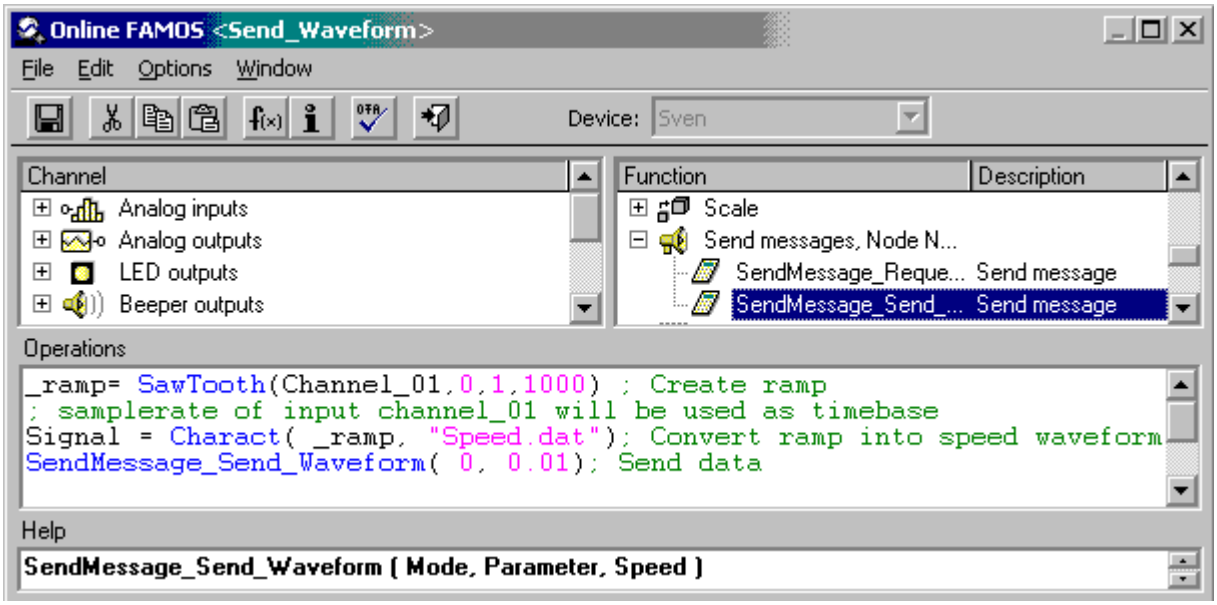
Create a send message with a channel

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

Important Note

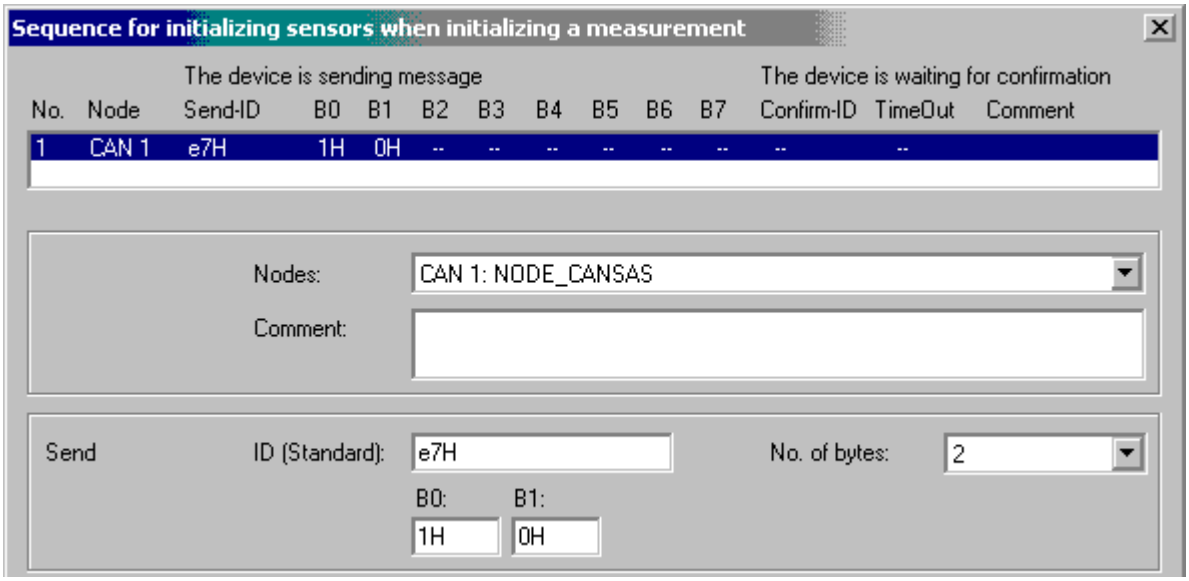
- The ramp must have the same length and same data rate as the data set to be outputted.
- In this example, an underline "_" was prefixed to the ramp. This causes the ramp to be retained in the signal processor as an auxiliary variable and cannot be found by the curve window.
- The data set to output 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.

7.2.4.7.2 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-Assistant. Once you are there, select the menu item Edit > Sensor initialization 393.

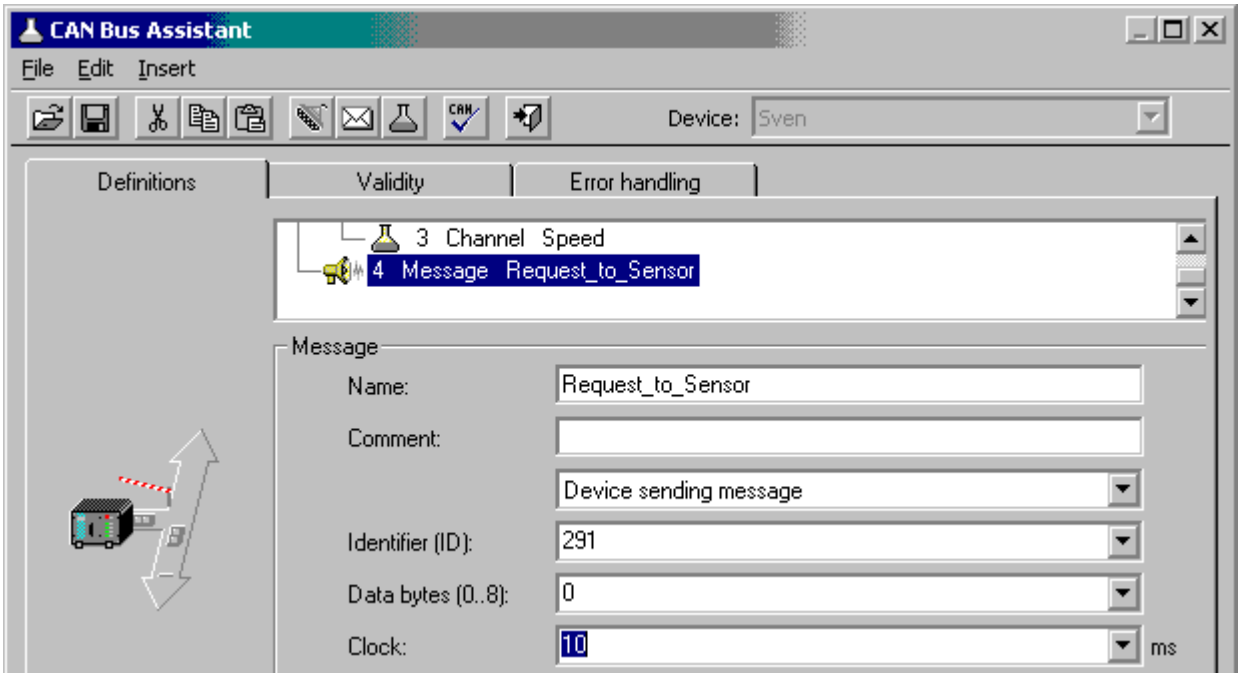
A possible sensor initialization code to enter could be as follows:



Example for initializing a sensor

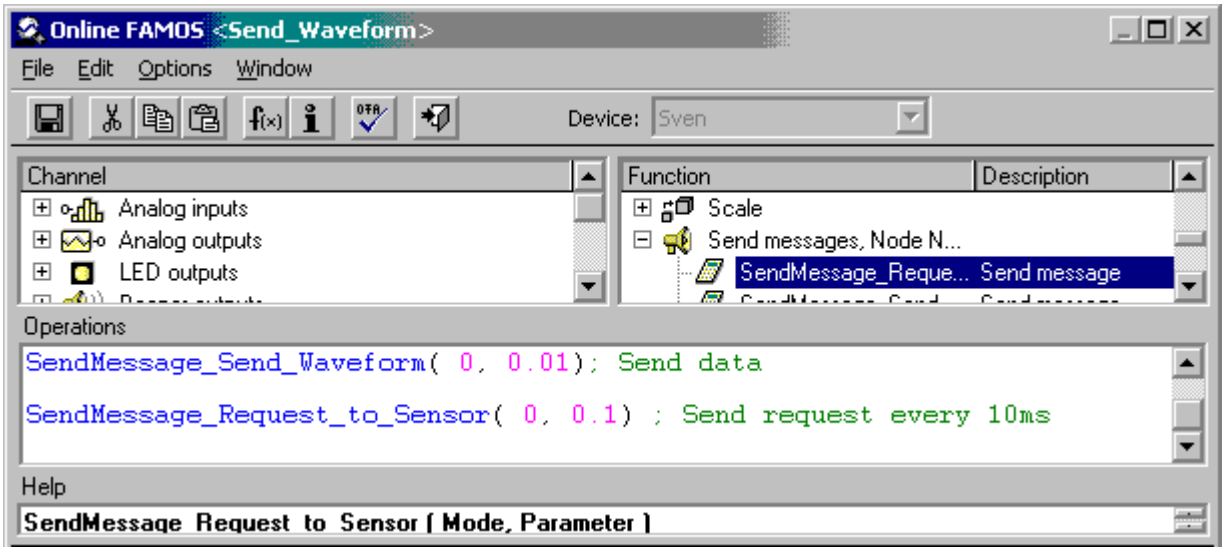
If your sensor requires a request for data, you will need to send a message to it. The ID and contents of the message are stated in the sensor's documentation.

In the CAN-Assistant, you must create a message and then put it in sending mode:

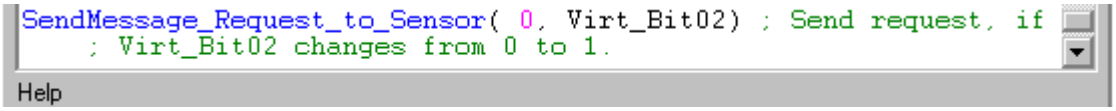


Create a send message

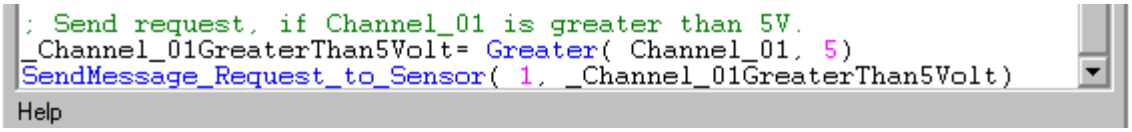
In imc Online FAMOS, select the SendMessage function and specify whether to make the data request cyclical or to make it depend on a state transition:



Example of cyclical sending at a 10ms rate



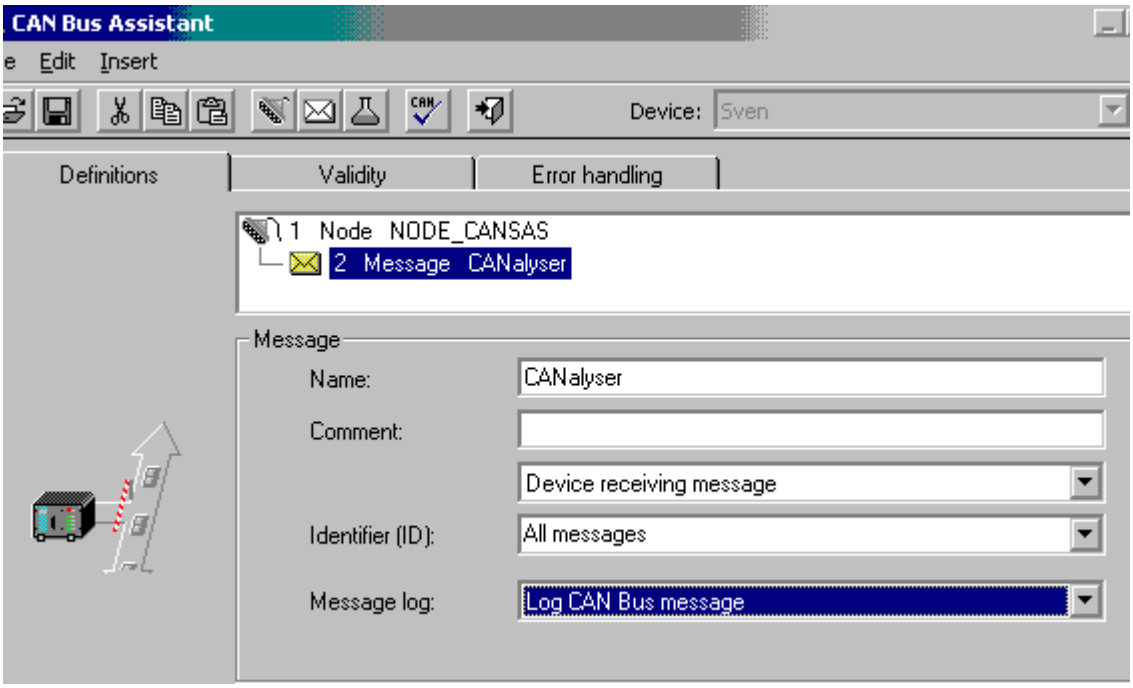
Example of an event-dependent request



Example of a request in response to a virtual bit, for instance by means of the DIODAC dialog.

7.2.4.7.3 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-Assistant which receives all the IDs. Furthermore you must set *Log CAN-Bus message* under Message log:

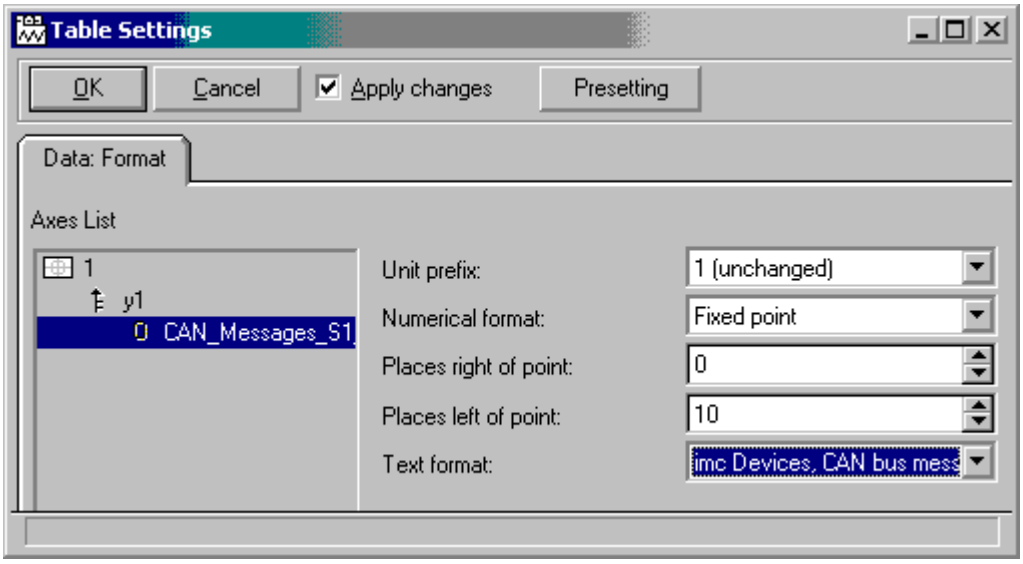


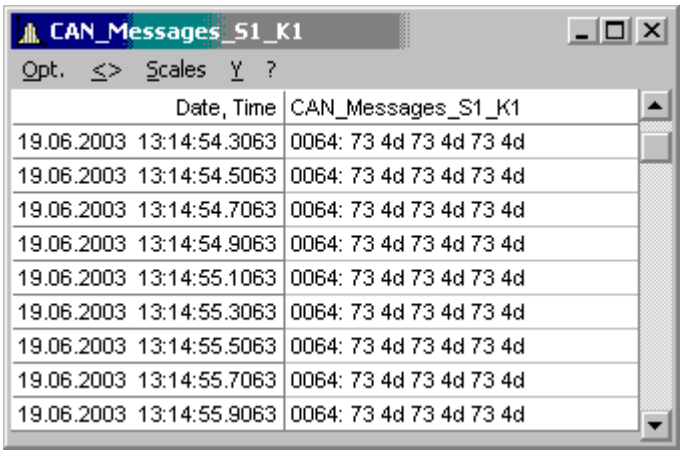
Create a log message

At this point, a channel is created whose tree-diagram entry appears together with the Field-bus 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*.





Display of CAN messages as a CAN analyzer

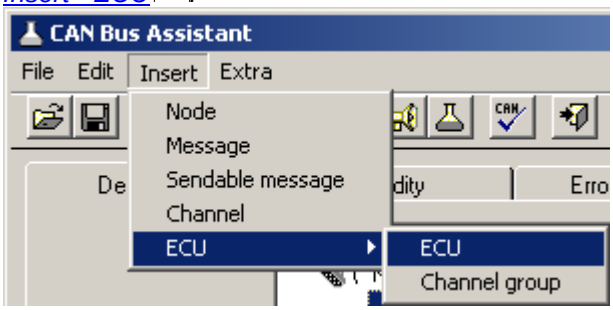
7.2.4.8 Control units in the CAN-Bus Assistant (ECU)

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.

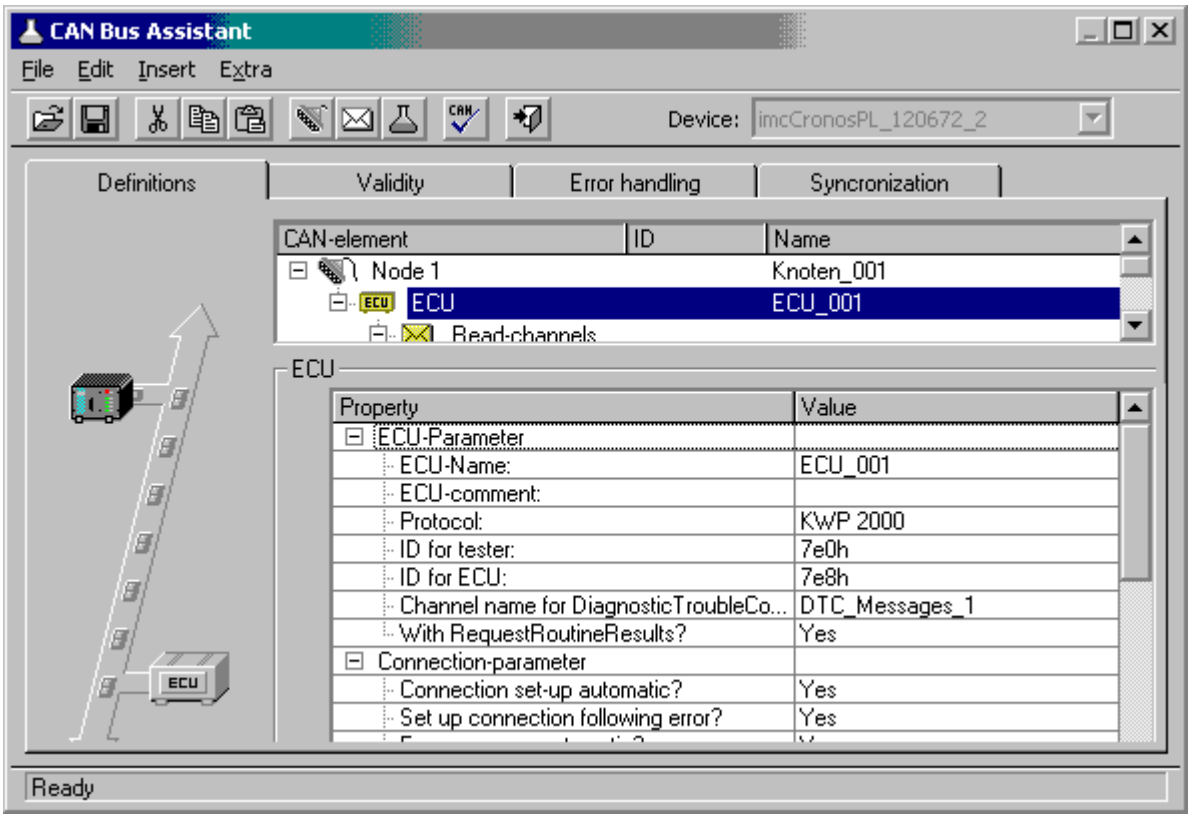
Requirements see here 390.

7.2.4.8.1 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* 424.

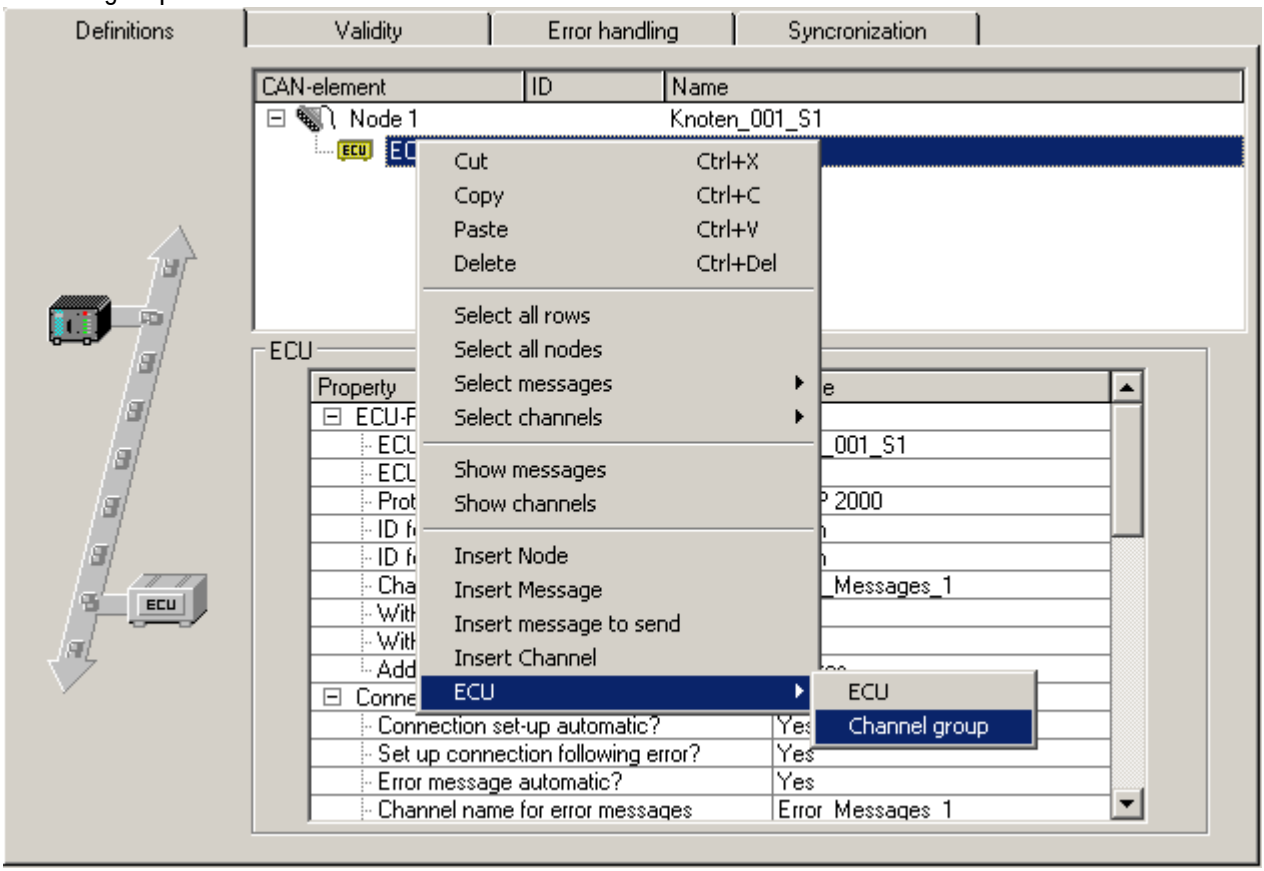


Open control unit properties



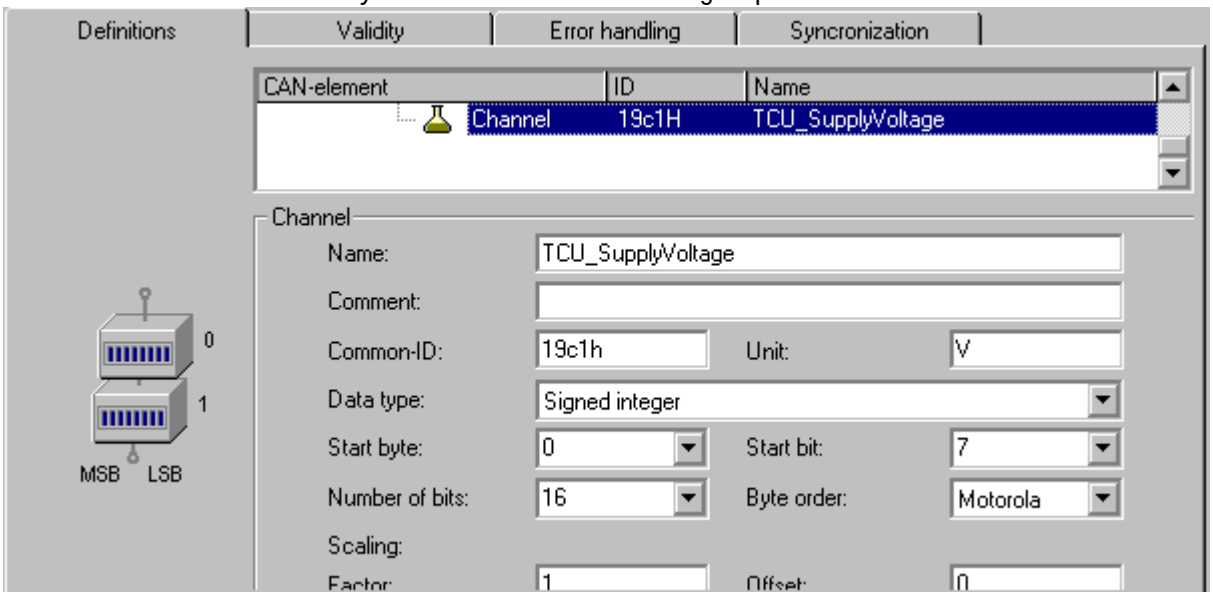
ECU properties

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

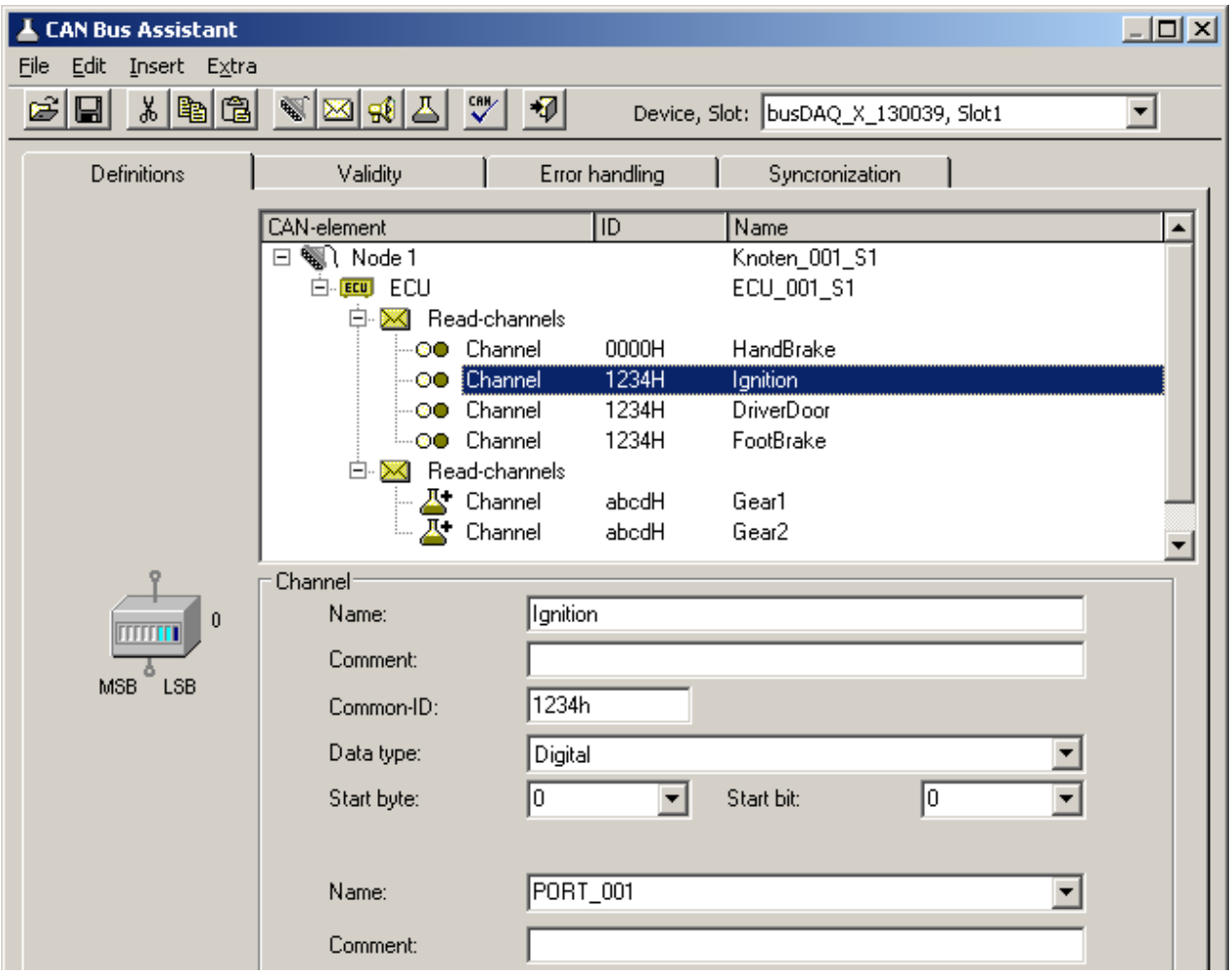
A new channel is automatically created for a new channel group.



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.

7.2.4.8.2 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

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

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

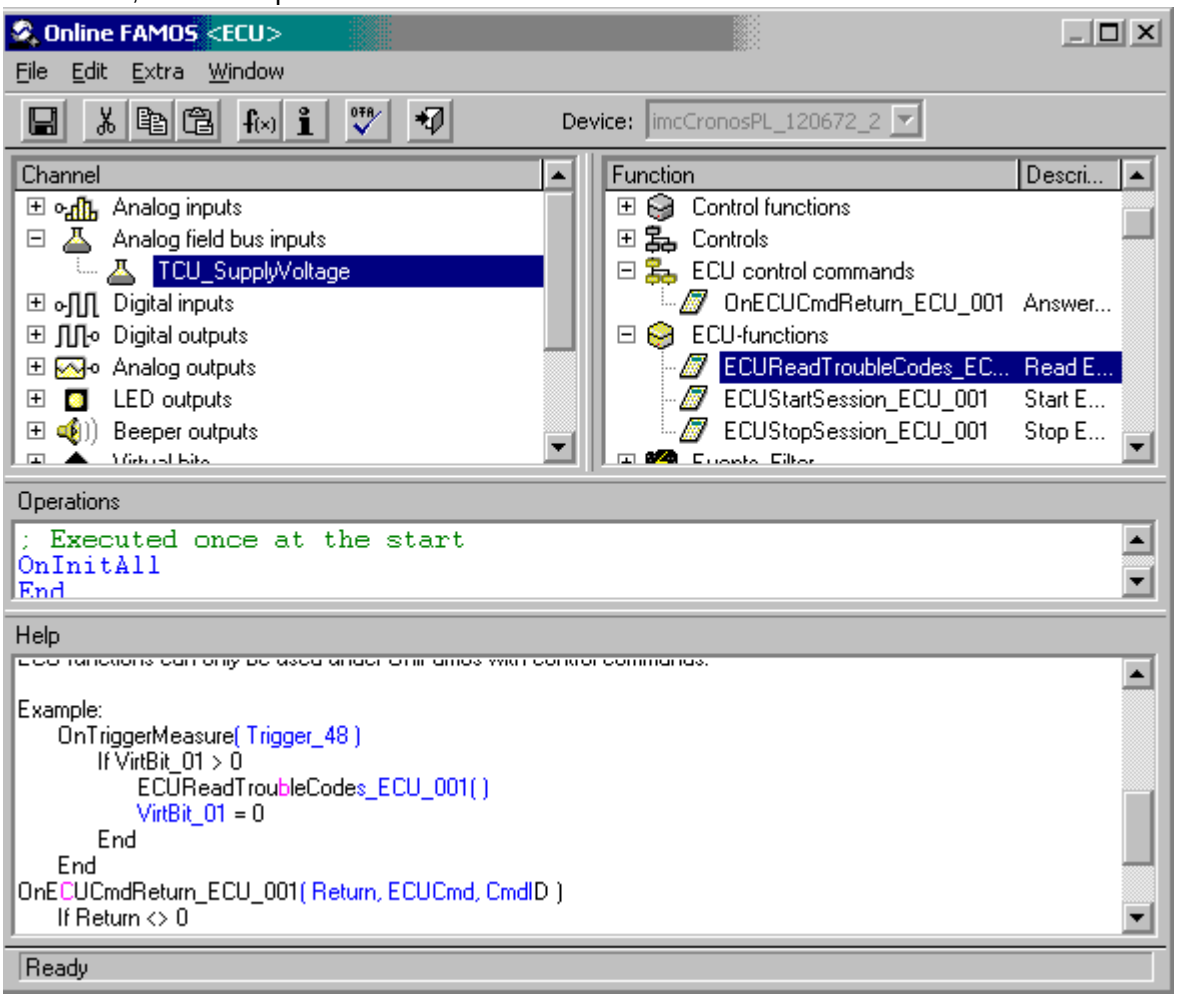
i: index of the ECU

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(Trigger_48)
    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
```

7.2.4.8.3 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.
- *With security access?*, *LogInKey:* If desired ECU functions are protected with Seed/Key, then *With security access?* must be answered with Yes and the key for the ECU must be entered for LogInKey. Only one permanent Seed/Key-pair is supported. The key must be stated in the property LogInKey.
- *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 398 all send messages can be send with higher baud rate.
- For control units which use a random Seed-value, algorithms for generating the corresponding key value are stored 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 4 88 and higher, MBus, ABus)
  so: under Linux, typical file extension for dynamically loadable libraries.
  This file must be located in the directory "*C:\Users\Public\Documents\imc\SeedKey*". imc DEVICES as of version 2.8R6 and older expect the file in subholder "*SeedKey*" which is to be created in the installation folder.
  In the property "*With security access?*", the selection "*No*", "*Fixed key*" and all filenames are offered. If a file has been selected, it is transferred upon preparation within the configuration. The Seed-Key algorithm can be set for all ECU protocols.
- As of imc DEVICES 2.8R6, "*.skb*" -files are also supported for "*SeedKey*". For this file type, *KeyIsSeedPlusOne* must be selected as the setting for the property *"With security access?*". The prerequisite is the presence of the file at the appropriate location 443.

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.

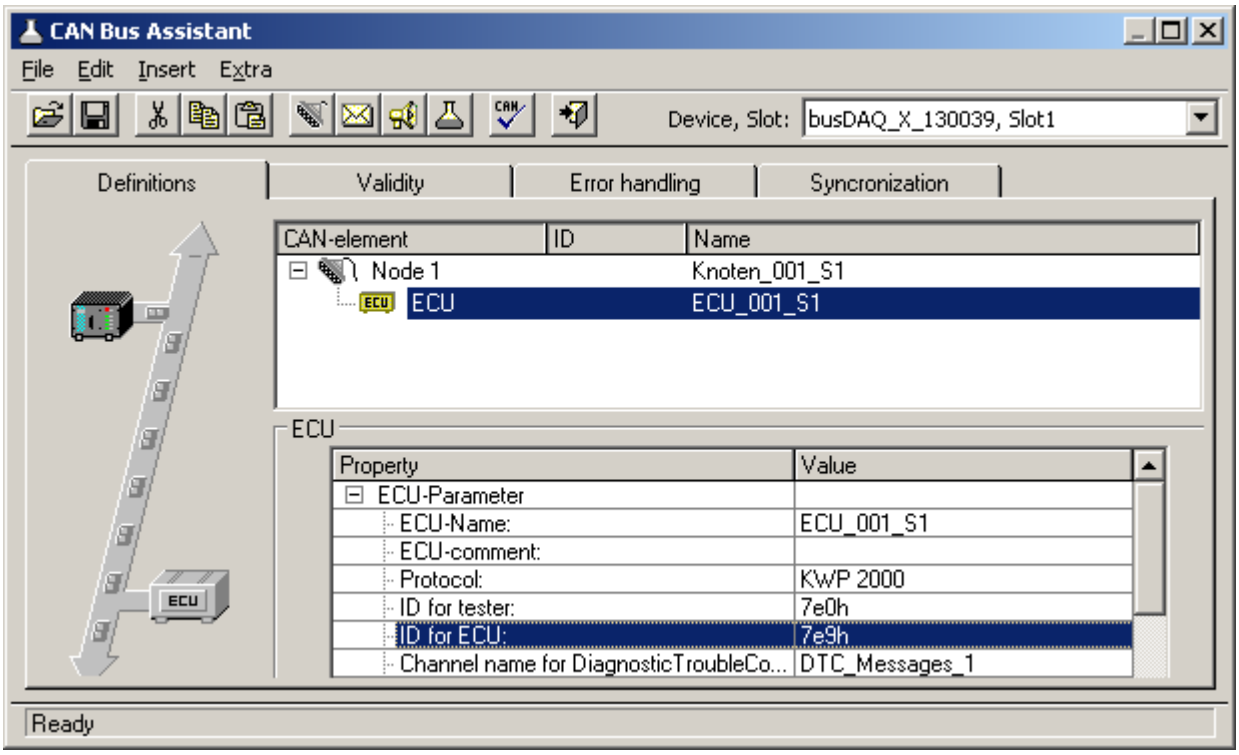
- *send at the start and always*: At the beginning there is no delay by the period "Tester present cycle"; instead the message is sent immediately.
- *Standard-timeout:* Time span in seconds within which the ECU must respond.
- *Expanded timeout:* Time span in seconds within which the ECU must conclude 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:

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)

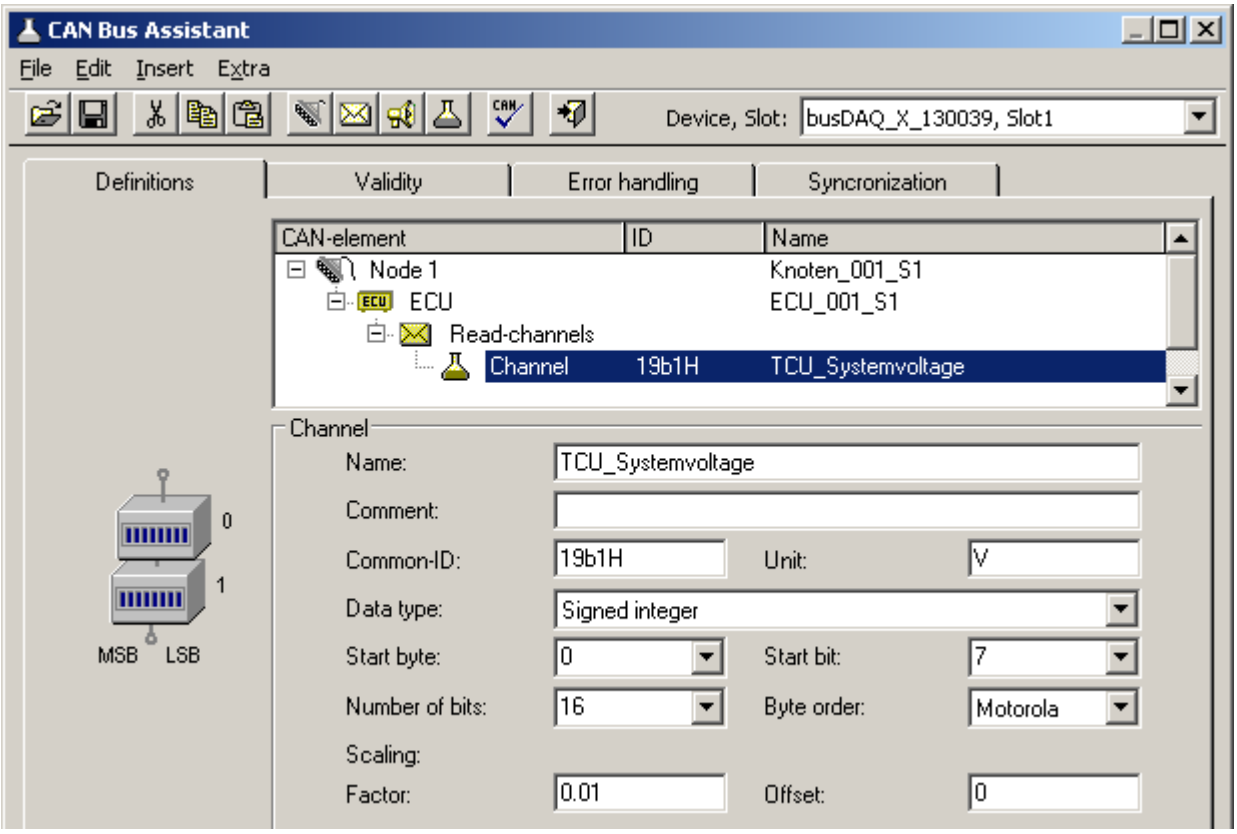
Once a new ECU has been added, the protocol is selected and the ID configured.



Set the identifier of the control unit

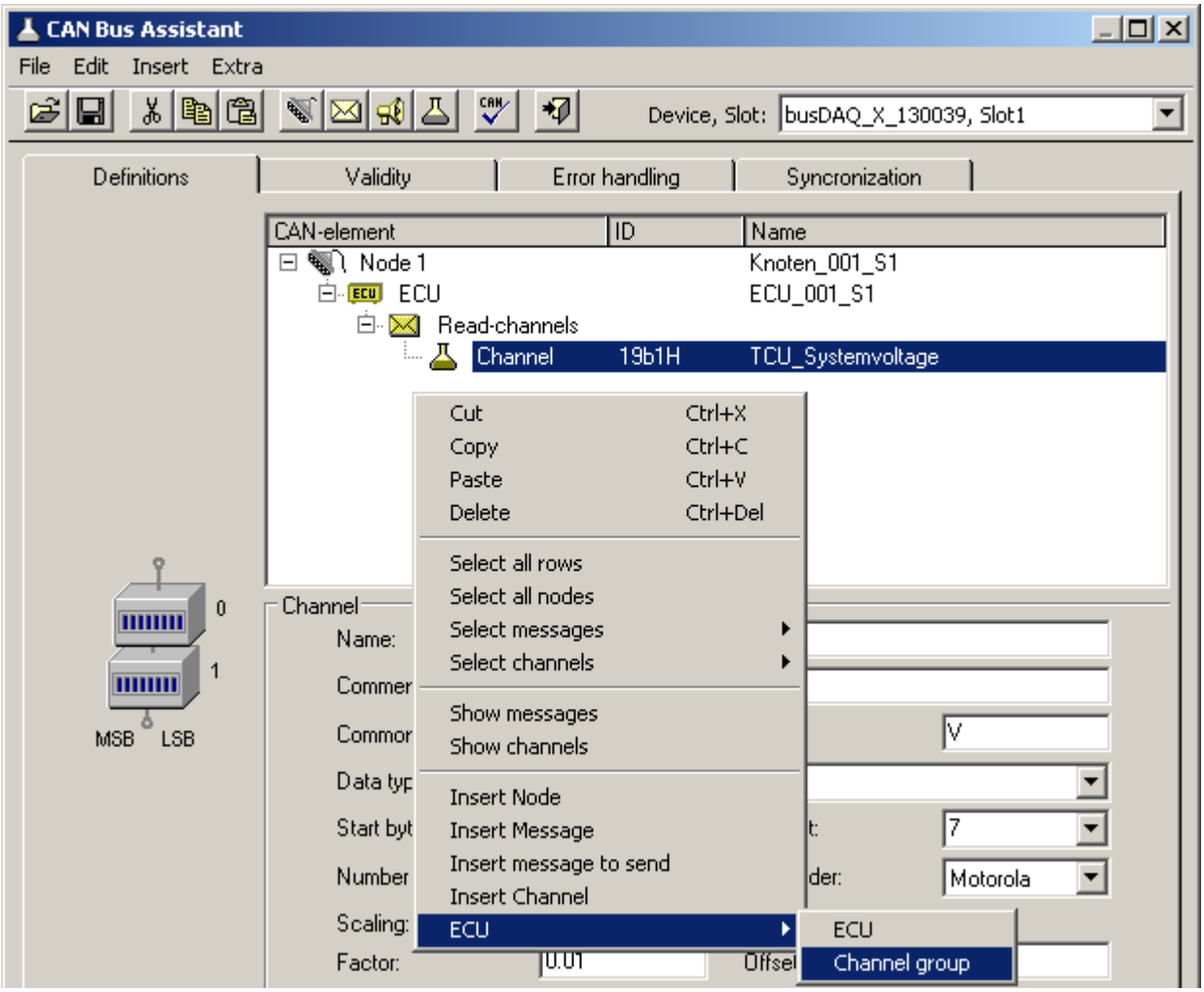
The value for StartDiagosticSession is left at the default setting.

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.

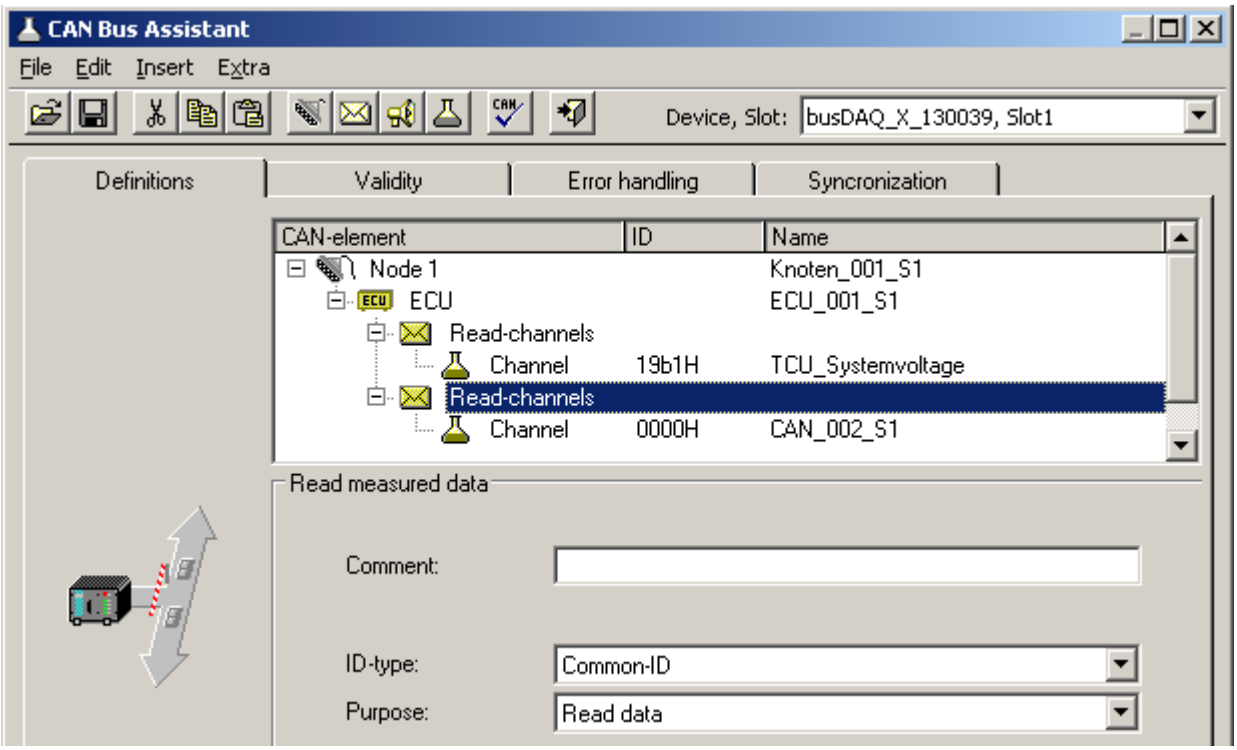


Configuring the channel

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.

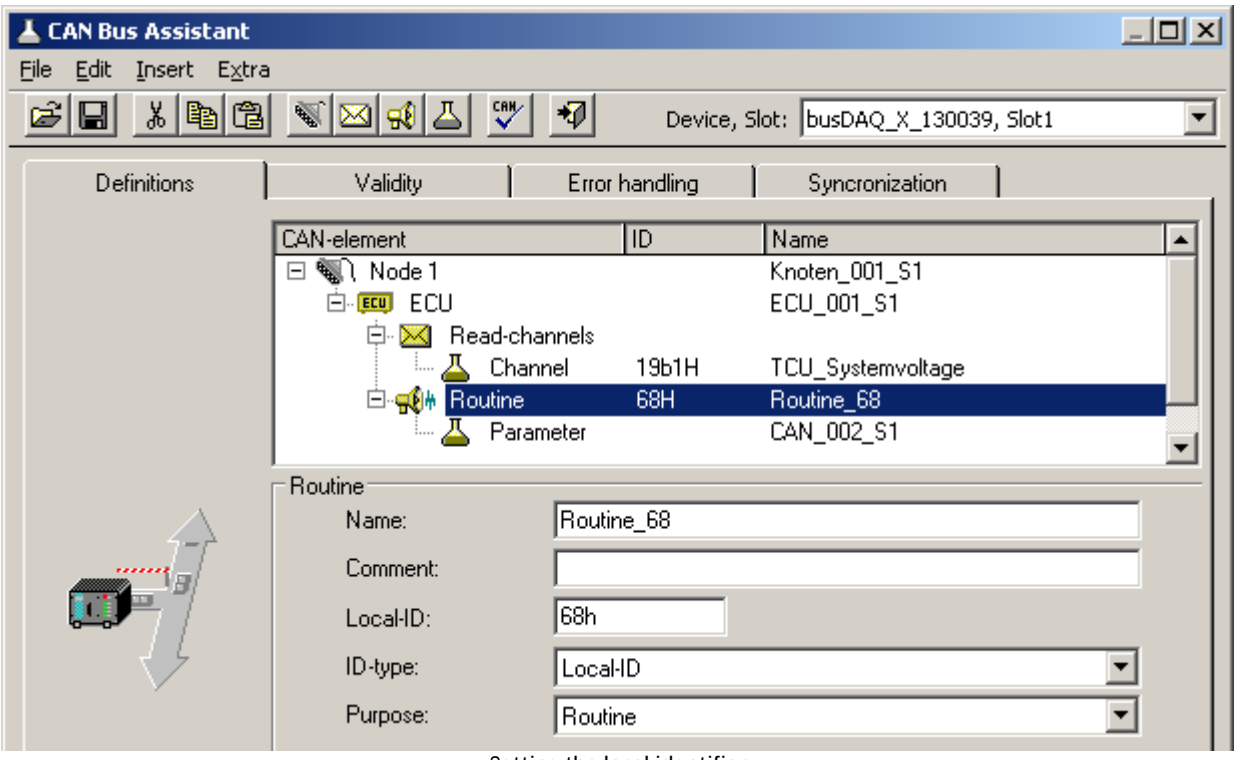


Create a channel goup



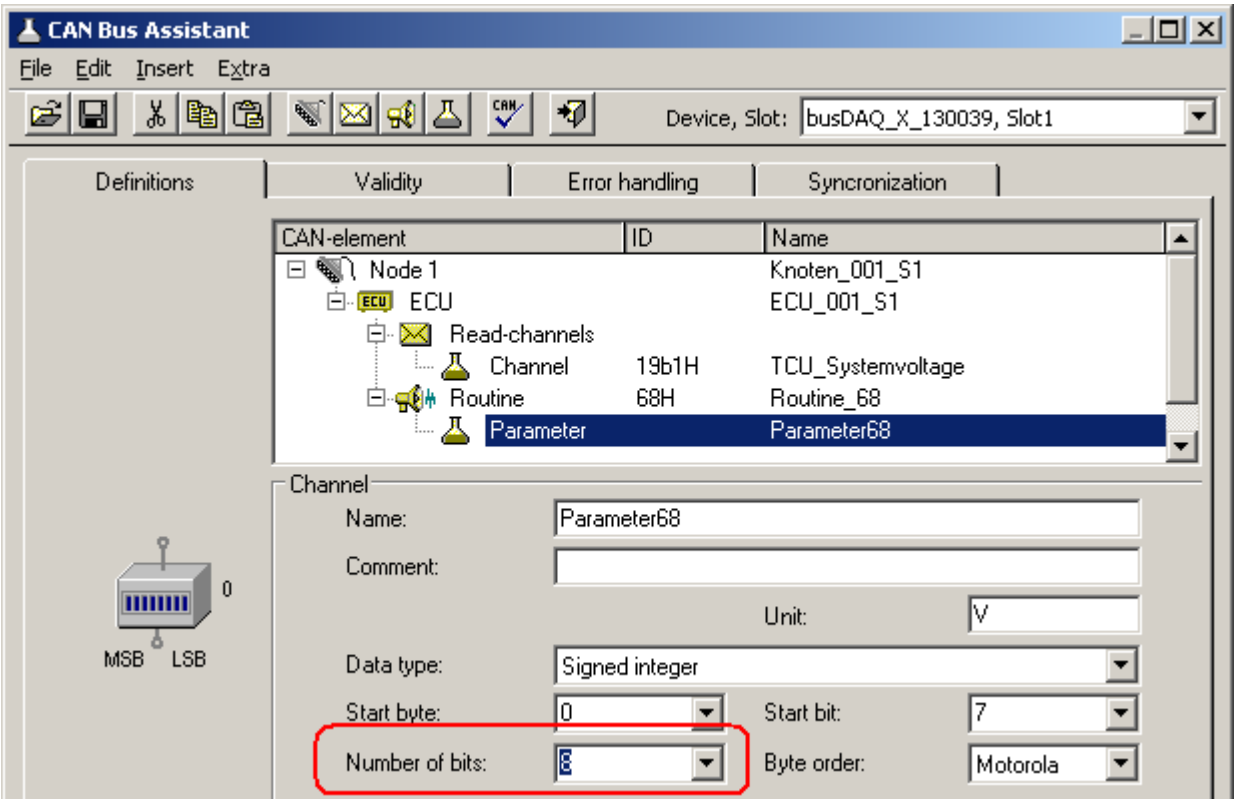
Channels of the channel group

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(Trigger_48)
End

; run during measurement
OnTriggerMeasure(Trigger_48)
    if Virt_Bit01 = 1
        Virt_Bit01 = 0
        ECUSend_Routine68( 9 )
    End
End

; run at end of measurement
OnTriggerEnd(Trigger_48)
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.

7.2.4.8.3.1 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.

7.2.4.8.3.2 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:

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.

7.2.4.8.4 OBD-2

OBD-2 enables reading of standardized ECU channels. The ECU protocol OBD-2 is implemented in accordance with the SAE J1979 standard (09.2010).

- 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

OnBoardDiagnostik, 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.

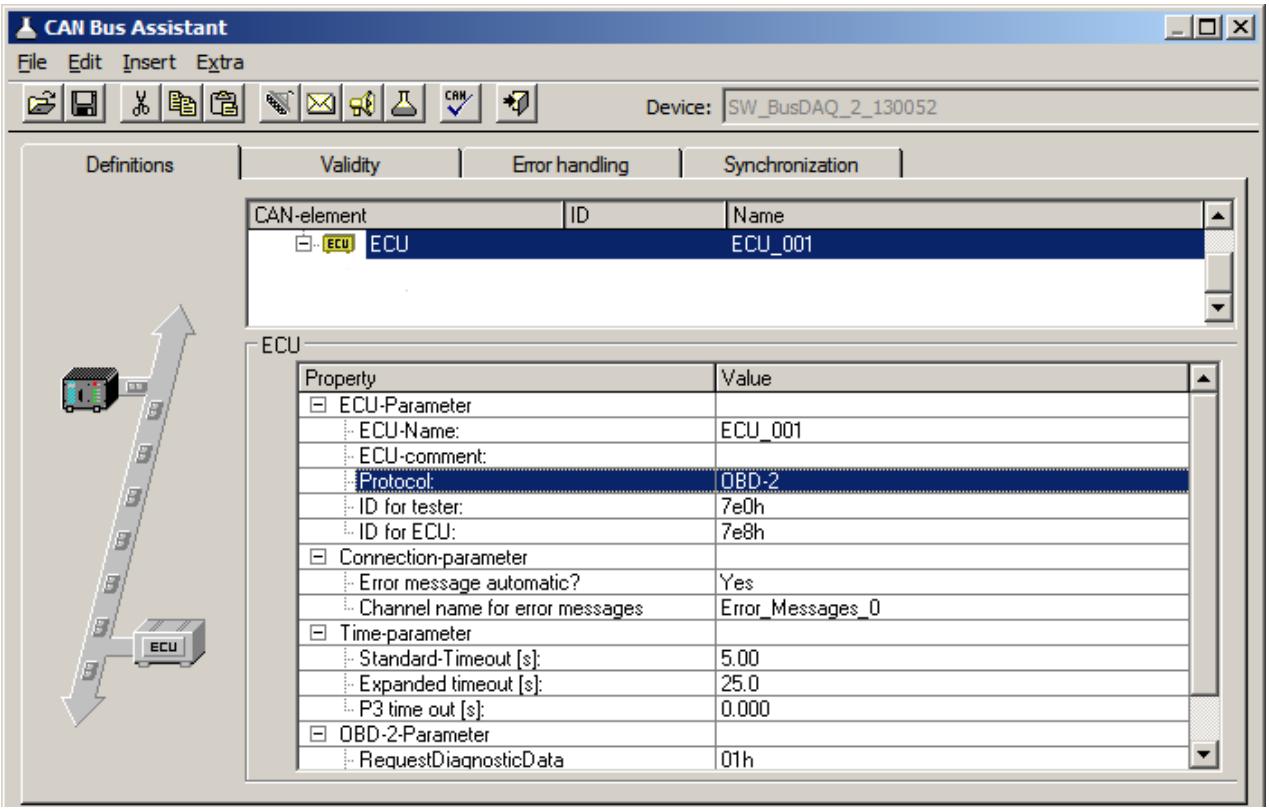
OnBoardDiagnostik monitors and saves even errors occurring during vehicle operation. Certain sensor data are recorded at fault condition.

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.

Settings in the assistant

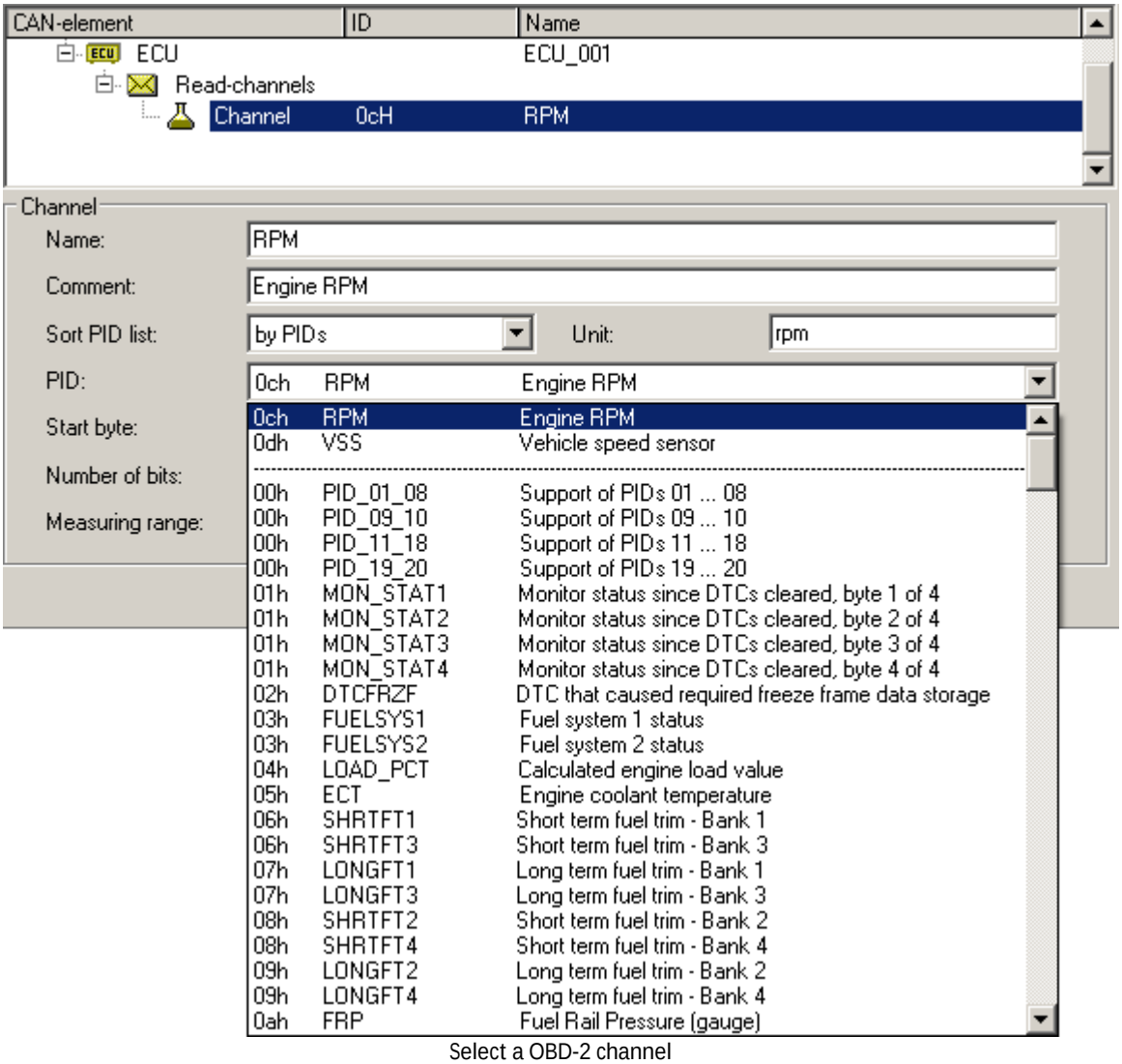
To select OBD-2 channels, add a ECU 437 to the configuration. Under ECU-Parameters, set the protocol type to OBD-2, before adding any channel:



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.

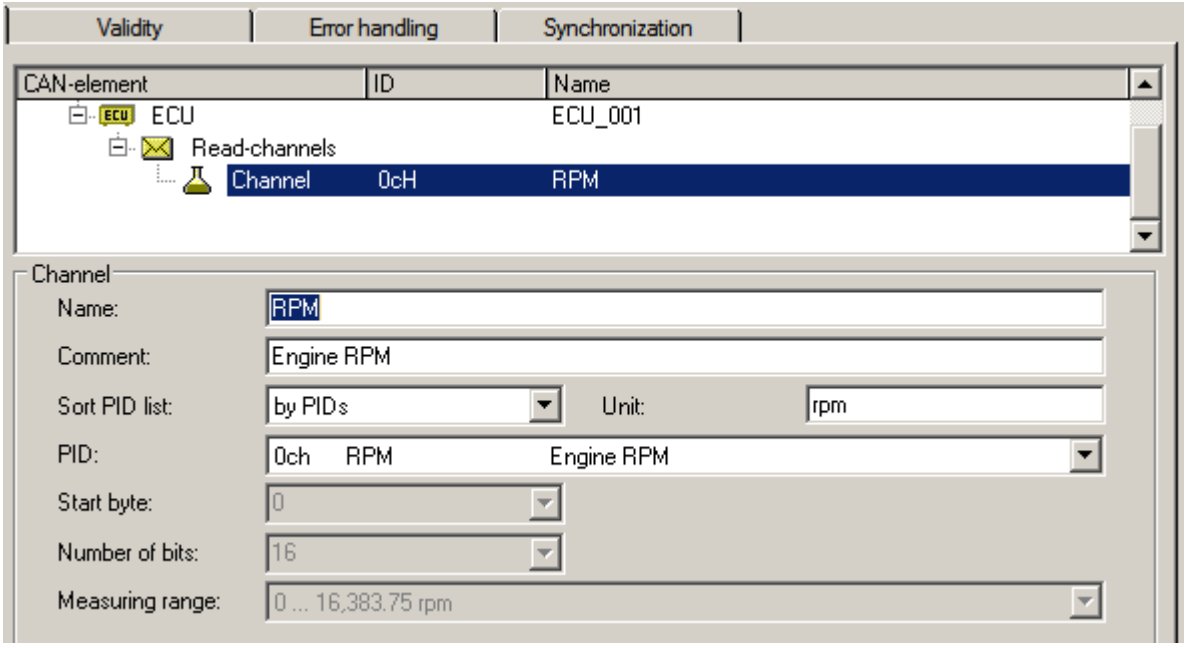
For selection of the ECU channels, permanent PIDs are set. Create a message and a channel and select the channel via the PID list:



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

7.2.4.8.5 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.
- *With security access?, LogInKey:* If desired ECU functions are protected with Seed/Key, then the property *With security access?* must be specified as *Yes* and for the LogInKey, the key for sending channels (export to the ECU) or for DAQ-lists must be specified.
- *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:

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
```

7.2.4.8.6 XCP

For XCP, the same notes apply as for CCP, with the following differences:

- XCP does not have station addresses
- *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.

7.2.4.8.7 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_**

7.2.4.8.8 UDS

The ECU protocol UDS is supported as of imc STUDIO 3.1.

The UDS protocol is similar to the KWP2000 protocol and is implemented according to the ISO 14229-1 standard (04.2007).

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

7.2.4.8.9 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* 425), 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(Trigger_48)
     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
```

7.2.4.8.10 Properties in common

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.

7.2.5 EtherCAT-Interface

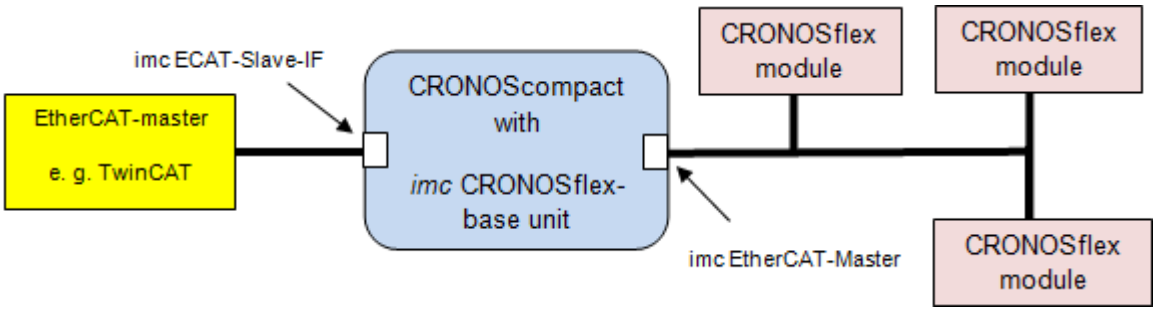
The fieldbus module imc ECAT-Slave-IF provide the integration of imc CRONOS*flex* and imc CRONOS*compact* devices into an automation system with an EtherCAT fieldbus.

The complete imc CRONOS device is integrated via the interface into the in den 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 are possible.

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 161 modules. imc CRONOS*flex* is a distributed measurement system, in which the EtherCAT Field-bus 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.



Configuration:

- For the configuration, a dedicated software assistant is provided.
- 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.
- Cyclical data are transmitted cyclically after the Fieldbus and the imc measurement system start.
- Irregular data can be queried sporadically by the EtherCAT Master by means of the protocol CoE (CANopen over EtherCAT), which is supported by the imc system.
- Configuration and parameterization of the imc measurement system are easy to perform using the device software imc STUDIO, by means of which an "EtherCAT-Slave-Information" document (ESI) can be created or exported in XML-format. This serves the purpose of system configuration for the EtherCAT Master.
- Alternatively or additionally, parameterization of the imc systems via the CoE protocol is possible.

7.2.5.1 EtherCAT general

EtherCAT (Ethernet for Control Automation Technology)

EtherCAT is an Ethernet-based Field-bus. EtherCAT is distinguished by high performance and real-time capability and thus enables open-and closed-loop control mechanisms which classical Field-bus 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 (CANopen over EtherCAT)

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/>
```

7.2.5.2 Prerequisites

For operation of imcECAT-IF the module CRC/ECAT-Slave is required.

The module is supported by imc CRONOS*compact* and imc CRONOS-PL.

7.2.5.3 Configuring the EtherCAT interface

Configuration of the Field-bus 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.

7.2.5.4 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

7.2.5.5 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 0:	Subobjekt 0
Subindex 1:	Status
Subindex 2:	imc Source
Subindex 3:	CANopen Data Type
Subindex 4:	Factor
Subindex 5:	Offset
Subindex 6:	Unit
Subindex 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 DEVICES Software.

Example

A PVV defined in the 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 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

Values: 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

Values:

6	=	not definied
7	=	Amplifier-channel
8	=	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

Values:

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

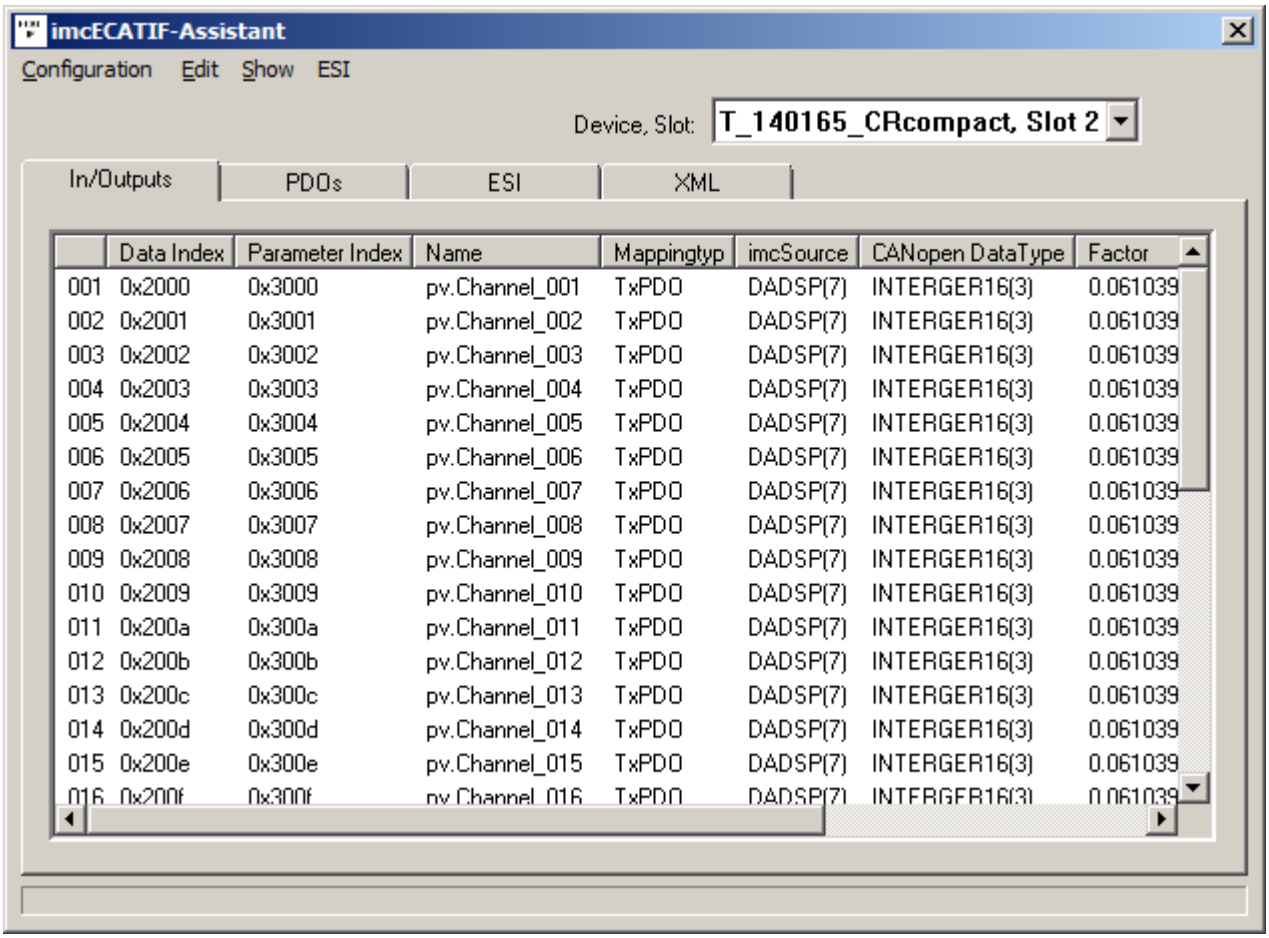
Sample Rate (only defined forResource-TypeAmplifier channel and Fieldbus-channel)

The content of the subobject pertains to the channel associated with the PVV.

Data type: UNSIGNED32

7.2.5.6 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
001	0x2000	0x3000	pv.Channel_001	TxPDO	DADSP(7)	INTERGER16(3)	0.061039
002	0x2001	0x3001	pv.Channel_002	TxPDO	DADSP(7)	INTERGER16(3)	0.061039
003	0x2002	0x3002	pv.Channel_003	TxPDO	DADSP(7)	INTERGER16(3)	0.061039
004	0x2003	0x3003	pv.Channel_004	TxPDO	DADSP(7)	INTERGER16(3)	0.061039
005	0x2004	0x3004	pv.Channel_005	TxPDO	DADSP(7)	INTERGER16(3)	0.061039
006	0x2005	0x3005	pv.Channel_006	TxPDO	DADSP(7)	INTERGER16(3)	0.061039
007	0x2006	0x3006	pv.Channel_007	TxPDO	DADSP(7)	INTERGER16(3)	0.061039
008	0x2007	0x3007	pv.Channel_008	TxPDO	DADSP(7)	INTERGER16(3)	0.061039
009	0x2008	0x3008	pv.Channel_009	TxPDO	DADSP(7)	INTERGER16(3)	0.061039
010	0x2009	0x3009	pv.Channel_010	TxPDO	DADSP(7)	INTERGER16(3)	0.061039
011	0x200a	0x300a	pv.Channel_011	TxPDO	DADSP(7)	INTERGER16(3)	0.061039
012	0x200b	0x300b	pv.Channel_012	TxPDO	DADSP(7)	INTERGER16(3)	0.061039
013	0x200c	0x300c	pv.Channel_013	TxPDO	DADSP(7)	INTERGER16(3)	0.061039
014	0x200d	0x300d	pv.Channel_014	TxPDO	DADSP(7)	INTERGER16(3)	0.061039
015	0x200e	0x300e	pv.Channel_015	TxPDO	DADSP(7)	INTERGER16(3)	0.061039
016	0x200f	0x300f	pv.Channel_016	TxPDO	DADSP(7)	INTERGER16(3)	0.061039

EtherCAT-IF assistant

It is possible to subsequently remove any EtherCAT channels from the XML file.

In the first step, pv.Variables appear as EtherCAT outputs. Process vector variables created in imc Online FAMOS can be converted to EtherCAT inputs.

7.2.5.6.1 Procedure - Configuration variants

Configuration variant I (no mapping)

1. Create an experiment with all channel settings, imc Online FAMOS etc.
2. Under Edit > Outputs 466 or Edit > Inputs 467, define which PVVs in the imcECAT-IF's CANopen object directory are to be mapped.
3. Using Edit > Add PDO 467, 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 467 automatically
4. Save the configuration created by means of Configuration > Accept changes 465.
5. With ESI > Save device specific ESI 469, 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.

Configuration variant II

1. Create an experiment with all channel settings, imc Online FAMOS etc.
2. Under Edit > Outputs 466 or Edit > Inputs 467, define which PVVs in the imcECAT-IF's CANopen object directory are to be mapped.
3. Using Edit > Add PDO 467, 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 467.
4. Under Edit > Mapping 468 create the mapping 468 of the cyclical data transfer.
5. Save the configuration created by means of Configuration > Accept changes 465.
6. With ESI > Save device-specific ESI 469, 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.

7.2.5.6.2 Menu

7.2.5.6.2.1 Configuration

Creation and editing of an EtherCAT 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.

New

This creates a new configuration. Existing settings are reset.

Next, the channels must be set as either outputs 466 or inputs 467 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

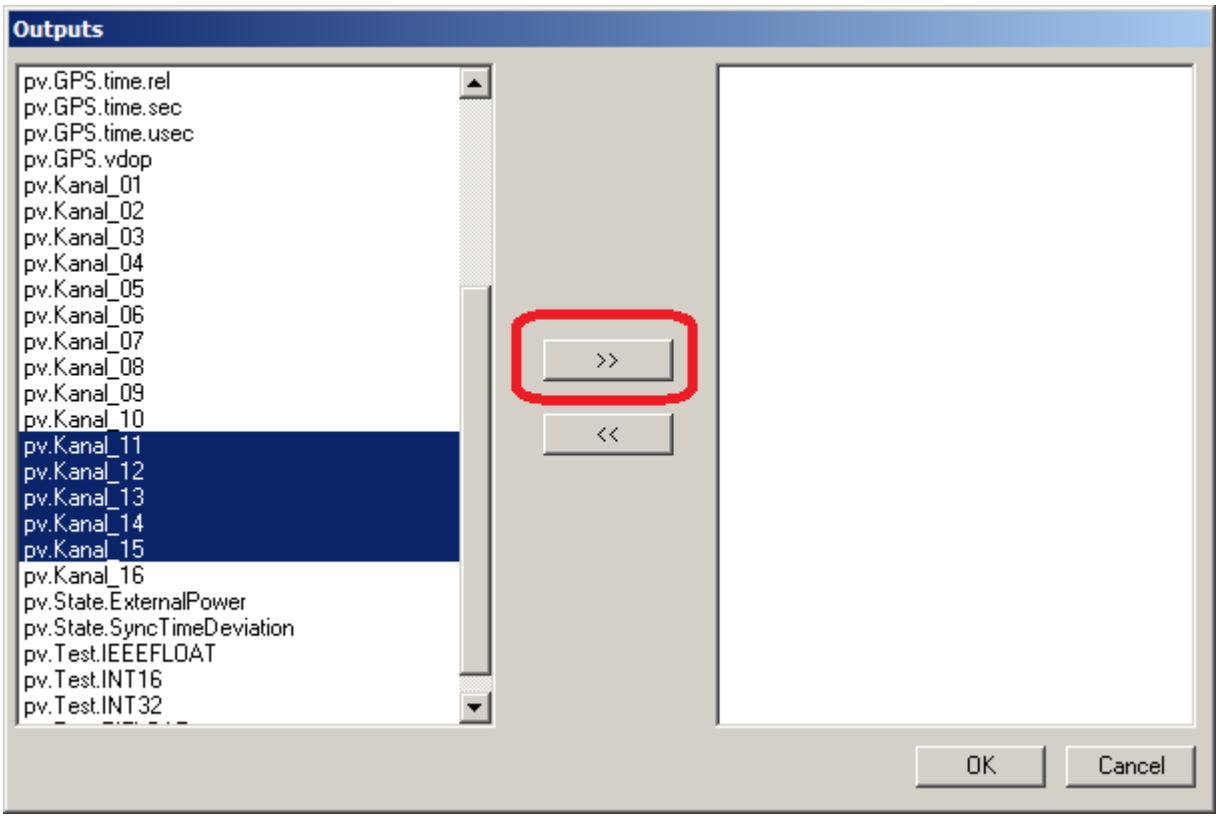
Saves the current EtherCAT configuration. Thus the configuration can be used for an experiment that will be loaded for a different device.

7.2.5.6.2.2 Edit

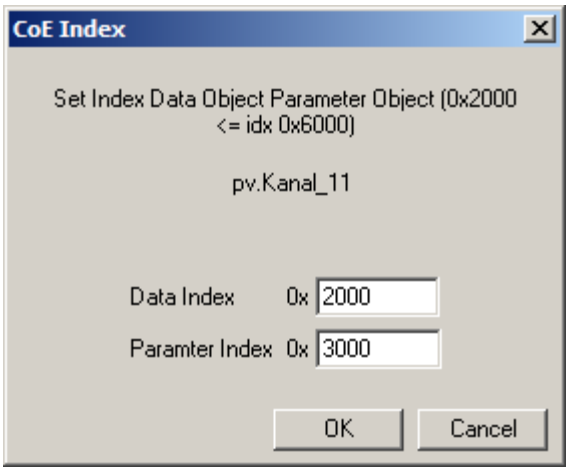
Once the configuration set has been prepared for the EtherCAT Interface by means of the menu item New 465, the channels can be set as either outputs or inputs with data and parameter addresses.

Alternatively, you can generate a complete EtherCAT configuration using the menu item Create PDOs automatically 467.

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.

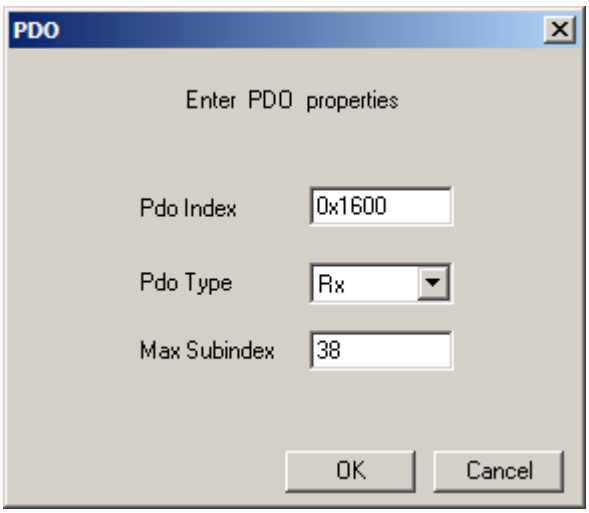
Outputs generated with *Create PDOs automatically* 467 can be brought to the left side using the button <<. Subsequently, these appear under *Edit > Inputs* 467 and can be configured as inputs there.

Device pv.Variables which are written via EtherCAT must be configured as inputs.

The PDOs act as the containers in which the inputs and outputs are organized. The menu item Create PDOs automatically 467 also generates these. The menu item *Edit > Add PDO* allows manual configuration.

PDOs for sending contain outputs and are called RxPDOs.

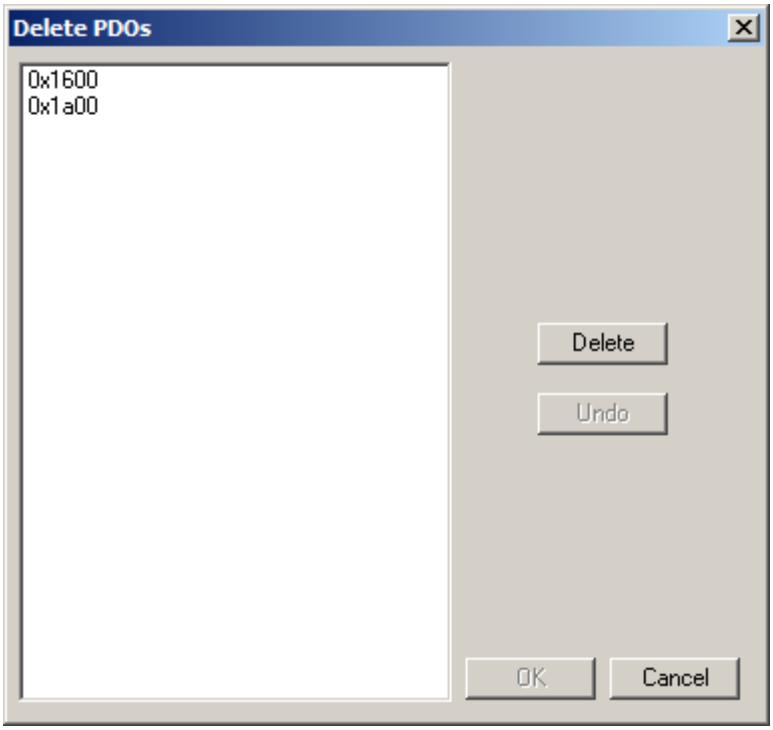
PDOs for receiving contain inputs and are called TxPDOs.



Add PDO

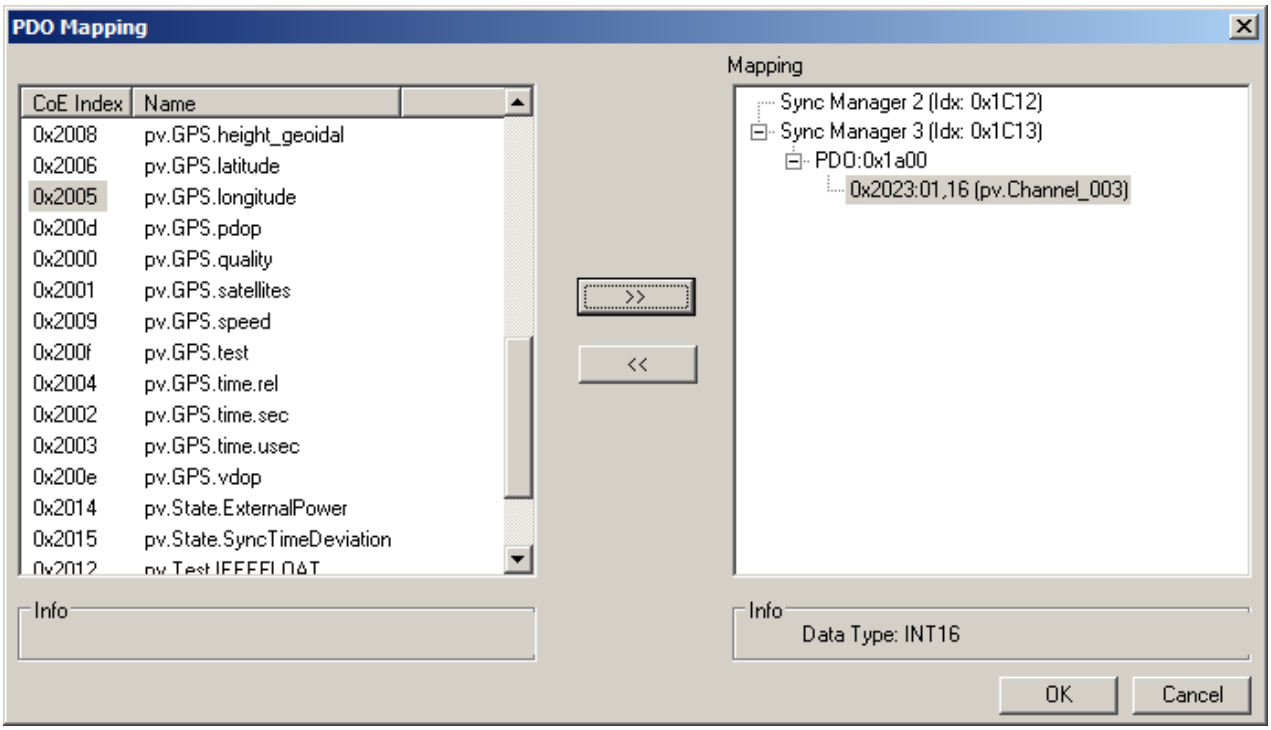
Creation of an RxPDOs with index 0x1600 and subindex 38.

Use this menu item to delete any PDOs no longer needed.



Delete PDOs

PDOs can also be generated automatically.



Mapping dialog

Create the configuration as described in the section Configuration variant II 465.

Subsequently, you can adapt the mapping.

7.2.5.6.2.3 Show

The items in the *Display* menu affect the display of certain parts of the configuration in the Editor.

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

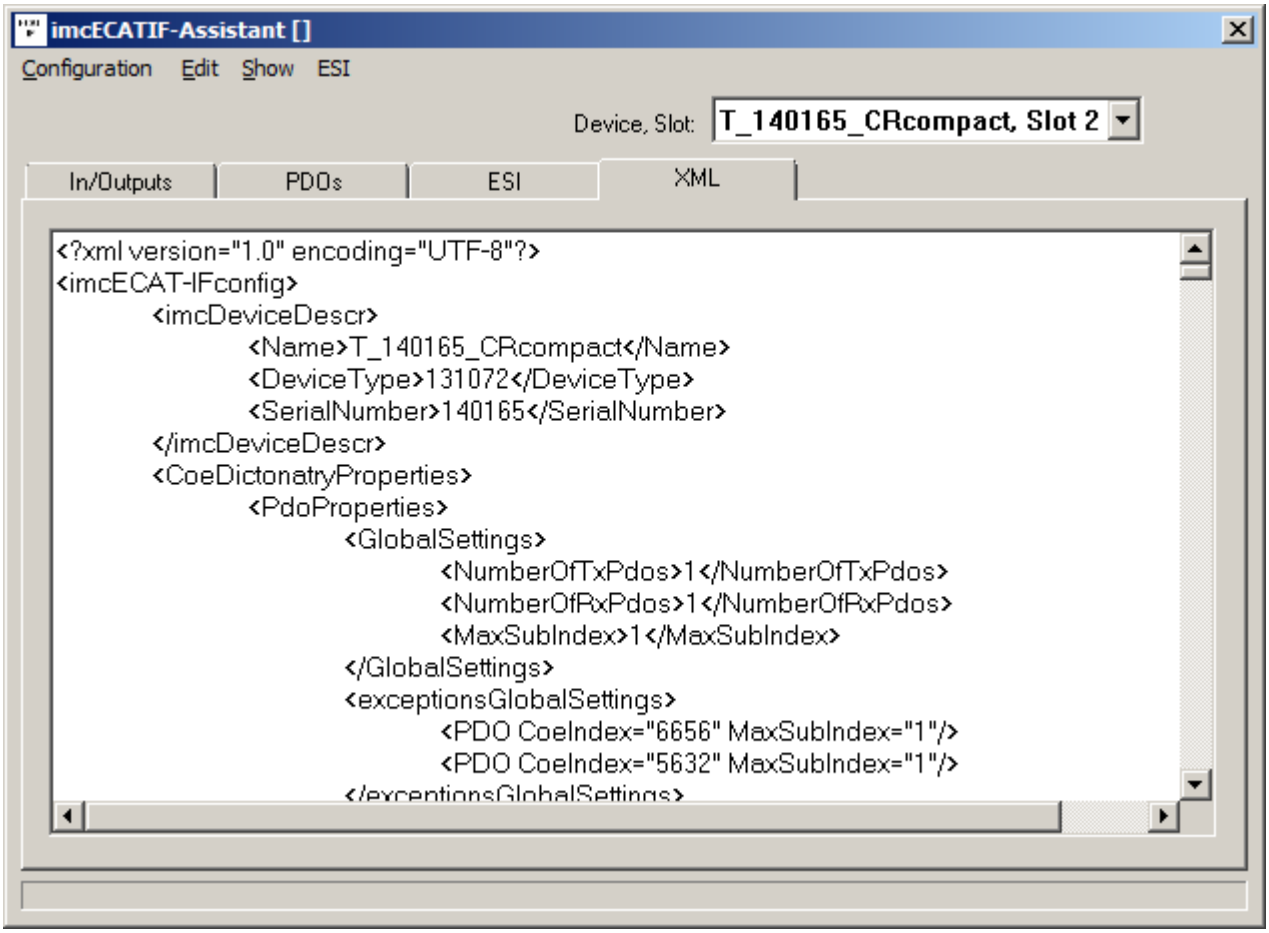
Display of all RxPDOs in the Editor.

TxPDOs

Display of all TxPDOs in the Editor.

Xml

Display of the XML-configuration generated.



Show configuration as XML

7.2.5.6.2.4 ESI

The EtherCAT interface's transfer properties can be transferred as an *EtherCAT-Slave-Information* document (ESI).

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.

7.2.6 Ethernet-Bus Interface

What is the Ethernet-Bus?

The Ethernet bus is used in modern aircrafts and uses the Ethernet protocol TCP/IP, familiar world-wide. Depending on whether the message protocol is IGH, HC or ACRA, then either UDP or TCP is used as the transfer type. The various message protocol types differ in the data format, in the header and in various other data processing attributes.

7.2.6.1 Ethernet Assistant

An Ethernet channel must be defined as a channel for the imc measurement device in a message. For this purpose, the Ethernet Assistant is used.

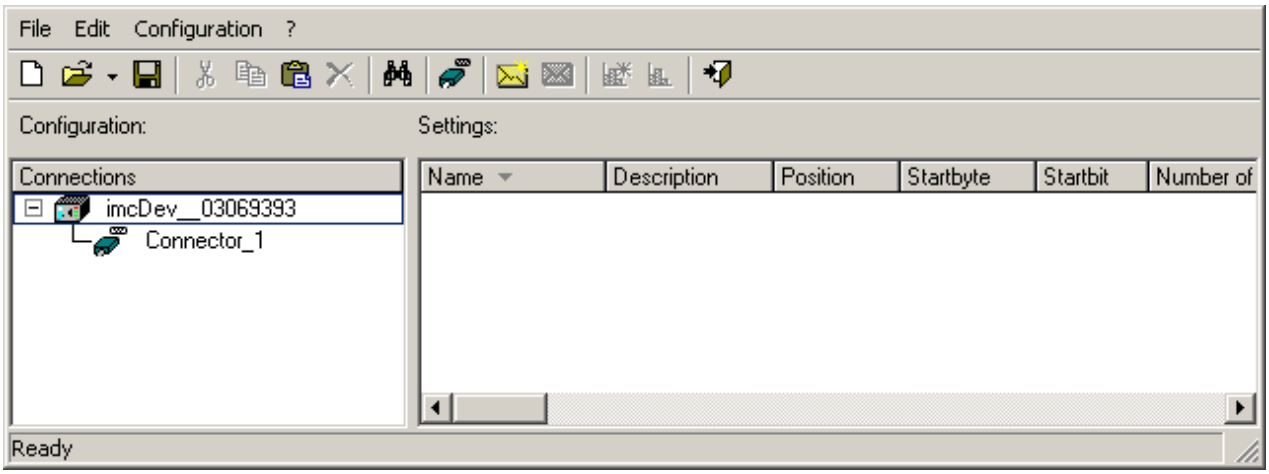
Ethernet channels appear in imc STUDIO as additional measurement channels with the properties of other fieldbus channels. This means that these channels can, for instance, be subjected to mathematical operations in imc Online FAMOS, or be recorded in response to a trigger working in conjunction with 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.

7.2.6.2 Starting the Assistant

Start the Assistant from within the imc STUDIO software via the Ribbon *Setup-Configuration* > *Ethernet-Assistant*.

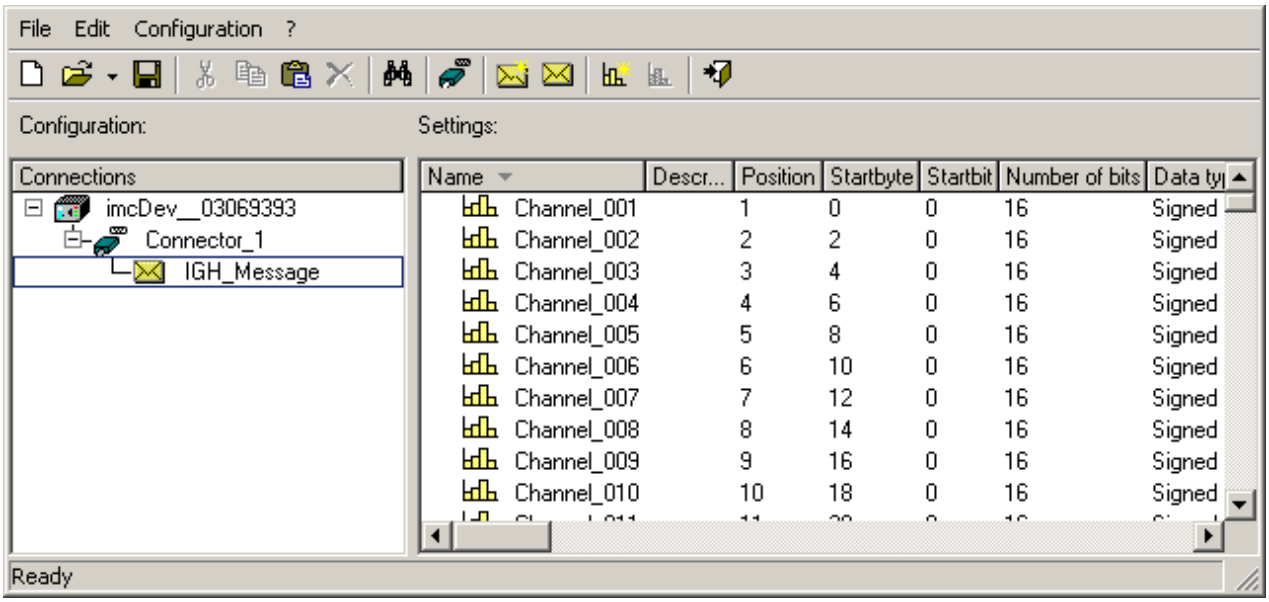
After the start, the Assistant appears with the following dialog:



Ethernet Assistant

The Assistant's main window is divided in halves:

- The left side, designated Configuration, lists the device, its receiver channels and the associated messages.
- The right side with the title Settings, lists the currently selected message and its associated, defined Ethernet channels.



Example for ethernet configuration

The definitions made here under Settings for the respective messages and channels are saved in a separate configuration file (.xml format) and thus are independent of the associated experiment. This configuration file can be considered a sort of database, in which all defined Ethernet channels with their assignment to the respective messages are recorded. Thus, to configure a experiment with the Ethernet Assistant, all that is needed is to load an Ethernet configuration file (.xml format) and to close the Assistant. All defined Ethernet channels then appear in the imc STUDIO operating software as channels and can be displayed or subjected to further processing in imc Online FAMOS.

7.2.6.3 Ethernet Configuration File

All Ethernet channel and message definitions are saved in a configuration file. This configuration file can be used as a central database by opening it for each experiment and deleting, copying or moving the Ethernet messages and Ethernet channels as desired.

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.

7.2.6.3.1 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*.

7.2.6.3.2 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 *. xml. It is also possible to load *.csv-files, in order to import one's own existing Ethernet channel databases.

7.2.6.3.3 Save Configuration File (as)

To save a configuration file, click on either the corresponding button in the toolbar or the menu item *File > Save*.

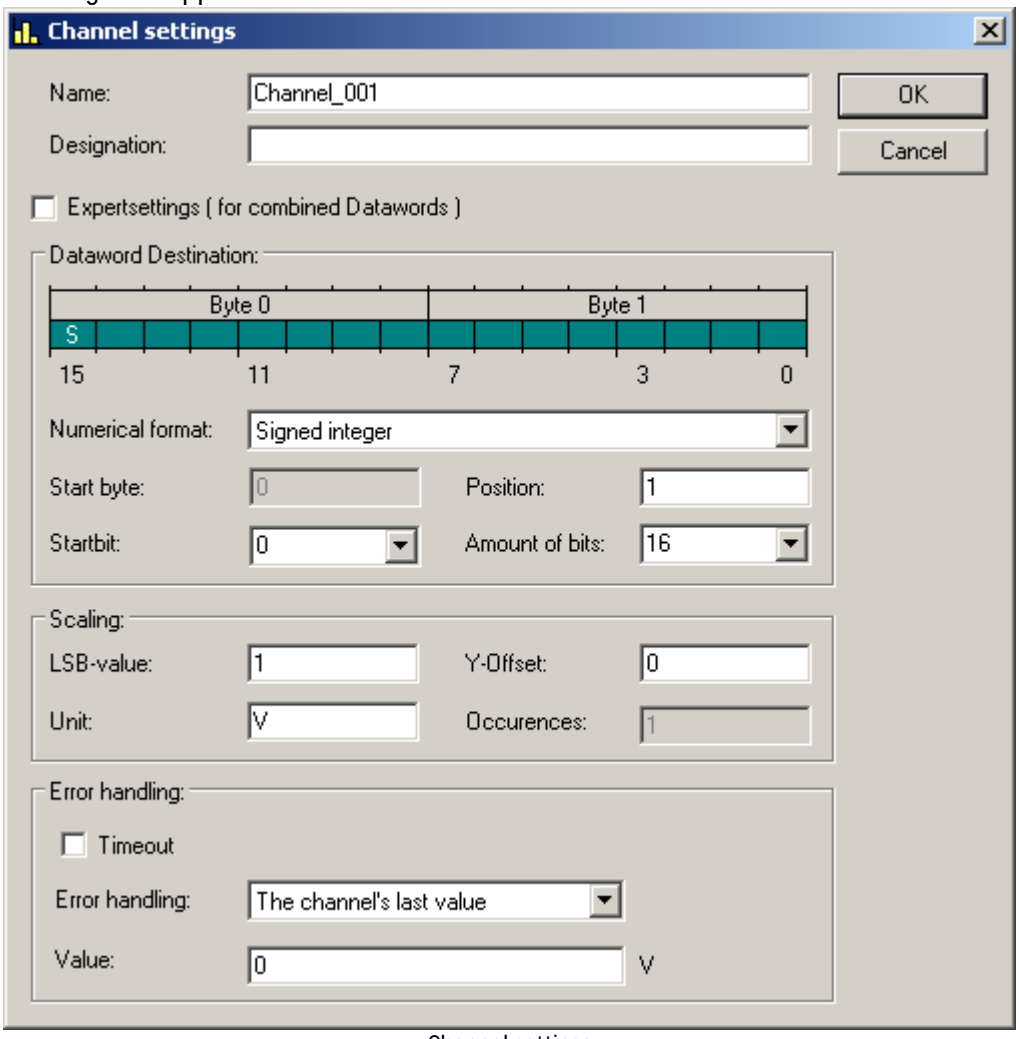
If the open configuration file is to be saved under a different name or in a different directory, menu item *File > Save as* is used. In the Windows file dialog which then appears, the name, without its extension, must be entered. The extension *.xml is appended automatically.

7.2.6.3.4 Creating a new Ethernet channel

A new Ethernet channel can only be created in the configuration file and never directly in the imc STUDIO: Setup.

- To create a new Ethernet channel, you either load an already existing configuration file or create a new one.
- Then, in the left portion, Configurations, you select the connector channel and the message in which the Ethernet channel is to be received.
- Afterwards, either click on the button in the title bar, or select the menu item *Configuration > New Channel*.

A dialog then appears:



Channel settings

The dialog has four portions. In the top portion, the channel name plus a designation for the newly created channel can be entered.

The portions Dataword 473 destination, Scaling 474 and Error handling 475 are described below.

Name:

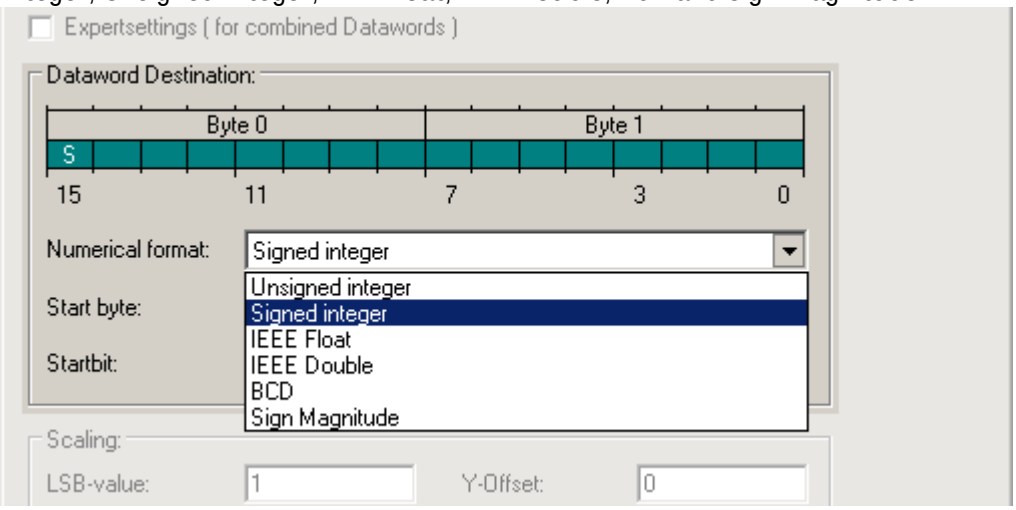
An arbitrary string of up to 255 characters. The general restrictions on channel names in in imc STUDIO apply. The name is assigned to this channel and is displayed in the Assistant interface. It is also used as the channel name in imc STUDIO.

Designation:

An arbitrary string of up to 255 characters, containing any arbitrary characters. The designation is assigned to this channel and is displayed in the Assistant interface. It is also retained in imc STUDIO as the channel comment.

7.2.6.3.4.1 Data word

A data word has different amounts of bits available depending on the numerical format selected. Bits not used are represented in the dialog by hatched boxes. Bits which are used are represented by filled boxes. Depending on the message protocol, the available numerical formats for the data word are Signed Integer, Unsigned Integer, IEEE Float, IEEE Double, BCD and Sign Magnitude.



Possible numerical formats for an ACRA message channel

Numerical format:

The available formats are Signed Integer, Unsigned Integer, IEEE Float, IEEE Double, BCD and Sign Magnitude. For all three numerical formats, the start byte can be selected.

Start byte:

In the pop-down list, the desired start byte can be selected. Due to applicable restriction, with the protocols IGH and HC, the start byte may not exceed 300 and with the message protocol ACRA it may not exceed 1500.

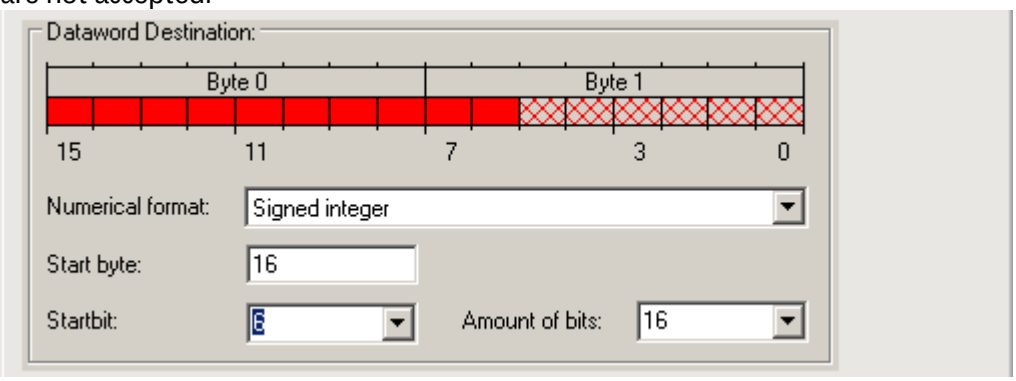
Start bit:

The start bit can not be selected for the numerical formats *IEEE Float* and *IEE Double*.

Number of bits:

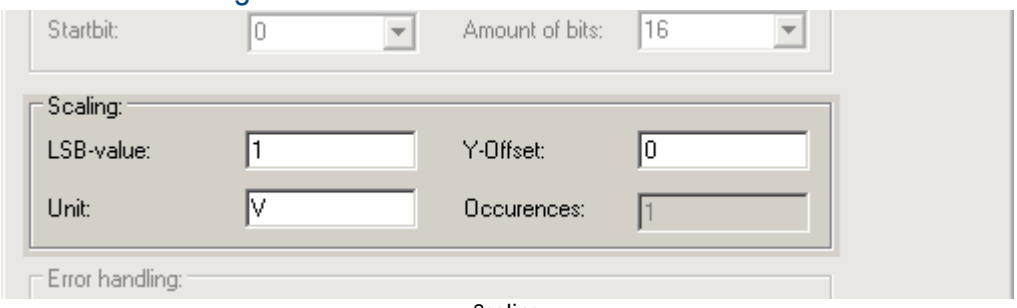
In the numerical formats *Unsigned Integer*, *Signed Integer*, *BCD* and *Sign Magnitude*, the number of bits can be selected.

If the sum of the start bit and the number of bits is greater than the number of bits available, the color of the marking in the bit boxes changes from green to red. In this case, the Ethernet Assistant settings are not accepted.



Impossible settings are marked in red color

7.2.6.3.4.2 Scaling



Scaling

LSB-value:

The LSB value specifies the factor by which the actual value is multiplied. Along with integer values, it is also possible to enter values in Exponential format such as 1.02e+3 or 1.02e-02.

Unit:

The signal's unit.

Y-Offset:

Here you can enter an offset which generates the ultimate result in conjunction with the LSB value.
Result = Value x LSB-Value + Offset

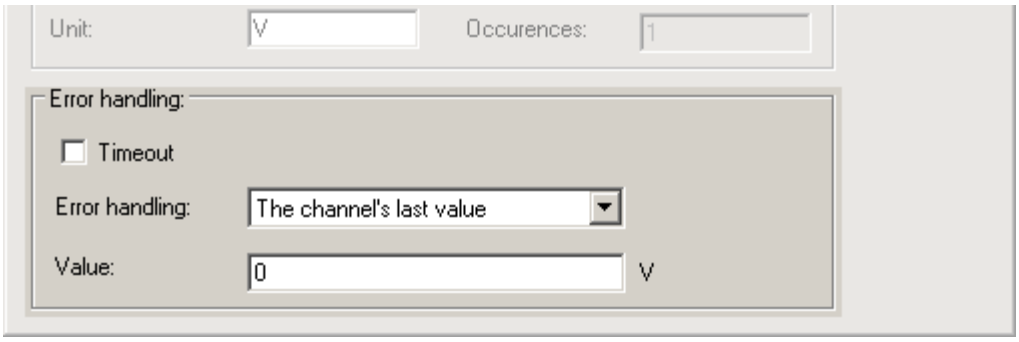
Occurrences:

Only for the message protocol ACRA, defining how many values are accepted in a message.

7.2.6.3.4.3 Error handling

Here is where you define the error handling procedure which takes effect in the case of a time limit violation.

If the box Timeout is not checked, then the setting in effect is by default The channel's last value as the error handling.



Error handling

Error handling:

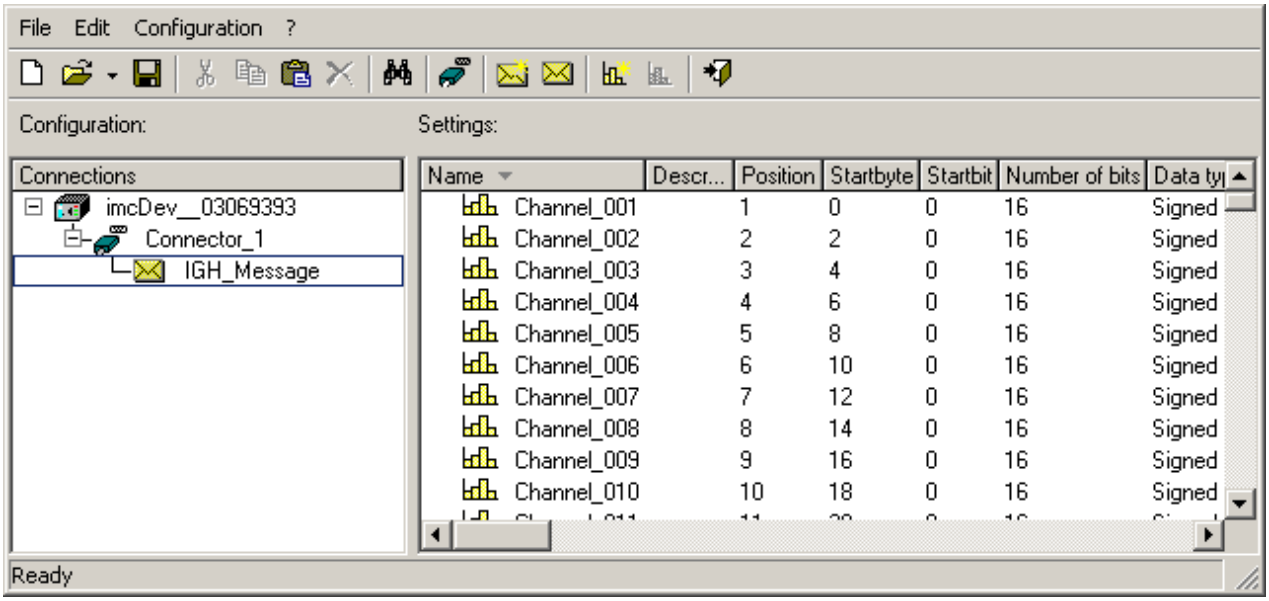
The available choices are *The channels last value*, *Substitute value* or *Ignore message's channel*.

Value:

A substitute value which can be recorded in the case of a time limit violation.

7.2.6.3.5 Configuration File List - Settings

If the Ethernet configuration file list was filled with Ethernet channels for the desired connector channels and their messages, the list may look like the following:



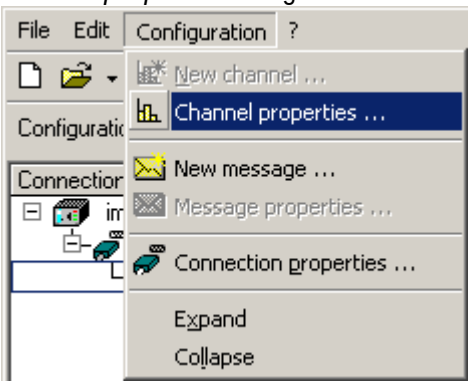
Configuration file list

All Ethernet channels can be ordered as desired by clicking on the corresponding column.

Each icon stands for a channel definition. After the icon the Ethernet channel's name, description etc. is displayed.

7.2.6.3.6 Editing an Ethernet Channel

To edit an existing Ethernet channel, select a channel and then select the menu item *Configuration > Channel properties* or right click to the selected channel.



Open channel properties

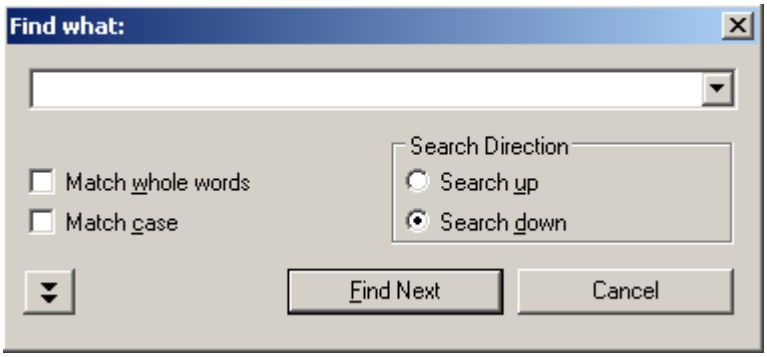
7.2.6.3.7 Cutting, copying, pasting or deleting an Ethernet channel

To cut, copy, paste or delete an Ethernet channel, 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 channel.

7.2.6.3.8 Searching for an Ethernet channel

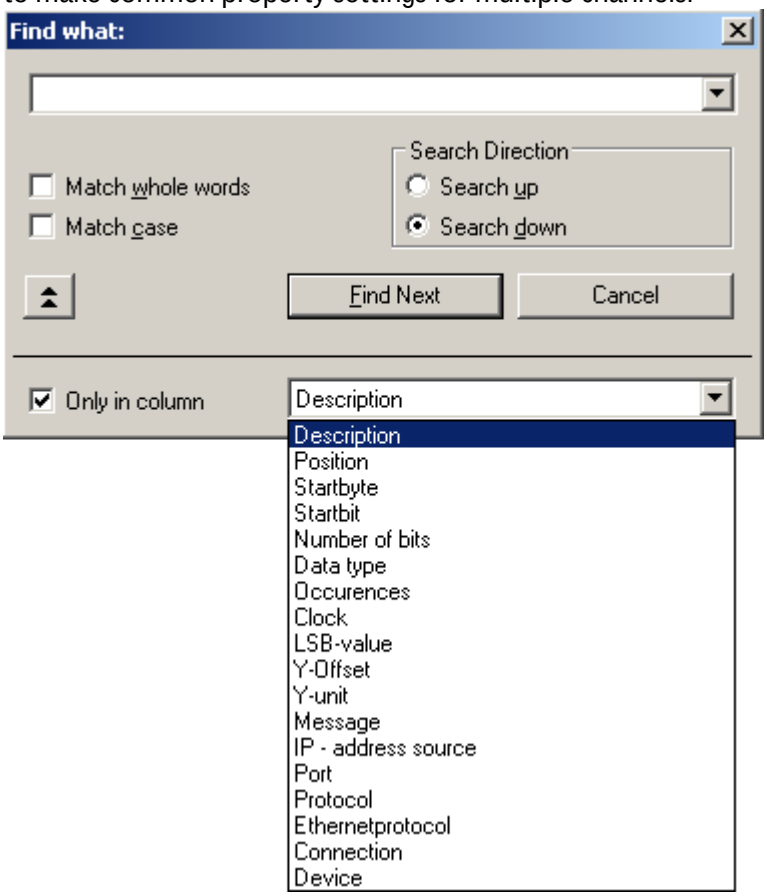
The context menu, called by right-clicking the mouse, also contains an item for the function Search and select. Using this function, you can search for specific Ethernet channels directly.

The Find function is available from either the *Edit* menu or by pressing the combination of keys *Ctrl+F*.



Searchning for an ethernet channel

Additionally, the option Only in column, which is accessed via the double-arrow button, offers the ability to make common property settings for multiple channels.



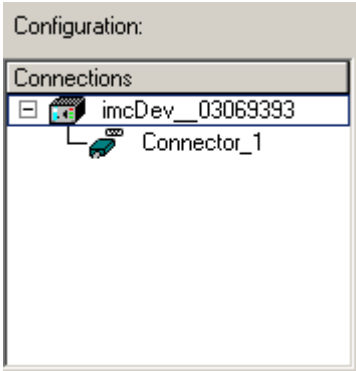
Search in column

7.2.6.4 The configuration

The next sections describe the left portion of the Assistant, which is designated Configuration, since everything which is set in this half is saved with the imc STUDIO experiment's configuration. In the simplest case, an Ethernet channel is added from a configuration file after the Assistant is started. When the Assistant is cllosed, this channel is adopted in imc STUDIO. How this is done in practise is described in detail below.

7.2.6.4.1 Configuration List

In the (empty) Configuration List, all imc devices along with the Ethernet interfaces are listed. The image below provides an example:



Configuration list

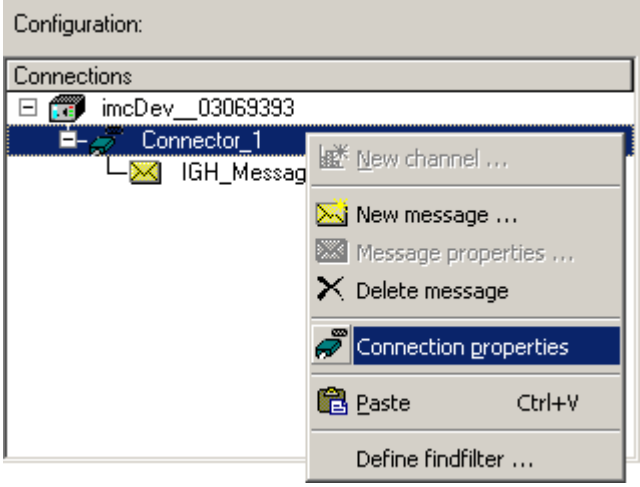
The icon represents a measurement device. Next to it is the device name.

The icons connected to it with lines symbolize the Ethernet connector channels and are numbered.

Important: A maximum of 500 channels per connector can be administered.

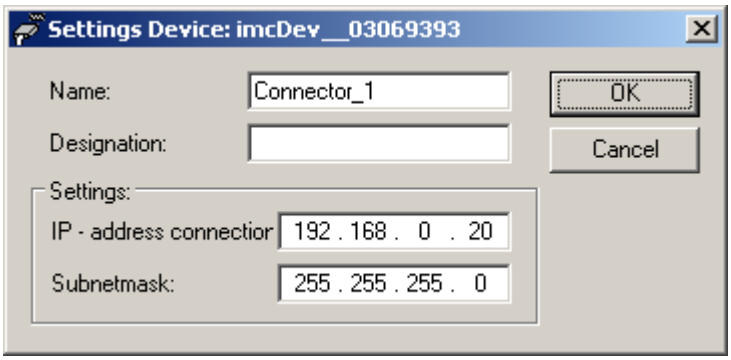
7.2.6.4.2 Connection settings

From the context menu called from over one of the connector terminals in the configuration list, or via the menu item *Configuration* > *Connection properties*, you reach a dialog in which the properties of the connector terminal are configured.



Connection properties by right click to connection

In this dialog the IP-address and the connector terminal's subnet mask can be set.



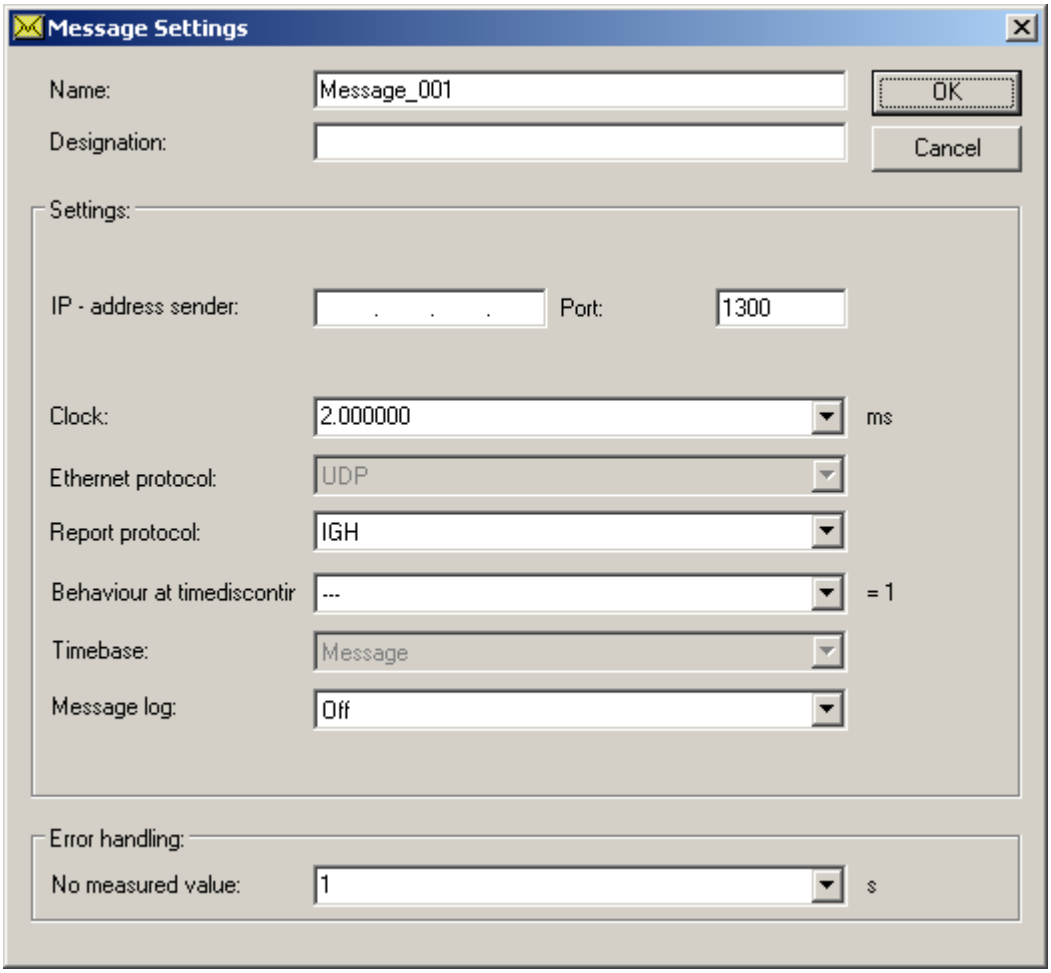
IP configuration

7.2.6.4.3 Creating a new Ethernet message

A new Ethernet message can only be created in the configuration file and never directly in the imc STUDIO: Setup.

- To create a new Ethernet message, select the connector channel in the left side (*Configuration*) of the Assistant on which the Ethernet message is to be received.
- Afterwards, either click on the button , or select the menu item *Configuration* > *New Message*, or select the corresponding item in the context menu.

A dialog appears:



Ethernet message settings

Important Note

On a connector channel, only messages having the same message protocol (IGH, HC, ARAC) can be defined and recorded.

Name:

An arbitrary string of up to 255 characters. The general restrictions on channel names in in imc STUDIO apply. The name is assigned to this channel and is displayed in the Assistant interface. It is also used as the channel name in imc STUDIO.

Designation:

An arbitrary string of up to 255 characters, containing any arbitrary characters. The designation is assigned to this channel and is displayed in the Assistant interface. It is also retained in imc STUDIO as the channel comment.

Settings:

IP address:

Here, the message sender's IP-address in the network is stated. The IP-address serves to identify a subscriber within a network.

Port:

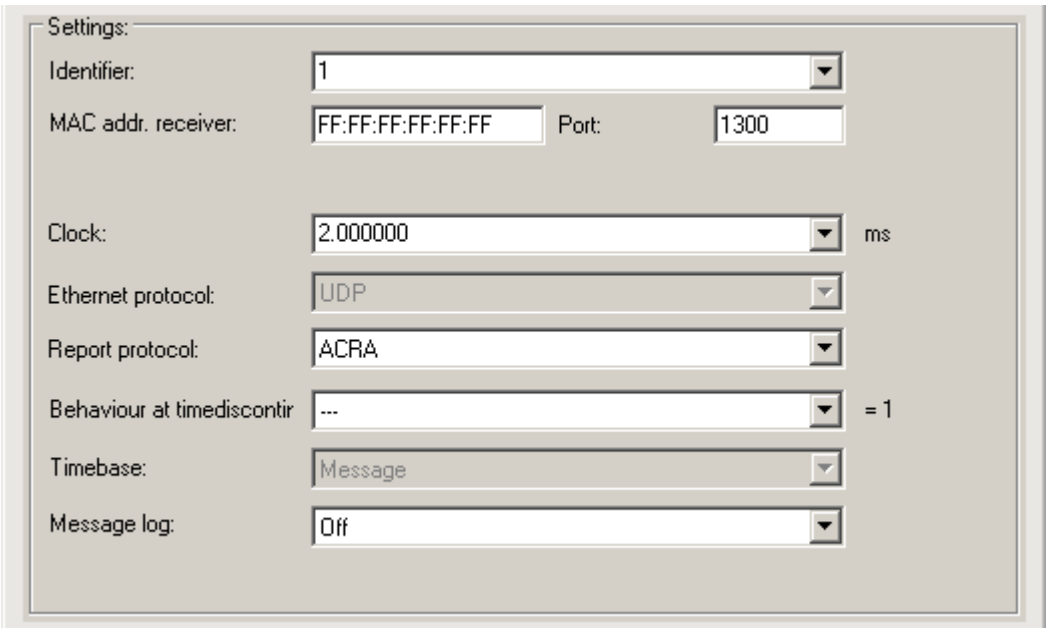
Here, the sender's port is entered.

Identifier:

This setting is only available with the message protocol ACRA. It lets you choose whether only messages having a specific identifier or all messages can be received.

MAC address receiver:

With the message protocol ACRA, the receiver's MAC address is specified here instead of an IP-address.



MAC address receiver

Clock:

For the message protocols IGH and HC, the message's clock is automatically set to 2 ms or 5 ms and cannot be changed. The clock can only be set by the user if the message protocol ACRA is selected. The clock is limited by the performance of the ethernet.

Report protocol:

Here you can select among the message protocols IGH, HC and ACRA.

Behaviour for leap in time:

In case of a leap in time it is possible to select among 32 available virtual bits, which is then set to 1. Using this virtual bit, the recording channels are linked in the imc STUDIO Trigger dialog, so that in case of a leap in time the recording can be stopped and restarted subsequently.

Timebase:

As the time base, either the device or the message can be used.

Message log:

When this setting is active, a new protocol channel is created in imc STUDIO which is similar to the protocol channel with the CAN-Bus. The protocol channel has the same name as the connector channel. This channel does not appear in the Ethernet Assistant.

Error handling:

No message since:

If the measurement device does not receive any measurement values during this number of time, the error handling routine to be defined in the channel settings is applied.

7.2.6.4.4 Changing channel parameters

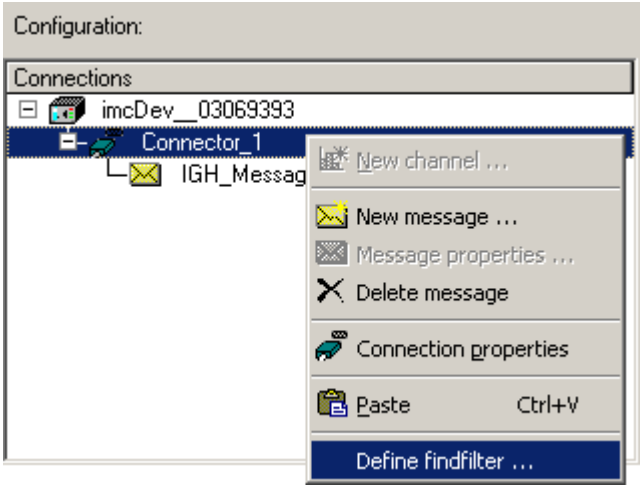
Channel parameters like channel name, sampling rate, measurement duration etc. are changed in the same way as with an analog channel.

All channels configured with the Assistant are also listed as field bus channels.

7.2.6.4.5 Defining search filters

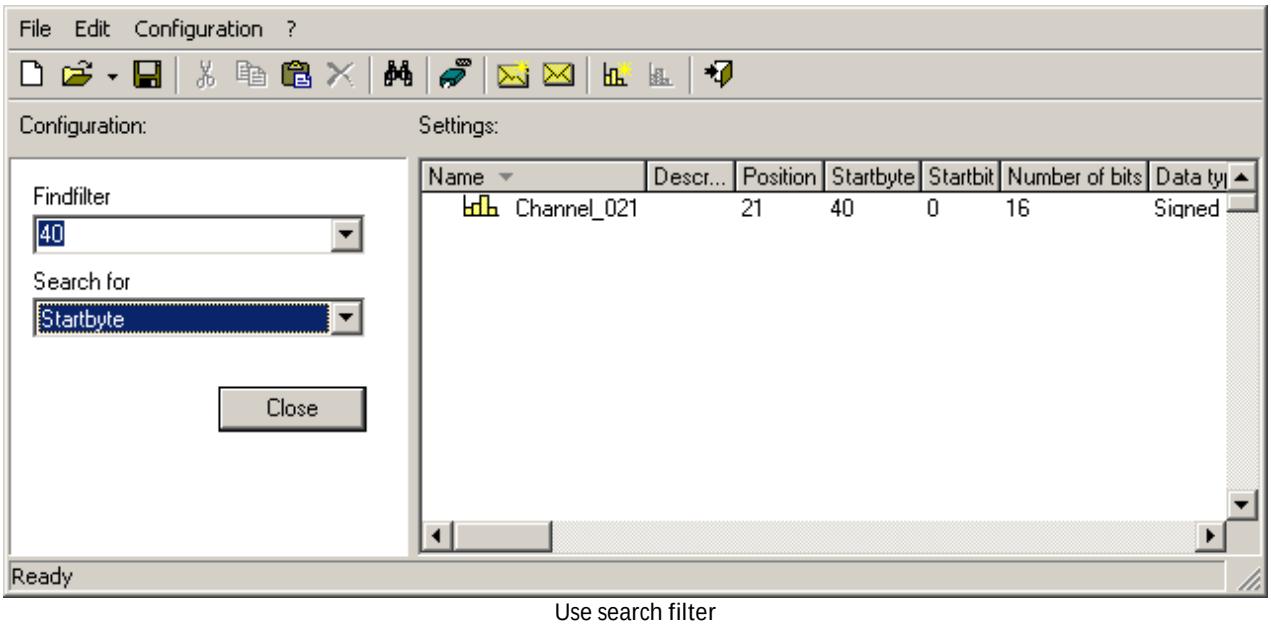
The context menu in the Configuration window contains an item for calling the Search filter dialog.

This filter supports the display and editing of a particular group's channels by means of wildcard searching.



Define search filter

Select the property from the pop-down list Search for and set the value in the box Search filter.



Use search filter

7.2.7 FlexRay-Bus Interface

What is the FlexRay-Bus?

FlexRay is a Field-bus system used for high-speed, deterministic data transfer in automobiles.

For configuration of imc devices connected with FlexRay, imc STUDIO offers the FlexRay Assistant 485.

7.2.7.1 Definitions

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 Field-bus 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
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] 483
Cycle-Multiplexing	This provides the ability to send different Frame types to the same slot of different cycles. See also [1] 483
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 Field-bus 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* 497 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 Field-bus 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

7.2.7.2 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

7.2.7.3 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 496, Version 2.0.1 (*.xml)
- Import of FlexRay-FIBEX-Plus 496 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 497-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

Observe the general Field bus notes regarding the operation methodology 338 of fixed sampling rates 340 versus timestamp 342.

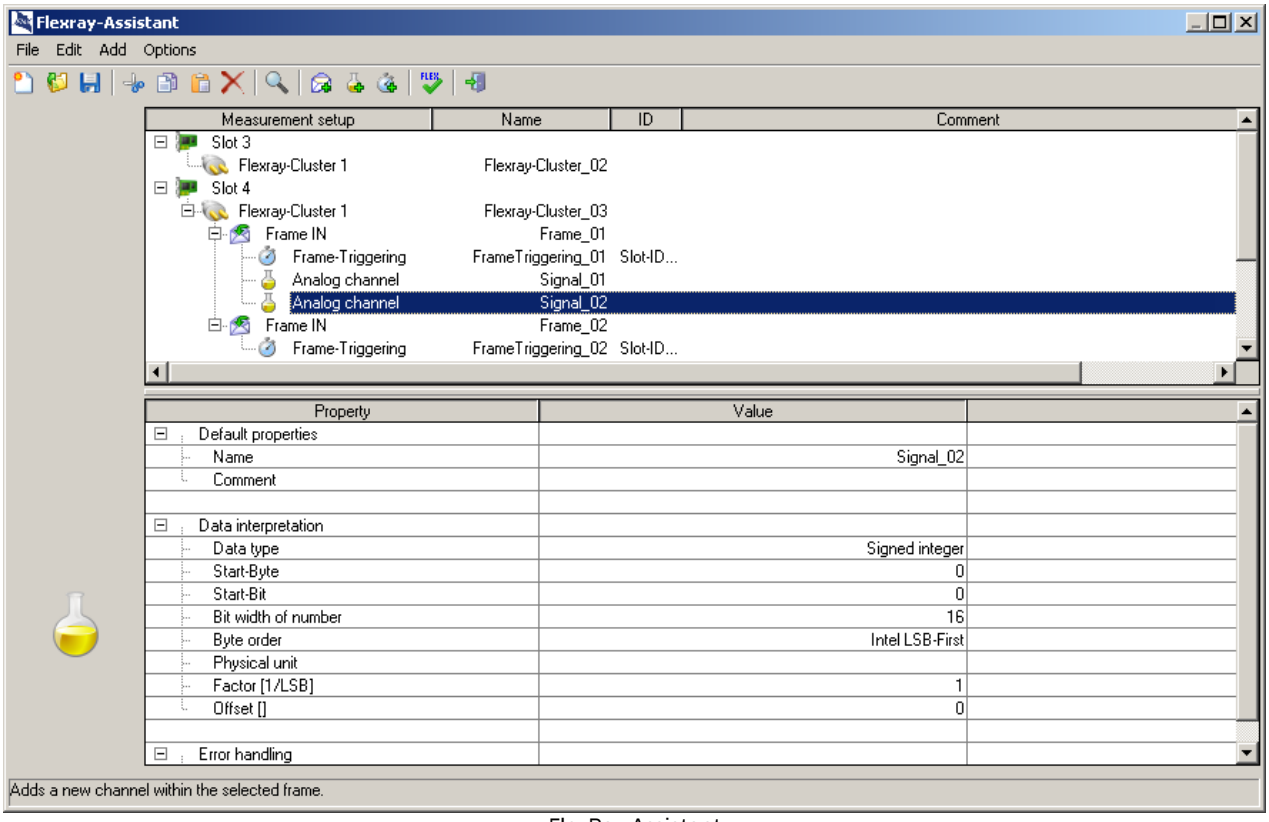
7.2.7.4 DSUB-9 Pin Configuration

Pin	DSUB-9	Pin	DSUB-9
1	n.c.	6	n.c.
2	BM Channel A	7	BP Channel A
3	GND	8	BP Channel B
4	BM Channel B	9	n.c.
5	GND		

7.2.7.5 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

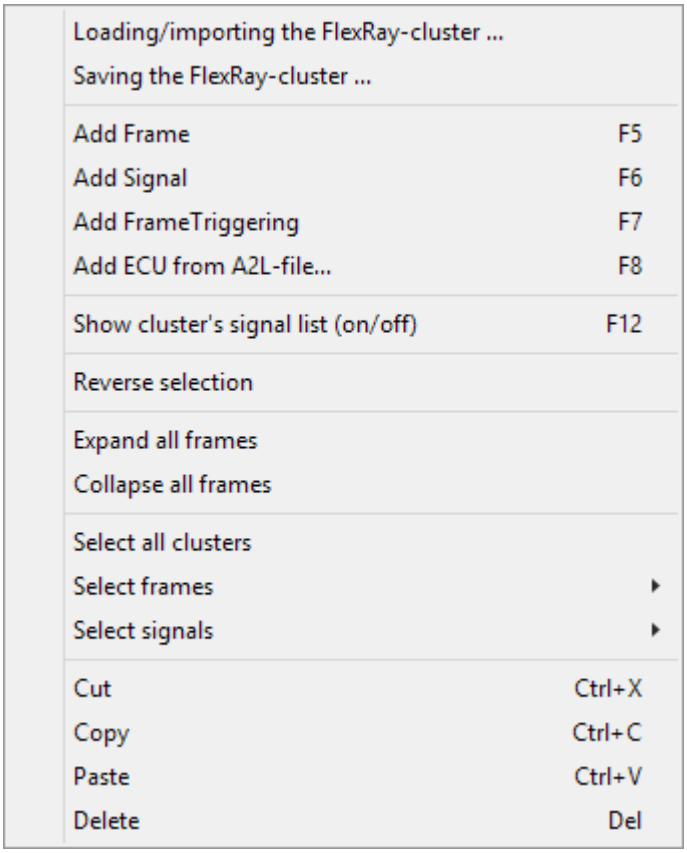
7.2.7.5.1 Overview Window

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

7.2.7.5.2 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.

7.2.7.5.3 Items in the main menu

File

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.

Two file extensions are supported.

- FRY: an imc proprietary format
- XML: FIBEX-XML, see Loading FIBEX-files 496

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

Copy, Paste, Cut, Delete

These commands only affect elements visible in the overview window.

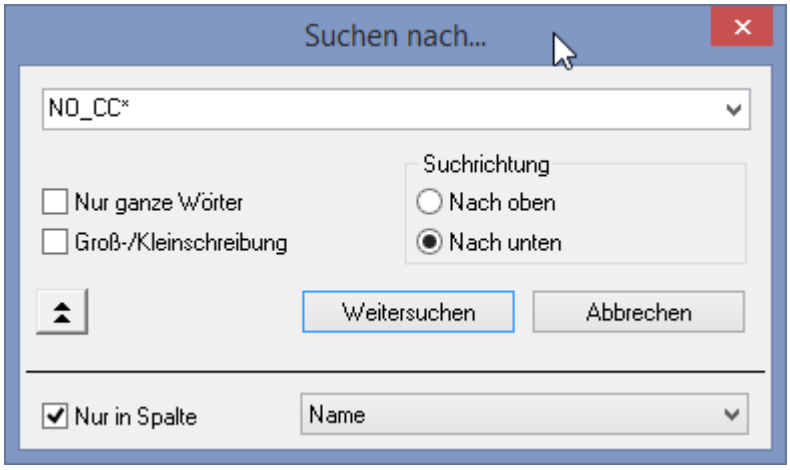
Frames, Frame-Triggerings and signals can only be copied, deleted and pasted.

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

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 497.

View

Display tree diagram und *Display the cluster's signal list*

Toggling between the tree diagram and the list of signals.

Configuration

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

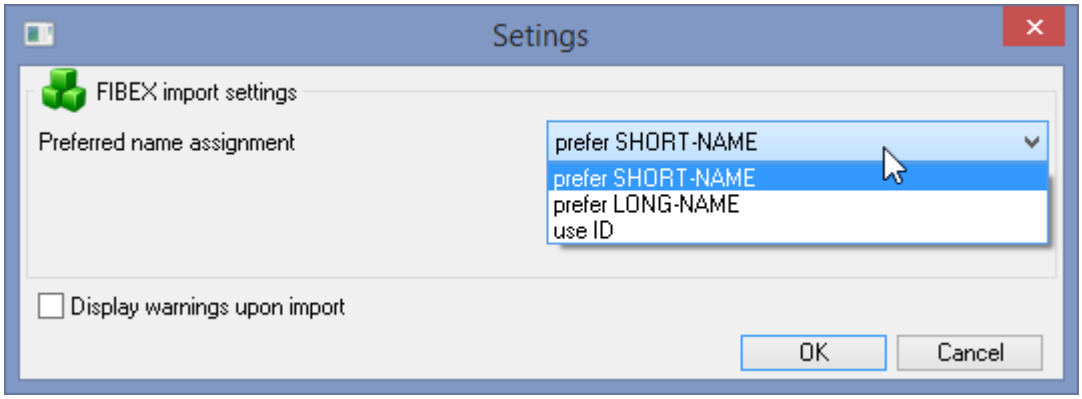
Font...

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 489.

To display warnings upon importing, activate the corresponding option.



Options menu: Settings

7.2.7.5.4 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 483.

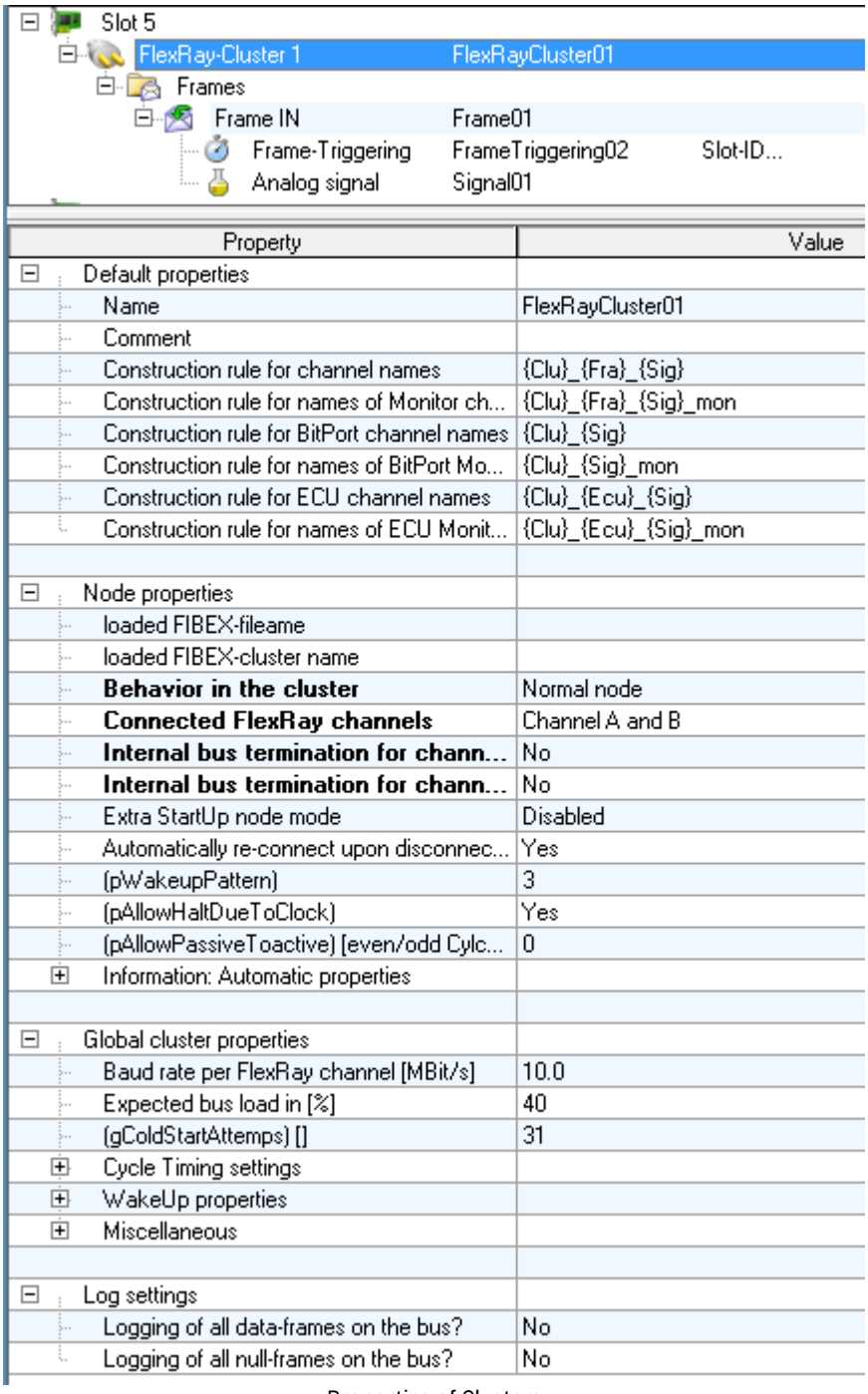
Multiple frames can be created in a cluster.

Construction rule for names of channels

The channel names can be composed from the name entries with placeholders: {Clu} for Cluster, {Frm} for Frame 491 and {Sig} for Signal 493 and ECU-channels 497.

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*"



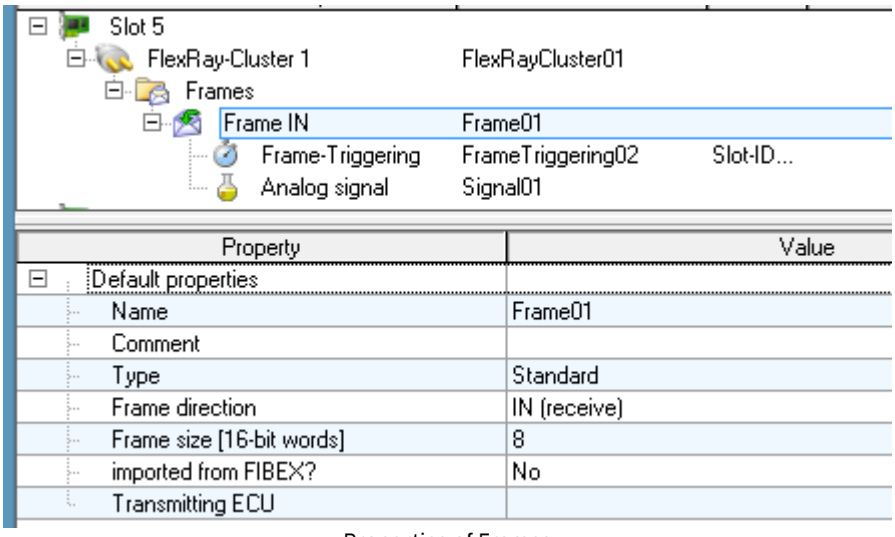
Properties of Clusters

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.

7.2.7.5.5 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

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

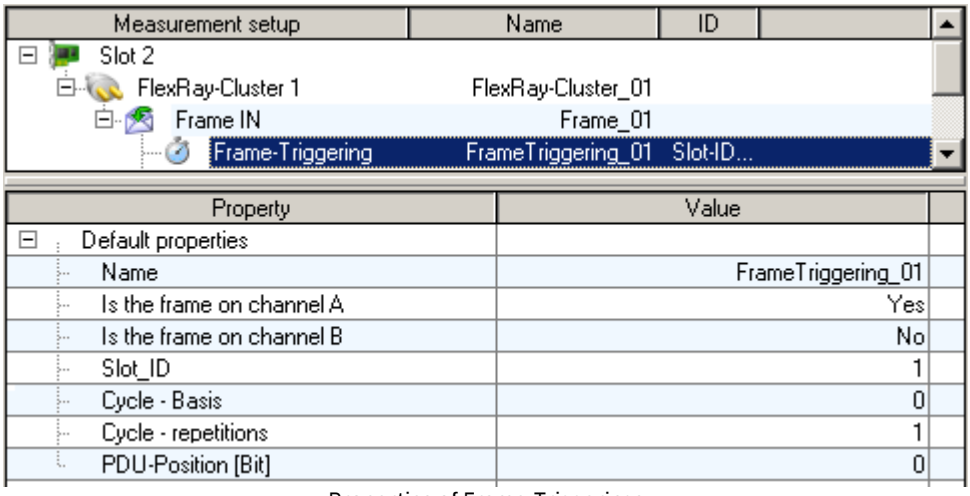
Designation of the ECU if available

7.2.7.5.6 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.

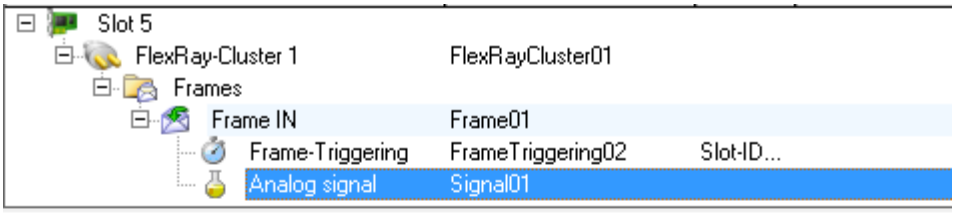
The separation of frames and the triggering is due to the fact that it allows one to specify multiple points in time at which to send / receive a frame, without needing to define (enter) new frames with all the associated signals each time.

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.



Properties of Frame-Triggerings

7.2.7.5.7 Properties of Signals



Property	Value
Default properties	
Name	Signal01
Resulting channel name	FlexRayCluster01_Frame01_Signal01
Create Monitor channel?	No
Comment	
Fibex SHORT_NAME	
Data interpretation	
Data type	Signed integer
Start-Byte	0
Start-Bit	0
Active/Passive Status	active
Bit width of number	16
Byte order	LSB-First
Physical unit	
Factor [1/LSB]	1
Offset []	0
displayed value range maximum []	32767
displayable value range minimum []	-32768
depending on multiplexer	<none>
Error handling	
Handle timeout error	No

Properties of Signals

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 489 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 489.

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 type

This sets the signals type, or rather, how the data on the bus are interpreted. The following data types are possible:

- *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.

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

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.

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	Variable definition
$y = f \cdot x + a$	$x = \frac{y}{f} - \frac{a}{f}$	y - physical value x - integer on the Field-bus f - factor a - offset

Example

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 V

Supposing the following integer were imported from the bus,

x = 13483 LSB

it would correspond to a voltage of

y = f * x + a = 12.057 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

This settings option determines whether error handling of any timeout error would be performed.

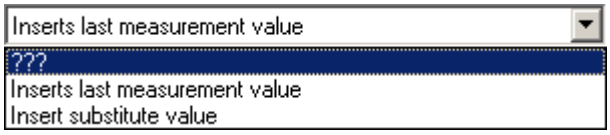
The term timeout-error refers to a case where no measured value is received from the field-bus after a certain period; error handling can be set to deal with such an occurrence.

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.

7.2.7.5.8 Loading FIBEX-files

Only a FIBEX file having FlexRay-Cluster description may be imported. Files with Field-bus 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

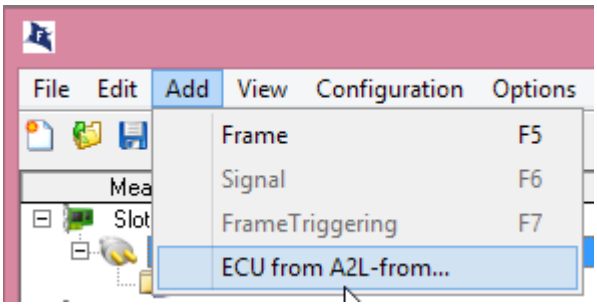
After loading a FIBEX-file, the following issues/ settings should be considered, since the imported FIBEX does not contain any information on the imc measurement system.

- 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 497.

These settings can be found in the properties of each cluster. To make them easier to find, they appear in bold letters.

7.2.7.5.9 Loading A2L files (XCPoFlexRay)

As of the version imc STUDIO 5.0R3 as of March, 2016 , XCP over FlexRay is supported.



The Import Assistant for A2L files displays all the available names and their associated address and comment.

7.2.7.5.10 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.

7.2.7.5.11 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.

7.2.7.5.11.1 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 497
- In the cluster-settings, set the Node-mode ("behavior of Cluster") to "Normal node".

7.2.7.5.11.2 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 497
- 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 497.

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 497).
- 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.

7.2.8 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. field busses)

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 DEVICES, 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

The Application Module Manual describe the use and initiation of the application module, such as the RS-232, RS-485, RS-422 and the RoaDyn Interface.

7.2.9 imc HiL: Hardware in the Loop

The imc HiL hardware is a supplemental component for the imc CRONOS measurement systems. imc HiL unites the functionalities of simulation, data capture, and open- and closed-loop control in a single compact system. The hardware basis for this component is a processor embedded in the measurement device, which is exclusively reserved for running MATLAB Simulink® models (Simulink Real-Time applications).

On the software side, the MATLAB Simulink® library is supplemented with imc connection blocks which represent an interface to the imc devices. By means of this expansion, it is possible to create Simulink applications in the customary way and to transfer them to imc devices. The interface for this purpose is the imc device software (imc STUDIO), in conjunction with the imc HiL Assistant, which is connected with all other imc components.

Highlights

- imc HiL integrates existing MATLAB Simulink® mdoels (Simulink Real-Time)
- Real-time simulation of application situations with direct feedback to the object under test
- Provides a dedicated processor integrated into the measurement system:
  - o CPU: Intel Atom
  - o Clock: 1.1 GHz
  - o Memory: 512 MB
- Expanded MATLAB Simulink ® library representing an interface to the imc devices
- imc STUDIO provides all settings and configurations for data acquisition and subsequent processing of signals within imc CRONOS measurement systems
- Support of tunable parameters
- No separate MATLAB license for the target system required

Prerequisites

You can download the current data sheet (PDF) from our website. This data sheet specify the prerequisites to the Developer (User)-PC of the MATLAB / Simulink models. You will also find the required licenses and the supported MATLAB versions and more details.

Reference

The imc HiL device manual describe the use and initiation of this interface.

7.2.10 J1587-Bus Interface

What is the J1587-Bus?

The J1587-bus is a special variety of a serial RS485 connection, which is deployed particularly in the field of freight motor vehicles and has been standardized by the SAE. The imc measurement system can be equipped with a J1587-Bus interface upon request. This enables the system to capture data along the J1587-Bus in parallel with other field bus systems (such as the CAN-Bus) or with analog measurement data, and to subject the various data jointly to data processing.

All channel definitions as per the J1587 standards are saved in a database.

Using the J1587 Assistant 508, this database can simply be edited and channels can be selected for data capture. The J1587 Interface offers the following features:

- One J1587 Bus connection per device
- Parallel operation with other Field-bus systems (e.g. CAN-Bus)
- Complete implementation of the SAE J1708/J1587 Standards in the capacity of a data capture and analysis system
- All data for defining J-Bus measurement channels are pre-defined in a database. All that is necessary is to make selections in the Assistant. This makes it possible to quickly configure measurements in a manner not prone to error.
- All channels which are encoded in binary or as (ASCII) text (PIDs), and all error and diagnosis codes can be captured and processed (PID, SID, Common SID), including channels which only deliver data upon request.
- Binary codes can be translated into understandable message texts.
- Every measurement value, error or diagnosis code can be assigned a time stamp.

The discussion in this chapter assumes that the reader is familiar with configuring analog and digital channels with imc STUDIO and with SAE J1587 standards.

Note

Observe the general Field bus notes regarding the operation methodology 338 of fixed sampling rates 340 versus timestamp 342.

7.2.10.1 Messages

According to J1587 protocol, a transmitted message consists of an MID (message identifier), a PID (parameter identifier), the data and the checksum. Messages may not exceed 21 Bytes in length, including the checksum.

Simple messages

There are different kinds of simple messages:

- Messages with one data word (MID, PID, data word, checksum)
- Messages with two data words (MID, PID, Dataword 1, Dataword 2, checksum)
- Messages with n data words, where n is transmitted after the MID and PID. (MID, PID, number of data words , Dataword 1, ..., Dataword N, checksum)

The PID can consist of one or two Bytes. PIDs consisting of two Bytes are called extended PIDs. Single-Byte PIDs are PIDs with a value of < 255. Two-Byte PIDs (extended PIDs) have a value of > 255.

In order to recognize the start and finish of a received message, there must be a time interval corresponding to ≥ 12 bits between two consecutive messages.

Packed messages

Messages can also be transmitted in a packed format. This format is used when an ECU must simultaneously transmit multiple PIDs. The format for packed messages is defined as follows:

MID, PID 1, data words, PID 2, data words, ..., PID N, data words, checksum

7.2.10.2 Reception and Evaluating Messages

The end of a message is established by a timeout of 12 bits. Once the end of the message has been recognized, the message is evaluated by the firmware. The first step of this process is the checking of the checksum. If it is correct, the data words are written to the channels specified by the user, in compliance with the gain factor and offset.

Packed messages are checked piece-by-piece for relevant PIDs. After the checking, the PID is deleted from the message buffer along with its data word.

The following conditions apply:

```
PID <  128:            1 data word
PID >= 128 and < 192:  2 data words
PID >= 192 and < 255:  n data words, where n must be transmitted as the 3rd Byte
PID >= 256:            1 data word
PID >= 384 and < 448:  2 data words
PID >= 448 and < 511:  n data words, where n must be transmitted as the 4th Byte
```

7.2.10.3 PID Query

Many (P)PIDs are continuously transmitted on the J-Bus, while some (P)PIDs must be queried by specific messages to be sent.

7.2.10.3.1 Periodic Queries

Query parameters

Query interval:

The user specifies an interval for the periodic issue of a (P)PID query.

Minimum query interval:

A minimum size for the time interval between queries is needed in order to prevent the queried ECU from becoming overloaded. Only after this interval has elapsed may a new query be issued.

Number of query repetitions:

If the queried ECU does not respond, a repeat query is issued after the minimum query interval elapses. P(PID)-querying is performed for a user-specified amount of times or else until the ECU responds, whichever comes first.

Firmware query procedure

This is the imc devices procedure for issuing queries via the J-bus:

- For each ECU the queries are ordered by time base.
- All (P)PIDs having the same query interval are assigned to a common time base.
- Each time the query interval elapses, the requested (P)PIDs having the corresponding time base are transmitted until either the number of repetitions is reached or the ECU responds with the PID requested.
- Once one time base's queries are all carried out to the end, the procedure is applied to the next time base's queries.

Limits on periodic queries

General notes

The number of queries is limited by the minimum query interval and the priority which the measurement device possesses for transmitting.

Limitations due to the minimum interval:

If for instance an ECU may handle a query only every 200 ms, then only up to 5 separate (P) PIDs each with a query interval of 1 s, or one (P)PID with a query interval of 200 ms may be defined. Any additional queries specified would be ignored.

Also note that the user-defined amount of P(PID)-query repetitions is counting down when any P(PID) is queried -even when there are different P(PID)s- and so no particular P(PID)'s query will be repeated for the full amount of times.

Limitations due to imc measurement device priorities:

The minimum possible time interval between consecutive queries issued by imc unit depends on these factors:

- imc measurement device priority for issuing a query message
- the time needed for sending a query message
- the priority of the queried PID
- the time needed for sending the requested message

Below is a computation of the repetition rate based on conditions given as:

imc devices priority: 1

Length of query message: 5 Bytes

PID priority: 1

Length of requested message: 4 Bytes

Baud rate: 9600Bits/s

Priority 1: 2*(12+2*1) bits / (9600bits/s)

Message of length 5 Byte : 5*10 bits / (9600bits/s)

Sum: 13.75ms

7.2.10.3.2 Queries in Response to Triggers

Trigger events can also cause a query to be issued, which have a higher priority than periodic queries.

For these queries "time bases" are also set up, but these are run through only once after the occurrence of the appropriate trigger event. As a rule, one-time queries have priority. The "time basis" assigned to the trigger with the lowest trigger number is processed first.

7.2.10.3.3 Querying the ECU-Versions

The ECU-versions are queried after the first measurement start. This query also is assigned a time base. This time base has top priority and is processed in the same way as the periodic queries and those made in response to triggers.

7.2.10.4 Logging ECU-Versions - PID 234 and PID 243

To correctly interpret measurements it is necessary to know both the hardware and software versions of the ECU. For this purpose, the ECU's hardware and software versions are queried at each measurement and logged in a global channel named J1587_EcuVersion.

The ECU-versions are communicated as ASCII-strings, therefore this channel is a time-stamped ASCII channel.

Procedure

The ECU-versions are queried after the measurement is prepared (Download) and started (see chapter PID query 502). If the ECU responds to the query, the strings received are written to the global channel in a certain format.

These stored data are kept in the device's memory.

When the same measurement is started again, the ECU-versions are not queried again, but the memory in the device is shared. Since it's important to know at what point in time the versions were queried for the first time, the corresponding time stamp of the strings is not changed.

Format for stating the ECU-versions:

The ECU-versions are stated in the following format:

ECU software-version:	ECU-Name (MID) SW version string
ECU hardware-version:	ECU-Name (MID) HW version string

Special notes

ECU not responding to query:

If the ECU does not respond to the query, the text: "+++ No data received +++" will appear in the version-string.

Missing ECU-name:

If the imc measurement device contains a MID, which is not defined in the database, the text "UNKNOWN ECU" appears in the channel in the place of the ECU-name.

Insufficient memory for the version channel:

The memory for the version channel in the device is not used as a FIFO (like normal channels), but is a purely linear storage space. If it is insufficient for storing all version strings, the channel will contain the announcement "Error!: RAM buffer time not enough for this type of data". In such a case, the RAM-buffer-time must be increased.

Time-stamp:

Since the version strings' time-stamp is never changed, a negative trigger time can appear in the curve window when the X-axis is labeled in terms of relative time. But in the representation style "Absolute hours, minutes", the point in time of the first data acquisition with the versions is shown.

Time-stamp and multi-event:

If the Version-channel is set as the trigger target for a trigger with multiple events, the relative trigger times for each release of the trigger are located ever further in the negative range.

The reason for this phenomenon is the special way this channel is handled. The ECU-versions are queried at the first trigger event and logged in the channel. At the next trigger event, there is no new query. Instead, the already stored ECU-version data is written to the channel again. Also, there is no change in the absolute trigger time. The original absolute trigger time is used at all subsequent trigger releases.

If the "Absolute hours, minutes" option is selected, all events have the same absolute start time, in contrast to the case of relative-time representation.

7.2.10.5 Multi-section Messages - PID 192, PID 448

According to the J1587 standards, only messages of maximum length 21 Bytes may be sent.

If a message containing more than 21 Bytes is to be transmitted, it can be done by means of PID 192. In this way, the message to be sent can be split up into multiple partial messages of maximum length 21 Bytes.

We will refer to the partial messages as sections.

Together with PID 192, the corresponding (P)PID must be specified. Ideally, the sections are transmitted one after the other and numbered. So that the receiver can reconstruct the total message from the sections, the protocol prescribes that information on the running number of the currently transmitted section and the total number of sections is to be encoded in a certain Byte.

For the J-bus firmware to reconstruct the total message from the received sections, intermediate storage in which the sections are joined up is required.

Since multiple (P)PIDs can be transmitted using PID 192, multiple intermediate storage spaces are needed (for instance, Section 1 of (P)PID A is first received, then Section 0 of (P)PID B). These are identified by the MID and the (P)PID.

For the purpose of transmitting extended PIDs (PID > 255), the PID 448 is provided. Its manner of operation is the same as that of PID 192.

Procedure for receiving a PID 192/448

Procedure for receiving a (P)PID with the PID 192/448

- The intermediate memory space is empty, no PID 192 has been received
- A PID 192 with the Section 0 is received
- A free area of memory is located
- The message is processed as if a normal (P)PID had been received
- The message is written to the intermediate memory. MID and (P)PID, as well as the reception time, are also stored in this memory for identification purposes
- A PID 192 of the same (P)PID of the same ECU, having the next section, is received.
- The corresponding intermediate memory area is found and the section is appended after appropriate processing, and the reception time for the section is noted in the record
- The last section is received
- The total message is released for further processing and evaluation.

Procedure for receiving multiple (P)PIDs with the PID 192/448

The procedure is the same as that for the reception of one (P)PID, but the sections are recorded in a different intermediate memory area and with a different identification.

Transmission errors and limitations

Number of intermediate memory areas:

50 intermediate memory areas are available for processing the PID 192.

Loss of a section or disregard of the sections' order:

The order of the sections absolutely must be followed. If, for instance, the 2nd section is followed by the 4th, the corresponding intermediate memory area is cleared.

If Section 0 of an incomplete message already held in intermediate memory is suddenly received, all previously received data will be deleted and Section 0 will be stored in their place.

Exceeding the number of intermediate storage areas:

If an additional (P)PID having Section 0 is received along with the PID 192 and all intermediate memory areas are occupied, the data in the memory area containing the first message stored are deleted and the new (P)PID is stored in their place.

7.2.10.6 Diagnostic Channel - (P)PID 194

If there is a malfunction in one of the truck's systems, e.g., the battery charge drops below a certain value, the J1587 standards provide PID 194 as a means for reporting this malfunction.

In such a message, the corresponding (P)PID must be included for identification. Also, certain combinations of bits in the corresponding data word can indicate which error (type) has occurred, and this bit-combination can also indicate whether an extended PID (PID > 255) or its SID is involved.

The user has the following options for evaluating PID 194:

- Evaluation of all messages sent with PID 194
- Evaluation of the messages from specific ECUs
- Evaluation of all PIDs/SIDs/PPIDs/PSIDs of specific ECUs
- Evaluation of one or more PIDs/SIDs/PPIDs/PSIDs of specific ECUs

Once a certain option has been selected, a virtual bit can be linked to the occurrence of this event.

Display of (P)PID 194

The data received are evaluated according to the option selected and written to a measurement channel with the name J1587_FaultCodes.

The error messages are written out explicitly. This text contains the following information:

- Name of the ECU
- PID/PPID/SID/PSID
- Name of the PID/PPID/SID/PSID
- Error code
- Active/inactive
- Number of times the error occurred
- MID as a value
- PID/PPID/SID/PSID with the following code:
  PID : 0
  PPID : 1
  SID : 2
  PSID : 3
- PID/PPID/SID/PSID as a value
- Data as a value

A sample explicit error message:

"E-ECU, PID, Engine Load, Abnormal update rate, Active, 5 (128, 0, 92, 9, 1)"

In order for this information to be conveyed, the names of the ECUs, as well as the names of the PID/PPID/SID/PSID must be defined in the database.

In order for the names of the ECUs and the PIDs/PPIDs/SIDs/PSIDs to be displayed, they are retained in memory by the firmware and accessed from there when a diagnosis message is received.

Common SIDs / PSIDs

According to J1587-standars one distinguishes between SIDs and Common SIDs. SIDs with a value greater than 150 are referred to as SIDs. These SIDs have the same name for all ECUs, but the SIDs with a value < 150 can have a different name for each ECU.

The firmware recognizes the difference and therefore accesses different name buffers.

Setting of virtual bits by error codes

If a particular error is linked to a virtual bit, the bit is set from zero-state to one, in order to provide the possibility of triggering by error. Once this bit has been set, the firmware does not reset it.

Missing definitions in the database

Missing ECU names:

If the imc measurement device contains a MID which is not defined in the database, the text "UNKNOWN ECU" appears in the channel instead of the PID/PPID/SID/PSID-name.

Missing PID/PPID/SID/PSID names:

If device contains a PID/PPID/SID/PSID which is not defined in the database, the text "UNKNOWN PID/PPID/SID/PSID" appears in the channel instead of the ECU-name.

7.2.10.7 J1587-Bus Assistant

Configuring the J-Bus interface is performed with the help of the Assistant.

Start the Assistant from within the imc STUDIO software via the Ribbon *Setup-Configuration* > *J1587-Assistant*.

Note

This item is only offered in the menu if the selected device is equipped with a J-Bus connection.

- When first started up or when a new experiment is created, the Assistant is empty. The first step is to import the J-Bus data definitions from the database.
- The Assistant administers all the J-Bus data by slot and device. As mentioned previously, the measurement system can be equipped with multiple Field bus connections (slots) simultaneously. Also, an experiment may involve multiple J-Bus devices. This means that if more than one J-Bus is present in the current experiment, the device selection box in the Assistant's toolbar becomes accessible. This list box, then, enables the user to select the desired J-Bus interface (slot). The Assistant's operating environment is adapted accordingly. The Assistant only changes settings for the selected slot!
- If there is only one slot, the device selection box appears gray to indicate that it cannot be accessed.

7.2.10.7.1 Importing the Database

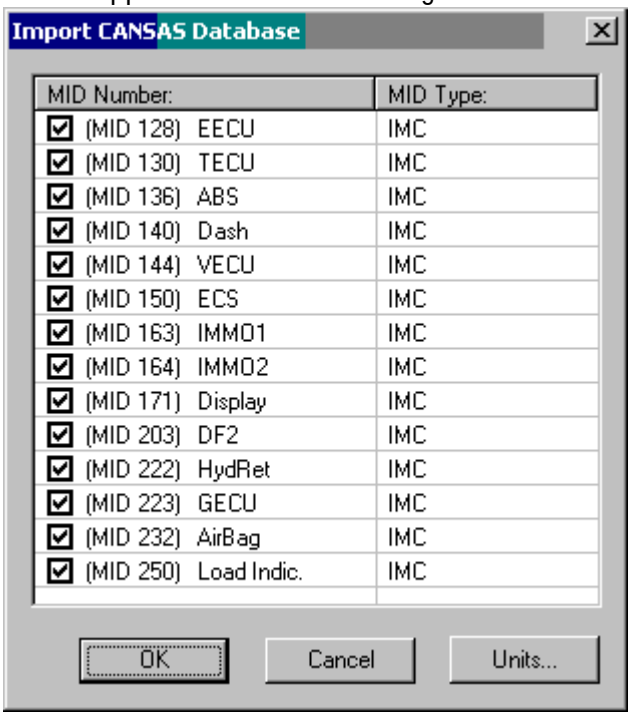
To start importing the database into the Assistant, select the menu item *File* > *Open DB*. Any configuration already open is deleted from the Assistant.

Note

If there are multiple J-Bus interfaces (slots), the database is only imported for the currently selected slot!

After clicking on the menu item, a file selection dialog appears for selecting the database file. After the database file is opened, the database is searched for all ECU's present and a selection list is made, since the data base can contain multiple ECU's from different manufacturers for a single MID.

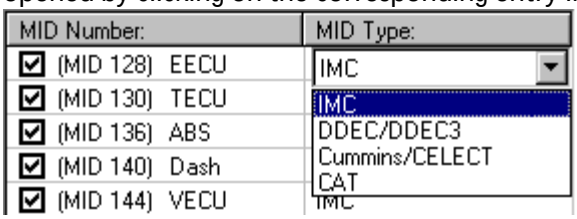
The list appears in a selection dialog as illustrated below:



Importing the database

ECU's are distinguished by their MID number and by their MID type (e.g. different manufacturers of an ECU with the same MID). All MIDs found are listed on the left side of the dialog. By clicking the checkbox at the extreme left of the list, an MID can be selected or de-selected for importation. The type of the MID selected appears on the right side of the list.

If multiple MID types are defined in the database for a single MID number, a list box for selection can be opened by clicking on the corresponding entry in the dialog.



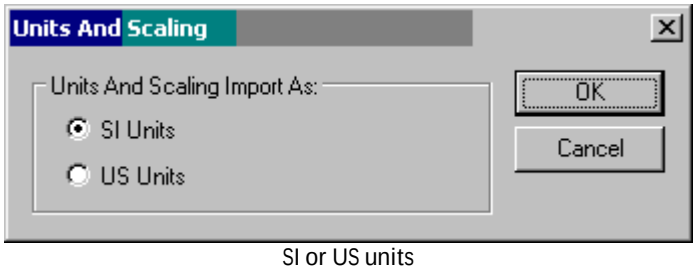
MID Type

Note

The list box for selecting the MID type only appears once the corresponding row in the right column of the dialog has been clicked on, and then only if more than one MID type is defined for a single MID number and if there is a check-mark in the corresponding checkbox at left.

Units

At the bottom right of the dialog there is a softkey called *Units* which accesses another dialog:



SI or US units

The choices here are SI and US units. The selection applies to all units and scaling factors imported from the database. Once the importation from the database has taken place, it is too late for changes! Pressing the softkey *OK* causes the setting made to be saved and the dialog closed.

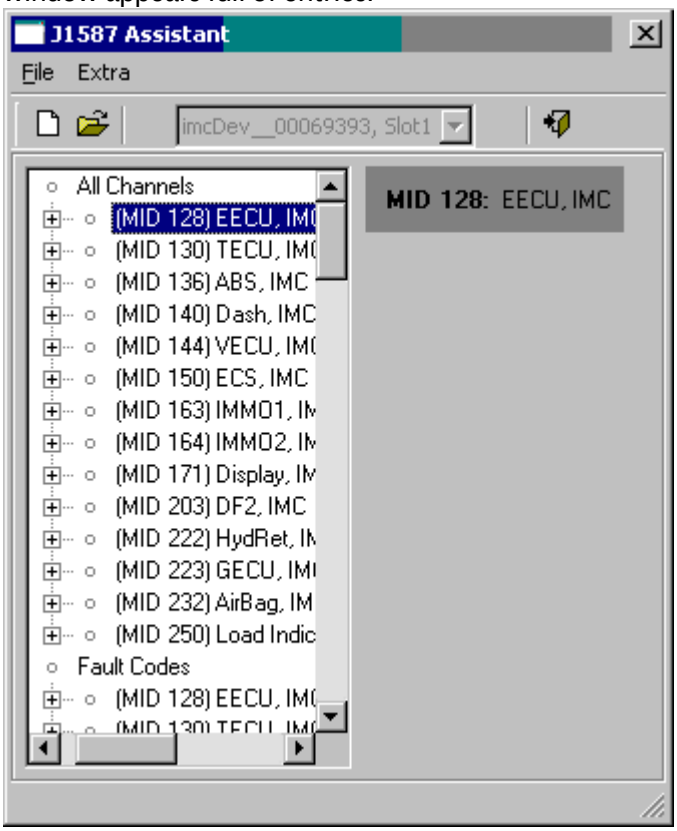
By pressing the *OK* softkey in the MID selection dialog, the database is imported.

Note

Pressing OK causes the current configuration to be deleted along with all channels included in it! Cancel has no effect on an existing configuration.

7.2.10.7.2 Main Window

Once importation form the database has been performed, the list on the left side of the Assistant's window appears full of entries:



J1587 Assistant

All information needed for setting up measurement channels is now available to the Assistant. Exact configuration of the bits and bytes isn't necessary; instead, one simply selects a certain channel or error code to be recorded. When the Assistant is closed, all measurement channels required are then created within the system.

7.2.10.7.3 J-Bus, imc STUDIO Channels and Data Formats

Data carried by the J-bus can belong to a variety of formats. They could be numeric data encoded in the binary system, they could be text data encoded in ASCII, binary digital (bitmapped) data or diagnosis and error messages. Some kinds of data are transmitted regularly, others only occasionally and some must even be specifically requested.

The imc unit, on the other hand, can process data encoded in binary or ASCII. Binary data can be recorded either equidistantly or with a time stamp, ASCII channels are time stamped as a rule.

Also as a general rule, the data ranges of the PIDs can be assigned to measurement channels in imc STUDIO. In the discussion below, the term "measurement channel" is used to denote the PID data range. Certain PIDs of special significance represent exceptions:

- Version numbers: The hardware and software versions of an ECU are transmitted via PID 234 and PID 243, respectively. For these version numbers, one extra measurement channel per device is created: *J1587_ECU_Versions*. The versions can be requested by imc STUDIO at the start of a measurement.
- Error and diagnosis codes: PID 194 transmits special diagnosis and error codes (PID, SID, Common SID). Since these messages can come from a variety of sources but only have a low data rate, all error and diagnosis codes (PID 194) are recorded in their own measurement channel *J1587_FaultCodes*. So that it is easier to read, this channel is encoded in ASCII, i.e. all messages are translated into plain text, as far as is possible.
- Bit-mapped PIDs: A number of PIDs contain messages which are binary bit-map encoded (discrete data, status messages). The bit-mapped data can be optimally encoded in the common, ASCII encoded channel *J1587_Protocol*. This helps to reduce the total number of channels.

These three measurement channels, in further discussions referred to as global channels, are automatically created by the Assistant for each device when importation takes place. All global channels are ASCII encoded. This setting cannot be changed.

Generally, the measurement data can be saved in their respective data formats. Additionally, binary data can be converted to ASCII and stored in ASCII channels. The reverse, from ASCII to binary, is not possible.

7.2.10.7.4 The List

On the left side of the Assistant's window, there is a list like a directory tree with all channels and error codes available for selection. Clicking on one of the entries in this list causes the corresponding settings dialog to appear on the right side of the window. In the settings dialog, a PID or an error code can be selected for recording. Channels or error codes activated for recording have a green check-mark next to their list entry. Activated channels which have been de-selected appear with a red cross-mark.

The list is always divided into two sections: in the upper part, all measurement channels appear under the heading All Channels. In the lower part under the heading Error codes, all error and diagnosis messages are to be found.

Right-clicking the mouse in the list box calls the context menu, which offers the ability to jump quickly between the two parts.

7.2.10.7.5 Measurement Channels (PID)

The channels in the list are ordered according to the following pattern:

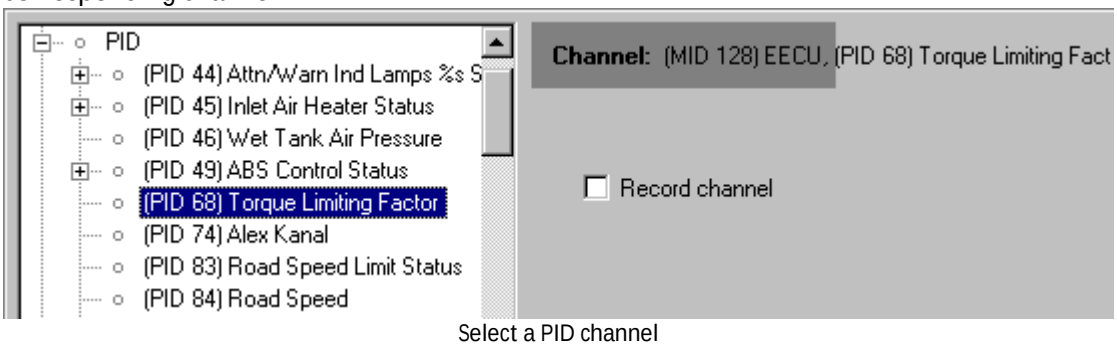
```
MID xxx
   |---- PID xxx
   |         |----- Channel 1
   |         |----- Channel 2
   |---- PID xxx
   |---- PID xxx
   ...
```

Channels are always below a PID in the tree structure. This signifies that a channel is always assigned to a particular PID. If a PID has more than one channel, the PID listing appears with a plus-sign, on which it is possible to click to expand the tree structure for that PID. This case usually occurs in connection with bit-mapped PIDs. Clicking on the PID only causes the PID properties to be displayed. The PID Properties dialog can be empty. Clicking on the entries of channels causes the respective channel properties dialogs to be opened.

If the PID cannot be expanded, only one channel can be defined for this PID. In this case, clicking on the PID calls the settings dialog, and the PID and channel properties are displayed together.

7.2.10.7.5.1 Selecting a PID Measurement Channel

By clicking on a PID or on one of the channels under it, a checkbox is called which appears on the right side of the Assistant window and which lets the user activate or deactivate data capture for the corresponding channel.

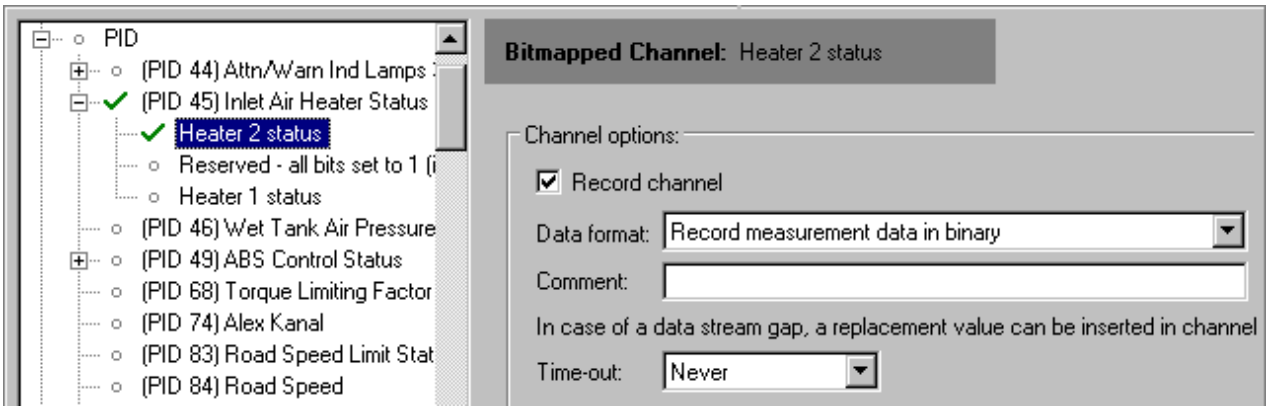


Select a PID channel

A green check-mark appears next to a channel's entry in the list when the option Record channel is activated. Also, the Channel Properties dialog appears on the right side (see illustration below). This signifies that the channel is activated for recording.

After the Assistant is closed, the channels involved are automatically set up in imc STUDIO. Before this happens, certain settings can be made for the PID and channel properties (see further below).

The names of the new channels created are formed by joining the MID names, PID names and bit-mapped name if appropriate. These are only default names and can be changed just like the names of any normal channel in imc STUDIO. Other channel properties such as sampling rate, measurement duration and so on can be changed there as well.

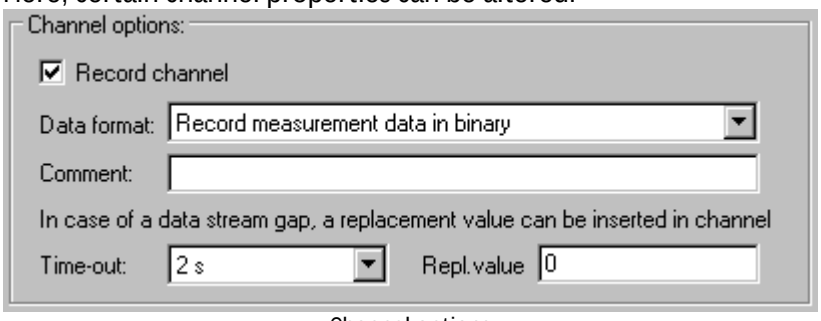


Channel options

The PID or channel properties dialogs are adapted to the particular PID and channel types. Thus, dialogs can be vary greatly in their appearance. However, the properties come in three categories, as represented by frames surrounding corresponding groups of dialog controls. Channel-options 513, Request options 514 and Bitmapped options 514.

7.2.10.7.5.2 Channel Options

Here, certain channel properties can be altered.



Channel options

Record channel:

This checkbox lets you specify whether the J-Bus channel involved is recorded or not.

Data format:

Sets type and format of the measurement channel. Normally, there are 3 choices offered:

Record measurement data in binary:

The data are recorded in their own binary measurement channel. If the PID in question comes with a Request, the measurement channel is set up as a time-stamped channel. J-Bus data in ASCII format can not be recorded in binary format. An error message appears if this option is selected.

Record measurement data as text:

The data are recorded in their own ASCII channel. If the J-Bus data are binary, they are converted to text. If the channel in question is bit-mapped, texts are assigned to the codes.

Record measurement data as text in a global channel:

No separate channel is created for these J-Bus data; instead, the data are recorded in the common (global) channel *J1587_Protocol* (ASCII format). In order to distinguish the J-Bus channels, the MID name, PID name and the bit-mapped name, if appropriate, are prefixed to each measurement value. If the J-Bus data are binary, they are converted to text. If the channel in question is bit-mapped, texts are assigned to the codes. This setting is to be preferred for bit-mapped PIDs.

Comment:

Optional channel comment. The channel comment is saved together with the measurement data and can be displayed using evaluation programs such as imc FAMOS.

Time-Out:

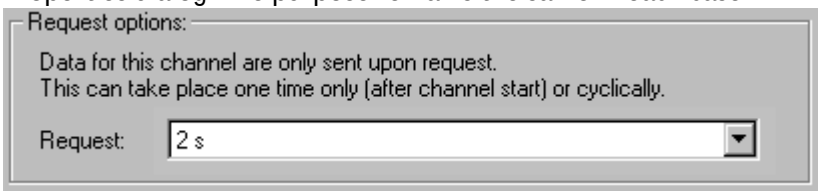
Activates monitoring for data stream gaps. If no new measurement value arrives during a time interval (of user-specified length), an Replacement value (see below) is written to the channel. Selecting *Never* causes the gap monitoring to be deactivated.

Repl. value:

This dialog control is only accessible if a Time-Out time interval has been specified. A replacement value can be specified here, which can be written to the channel in case of a drop-out. No unit need be specified; it is defined in the database.

7.2.10.7.5.3 Request Options

The request options are a PID property featured only by certain PIDs. These PIDs only transmit data upon "Request". For PIDs which come with only one channel, the Request options are indicated in the Channel Properties dialog. The purpose remains the same in each case.



Request options

Requests have two operational modes: either the Request is triggered by a one-time event, or the Requests are transmitted periodically by the measurement device at fixed intervals.

Which mode is active is selected via the list box:

One-time, after trigger start:

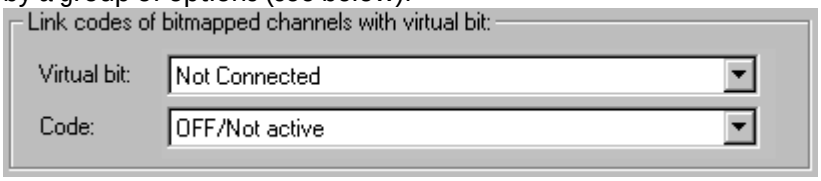
In this operational mode, the Request is transmitted only once after the data-capture channel is started, and then no longer. Only a new startup of the measurement channel can trigger a new Request.

Specified time:

Requests are transmitted cyclically with the repetition interval set. Note that this interval must not be set at a value too small to accommodate the bus load. After too many unsuccessful attempts, the Request is dropped. Refer to the section on queries in the Firmware-documentation.

7.2.10.7.5.4 Bitmapped Options

Some PIDs can carry data of the bit-mapped type. In this case, the Channel Properties dialog is extended by a group of options (see below).



Bitmapped options

Bitmapped channels hold discrete data (status messages) which are usually encoded as a number of bits (a code). These codes can be linked with a *virtual bit* by the imc measurement device. A virtual bit can be an event bit, by which, for instance, a trigger is released. Thus, the occurrence of certain bitmapped codes on the J-Bus can trigger an event in imc STUDIO. The link between the bitmapped Code and the virtual bit is set up using the Options box's two list boxes.

Virtual bit:

List of the devices 32 virtual bits are listed by name. If a virtual bit is selected, it is connected with the code in this options box. The virtual bit is set to 1 in case the code is detected and to 0 as soon as the code no longer is being received. A virtual bit's name can be set in imc STUDIO: Setup.

Code:

The Code with which to link the virtual bit is selected here.

Note

The virtual bits are available throughout the imc STUDIO system, for instance for the trigger or in imc Online FAMOS. In the J1587 Assistant, they can be used to signal bitmapped codes and error codes. Since this makes it possible to lose track of things, caution must be used when assigning virtual bits. For this reason, the J1587 Assistant does not support multiple assignment of virtual bits.

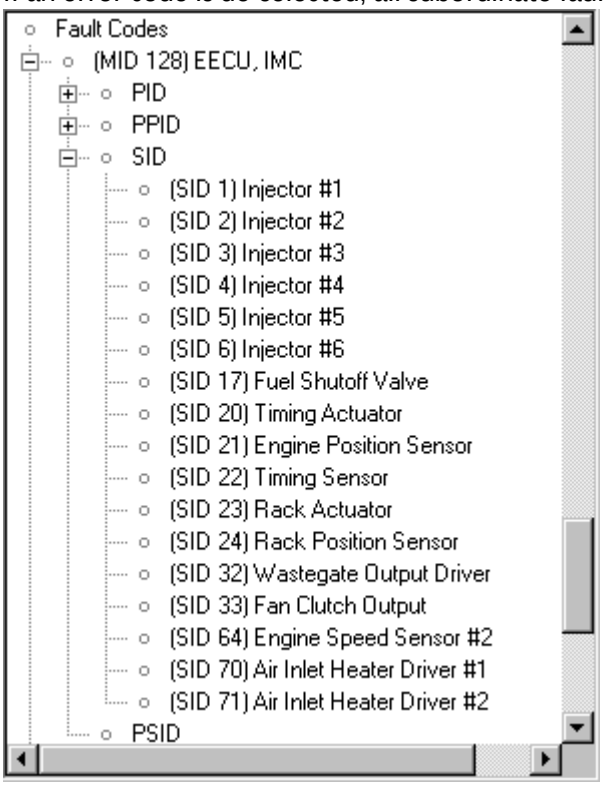
7.2.10.7.6 Fault Codes

Fault codes here refer to diagnosis messages transmitted via PID 194. For this reason, PID 194 is not listed among the channels.

Fault codes are ordered according to the same hierarchical structure as the measurement channels. Additionally, there are general fault codes (SID 151 to SID 255) which are not assigned to any particular MID (referred to here as Common SID).

Any fault code settings made apply to all subordinate fault codes in the tree structure. If All Fault codes is selected, all fault codes are activated for recording; this is indicated by the green check-mark in the list. Any subordinate fault codes then do not get their own properties dialogs.

If an error code is de-selected, all subordinate fault codes are reset.



Fault codes

7.2.10.7.6.1 Fault Code Properties

The Properties dialog for error codes always features the same appearance. The settings made apply to the currently selected error code and any error codes which are subordinate in the tree structure. In the illustration below, for instance, all SIDs of MID 128 are recorded.

Record error code

Fault codes can be lined to a virtual bit in imc STUDIO. A virtual bit can be considered an event bit which releases a trigger. Thus, the appearance of certain fault codes on the J-Bus can trigger an event in imc STUDIO.

Record fault code:

Activates/ deactivate recording of the selected fault code.

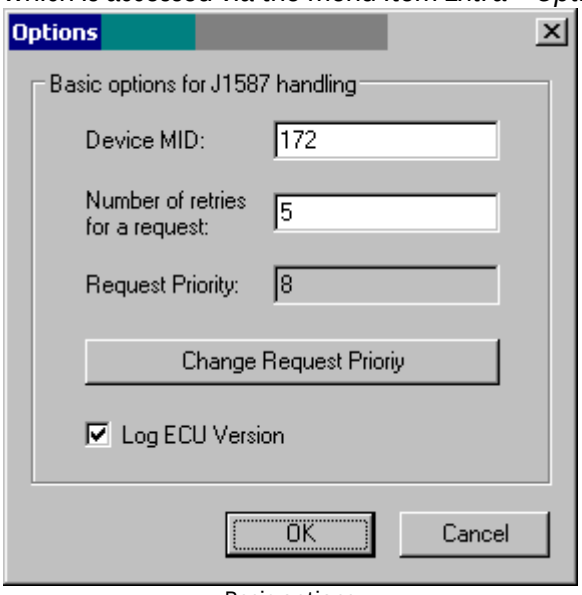
Virtual Bit:

Here, all of devices' virtual bits are listed by name. The virtual bit is set to 1 in case the code is detected.

A virtual bit's name can be set in imc STUDIO: Setup.

7.2.10.7.6.2 Basic Options

There are PIDs on the J-Bus which only transmit data upon "Request". These Requests must be sent by the device along the J-Bus. For this reason, the measurement device must be able to act as a normal J-Bus subscriber and therefore needs, for instance, an MID. These Basis-Options can be set in a dialog which is accessed via the menu item *Extra* > *Options* in the Assistant.



Basic options

Device MID:

MID under which the device poses as a J-Bus subscriber. This MID-number (1...255) must not collide with other MIDs on the bus. The system sends data an the bus in the form of Requests.

Number of retries for a request:

If a Requested PID does not return any data within the Request-interval specified, this number of repeat tries is performed before the requesting of the PID is abandoned. That means: if an ECU does not respond in (number of repetitions) times (request interval) seconds, recording of this channel is stopped.

Request Priority:

Message priority of PID 128, which is sent by the device as a Request. Refer to SAE J1708 Paragraph 4.2.1.1. This box is write-protected by default. Changing the setting is only possible after entering a password.

Change request priority:

The Request Priority is normally write-protected and can be changed to shared after entering a password.

Log ECU Version:

See section above.

7.2.10.7.6.3 Recording the ECU-Version

The ECU version query (PID 234, PID 243) is performed automatically when a measurement is started and is active by default. There is a separate measurement channel for the versions: *J1587_ECU_Versions*. This global channel is automatically created by the J1587 Assistant when the database is imported. There are two ways to prevent the version query from happening.

1. Set the channel *J1587_ECU_Versions* to "passive" in the plug-in: Setup.
2. There is a special check box, *Log CU Version*, in the J1587 Assistant in an options dialog reached via the menu item *Extra* > *Options*.

7.2.11 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

Observe the general Field bus notes regarding the operation methodology 338 of fixed sampling rates 340 versus timestamp 342.

Reference Literature and Standards

LIN specification Revision 1.3 and 2.0, 2.1
"LIN-Bus", Andreas Grzemba, Hans-Christian von der Wense, Franzis Verlag, 2005, Poing (a German language book)
ISO / OSI reference model

7.2.11.1 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.

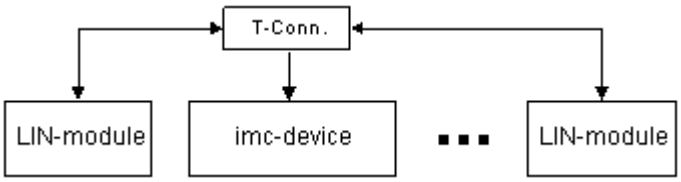
DSUB-PIN	Signal	Description
1	NC	
2	NC	
3	LIN_GND	LIN Ground
4	NC	
5	NC	
6	LIN_GND	Optional LIN Ground
7	LIN_INPUT/OUTPUT	LIN bus line
8	NC	
9	NC	

Note

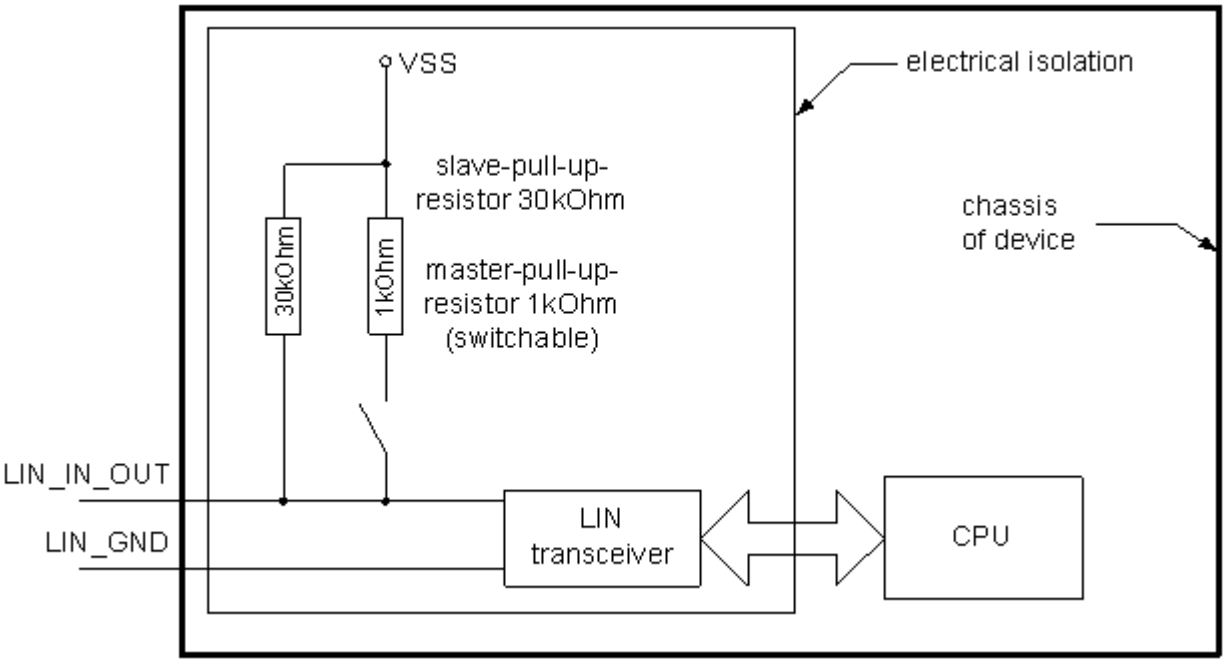
Since the Pin configuration is similar to that of the CAN-Bus, you can also use any CAN-Bus cables already available for the LIN-Bus. However CAN- and LIN-modules may not be connected together to one node!

7.2.11.2 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.

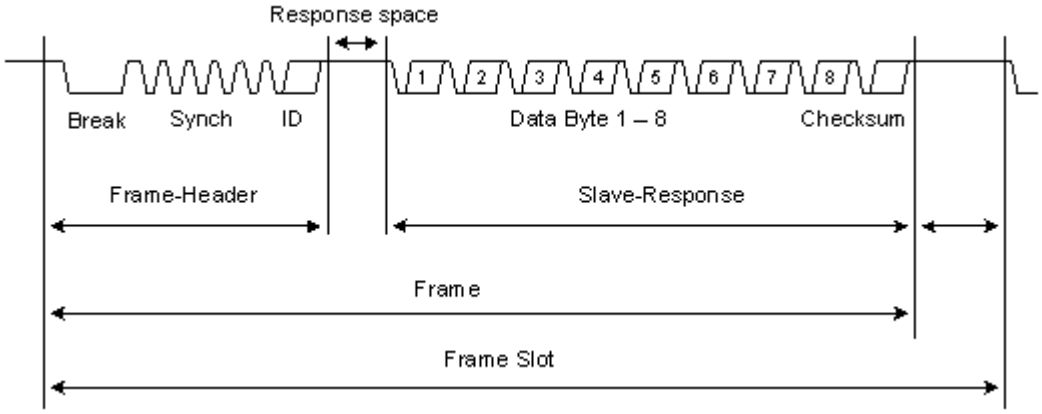
Warning

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.

7.2.11.3 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.

7.2.11.4 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 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.

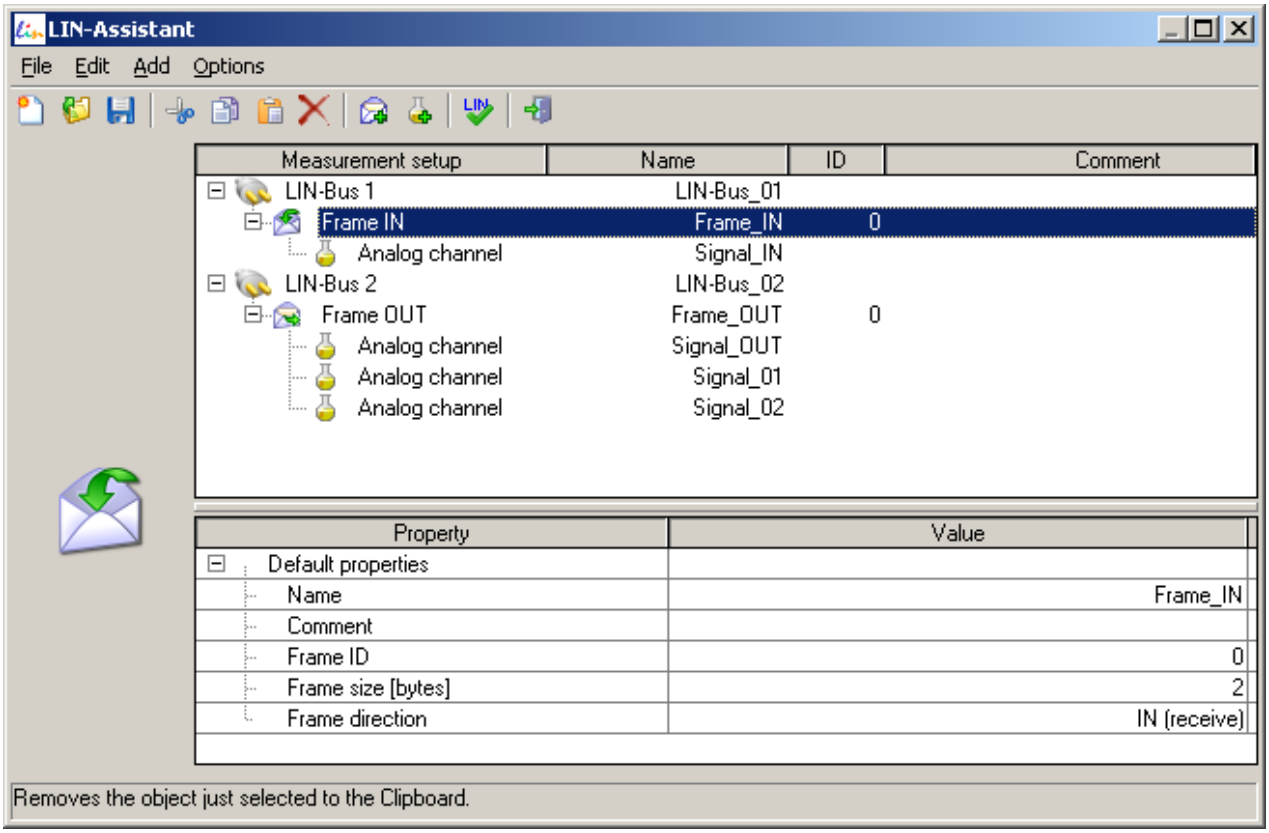
Combination

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.

7.2.11.5 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

7.2.11.5.1 Properties of the LIN-Assistant

- Simultaneous functioning as LIN-Slave and as LIN-Master (the master is limited, see Limitations of the LIN-Assistant 524)
- 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 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

7.2.11.5.2 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

7.2.11.5.3 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.

7.2.11.5.4 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.

7.2.11.5.5 Items in the Main Menu

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.

Note

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...)

7.2.11.5.6 Further Menus

Edit - menu

These are the familiar Windows Clipboard commands:

Copy, Paste, Cut, Delete

Add - menu

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

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.

7.2.11.5.7 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

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.

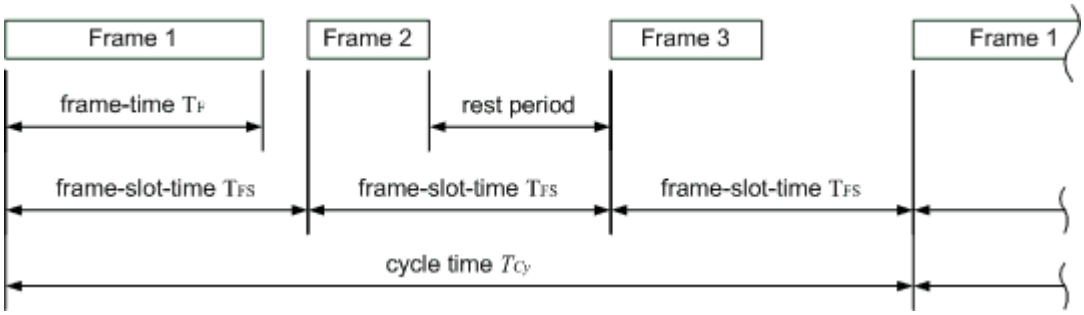
Baud rate: This property specifies the LIN-bus's bit transfer rate. There is a pop-down list containing pre-defined standard values, but arbitrary values can also be entered. According to LIN-specifications, Baud rates of 1000 to 20000 bits per second are permitted.

Master settings

Master 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 521.

The following parameters only appear if *Master type* is set to *Master*.

Time for one polling cycle [ms]



Setting the polling cycle time 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 T_{FMax} determines the minimum frame-slot-time T_{FSMin}. Taking into account the frame count n, the smallest possible cycle time T_{CyMin} can be found according to the following equation:

Where:

$$T_{CyMin} = n \cdot T_{FSMin}$$

$$T_{FSMin} = T_{FMax}$$

The transfer time T_F for one frame is calculated according to the equation below:

$$T_F = 1,4 \cdot \frac{34\,Bits + 10 \cdot \frac{Bits}{Byte} \cdot \left(n_{DataBytes} + 1\,Byte\right)}{v_{Bit}}$$

Key to variables and numbers:

1,4	Safety factor as per LIN specifications
34 Bits	The frame header requires approx. 34 bit times
$10 \cdot \frac{Bits}{Byte}$	10 bits per byte (8 bits + Start- and Stop-bit)
$n_{DataBytes}$	Number of data bytes to be transferred in the frame *[Byte]*
1 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.

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]*".

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.

Sleep-WakeUp-Command-DisplayVariable: A Display variable selected in the LIN-Assistant can be used to activate the Sleep- or WakeUp-mode. 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. 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.

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** *Command*			

Description of the bitfields:

Bits	Definition
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. 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.

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).

Possible combinations

	Break=13 BT	Break=14 BT	Break=15 BT
Delimiter= 1BT	x		
Delimiter= 3BT	x	x	x

Channel name for the log

This property sets the name for the log channel.

7.2.11.5.8 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

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.

Note

- 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.

7.2.11.5.9 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

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

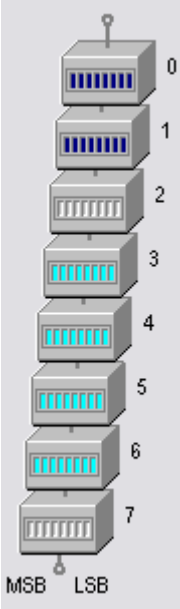
Specifies the comment for this channel.

Data type

This sets the signals type, or rather, how the data on the bus are interpreted. The following data types are possible:

- *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.

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	Outgoing signal	Variable definition
$y = f \cdot x + a$	$x = \frac{y}{f} - \frac{a}{f}$	y - physical value x - integer on the Field-bus f - factor a - offset

Example

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 V

Supposing the following integer were imported from the bus,

x = 13483 LSB

it would correspond to a voltage of

y = f * x + a = 12.057 V

Timeout Setting

This settings option determines whether error handling of any timeout error would be performed.

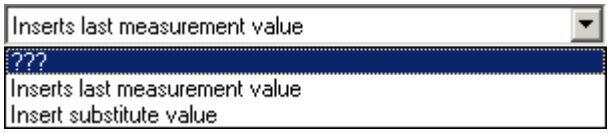
The term timeout-error refers to a case where no measured value is received from the LIN-bus after a certain period; error handling can be set to deal with such an occurrence.

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.

Substitute value

Here, the substitute value can be defined, which is to be used for the respective error handling procedure. This settings control is only visible (enabled) if the setting **Error handling action** is set to **Insert substitute value**.

7.2.11.5.9.1 Special Properties 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:

- 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.

7.2.11.5.9.2 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 536.
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 the imc Hotline.

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.

7.2.11.5.10 LIN-PowerManagementCommands

Issuing Sleep-/ WakeUp-commands via a Display-variable

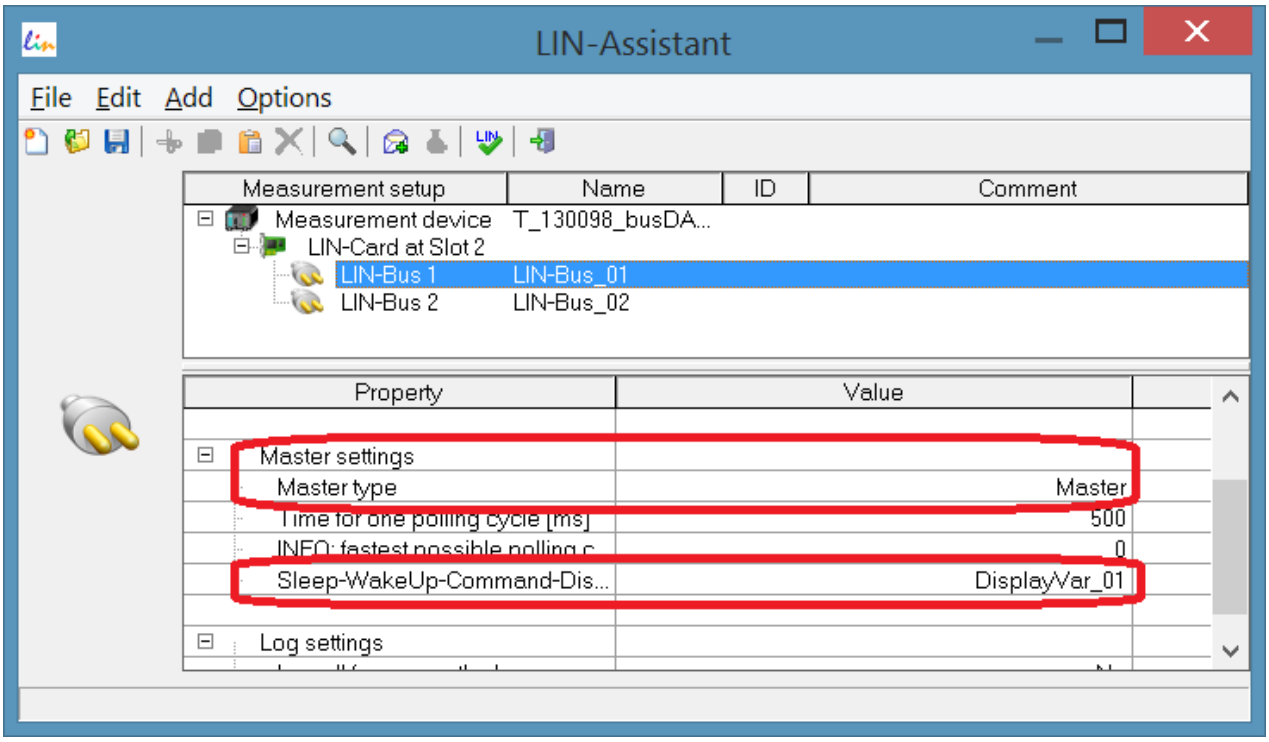
General notes

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.

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 Online FAMOS, or the input/output dialog under imc DEVICES, 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** *Command*			

Description of the bitfields:

Bits	Definition
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 field buses are counted. Available values: 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

7.2.12 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). See also here 539.

7.2.12.1 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"
"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"
Field-bus 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 Field-bus.
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.

7.2.12.2 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.

7.2.12.3 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 548 is available. When in doubt, contact the Hotline 13.

7.2.12.4 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

7.2.12.5 General description of MVB

The Multifunction Vehicle Bus (MVB) is a "multi-purpose"-Field bus for use within a rail vehicle. In the 1990s, this Field-bus 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 (Electrical Middle Distance bus): Twisted wire pair with insulation of the bus interfaces by means of transducers.
- ESD (Electrical Short Distance 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 (Optical Glass Fibre): 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 Field bus. 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

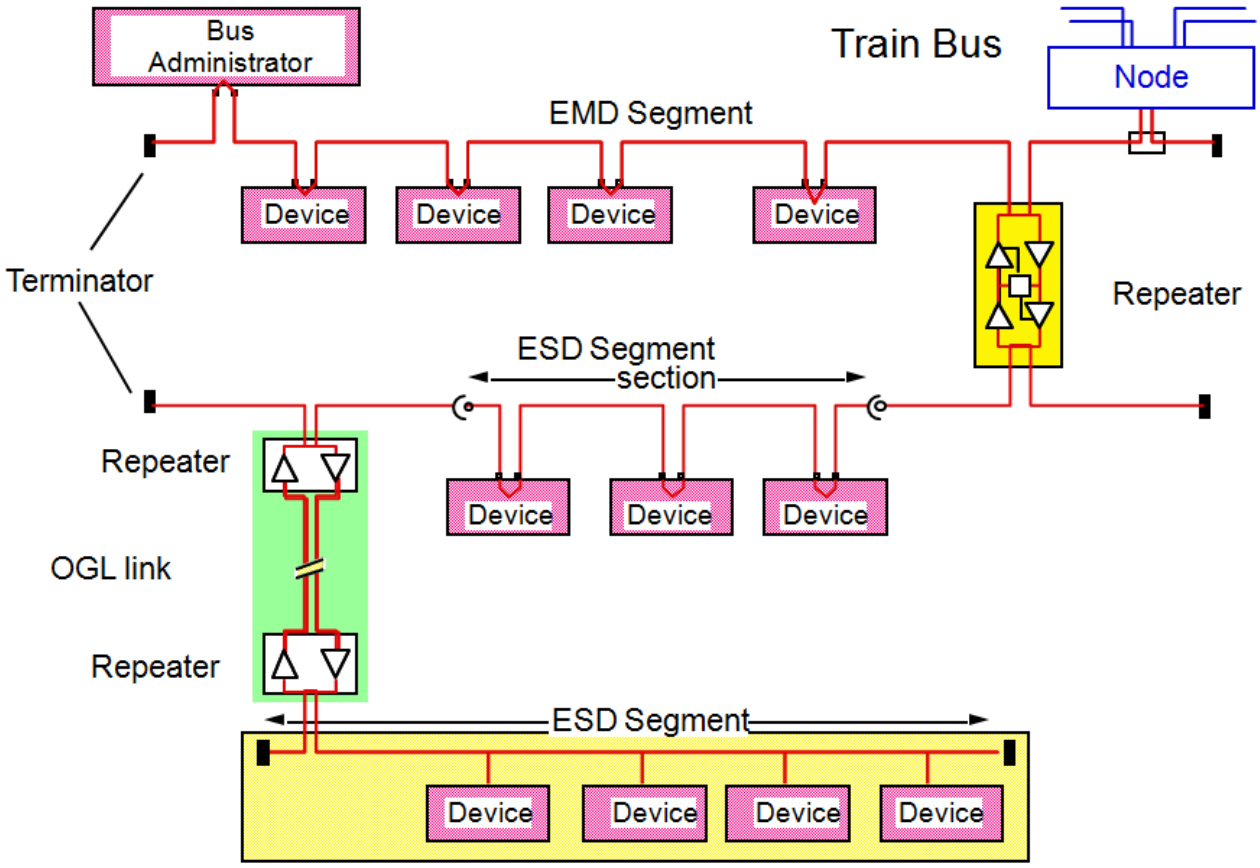
Rail vehicles present electronic equipment with difficult conditions. For this reason, the MVB bus has properties not possessed by normal computer systems' busses.

- 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	

7.2.12.5.1 MVB Topology



MVB uses one baudrate for the complete system.
The segments are connected via repeater.

7.2.12.5.2 Connection and Activation

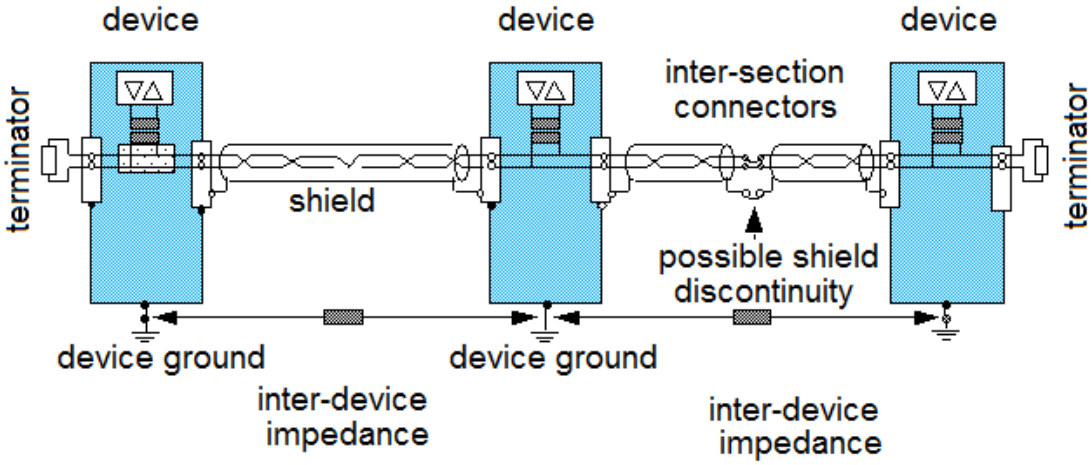
- The MVB-Bus may only be connected with the measurement device once it has been completely wired up.
- Only then may the device be activated.
- During operation, the MVB connectors must not be unplugged.

7.2.12.6 Cabling and Pin Configuration

7.2.12.6.1 EMD Cabling

EMD (Electrical Medium Distance) with duplicate interconnection providing redundant transmission for the bus via two differential lead pairs.

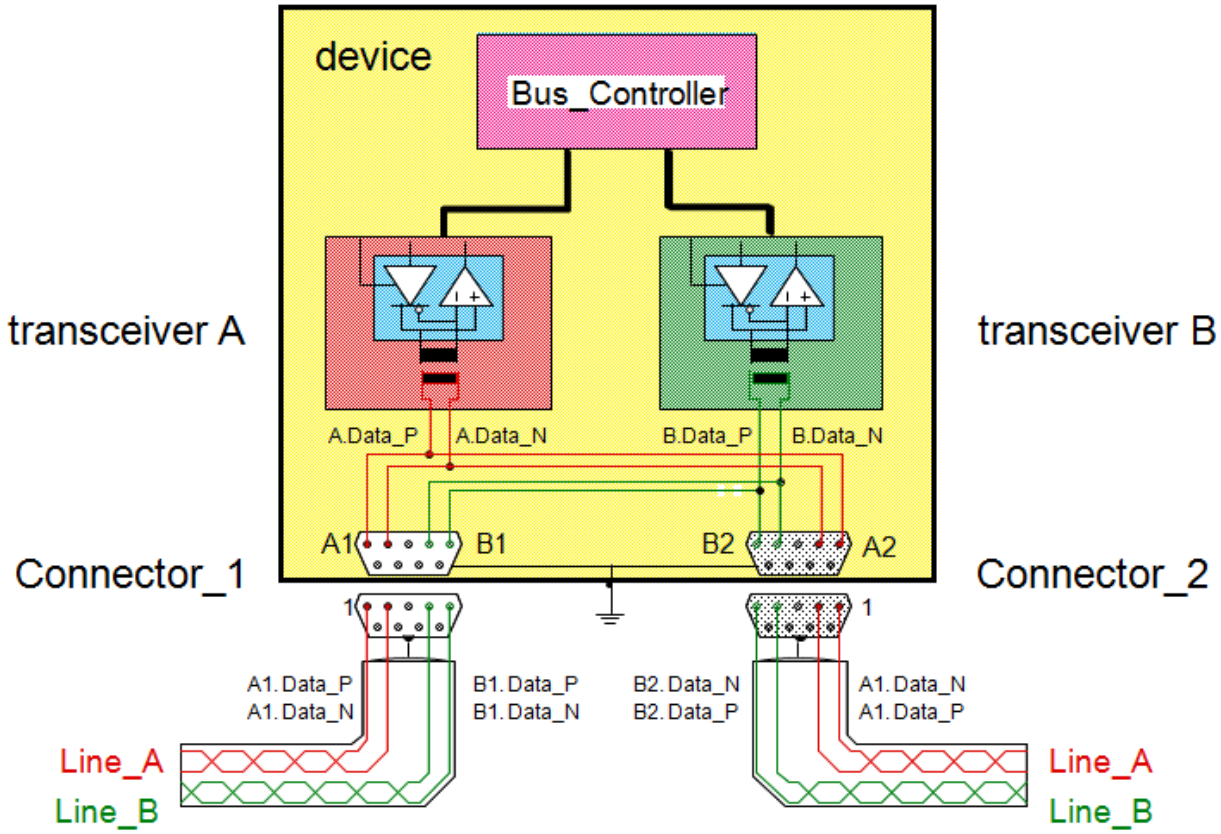
- Up to 32 devices can be connected across a distance of max. 200 m.
- The cables used are standard 120 Ω lines.
- The signals are connected by means of two DSUB9 terminals.



MVB-bus cabling for EMD

The shielding is connected directly to the device housing. The housing should be grounded if possible.

Internally, the bus is electrically insulated from the device connected.

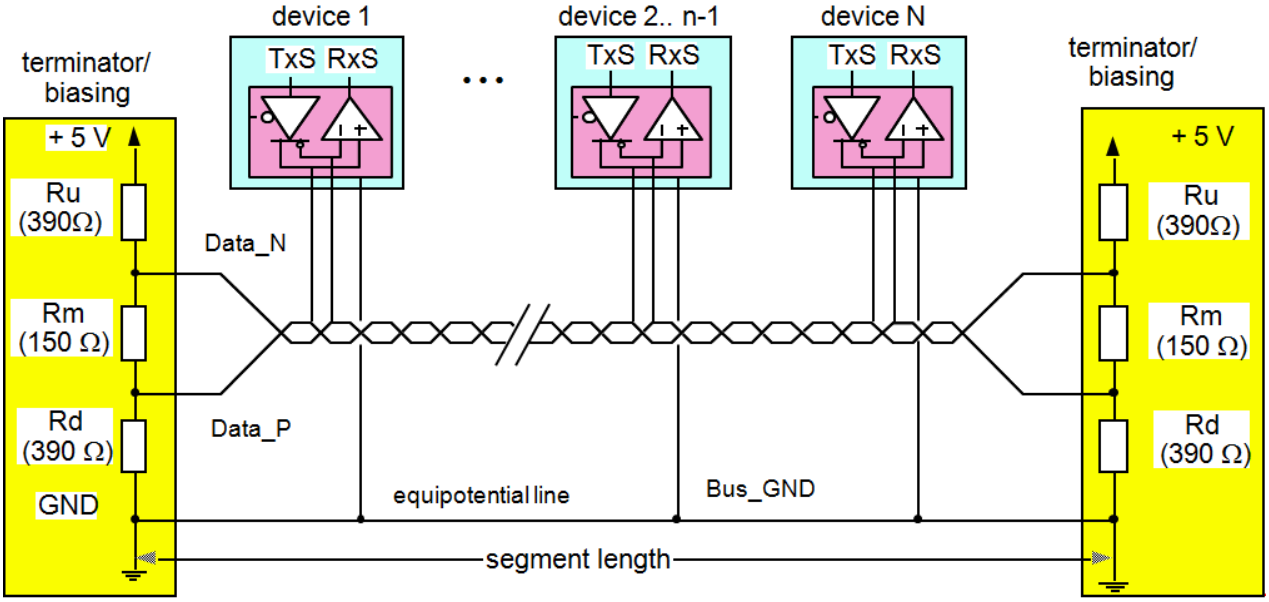


Internal wiring for EMD connection

7.2.12.6.2 ESD Cabling

ESD (Electrical Short Distance) RS485 sets up the connection without galvanic isolation. The variant ESD+ comes with galvanic isolation and is available from imc.

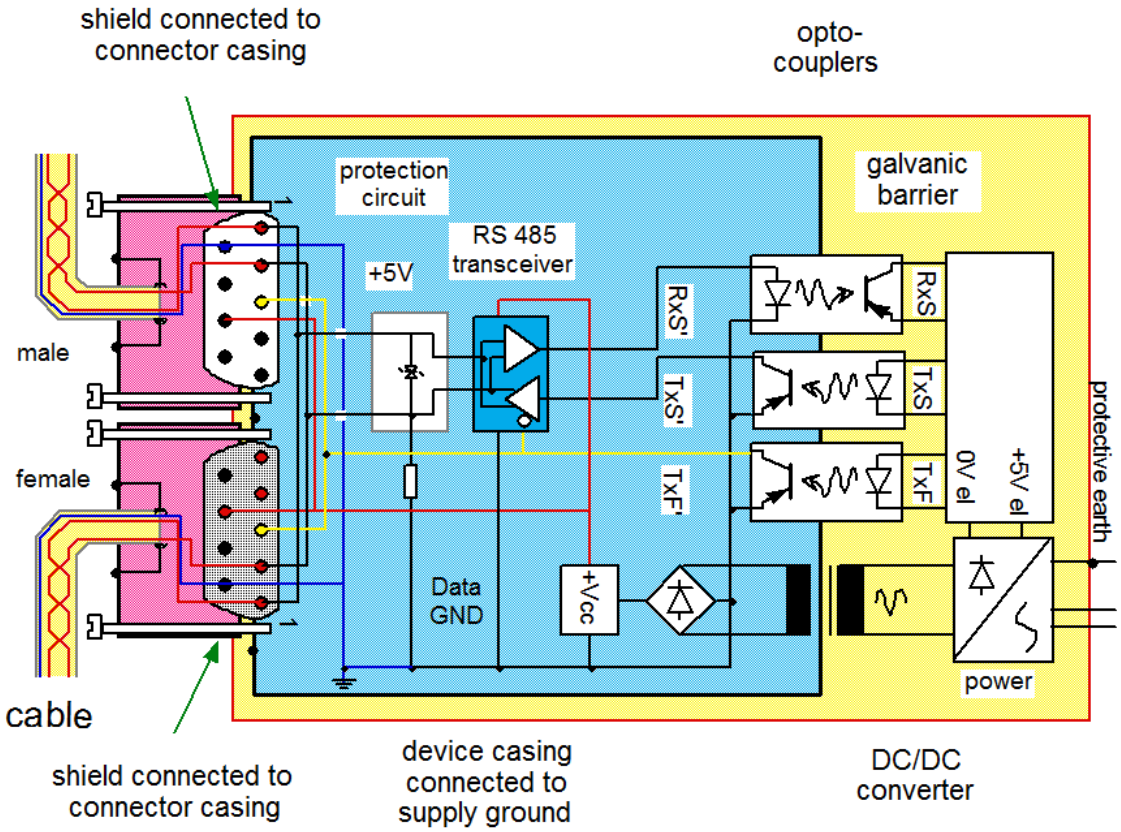
- Up to 32 devices can be connected across a max. distance of 20 m.
- Standard 120 Ω cable leads are used.
- Lines are connected via DSUB9 terminals.



MVB-Bus cabling for ESD

The shielding is connected directly to the device chassis, which should be grounded if possible.

Internally, the bus is galvanically isolated from the connected device.

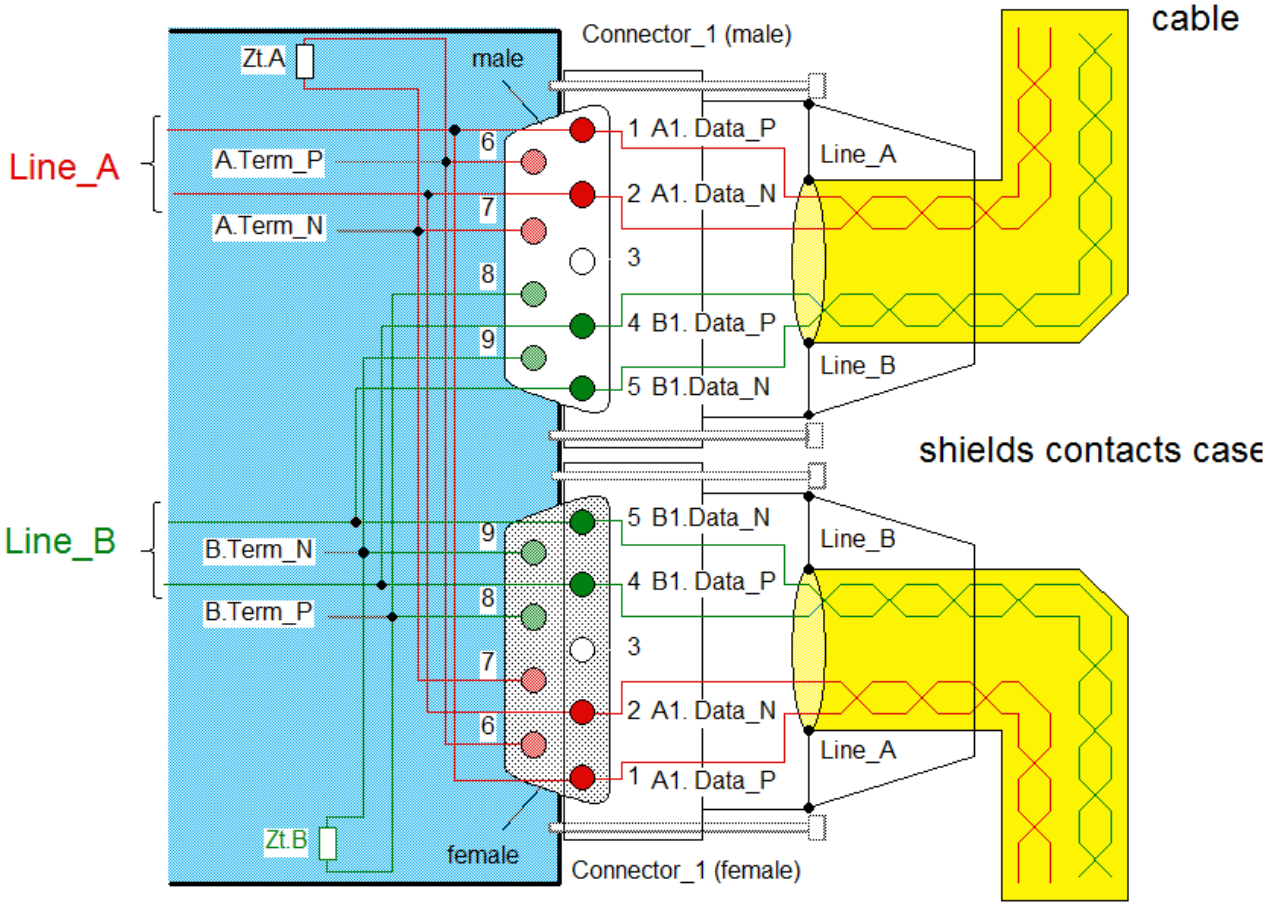


ESD+ device with galvanic isolation

7.2.12.6.3 EMD Pin Configuration - DSUB-9

EMD connection with double-occupancy. Standard DSUB9 terminals are used.

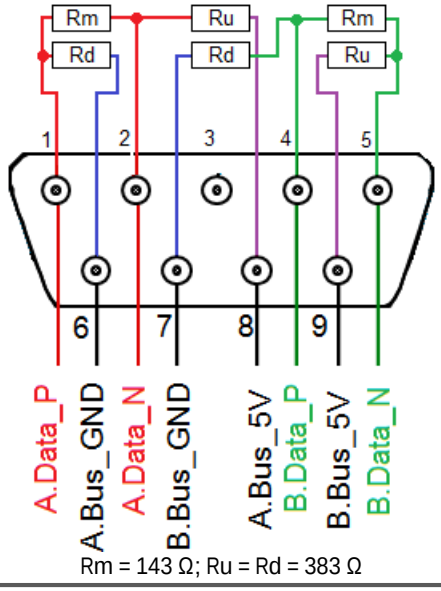
DSUB-PIN	Signal	Description	as termination terminal
1	A1. Data_P	data line A	jumper to 6
2	A1. Data_N	data line A	jumper to 7
3	NC	not connected	
4	B1. Data_P	data line B	jumper to 8
5	B1. Data_N	data line B	jumper to 9
6	Terminator A	internal	jumper to 1
7	Terminator A	interna	jumper to 2
8	Terminator B	interna	jumper to 4
9	Terminator B	interna	jumper to 5

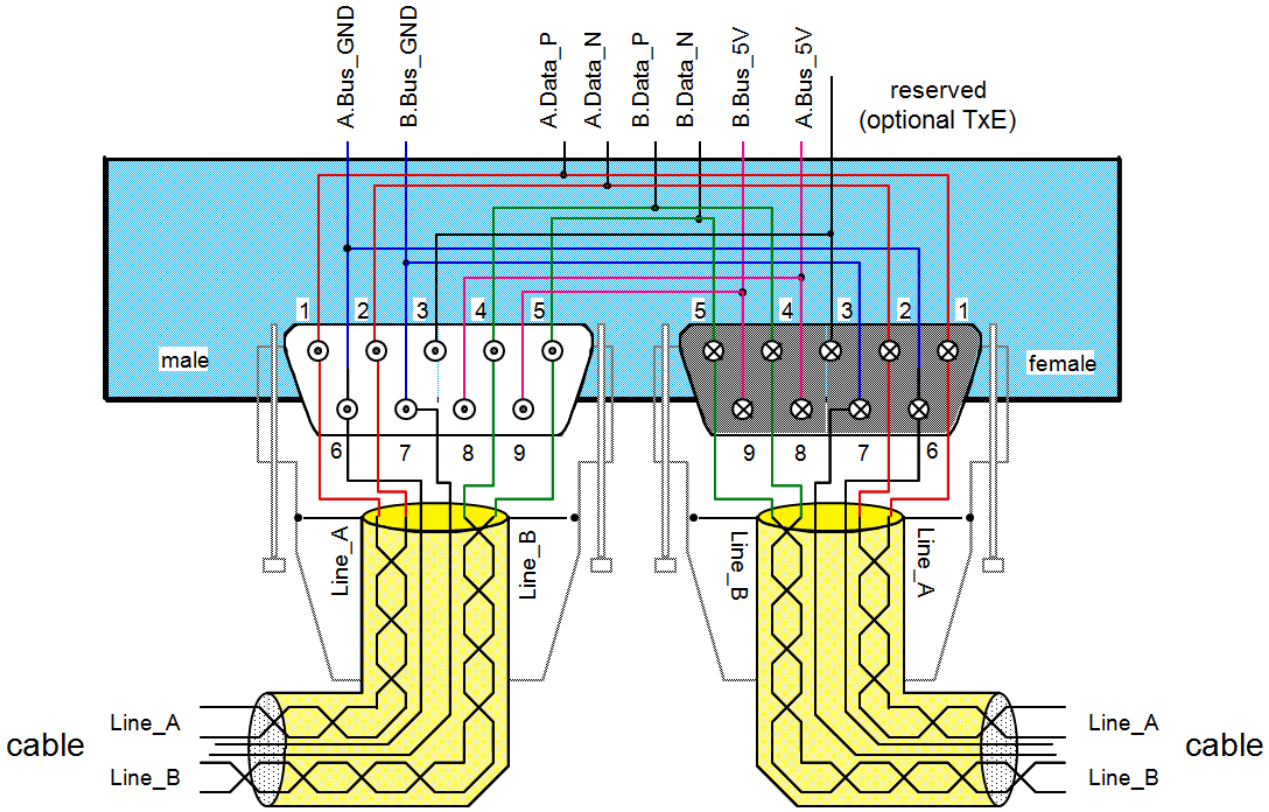


Internal wiring for EMD connection

7.2.12.6.4 ESD Pin Configuration - DSUB-9

ESD Connection. Standard DSUB9 terminals are used.

DSUB-PIN	Signal	Description	Termination
1	A. Data_P	Data lead A	
2	A. Data_N	Data lead A	
3	NC	not connected	
4	B. Data_P	Data lead B	
5	B. Data_N	Data lead B	
6	A.Bus_GND	Ground A	
7	B.Bus_GND	Ground B	
8	A.Bus_5V	5V Supply A	
9	B.Bus_5V	5V Supply B	



DSUB connection for ESD

7.2.12.7 MVB-Assistant

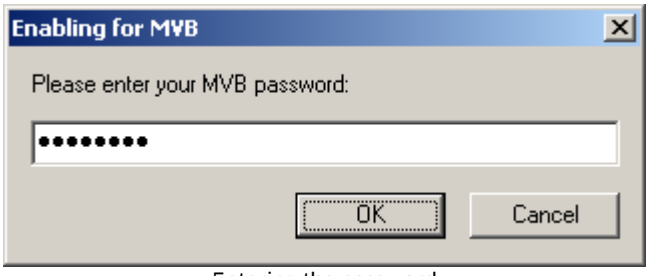
The MVB-Assistant offers the ability to make settings for the imc device's MVB terminals. This includes definitions for the frames and channels (signals) to be received.

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*.

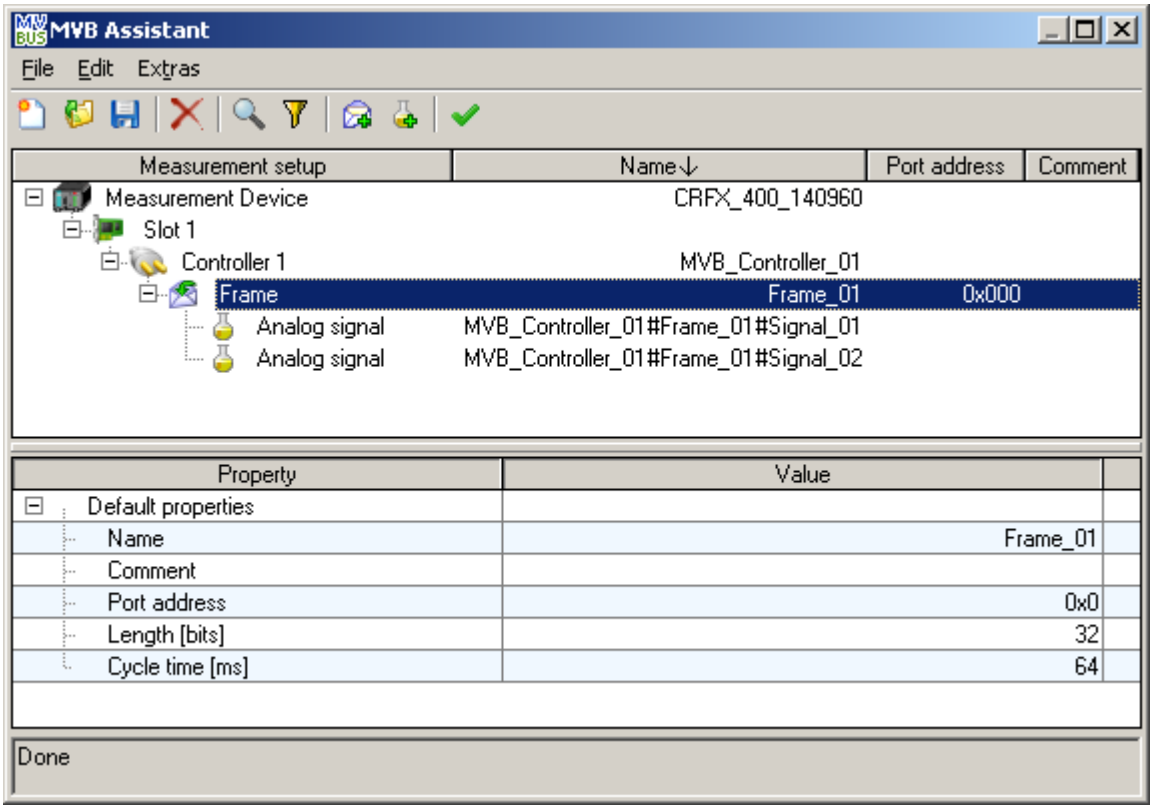
Password

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 540 is demanded, the first time the Assistant is called.



Entering the password

7.2.12.7.1 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.

7.2.12.7.2 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.

7.2.12.7.3 Menu

7.2.12.7.3.1 File

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.

7.2.12.7.3.2 Edit

Delete

Removes the selected entry

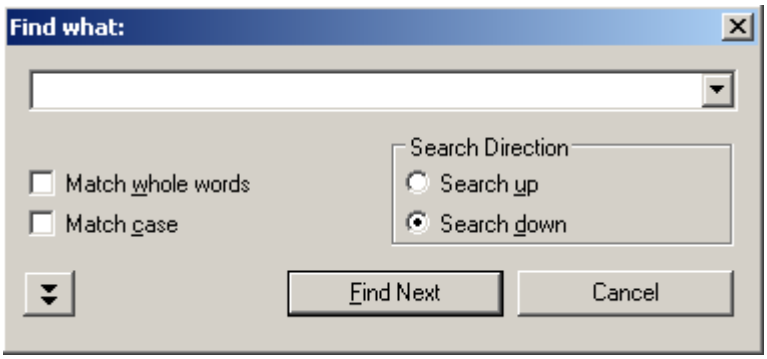
Add new frame

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

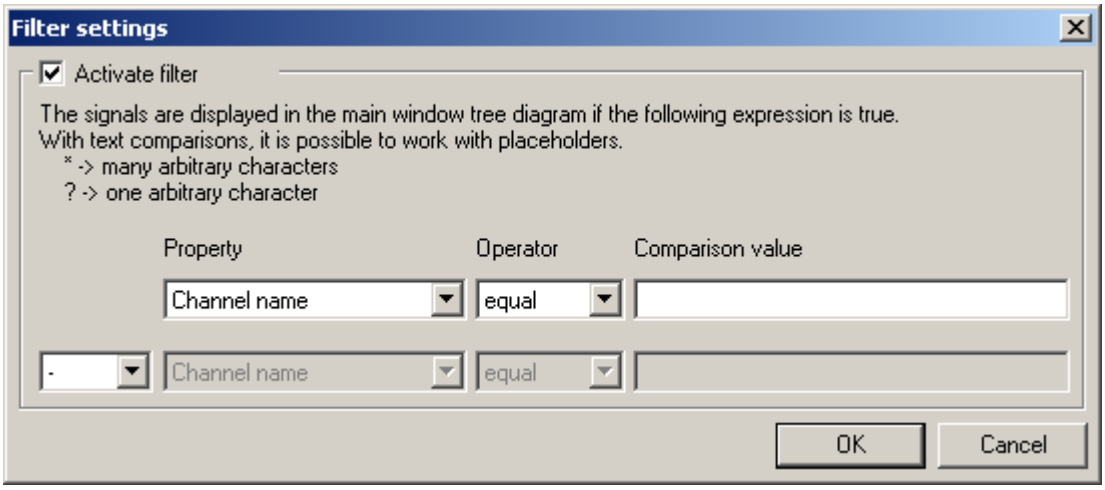


It is possible to search for words or word fragments in the overview window. When many frames/signals have been defined, it is helpful for searching for them by name.

Find Next

Searches for the next instance of the text entered.

Filter



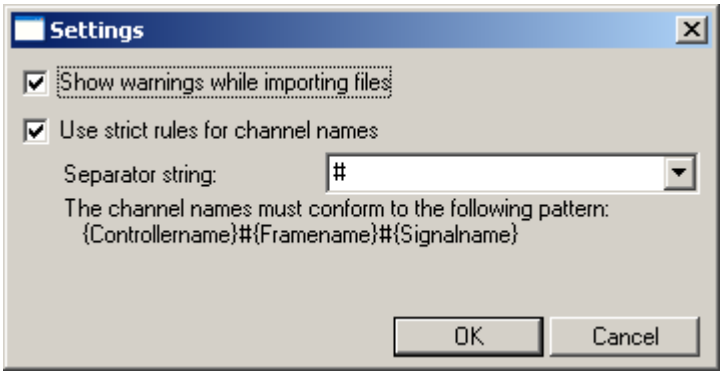
The signals are displayed in the main window's tree diagram if a logical expression's condition is met.

7.2.12.7.3.3 Extras

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.

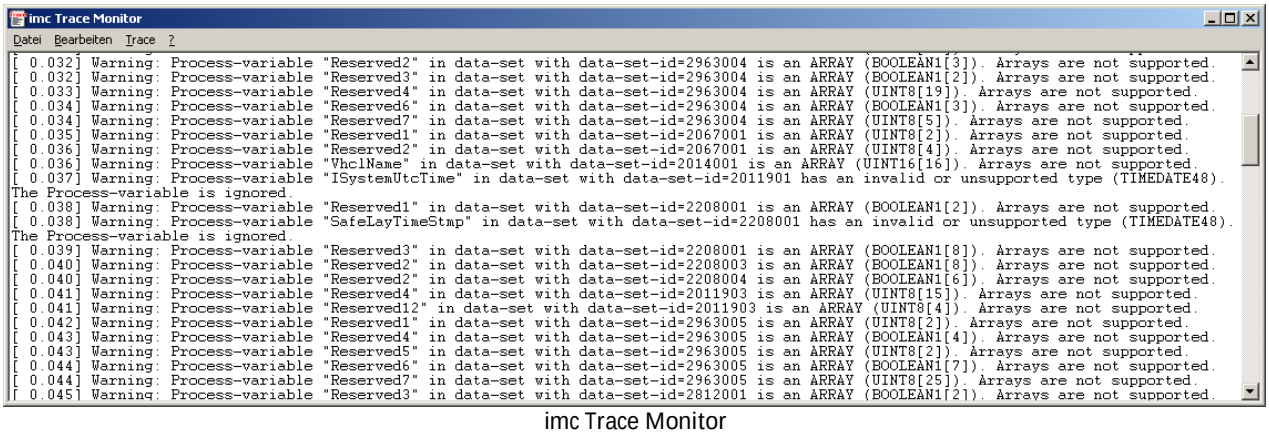
Settings



Extras > 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

7.2.12.7.3.4 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.

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.

7.2.12.7.4 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

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 553, [FRM} for Frame 554 and {SIG} for Signal 555. 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*"

Logging of all configured and received data

MVB_Controller_01#Protokoll01

t [s]	MVB_Controller_01#Protokoll01
0.00561	0000: 98 12 01
0.01409	0001: b0 f 9
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 f 9
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 f 9
0.09647	0002: 00 00 00 a0 14 15 8f c0
0.10547	0003: 00 00 00 40 03 db 7b 40
0.11359	0000: 98 12 01
0.12206	0001: b0 f 9
0.13246	0002: 00 00 00 a0 14 15 8f c0
0.14146	0003: 00 00 00 40 03 db 7b 40
0.14962	0000: 98 12 01
0.15810	0001: b0 f 9
0.16851	0002: 00 00 00 a0 14 15 8f c0

Bus log example

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.

Channel name for the log

Freely definable channel name of the bus log channel.

Resulting name of log channel: The log name resulting from the Construction rule 553.

7.2.12.7.5 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

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

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.

7.2.12.7.6 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

Name

Freely definable name for the signal.

Resulting channel name: The channel name which results from the construction rule 553 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.

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 type

The data type determines the signal's binary encoding.

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	Variable definition
$y = f \cdot x + a$	$x = \frac{y}{f} - \frac{a}{f}$	y - physical value x - integer on the Field-bus f - factor a - offset

Example

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 V

Supposing the following integer were imported from the bus,

x = 13483 LSB

it would correspond to a voltage of

y = f * x + a = 12.057 V

Handle timeout error

This settings option determines whether error handling of any timeout error would be performed.

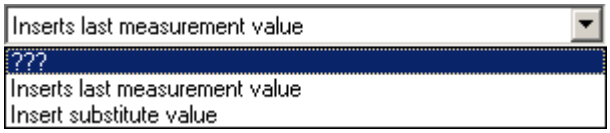
The term timeout-error refers to a case where no measured value is received from the field-bus after a certain period; error handling can be set to deal with such an occurrence.

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.

Substitute value

Here, the substitute value can be defined, which is to be used for the respective error handling procedure. This settings control is only visible (enabled) if the setting **Error handling action** is set to **Insert substitute value**.

7.2.13 PROFIBUS Interface

Using the PROFIBUS Configurator 558, you can create a configuration for the Assistant in imc STUDIO

With PROFIBUS Assistant 565 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.

7.2.13.1 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

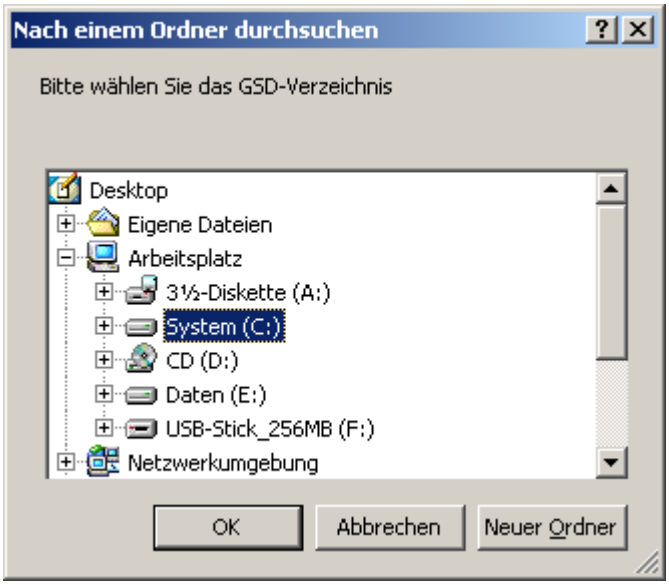
This program is not automatically installed with the rest of the package. It is on the imc STUDIO installations medium in the folder: Products\imc DEVICES\Tools\imcPROFIBUSConfigurator.

Start the setup and follow the instructions which appear. The program can then be installed at an arbitrary location on your hard drive.

7.2.13.1.1 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.

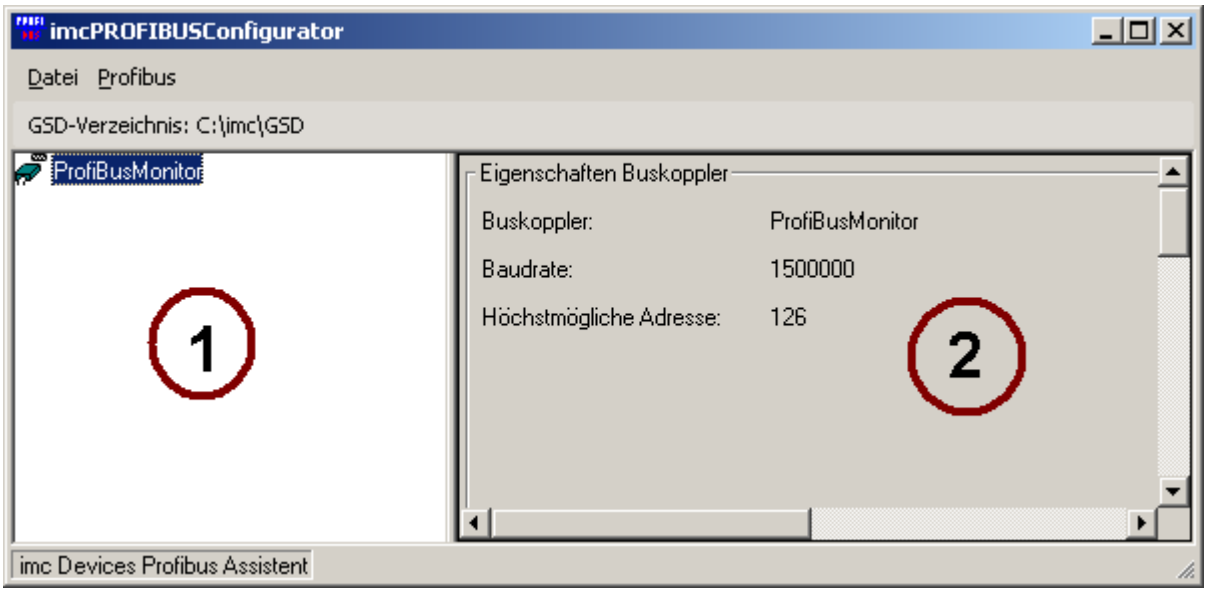


Selecting the GSD folder

7.2.13.1.2 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

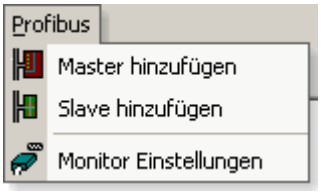
Open: Loads a previously stored configuration in XML format.

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 564.

GSD Verzeichnis wählen: Selects the GSD (General Station Description) 558 files.

Profibus menu



Add Master: Opens the dialog for selecting the Master 560.

Add Slave: Opens the dialog for selecting the modules 560.

Monitor settings: Settings dialog for the Bus monitor 564.

7.2.13.1.3 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 559 or not contain any GSD files, the list remains empty.

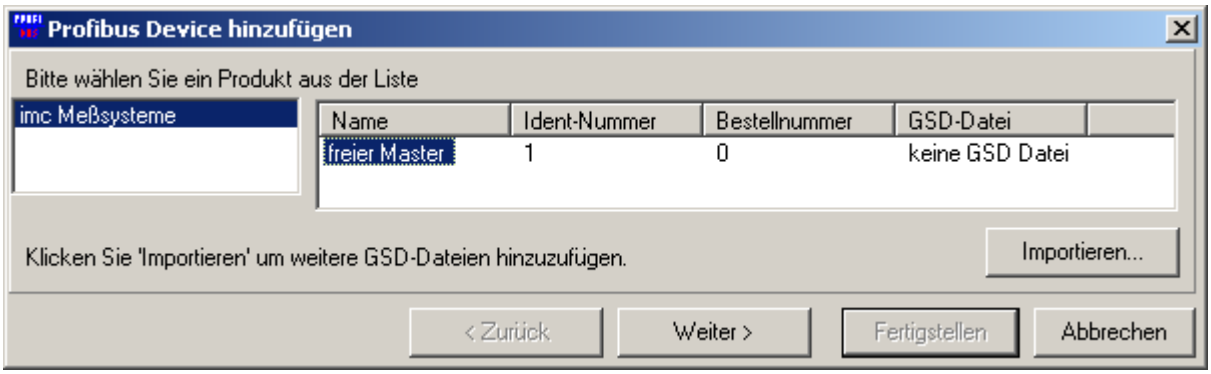
Click on the desired system and then on *Next*. In the next window, you can set any desired properties, such as the name or PROFIBUS addresses. Using the *Import* button, you can add additional GSD files.

Note

Notice that only such elements are listed whose GSD files are located in the GSD folder.

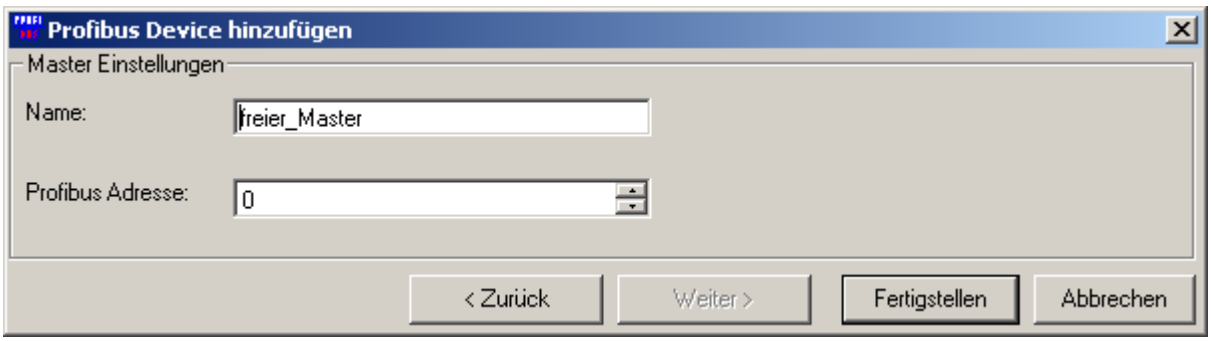
Configuring the Master

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.



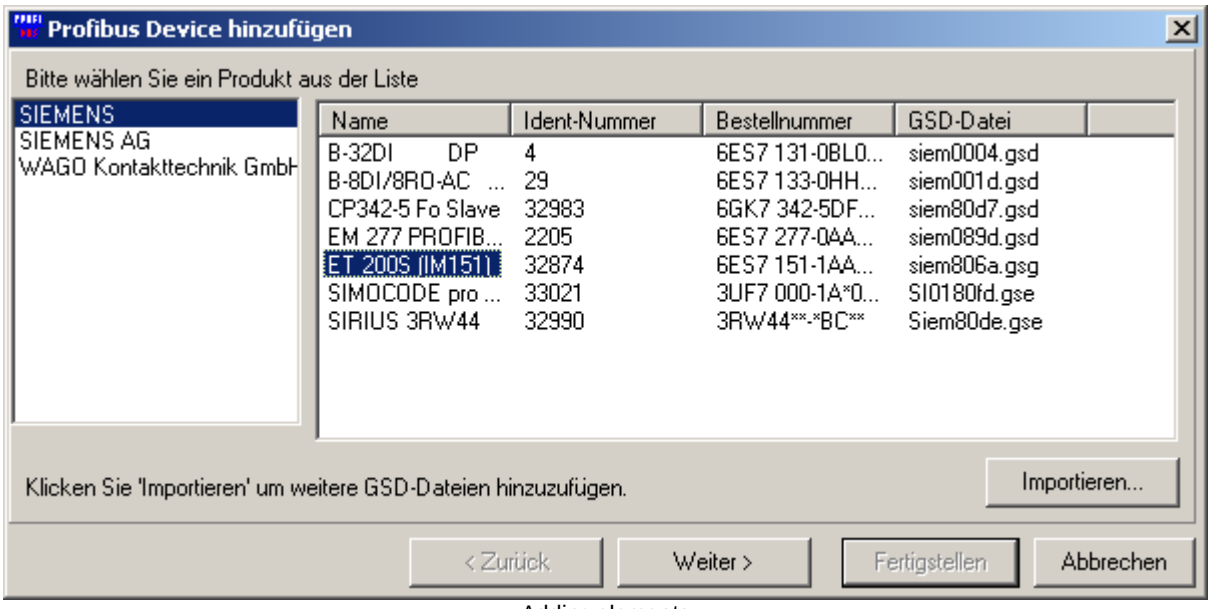
Configuring the Master

Click on *Next >* and set the name and ID.



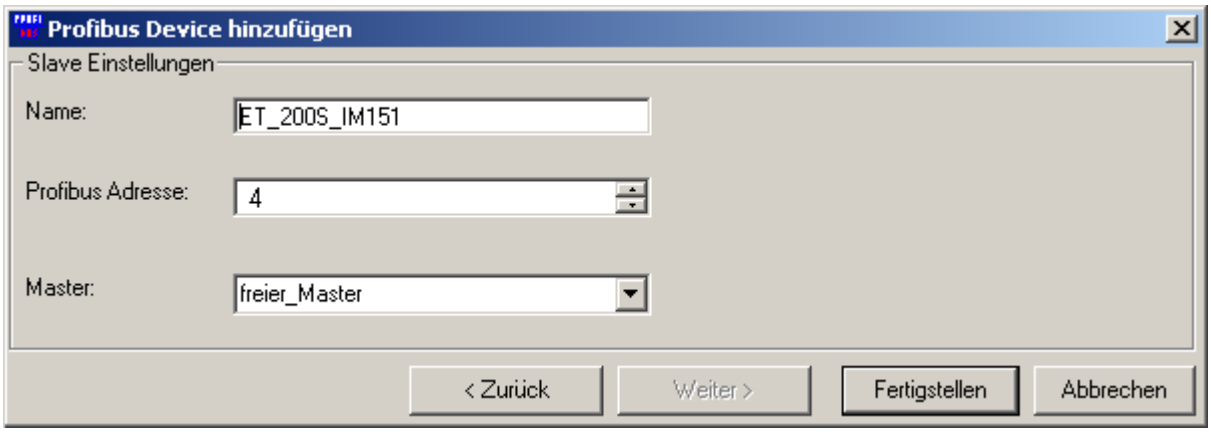
Setting the Master ID

Configuring the Slave



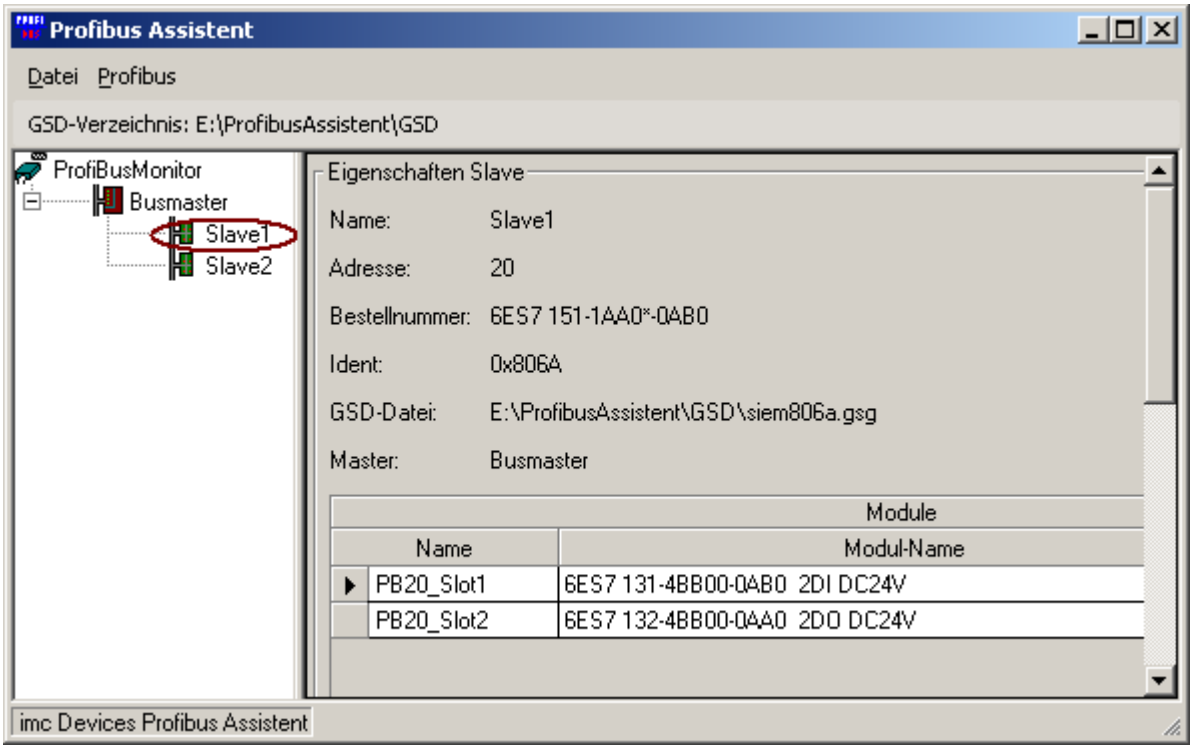
Adding elements

Select the connection terminal and click on *Next*>.



Set the Profibus address and the designation here

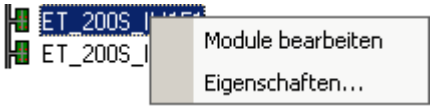
The modules entered now appear in the main window:



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.

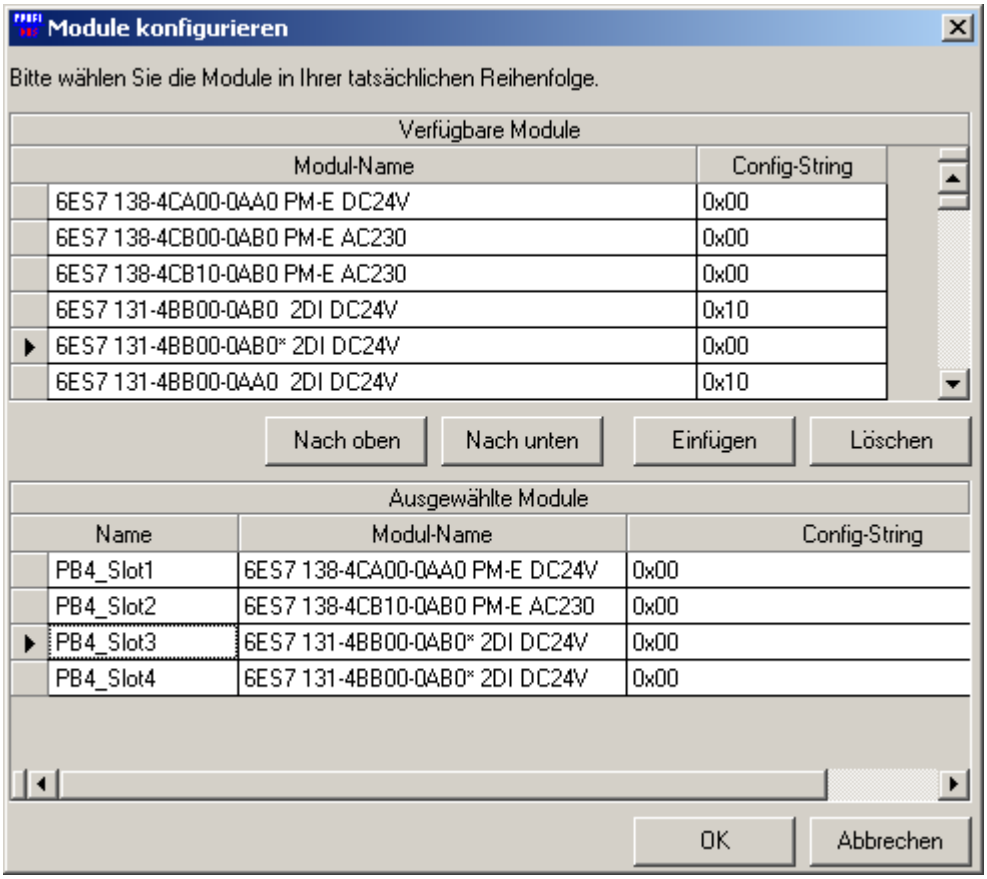
Right-clicking over the Master or the Slave opens the context menu:



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

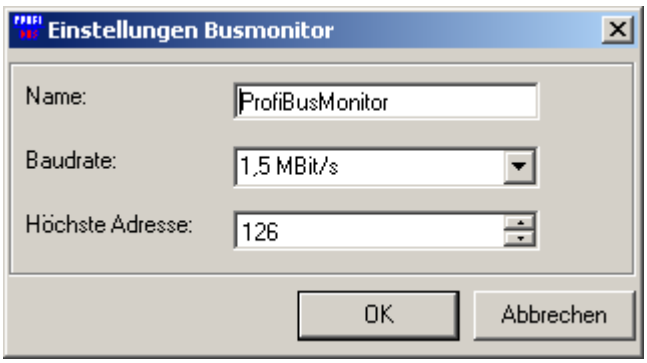
Note that it is absolutely necessary to order the modules in their actual order.

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.

7.2.13.1.4 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 566.



Monitor settings

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.

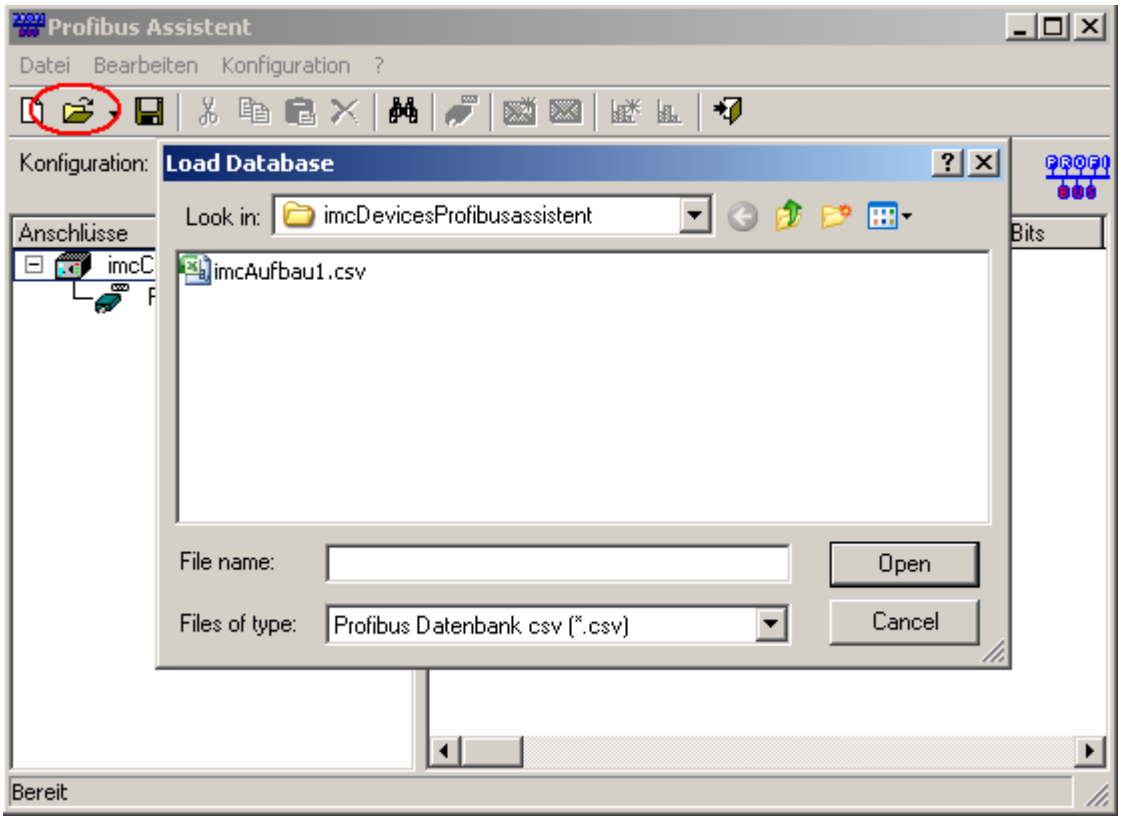
7.2.13.2 Transferring the configuration to imc STUDIO

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 565
5. Open the file saved in step 2 above

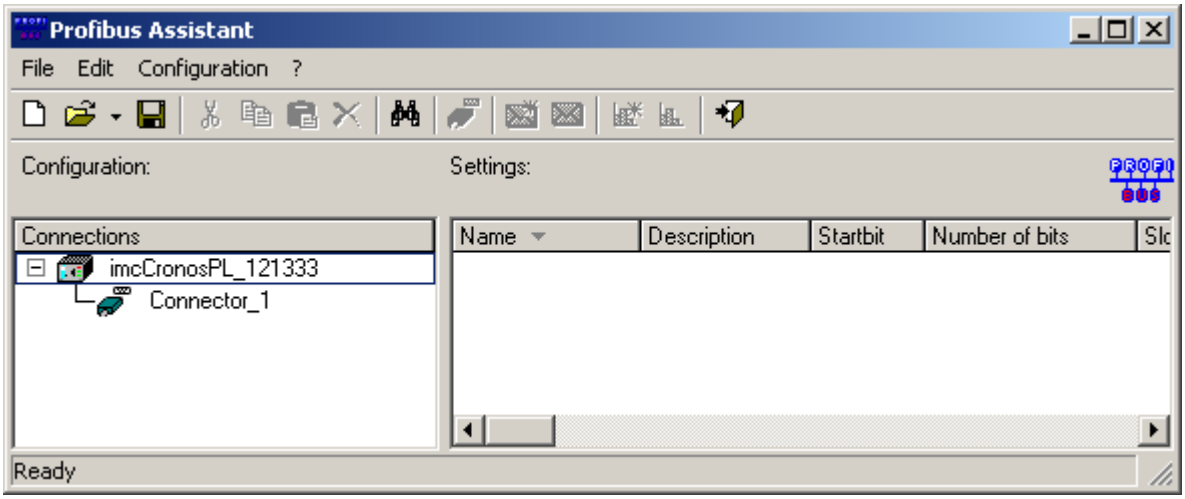
Exporting



Loading the configuration

7.2.13.3 PROFIBUS Assistant

Start the Assistant from within the imc STUDIO software via the Ribbon *Setup-Configuration* > *Profibus-Assistant*.



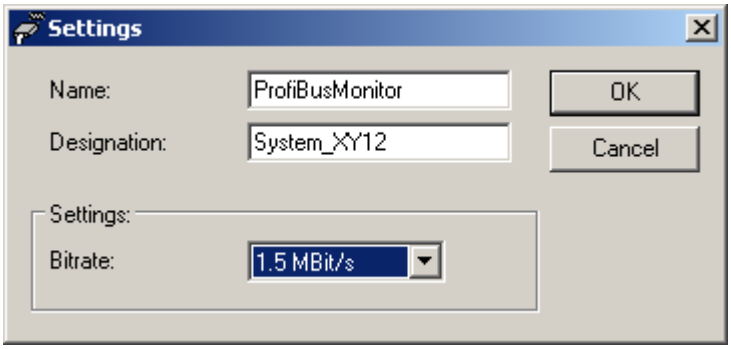
PROFIBUS Assistant

File menu: Here you will find the usual items for opening and saving a configuration in the CSV format.

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.

7.2.13.3.1 Connection settings

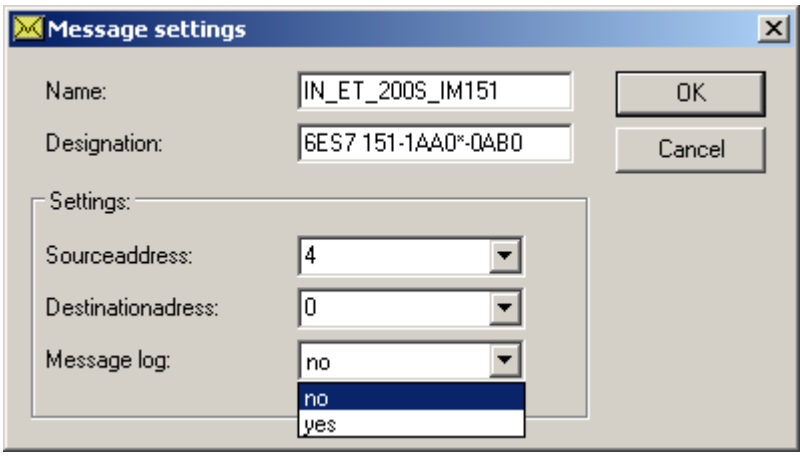


Connection settings

Name, Designation and Bitrate of the connection. The settings 564 created in the PROFIBUS Configurator appear here automatically.

7.2.13.3.2 Message settings

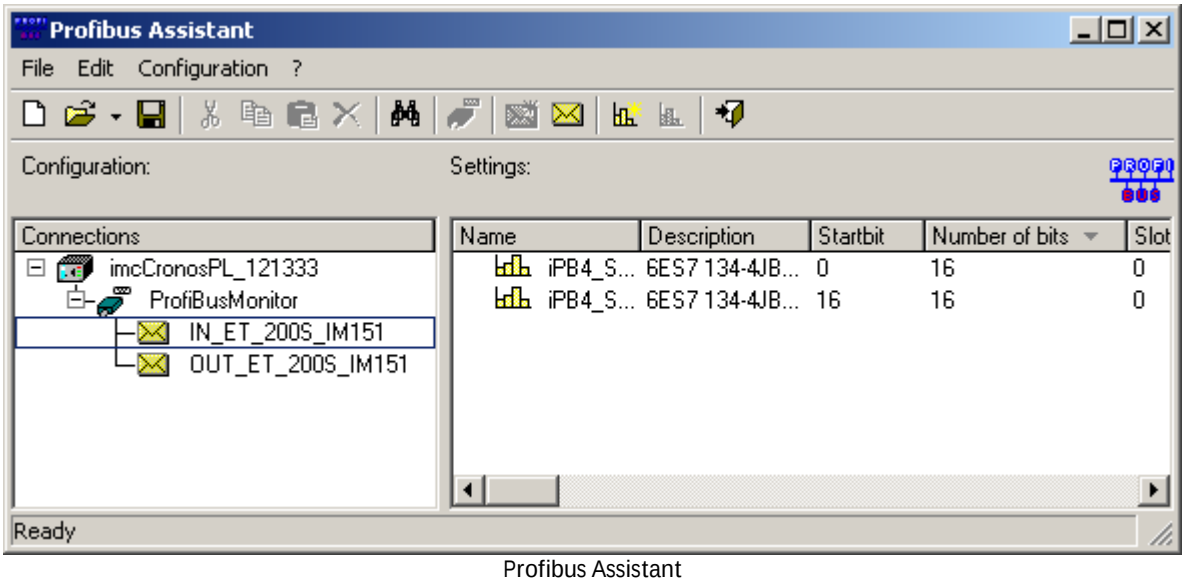
The configuration file created with the PROFIBUS Configurator 558 already contains definitions for all available messages in the system.



Message settings

If you wish to view all of a message's raw data, activate the control Message log in the message settings. The protocol channel is then available as a separate channel in imc STUDIO.

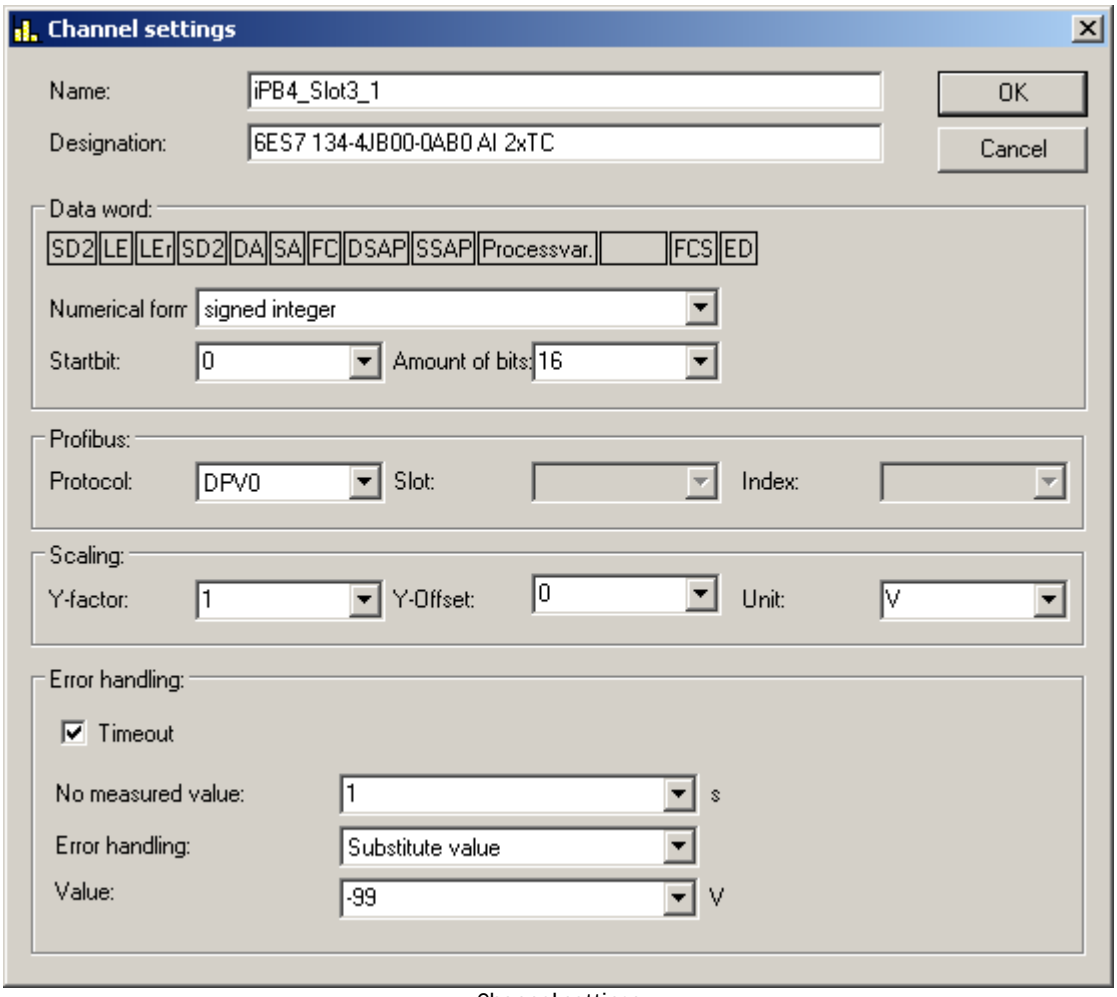
7.2.13.3.3 Channel settings for imc STUDIO



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.

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

7.2.14 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.

7.2.14.1 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

Die KiRoad Performance Steuereinheit übernimmt nicht nur die Energieversorgung aller RoaDyn Radkraftsensoren, sondern sie bereitet die Rohsignale der Messzellen bezüglich übersprech- und hebelarmkompensiert auf. Die Daten werden digital und analog zur Verfügung gestellt und sorgt für präzise Messresultate. 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.

7.2.14.2 Prerequisites

imc Hardware (imc measurement device)

- device with RoaDyn Interface

Signal interconnections

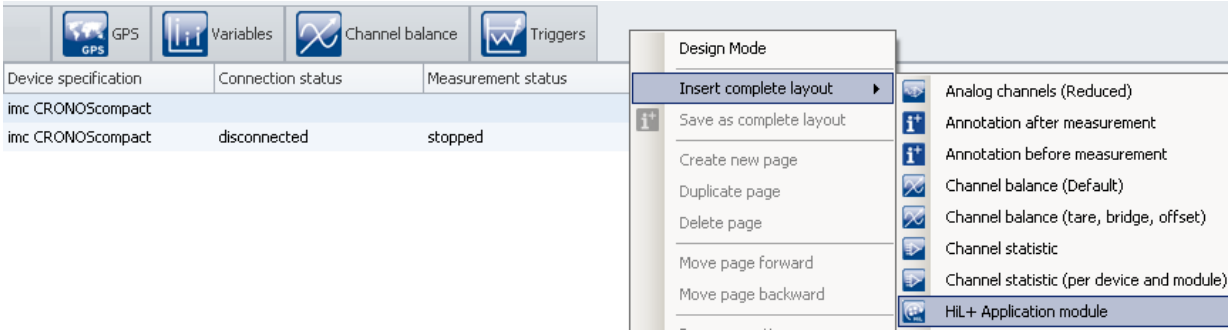
- RoaDyn 2000's BNC terminals Clock and Trigger (CLK and TRG) are to be connected to the application module's respective BNC terminals of the same name CLK and TRG.
- The CRFX/RoaDyn application module's network jack (RoaDyn) is to be connected to the RoaDyn-On-board electronics System 2000's Ethernet interface (ETH).

Note

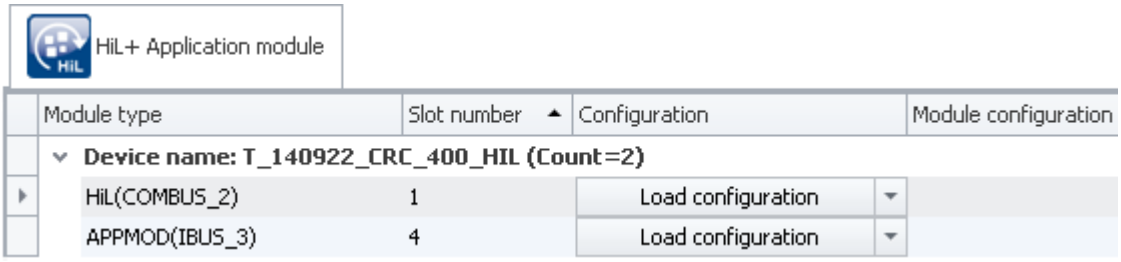
Carefully check the signal connections. The application module's Ethernet socket must not be confused neither with the imc measurement device's standard terminal (LAN) nor with EtherCAT OUT.

How to open the assistant with the desired application?

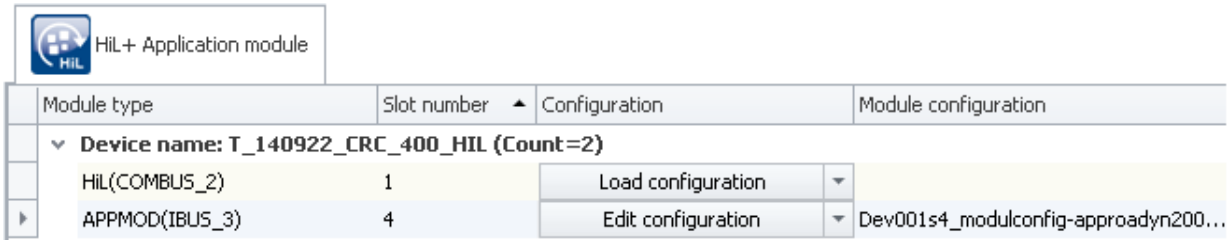
First of all it is necessary to add an additional pre-configured setup page. In order to add the needed page to your user interface open the context menu over a page-tab or over the empty space to the right. Select the following page entry in the list under *Insert complete layout*: *HIL + Application 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-AppKistler.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*".

7.2.14.3 Configuration mode

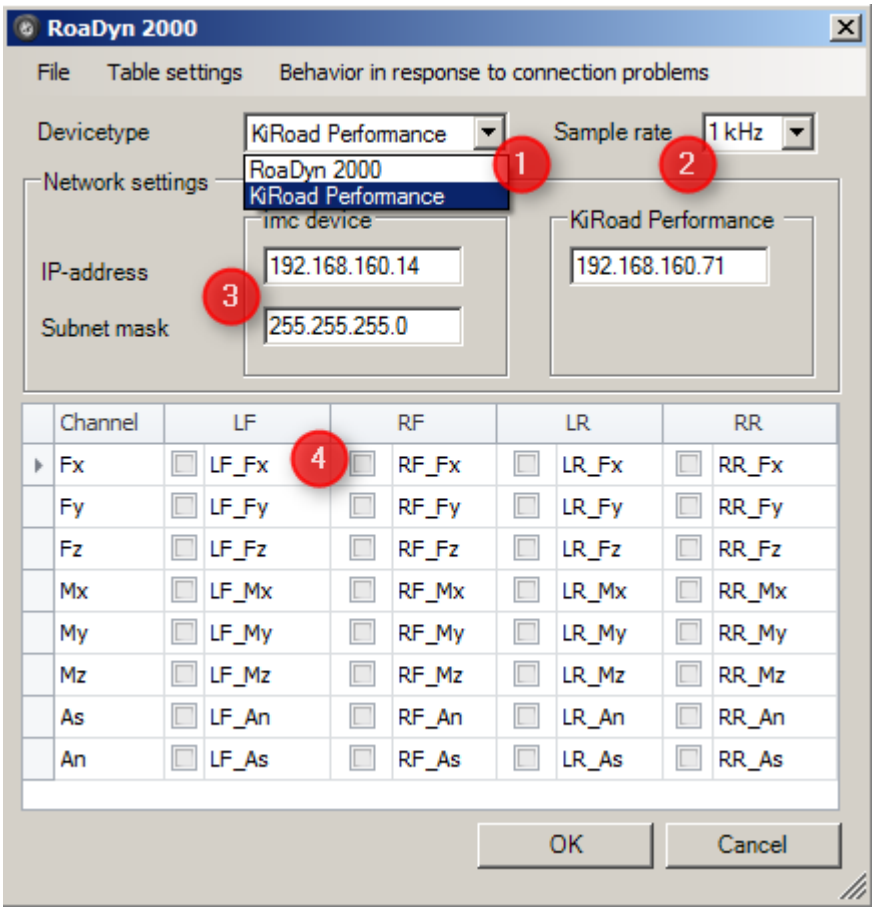
In Configuration mode of the AppRoaDyn2000 application, the following settings are available:

[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

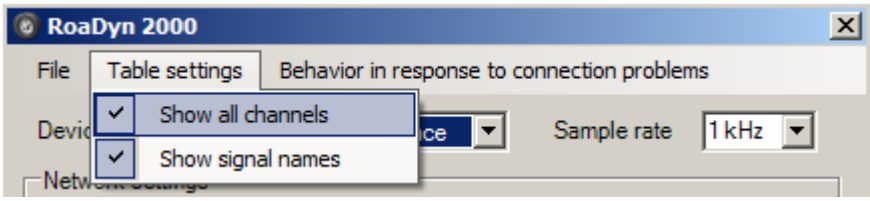
In the Configuration dialog (see above), the imc measurement device appears above the box in which you must enter the application module's IP-address.

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.



7.2.14.4 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.

7.2.14.5 Troubleshooting

Possible messages appearing during a measurement procedure

The following errors affecting the Kistler- System 2000 can cause an error message to be posted:

(-)5408	The configuration is not in order; —> Check the configuration using the Assistant.
(-)5410	Initialization of RoaDyn® 2000 is not possible; —> Check the RoaDyn® 2000's network settings and the configuration, restart RoaDyn® 2000
(-)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 HELLO message: RoaDyn® 2000 not responding to communication attempts (no response to HELLO-message) —> Check the RoaDyn® 2000 configuration; switch RoaDyn® 2000 off and on again.
(-)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.
(-)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.

(-)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.
(-)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.

7.2.15 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 is opened via the plug-in Data Processing. A detailed description of the Bus Decoder is under construction.

7.2.15.1 Definition of terms

Term	Description
Bus Decoder	The Bus Decoder shows the channels created upon selection of the data log channel.

7.2.15.2 Prerequisites

Hardware:

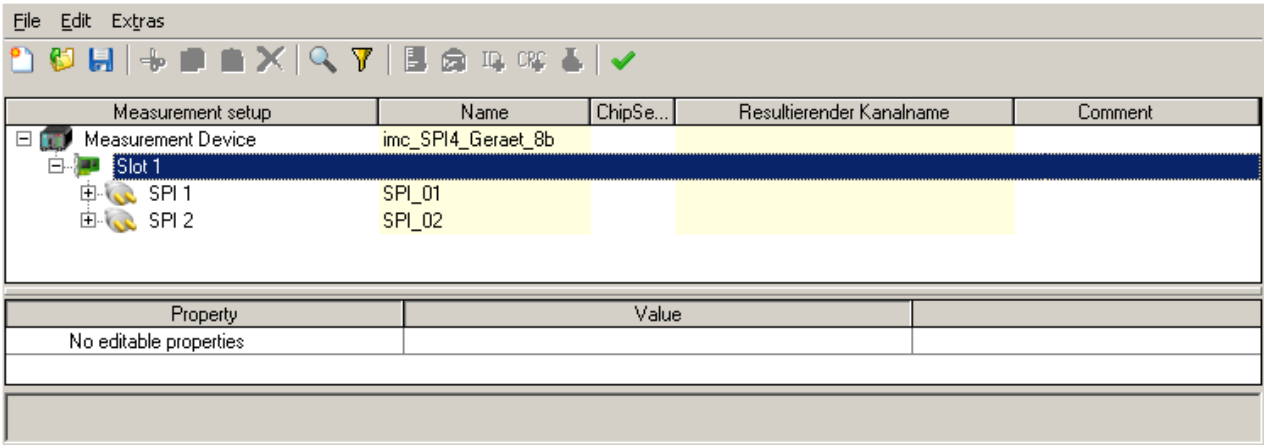
An imc device with a SPI-Bus interface.

Software:

The system requirements are described in the data sheet of our fieldbus connection options.

7.2.15.3 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.

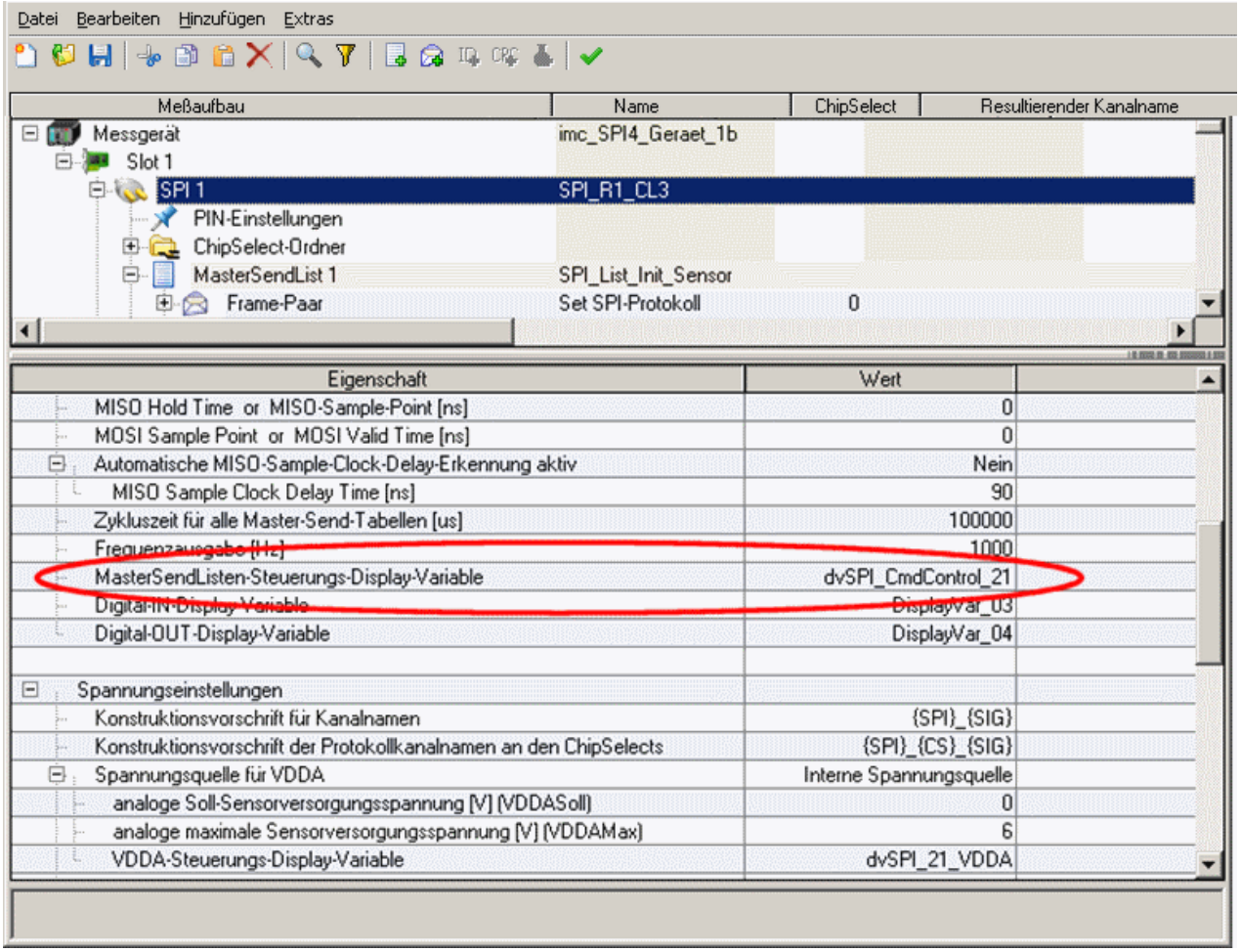


The Assistant is structured in tables, in order to present a large number of parameters in a clear way and to make them available for editing. The Assistant graphically displays the structure of the frame currently being edited.

7.2.15.3.1 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:

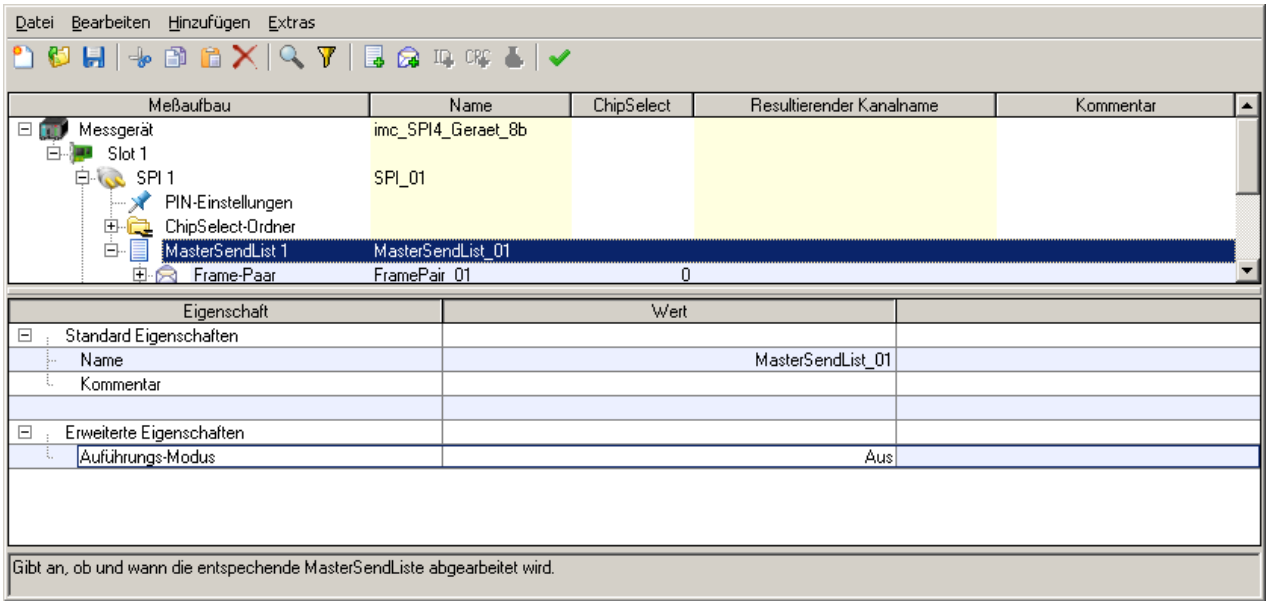
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
4-bit Port				4-bit command				8-bit list index							

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 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 Initialize (Download):	0xF100 or 0xF500	only 0xF500
Switch on List 1 in Port 1 as with Initialize (Download):	0x1101 or 0x1501	only 0x1101
Switch on List 1 in Port 2 as with Initialize (Download):	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	

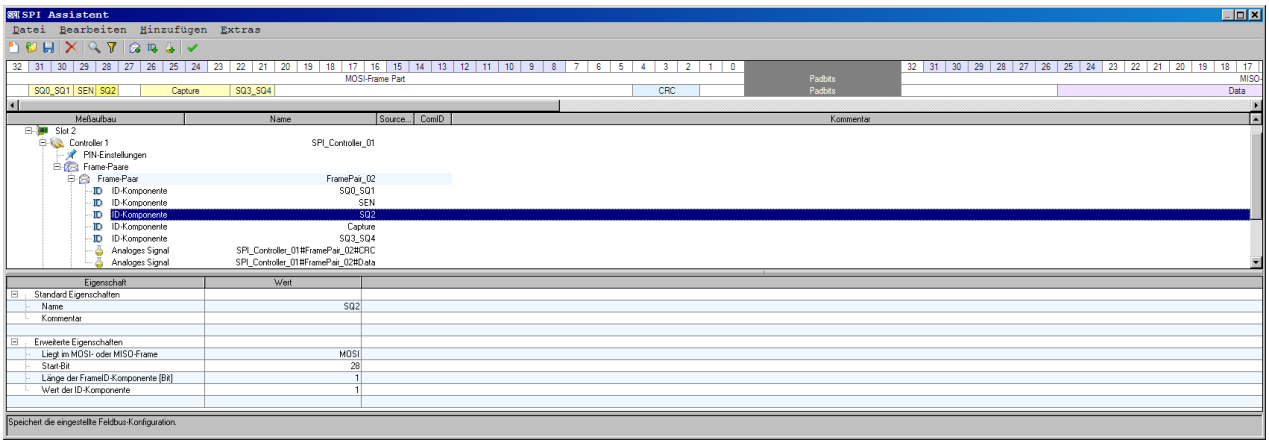
7.2.15.3.2 Chip select settings

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.



7.2.15.3.3 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.

7.2.15.3.4 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.

7.2.16 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 580 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.

7.2.16.1 Definitions

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

7.2.16.2 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

7.2.16.3 Prerequisites

Hardware:

- XCPoE Interface Slave (for sending imc channels via XCP)
- XCPoE Interface Master(for receiving XCP with the imc device)

7.2.16.4 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!

7.2.16.4.1 Variables-Binding

The Assistant for the XCPoE-Slave Module, offered by imc STUDIO, provide a global option 582 with which all of the imc system's "variable" are activated for transfer. These are embedded in an automatically generated *.A2L 582 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 "Trigger_48"! 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.

7.2.16.4.1.1 Update von imc STUDIO vor 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.

7.2.16.4.2 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.

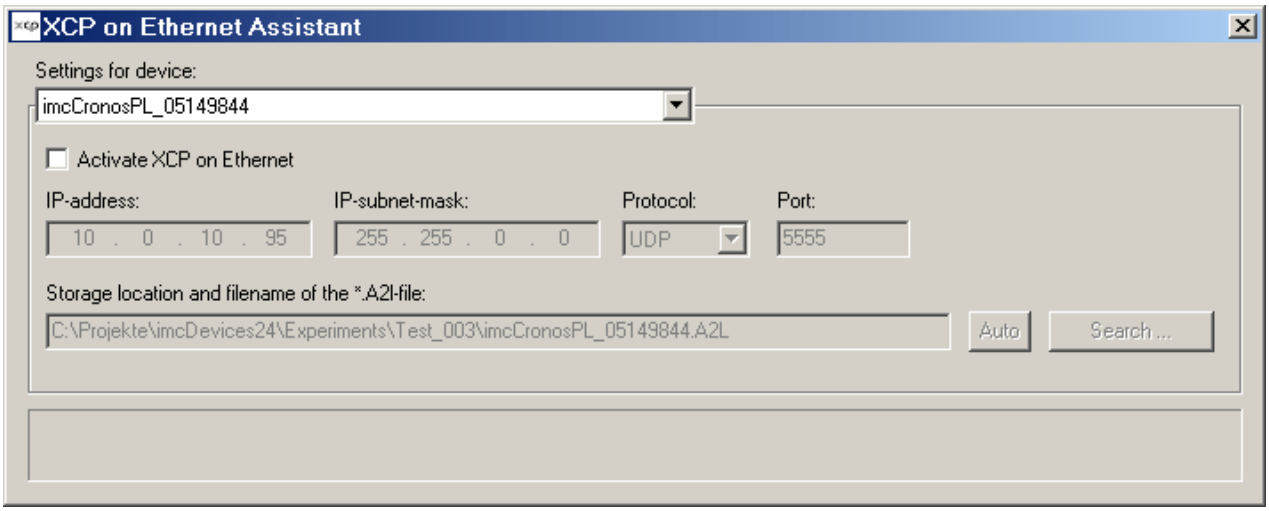
This is the case when the experiment is prepared. In the preparation process, one A2L-file per XCPoE-capable device used in the experiment is written to the storage location specified in the XCPoE-Assistant.

There, the XCP-receiver can retrieve the file.

7.2.16.4.3 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

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.

Note

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.

7.2.16.5 XCPoE Master

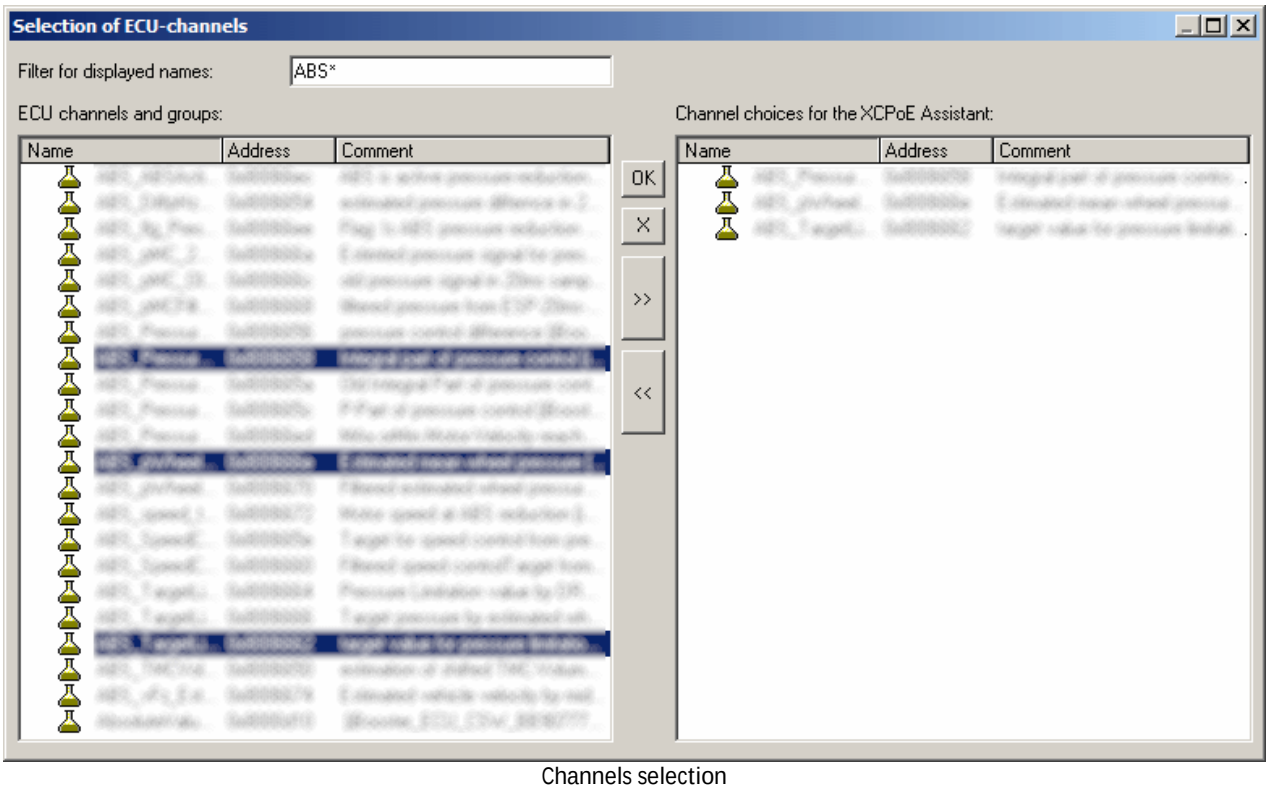
Capture of field bus channels via XCP

7.2.16.5.1 XCP on Ethernet Assistant - Master

Start the Assistant from within the imc STUDIO software via the Ribbon *Setup-Configuration* > *XCPoE-Master*.

7.2.16.5.2 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.

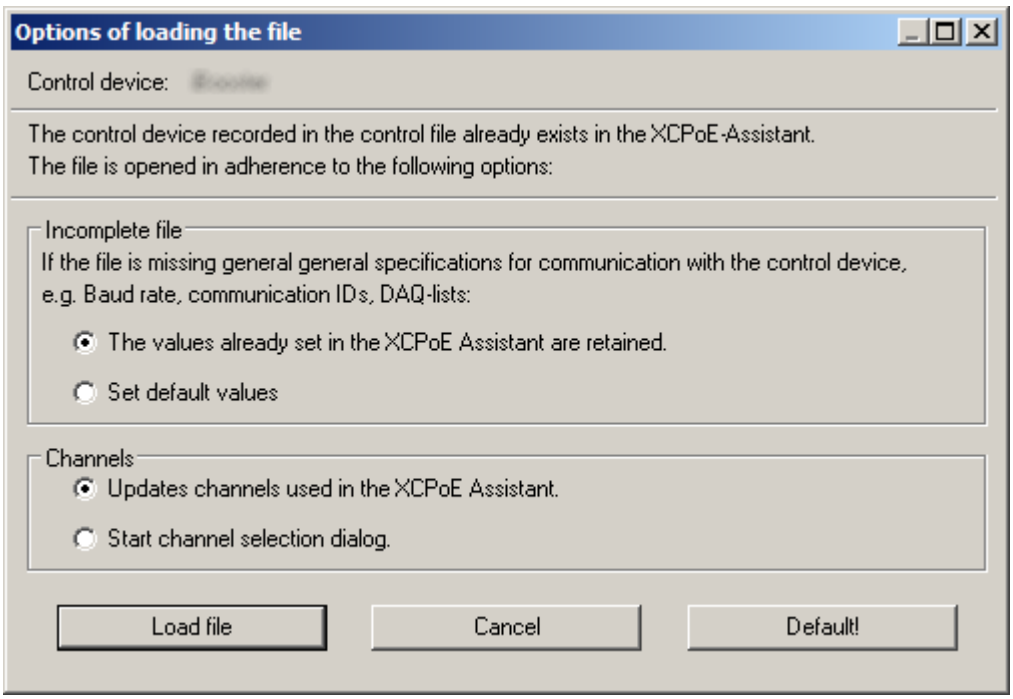


Channels selection

Note

As of imc STUDIO 5.0R8 the extended format XCPplus is also supported.

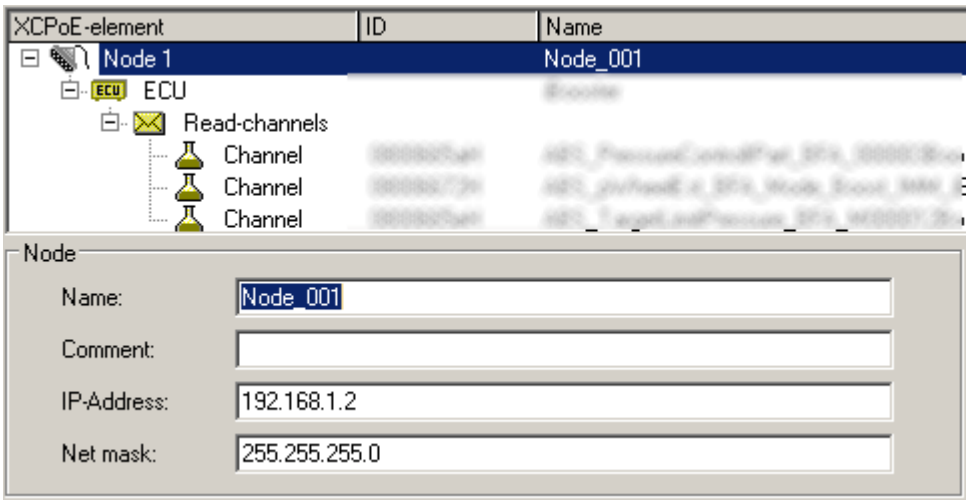
If an ECU branch exists already, an additional dialog appears initially:



Import dialog, if another ECU is already configured

7.2.16.5.3 Node settings

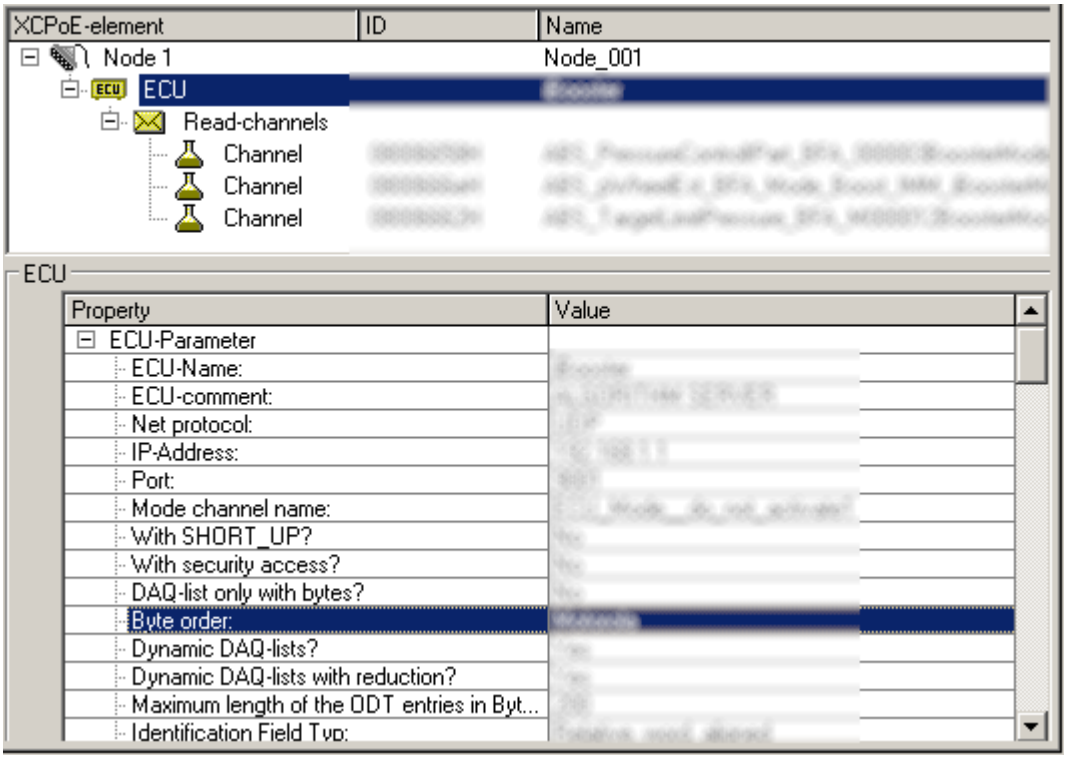
At the node, set the IP-address and the subnet mask.



XCPoE settings: node

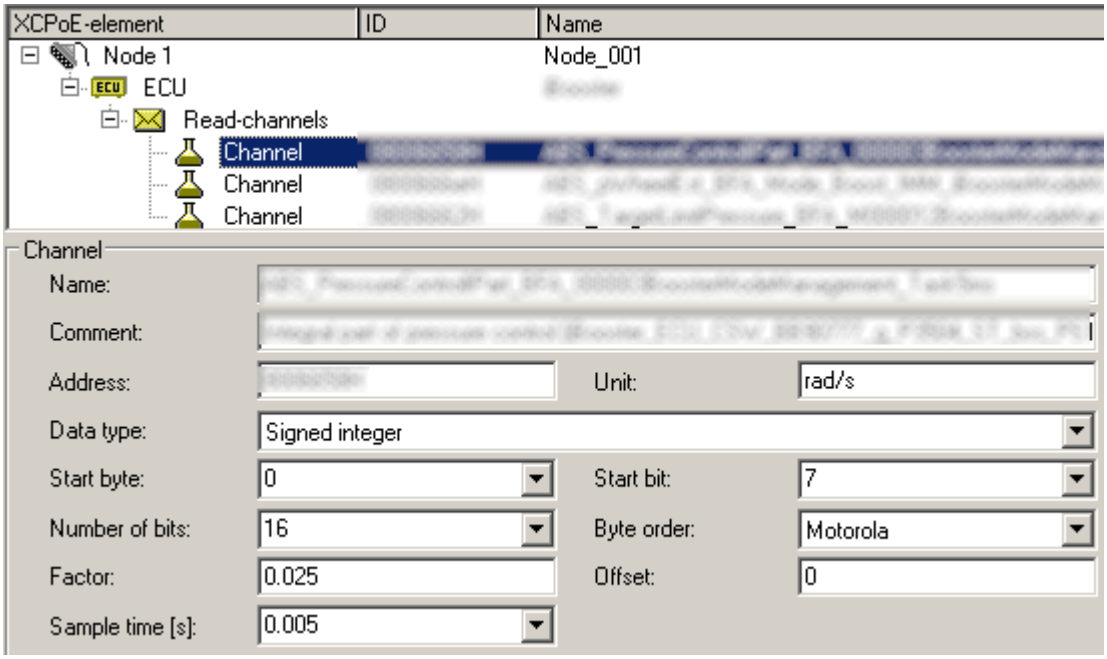
7.2.16.5.4 ECU-settings

Upon selecting the ECU branch, an editable table of the ECU's settings appears.



ECU-settings

7.2.16.5.5 Channel settings



Channel settings

7.2.17 GPS

GPS channels

All GPS signals are available directly as Field-bus channels.

7.3 Storage Options and Directory Structure

This chapter describes the different options for saving measurement data with imc STUDIO.

Data stream

Once the measurement starts, the measurement device generates a data stream.

The goal in every measurement is to process this data stream without losses and to observe and or save it.

- The data stream is first written to RAM buffer memory 592 in the device.
- This RAM buffer is read by a signal processor ("DSP").
  - If you use an imc Online FAMOS 739 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* 587).
- 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.
  - To save a channel, you must always set the appropriate options both for the channel and for the device.

Saving by the device

The device can save the measured data to the internal storage medium 592 or to a network computer 601 (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 99.

Saving and displaying data on the PC 594

Here, the data stream is transferred to the controlling PC via the connection (mostly LAN) (Transfer to the PC 587). By default, this option is activated.

The data can be displayed using the plug-in Panel (particularly the curve window); see the documentation on imc STUDIO Panel 813).

Additionally, the transferred measurement data can be saved on the PC.

By default the data are saved in the PC to a circular buffer. The use of the circular buffer prevents the data stream from totally exhausting the PC's available working memory.

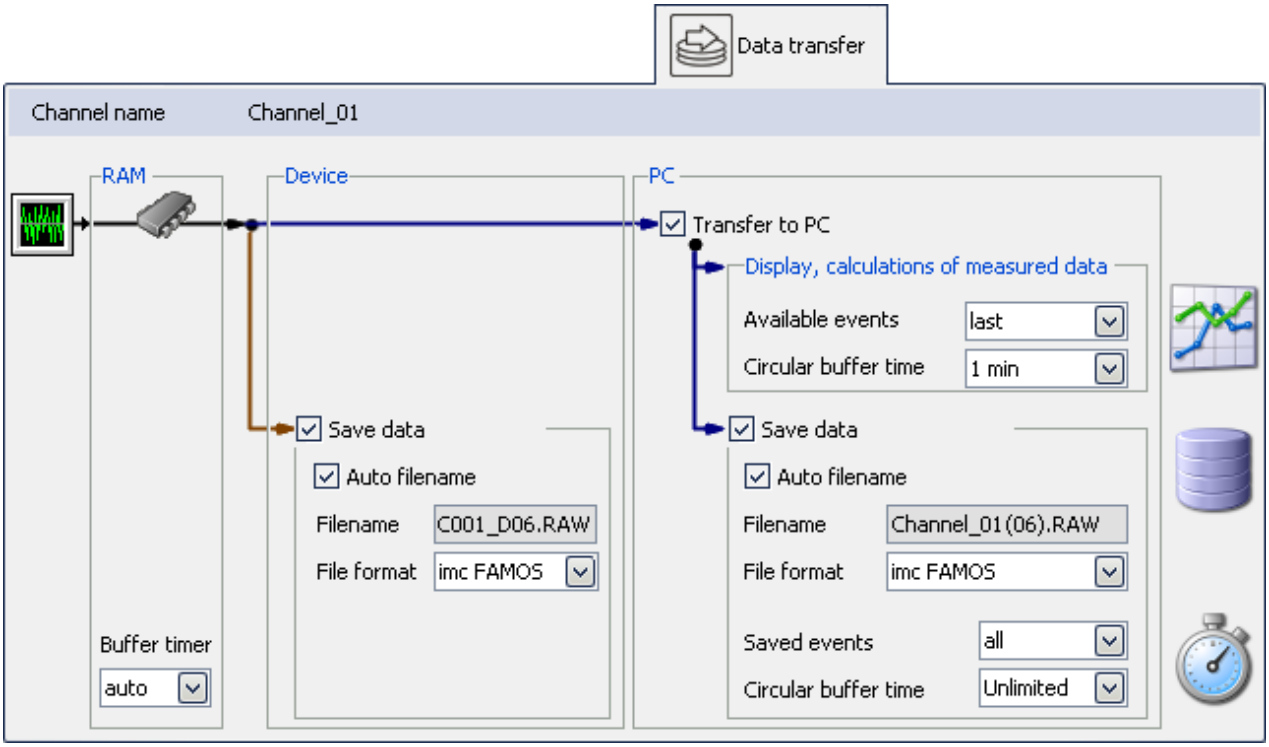
Data saving by the PC and by the device

By default, data storage is deactivated. There are a variety of data storage options:

- Name/Format 587
- Circular buffer memory 587
- Available events 587
- Interval storage 598
- Folder naming 603
- Save trigger events in individual files 603
- Storage location 592

7.3.1 Storage Options - Overview

Setting the storage options for the channels is performed in the plug-in Setup on the 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. So that the data can be saved, certain settings must be made in the user program. After restarting, the default setting is *Transfer to PC*.

imc STUDIO offers the following saving 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 (with or without Circular buffer[1])

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 (first in first out).

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 in imc Online FAMOS for processing or display in the device. No display of the measured data in the curve window or in the tool window Data Browser.

Save data:

- *Saving on the PC* 594: Saving of the measured data on the PC hard drive according to the folder structure 603 set.
- *Saving on the device hard drive* 592: Saving of the measured data device hard drive according to the folder structure 603 set. Channels using this option are recorded under the current experiment name on the device's internal drive.

7.3.1.1 Other Options

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

Circular buffer memory on the PC:

For the amount of time specified, the data are held in the PC. When using this option, the required memory is known from the beginning of the measurement.

- The circular buffer memory can be set separately for display on a Panel-page and calculation, or for data storage.

- This option is helpful in cases where you only wish to observe data on the PC or when the capacity of your PC drive is insufficient for longer measurements. Only the data most recently acquired within the circular buffer duration set are saved or displayed.
- Enter a time in seconds. You can also make a formatted entry. To do this, append one of the following suffixes to the numerical value:

s	seconds
min	minutes
h	hours
d	days

  Examples: "3 min" = 3 minutes, "2 h" = 2 hours

  If you enter *0*, circular buffer operation is deactivated and all data are saved.

Note

- Measured data intended for display and for calculations should always be handled in circular buffer mode.
- One cause of there being too few data in the circular buffer memory is an overflow of the measured data.
  Since in contrast to a continuously recording signal the circular buffer enables correctly timed display and recording of data following a data overflow, there is no overflow notification. It doesn't matter whether there was a data overflow yesterday if the circular buffer memory is one hour long. The correctly timed display of current data does, however, mean that the data from before the overflow are deleted. In this case, the circular buffer memory is not full even though the measurement duration is greater than the buffer duration.

Available / Saved events:

With this setting, you determine whether all triggered events are to be saved, or only the last one.

- If all events are saved, the result is an event-based data set. For storage of the measured data on the PC, each event can be saved to a separate file (see Save trigger events in individual files 604).
- If all events are avctivated for display and calculation, it would be possible, for instance, to display all the events in one curve window.

Note

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*.

File format:

Format of the saved measured data (see: Data file format 590).

7.3.1.2 Data Display without Data Storage

Data storage isn't necessary in such settings as system setup or trial readings. Using *Transfer to PC*, you can view the data without storing them.

7.3.1.3 Data File Format

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 590	Compressed format for many Field-bus variables
CANAlyse-MDF 591	CANAlzser-MDF format for CAN-Log data.

imc FAMOS

The file format imc FAMOS is based on the file format.

Each cannel is saved in a separate file with the extension *.RAW. RAW stands for raw data, meaning the original data returned by the measurement device.

The header in each file of measured data contains complete *version information* on imc DEVICES. The version information on imc STUDIO, imc POLARES, imc WAVE and imc COM programs is passed along with the measured data, so that it is also reflected in the respective file header. When *Saving to device* w, the devices serial number is additionally recorded.

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 (5.0 Revision 8 or higher).

Note

If you want to convert the imc measurement data into a EXCEL table or a ASCII file see the ImcFrmtCvrt program located on the imc DEVICES installation CD in *Tools*, or use the command: Format Converter 1616.

imc FAMOS ZIP

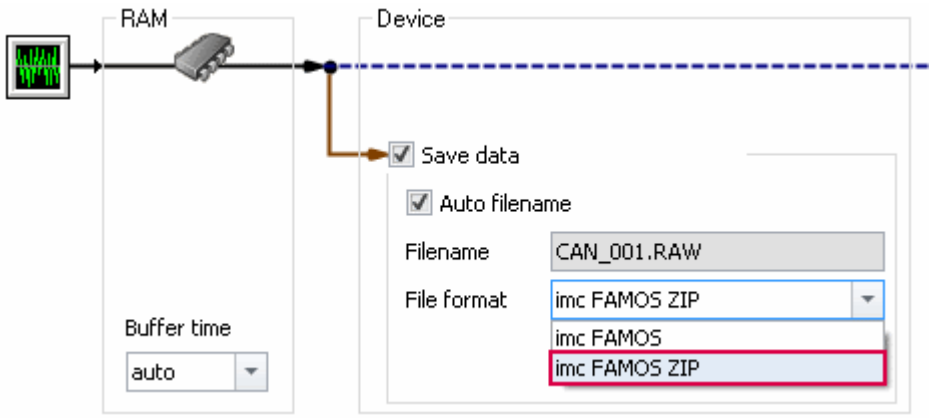
Data storage capacity consumption of Fieldbus channels on the device hard drive can be reduced by means of the file format: imc FAMOS ZIP. In the background, the file size is minimized by means of zip-compression. The result depends accordingly on the nature of the signal.

The filename remains unchanged. As well, the handling of the file, for instance with imc FAMOS is still the same.

The file can be loaded with imc FAMOS from version 7.2R4 onward. Loading with older versions can be possible in some cases when the current imc STUDIO version is installed on the computer. Contact our hotline to find out about such cases.

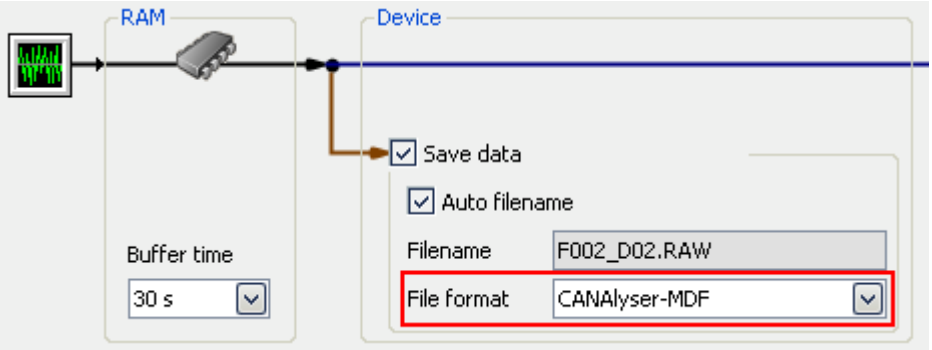
Data compression is possible for the following channel types:

- analog Fieldbus channels
- digital Fieldbus channels
- Log channels of the CAN-Field-bus



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

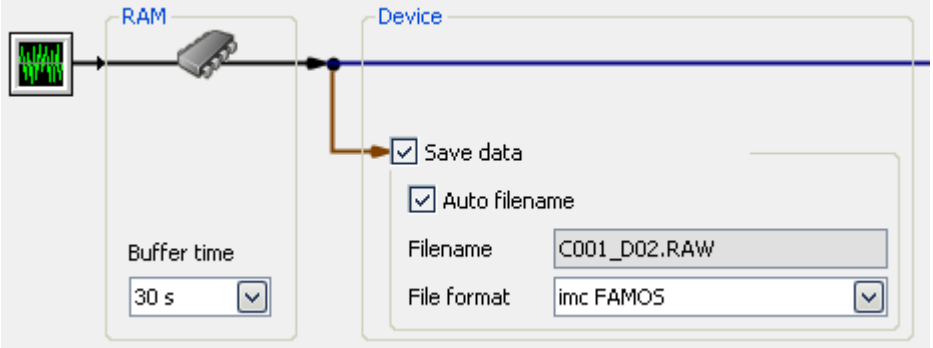
- The maximum possible measurement duration is 11 hours, 55 minutes and 49.67 seconds.
- This format is only available, if in the CAN Assistant the control *Message Log*: for the message is set to *Log CAN Bus message*.

7.3.2 Data Processing and Storage in the Device

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 Internal Storage Media 617).

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 99.



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 603 set.

Storage location

There are four 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 (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 plug-in Setup's page: Devices.

7.3.2.1 RAM Buffer Duration

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. Adjusting the setting for this time interval would make sense for working with very elaborate online programs. The same applies if you intend too switch the internal storage medium during the course of a measurement (Hot-Plug).

Example

The active channel's data are assigned a RAM buffer duration of 10s. If the PC doesn't retrieve any data from the Ethernet-interface for 10s, the buffer overflows, i.e. data to be transferred to the PC are lost. The message *Data overflow* is posted. Therefore it is best to set the buffer duration to an adequately high value.

A channel's RAM buffer volume is computed by multiplying sampling rate x buffer duration. For instance, a channel sampled at 5kHz for the recommended RAM buffer duration requires 10s 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.

Also observe the notes on the RAM buffer duration with time-stamped CAN data, in the chapter Time stamping 342 (CAN-Bus Interface).

RAM Buffer - Data Overflow on the Internal (removable) Storage Medium

Data transfer to the PC is performed by means of the same procedure which also writes data to the internal storage medium. For this reason, interrupted communication with the PC can cause a data overflow on the internal data storage no matter whether or not the channels involved are transferred to the PC.

If **connection problems** are anticipated, we recommend the following settings:

Case 1: At an aggregate sampling rate of about 50% (200kHz), a RAM buffer duration of 10s per channel is sufficient to avoid data overflow.

Case 2: With the maximum aggregate sampling rate (400kHz), it is recommended to run the measurement in autostart mode.

RAM Buffer Duration 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 duration*. Assign these channels to the same trigger.
2. Avoid multiple triggers and perform *Download* of the measurement before starting it, even if the configuration has not been changed. To force a repeat *Download*, select the Ribbon *Setup-Control > Reconfigure*.
3. Select a *RAM buffer duration* resulting in a *buffer volume less than 65,536*. Example: 20kHz sampling rate: 65,536 / 20,000= 3.3s. You can use the keyboard to enter this value in the input box next to the suggested values. At very high sampling rates, the measurement system may need to be run in autostart mode in order to avoid data overflow; see "RAM Buffer - Data Overflow on the Internal (removable) Storage Medium" 593, above.

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 duration, the RAM buffer is organized in blocks.

Example

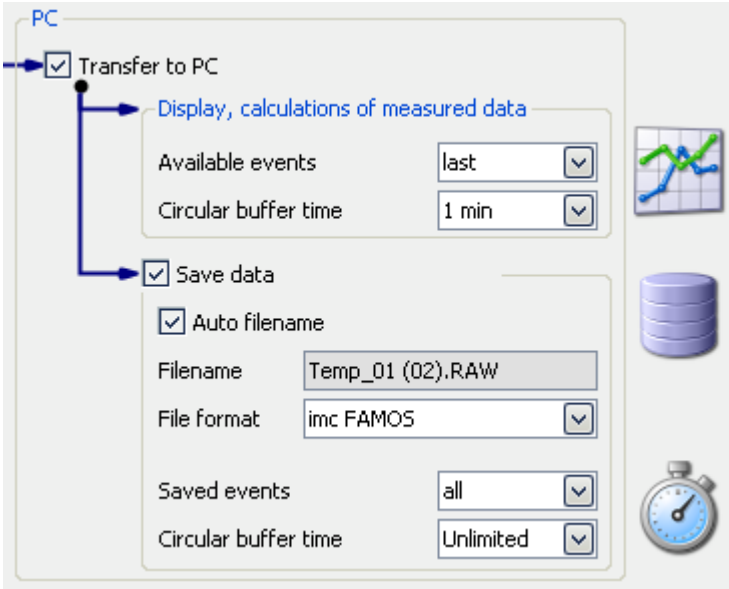
A sampling rate of 10kHz x 10s buffer duration results in 100,000 > 65,536 samples. Here, the RAM buffer must be organized in 50,000 blocks of 2 samples apiece. This results in a tolerance in the addressing of magnitude block size-1, in this example, 1 sample.

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.

7.3.2.2 Data Transfer

The transfer of the internal data to the PC is described in chapter Device Hard Disk, removable drive 621.

7.3.3 Data Processing and Storage on the 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 are not saved automatically, but rather are available for display in the plug-in: Panel 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 permanent folder structure together with the experiment.

After conclusion of the measurement, display the data thus saved using the plug-in: Panel or with imc FAMOS.

Note

The measurement data is saved with the experiment. Make sure, that there is enough free memory on the selected drive for the experiments and for the expected measurement data.

Reference

- Information on the options for display and storage of the measured data is presented in the channel: Storage options - Overview 587.
- Information on display of the stored measurement data is presented in the chapter: Viewing saved measurement data 616.

7.3.3.1 Introduction (PC Data Storage)

An Experiment constitutes the framework of every imc STUDIO measurement procedure. Among other things, it comprises a record of all settings for the plug-in imc STUDIO Setup such as the device selection, parameterization of the channels, etc. Experiments are saved on the PC hard drive in the imc STUDIO database.

Definitions

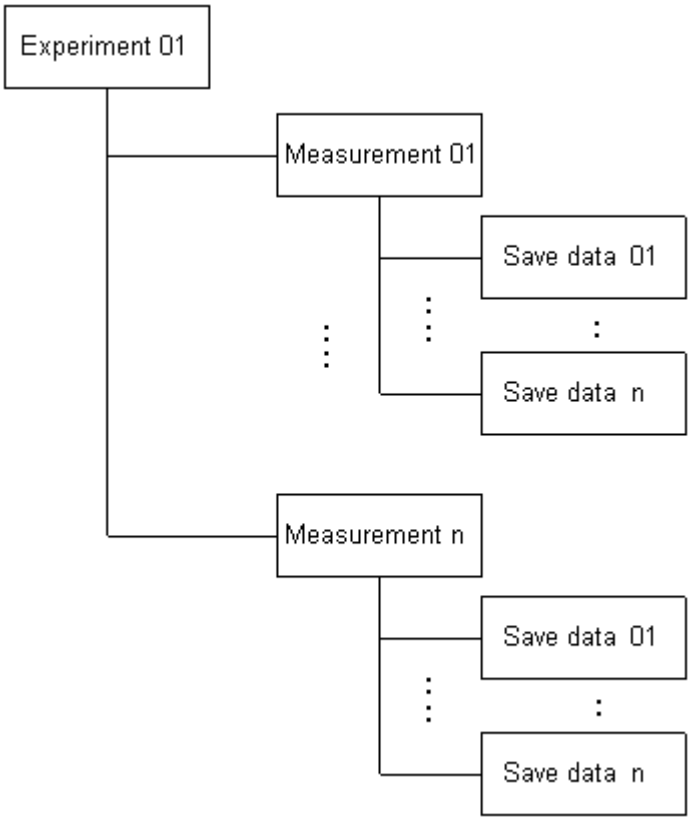
Experiment: An experiment contains all information needed for restoring a measurement

Measurement: In each experiment, multiple "measurements" can be taken. This means that a particular configuration has been set up and it is used to take several measurements in succession.

- The names of the measurements are either numbered incrementally or adopt the time and date. (See Data storage and Directory 603 Structure 603)

7.3.3.2 Tree Structure

The following directory structure is used when saving data:



Information on the folder structure for saving the measurement results is presented in the chapter: Data storage and Directory Structure 603.

7.3.3.3 Overview of PC Storage and Display Options

--- = Option is ignored
0/1 = deactivated / activated

SL = Segment length (length of the FFT or similar)
MS = Matrix size
HS = Histogram size

Analog, digital, or other equidistantly sampled or time-stamped channels (e.g data-reduced channels)

- For display and calculation (with or without storage of measured data)

Triggering	Circular buffer		Available events
Multiple triggering?	Circular buffer	Time	Last or All
No	0	---	---
	1	any (6)	---
Yes	0	---	Last / All (5)
	1	any (6, 7)	Last (1)

- With storage of measured data

Triggering	Circular buffer		Saved events	Storage interval	
Multiple triggering?	Circular buffer	Time	Last or All	Storage interval	Storage interval count
No	0	---	---	any (6)	1..n or all
	1	any (6)	---	any (6)	1..n or all
Yes	0	---	Last / All	End of measurement	1..n or all
				intervals (3)	---
	1	any (6, 7)	Last (1)	End of measurement	1..n or all
				--- (8)	---

FFT or similar virtual channels

- For display and calculation (with or without storage of measured data)

Triggering	Circular buffer		Available events
Multiple triggering?	Circular buffer	Time	Last or All
No	0	---	---
	1	n*SL	---
Yes	0	---	Last / All (5)
	1	n*SL	Last (1)

- With storage of measured data

Triggering	Circular buffer		Saved events	Storage interval	
Multiple triggering?	Circular buffer	Time	Last or All	Storage interval	Storage interval count
No	0	---	---	any (6)	1..n or all
	1	n*SL	---	any (6)	1..n or all
Yes	0	---	Last / All	End of measurement	1..n or all
				intervals (3)	---
	1	n*SL	Last (1)	End of measurement	1..n or all
				--- (8)	---

Matrix or histograms

- For display and calculation (With or without storage of measured data)

Triggering	Circular buffer		Available events
Multiple triggering?	Circular buffer	Time	Last or All
No	---	---	---
Yes	0	---	Last / All (5)
	1	MS or HS	Last (1)

- With storage of measured data

Triggering	Circular buffer		Saved events	Storage interval	
Multiple triggering?	Circular buffer	Time	Last or All	Storage interval	Storage interval count
No	---	---	---	End of measurement	1..n or all
				intervals (3)	---
Yes	0	---	Last / All	End of measurement	1..n or all
				intervals (3, 4)	---
	1	MS or HS	Last (1)	End of measurement	1..n or all
				--- (8)	---

(1) For channels with multi-triggering, circular buffering pertains to the last triggering. The circular buffer IS NOT applied to multiple triggered measurement runs!
The option *Last or all shots* is the simplified form of the option *Circular buffer of size of one trigger event*.

(2) Upon saving, all trigger events are available and thus can be displayed. In the curve window, the display modes All/last/some events can be used.

(3) Multiple saving intervals during a measurement are not possible for multi-triggered channels, matrices and histograms. All data are saved to the first folder set up and the file is closed at the end. With triggered channels in the circular buffer and saving intervals on the PC, only the last event is saved!

(4) Histograms and matrices are saved at the saving interval set for the channel in the first folder created.

(5) The default setting for channels with endless triggering is 'Display of last event'. By this means, overflow of memory during long measurements is to be avoided. The user can switch to display of all events. By "display" what is mainly meant is "remembering", because when you switch between Display all and Display last events, it ONLY affects display. You can switch as often as desired during a measurement. However, the setting for the channel is fixed throughout the entire measurement; only the last /all events are available, independent of the curve window settings.

(6) 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.

(7) 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.

(8) Interval storage is ignored for channels saved to a circular buffer.

7.3.4 Saving Interval

Saving interval - What is it (for)?

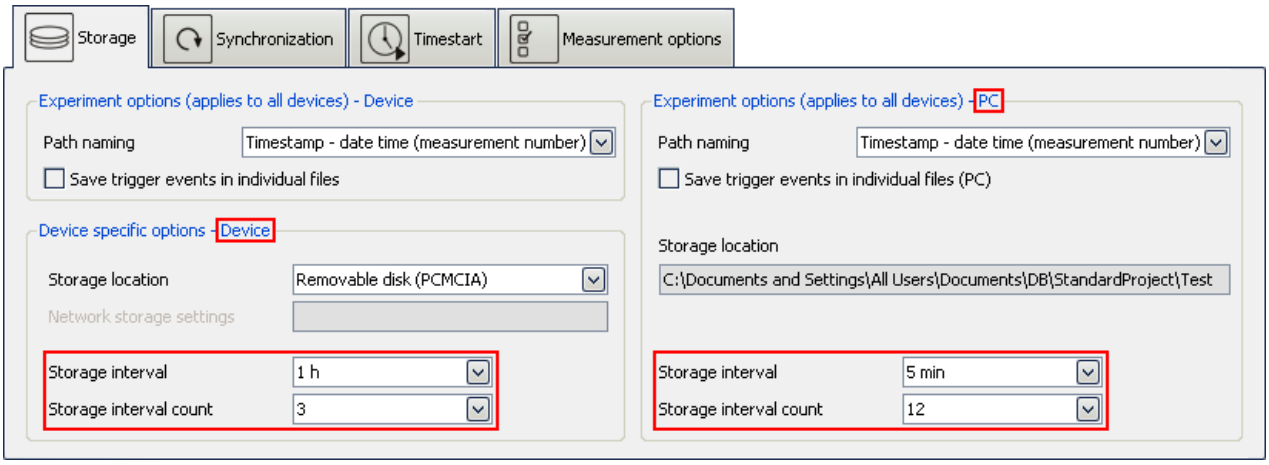
Upon the elapse of a certain time interval, a running data set is closed and a new file created.

- 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 plug-in Setup's 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 599: The interval saving can be set for each device separately.
- On the right side, you see the settings for storage on the PC: the storage interval set applies to all the devices.
- 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 587 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.

Important Notes: Storage interval count

- Once the measurement is done, you will normally obtain one save interval more of data than was set.
- When interval storage is active, all folders are deleted except for the specified number of intervals!
- If you download 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.

7.3.4.1 Saving Interval on the Internal Device Storage Medium

For devices as of the group 2 and higher

- By means of the settings *Storage interval* and *Storage interval count*, you specify a defined storage space.
- In case there is no space available before the number of intervals has been reached, the devices switch to circular buffer operation. For instance, if the number of intervals is 30 and the storage medium space is not enough, a circular buffer operation with fewer intervals will be executed.
  - 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 605.

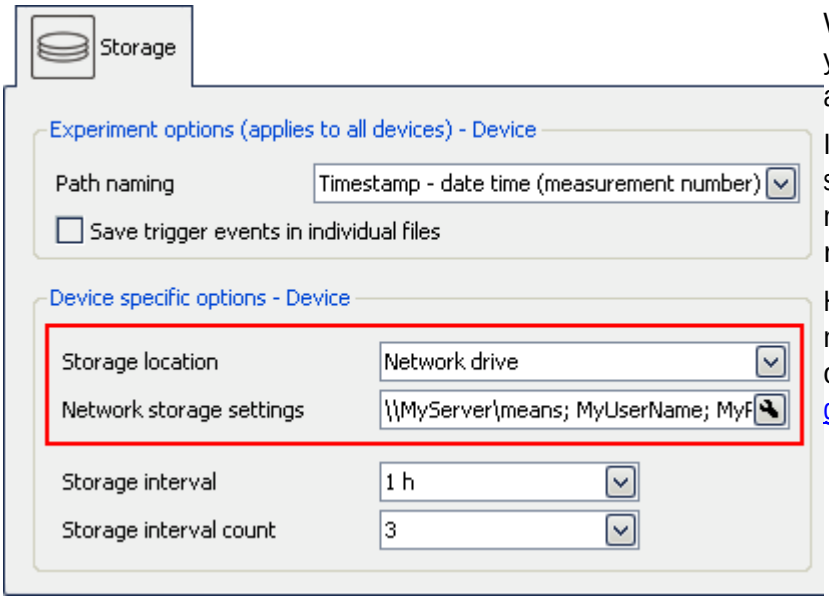
7.3.4.2 Standardization of Intervals

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.
- If channels haven't been triggered during an interval, the interval folder remains empty and will be deleted when the next interval folder will be created.

7.3.5 Saving Data on the Network



"Storage" dialog
Network storage

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. (Devices as of group 4 88 and higher)

The data are not saved in the device, but on a centralized data collector.

Systems which use the Common Internet File System (CIFS), such as LINUX, WIN2000 and Windows XP/Vista/7/8, can be addressed as the target location.

In order to set data saving on a network drive, open the dialog: Storage on the plug-in Setup's 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.

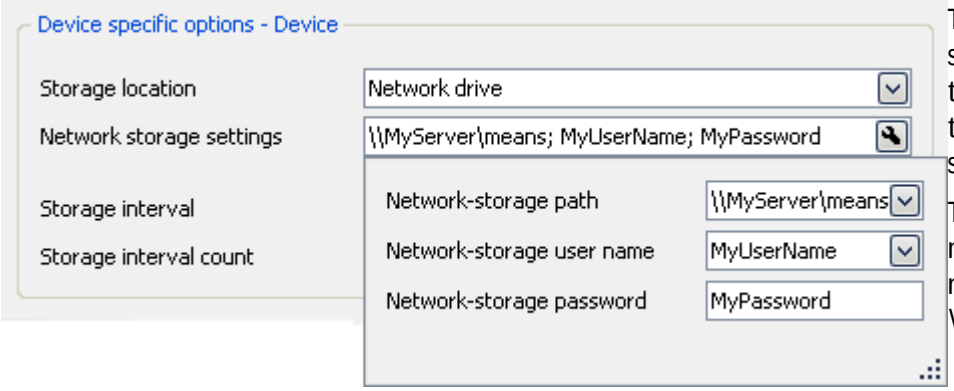
Advantages:

- 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 (imc BUSDAQ 88).
- 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:

- Devices as of group 4 and higher 88
- Sharing of a network drive with writing permissions
- Secure network
- Target system using the Common Internet File System(CIFS): LINUX, WIN2000 and Windows XP/Vista/7/8

7.3.5.1 Setting



Dialog "Storage"
Network storage settings

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\).

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 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.

7.3.6 Data Storage and Directory Structure

To change the folder structure in which measurement results are saved, open the dialog: Storage in the plug-in Setup's page: Devices.

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.

Note

- 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.
- If a smaller data overflow occurs than as under Point 1 above, the data already stored in the file are overwritten, leaving only those data stored after the overflow. These data do have the correct time stamp, however.
- 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. *2014-08-05 14-30-00 (1).002*.
  If this new file, too, reaches 2 GB in size, then a new file is created again, this time with the suffix 002.

7.3.6.1 Designation of the Measurement Data Folders

By default, the folders are named along with their date and time. Additionally, there can be a running number. It is also possible to create subfolders for each trigger event 604.

Using the parameter Path naming, you set the folder structure for the measured data.

Timestamp - date time (measurement number)

The path name is formed from the start time

\2011-06-12 17-01-30 (1)	\Channel_001.raw
	\Channel_002.raw
	\Channel_003.raw
	\---.raw
\2011-06-16 14-01-30 (2)	\---.raw
\Starttime (measurement number)	\Filename.raw

Continuous numbers

\00000001	\Channel_001.raw
	\Channel_002.raw
	\Channel_003.raw
	\---.raw
\00000002	\---.raw
\Number	\Filename.raw

Notes on parallel saving on the PC and in the device

If parallel data storage both in the device and on the PC is selected, it makes sense to provide identical names to the two corresponding measurement data folders.

This is accomplished by means of the path naming selection: "Timestamp".

With the path naming option: "Continuous numbers", discrepancies can occur.

1. The consecutive numbering reflects existing measurement data folders. For example, if some measurements were deleted from the PC, there can be discrepancies in the naming of subsequent measurements.
2. In the initialization (downloading) process, the system finds the highest number used. Then when the measurement is started, the next higher number is used.
   The process of data storage on the PC includes inspecting upon each start which number is the highest. For instance, if following a measurement, the folder with the highest number is deleted, then this number is used again upon the next start.
   Searching the storage medium on the device for the highest number - before each start - cannot be performed quickly enough. For this reason, the number is incremented for each measurement. Even if the last measurement was deleted. The system only checks what the highest number is upon the next initialization.

This can lead to different results if measured data are deleted after a measurement.

7.3.6.2 Save Trigger Events in individual Files

The option: Save trigger events in individual files lets you set the system's data saving behavior with triggered Events:

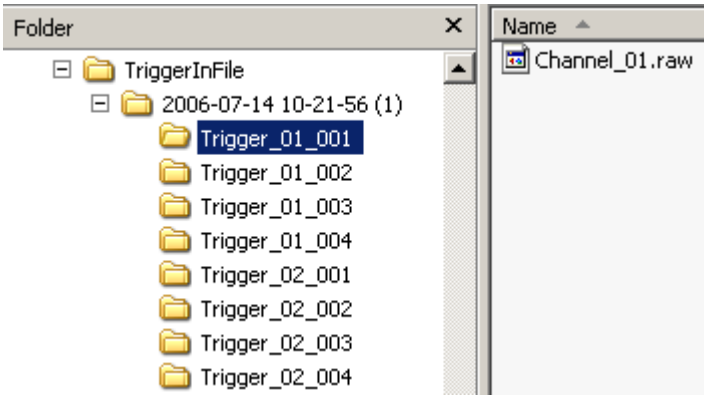
Not activated	Activated
Every triggered event is saved to the same file.	Each occurring trigger event is saved to a new file.
• If multiple triggered events are saved (see: *Storage Options* > *Other options* 588) , an event-based data set is generated. • If only the last event is saved, the previous trigger events are deleted.	• 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.

Each trigger event will be saved in an individual folder:

\2011-06-12 17-01-30 (1)	\Trigger_48_0001\Channel_001.raw
	\Trigger_01_0001\Channel_002.raw
	\Trigger_01_0002\Channel_002.raw

	\Trigger_02_0001\Channel_003.raw
\2011-06-16 14-01-30 (2)	\Trigger_48_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 and limitations

- All triggers with the combination: *1-Trigger* are always saved in the folder of *Trigger_48*. They are not assigned a separate folder.

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, when 1000 trigger folders accumulate, a new measurement data folder is is created.

7.3.6.3 Supplemental Files: DirClosed and imcSyslog.txt

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

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 626 occurs during online processing or when saving the data, this information must be saved in the device.

Such information is saved in the file imcSyslog.txt, 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 2.8R3 SP6 (8.4.2013)
Nothing to report!
```

Structure of the file in case of data overflow

```
imcSyslog Version 1.0
2010-12-21 11:08:06.929190 M#:Data overflow on disk! F#:V006_G01.raw E#:1 FE#:0 O#:206 L#:1253
2010-12-21 11:08:08.147645 M#:Data overflow on disk! F#:V006_G01.raw E#:1 FE#:0 O#:1459 L#:1800
2010-12-21 11:08:11.029787 M#:Data overflow on disk! F#:V006_G01.raw E#:1 FE#:0 O#:3259 L#:1113
2010-12-21 11:08:16.071192 M#:Data overflow on disk! F#:V006_G01.raw E#:1 FE#:0 O#:6918 L#:1304
2010-12-21 11:08:22.906673 M#:Data overflow on disk! F#:V006_G01.raw E#:1 FE#:0 O#:10111
L#:1054
2010-12-21 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.

2010-12-21 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. 2010-12-21 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

7.3.6.4 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.

Note

Once all files have been concluded, the system creates an additional file DirClosed 605. As soon as it is present, you have reliable access to the measured data.

7.3.7 Saving - Miscellaneous

7.3.7.1 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.
- 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 1536 byte* + 1 x 2000 x 2 byte = 5,5 KB => 8 KB 1536 byte* + 1 x 2000 x 4 byte = 9,5 KB => 16 KB	 8 KB 8 KB 16 KB
Next events: 512 byte* + 1 x 2000 x 2 byte = 4,5 KB => 8 KB => 99 x 8 KB 512 byte* + 1 x 2000 x 2 byte = 4,5 KB => 8 KB => 99 x 8 KB 512 byte* + 1 x 2000 x 4 byte = 8,5 KB => 16 KB => 99 x 16 KB	 792 KB 792 KB 1584 KB
1 cluster for directory and 3 file allocation infos	8 KB
True memory requirement:	= 3208 KB

* File header for trigger events (size is variable)

7.3.8 Controlling Data Storage

7.3.8.1 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 menu item *Project > Save current measurement data/as*.

Constraints

- 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 174 are applicable. The storage type: *Display, calculation of measured data* is used.

Save current measurement data

By default, the measured data are saved in the database. If the automated 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 automatically and whatever measurement results are already available at the time are saved.

Save current measurement data as

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.

Options

In the Options 66, 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.

- 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

  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.

7.3.8.2 Suspend and Resume Data Storage

It is possible to control the data storage to the devices and to the PC hard drives.

The current folder can be concluded - this stops data recording of all channel data. Or, data recording for all channels can be resumed.

There are a number of ways to interrupt/resume the data storage:

- Open the Devices menu and click on the item Suspend data storage or Resume data storage
- Click on the button or in the toolbar of the plug-in Setup.
- By means of the command: Execute menu action 1604 > Devices > Suspend (Suspend data storage) or Resume (Resume data storage)

Note

Data storage con only be controlled if it was previously activated.

7.3.8.3 User-defined Measurement Storage Area

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 603 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* > *Project Management* > *Measurement storage area* 63.

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. For example, the *Path naming* (e.g. *Continuous numbers*). 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

Measurement storage path

- By default, this is the Experiment-path. This denotes the folder in which all measurements are saved (according to the measurement folder structure specified).

Measurement folder structure

- This refers to the structure below the measurement storage folder. For instance, folders are set up here according to the settings for *Path naming* (e.g. *Continuous numbers*).

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 *Path naming* (e.g. instead of *Continuous numbers*, the time stamp is displayed)

Use a user-defined measurement storage path

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 of measurement storage paths

Example 1

C:\Measurement\<EXPERIMENT.NAME>\Results

- Measurement results for the experiment: *Experiment_0001* are saved as follows, for example:
  C:\Measurement\Experiment_0001\Results\2013-01-01 08-00-00 (1)
  C:\Measurement\Experiment_0001\Results\2013-01-01 09-00-00 (2)
  C:\Measurement\Experiment_0001\Results\2013-01-01 10-00-00 (3)

Example 2

C:\Measurement\<EXPERIMENT.NAME>

- Measurement results for the experiment: *Experiment_0002* are saved as follows, for example:
  C:\Measurement\Experiment_0002\2013-01-02 08-00-00 (1)
  C:\Measurement\Experiment_0002\2013-01-02 09-00-00 (2)
  C:\Measurement\Experiment_0002\2013-01-02 10-00-00 (3)

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\Results\2013-01-03 08-00-00 (1)
  C:\DB\StandardProject\Experiment_0003\Results\2013-01-03 09-00-00 (2)
  C:\DB\StandardProject\Experiment_0003\Results\2013-01-03 10-00-00 (3)

Use a user-defined measurement folder structure

Default setting: `\<STORAGE.FOLDERNAME>\`

- Returns the experiment's data saving settings (e.g. the *Path naming* such as *Continuous numbers*)

If you specify a different folder structure, all measurement results will be save according to the specified folder structure.

Useful placeholders for the measurement folder structure:

- `<STORAGE.FOLDERNAME>`

  Generates a folder from the data storage settings (e.g. *2013-01-01 08-00-00 (1)*)

  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. *2013-01-01 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 of measurement folder structure

In all examples, no user-defined data storage folder for the measurements 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\2013-01-01 08-00-00 (1)
  C:\DB\StandardProject\Experiment_0004\Results\2013-01-01 09-00-00 (2)
  C:\DB\StandardProject\Experiment_0004\Results\2013-01-01 10-00-00 (3)
- In the Data Browser, the results are displayed as follows:
  Results\2013-01-01 08-00-00 (1)
  Results\2013-01-01 09-00-00 (2)
  Results\2013-01-01 10-00-00 (3)

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\2013-01-02 08-00-00\2013-01-02 08-00-00 (1)
  C:\DB\StandardProject\Experiment_0005\2013-01-02 08-00-00\2013-01-02 08-01-00 (1)
  C:\DB\StandardProject\Experiment_0005\2013-01-02 08-00-00\2013-01-02 08-02-00 (1)
  C:\DB\StandardProject\Experiment_0005\2013-01-05 08-00-00\2013-01-05 08-00-00 (2)
  C:\DB\StandardProject\Experiment_0005\2013-01-05 08-00-00\2013-01-05 08-01-00 (2)
  C:\DB\StandardProject\Experiment_0005\2013-01-05 08-00-00\2013-01-05 08-02-00 (2)
  C:\DB\StandardProject\Experiment_0005\2013-01-10 08-00-00\2013-01-10 08-00-00 (3)
  C:\DB\StandardProject\Experiment_0005\2013-01-10 08-00-00\2013-01-10 08-01-00 (3)
  C:\DB\StandardProject\Experiment_0005\2013-01-10 08-00-00\2013-01-10 08-02-00 (3)
- In the Data Browser the results are displayed as follows:
  2013-01-02 08-00-00\2013-01-02 08-00-00 (1)
  2013-01-02 08-00-00\2013-01-02 08-01-00 (1)
  2013-01-02 08-00-00\2013-01-02 08-02-00 (1)
  2013-01-05 08-00-00\2013-01-05 08-00-00 (2)
  2013-01-05 08-00-00\2013-01-05 08-01-00 (2)
  2013-01-05 08-00-00\2013-01-05 08-02-00 (2)
  2013-01-10 08-00-00\2013-01-10 08-00-00 (3)
  2013-01-10 08-00-00\2013-01-10 08-01-00 (3)
  2013-01-10 08-00-00\2013-01-10 08-02-00 (3)

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

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.

Why do changes to the metadata displayed at the start of the measurement (via the Metadata Assistant) not affect the path?

Answer: The measurement data folder is determined upon clicking on the Start button. The dialog for entering the metadata is displayed shortly before the measurement starts. This is at a later time.

In consequence, changes made in this way are not yet available at the right time and can not be processed. Instead you can use, for example, the command: *Display Setup as dialog* or alter the columns directly via the Setup page. Then start the measurement.

You can also do this automatically by means of commands linked to a Panel-button.

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.

7.3.8.4 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.

7.3.9 Viewing Saved Measurement Data

imc STUDIO enables the viewing of measured data during the running measurement by means of the Panel. As well, it is possible to view measurements previously saved to the PC (by means of the tool window Data Browser) or to the device drive (by means of Command Load/Import Variable 1647 via the Data Browser).

7.4 Device Hard Disk, removable drive

This chapter describes details of the internal data carriers of imc measurement devices and how to operate them with imc STUDIO.

7.4.1 Internal Storage Media

For saving measured data, all devices support an internal storage medium. Depending on the device type, this storage medium could be a removable drive, e.g. PCMCIA-HD, PCMCIA-Flash-Card, CF-Card, ExpressCard or USB-storage medium. Card-Bus interface are not recommended. Some devices can be equipped with an hard drive (see Device overview 88).

The internal storage medium is exclusively for data acquisition with imc STUDIO. Removable storage media can of course be purchased subsequently. The hard drive is not a removable drive, it is ordered along with the device and can only be retrofitted by imc.

Note

The functional capabilities of imc STUDIO depend on both the model and production date. A list of all device types and their distinctive properties is presented in a list in in the Device Overview 88.

Swapping the Storage Medium

Pressing the LED-button 618 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. And if you were to remove the storage medium while a measurement is running, the waveforms would be incomplete! Therefore, always proceed as follows when swapping the storage medium:

1. Press the LED-button.
2. Once the button blinks, remove the storage medium.
3. Insert the new data carrier. Devices indicate by a short flash that the new drive has been successfully recognized.

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-button. 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-button 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***.

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 592"). 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 LED-button, in order to avoid data loss and damage to the storage medium. The LED button will **shine continuously** in green.
2. Once the device is ready for removal of the storage medium, the LED-button **blinks**.
3. Remove the already full storage medium.
4. No announcement is necessary for inserting a storage medium.

Note

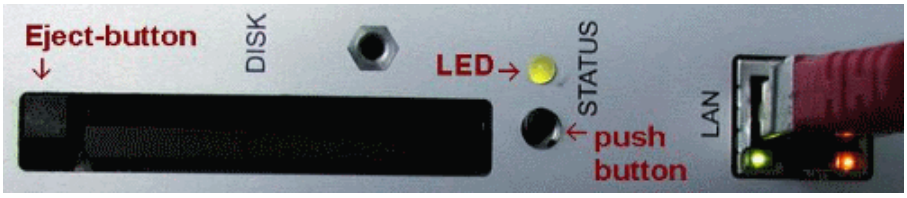
The Hot-Plug function is not possible with *Compact Flashcards (CF)* if the device is equipped with a PCMCIA slot, too! To exchange the CF-Card, the device must be deactivated! Otherwise the data carrier may be damaged!

7.4.1.1 LED-Button / Push Button

The LED-button at the storage medium slot serves multiple purposes.

PCMCIA-slot

With devices having a Compact Flash slot, ExpressCard or USB, the function is divided between a push-button and an LED.



Compact Flash slot

7.4.1.2 Memory Volumes

Storage media larger than 16 GB are possible as of group 2 an higher (see Device Overview 88), but see the notes on the use of the storage media 626. When the formatting is FAT32, the maximum address space is 8.8TB.

7.4.1.3 CF Cards (Compact Flash)

For devices of group 2 and 3 88:

- *CF-CARD in the PCMCIA slot:* The devices are in principle designed for CF-Cards in the PCMCIA-adapter. However, due to the large number of CF-card manufacturers, correct functioning cannot be assured for all of them. When in doubt, you can consult with the imc-Hotline.
- *CF-CARD at the internal IDE port:* If the PCMCIA-slot is already being used for a WLAN connection, a CF Flash Card can be connected at the internal IDE port using the *CF-Adapter (SL 836)*.
  This option is subject to the following constraints:
  - With CF Cards in the CF-Adapter, no *Hot-Plug* 618 function is possible. It isn't sufficient to just stop the measurement! The device must be deactivated in order to exchange the CF Card!
  - Only CF cards supporting the *PIO mode 4* work in the CF-adapter. These CF cards (as of Summer, 2004) work with the so-called "J"-controllers, e.g. *SimpleTech SLCFxxxJ* and *SanDisk SDCFB-xxx* or *SDCFBI-xxx* for -40 to 85°C. The previous versions *SimpleTech SLCFxxxH* do not support this! We recommend the use of *SanDisk SDCFB-xxx*, since these also support MW-DMA.
  - To avoid data loss, CF cards which don't support the PIO Mode 4 are neither recognized nor displayed.
  - The card appears in the software interface as an IDE hard drive

For devices of group 4-6 88:

The device group exclusively uses CF cards for storage medium.

7.4.1.4 USB Storage Medium

Concerning devices with USB (see Device overview 88). Memory sticks or external hard drives can be connected at this terminal.

Remarks:

- Always set only one storage medium unit! Devices belonging to group 6 88 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* 618 functionality is available. Make sure that sufficient time is available for swapping the data carrier. How much time is required for de-registering and re-registering with the system depends on the particular data carrier and on the number of channels. As an orientation value, we recommend at least 30s, even for simple configurations!

7.4.1.5 ExpressCard

Concerning devices with ExpressCard slot (see Device overview 88).

Remarks:

- Remove any connected USB storage medium! Devices belonging to group 6 88 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* 618 functionality is available.

7.4.1.6 CFast

Concerning devices with CFast slot (see Device overview 88).

Remarks:

- 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* 618 functionality is available.

7.4.2 Data Transfer

For devices as of group 2 and higher the Microsoft Explorer is used.

a) Retrieving data with the PC connected

Normal procedure. This is the fastest method for small data volumes.

- Using the Explorer, copy the data from the internal storage medium.
- Erase the storage medium.
- If appropriate, run Scandisk.

b) Retrieving data via modem

Remote data retrieval. Can take a long time if the data volume is correspondingly large.

- Like a) only connected via modem

c) Retrieving data from the storage medium of a disconnected device (measuring under Autostart) (see Hot-Plug 618)

E.g. in long-term remote operation. Copy data using another imc measurement device a) or in the PC d).

d) Copying storage medium via the card reader of a PC

Suitable for large data volumes due to the high-speed transfer.

1. Insert storage medium into the PC-slot.
2. Use Microsoft Explorer to copy the storage medium's data.
3. Erase the storage medium.
4. If appropriate, run Scandisk.
5. Insert storage medium back into the imc device.

Warning

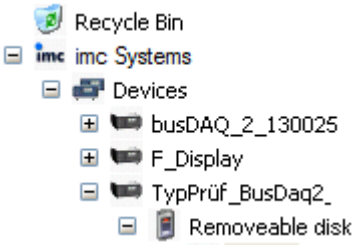
Do not use force to insert or remove the device storage medium.

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 chapter Saving Interval 598).

7.4.2.1 Shell Extension

For devices as of group 2 88 and higher:



While installing the operating software, if the option "*Windows explorer extension*" is activated, you are able to copy, display or delete the files of measured data saved on the device's removable and/or hard drive. 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 Adding a Device 622). Then, double-click on the entry for the device. Two additional branches appear for the hard drive and the removable disk. Now it is possible to work with both of these storage media just like with any other partition on your computer.

Note

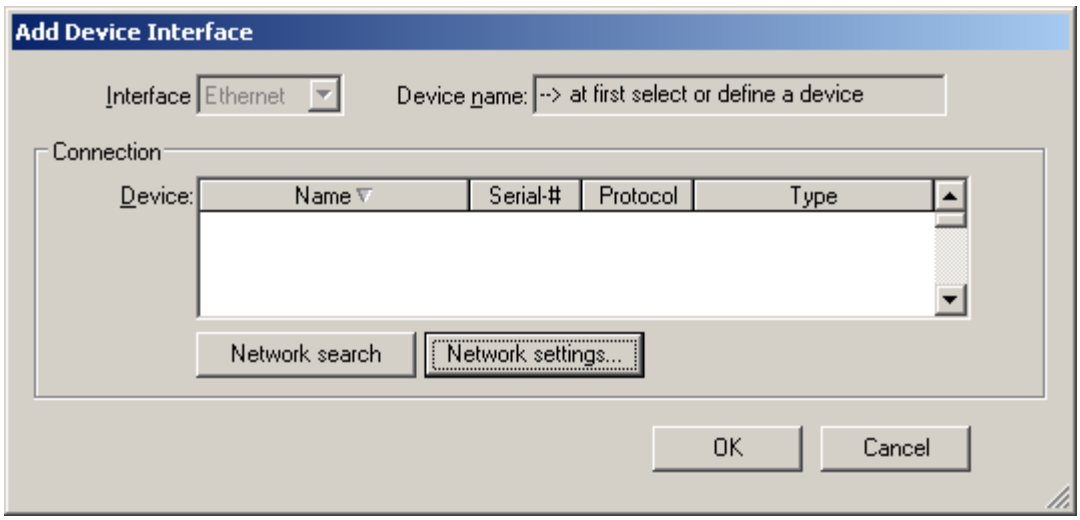
For imc DEVICES up to version 2.7: With the default 64 bit version of the Windows XP x64 explorer, the imc shell extension is not working. Therefore the 32 bit version has to be started: "\windows\syswow64\explorer.exe /separate". A direct access to the internal drive is only possible by FTP 623.

Adding a Device (New...)

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.

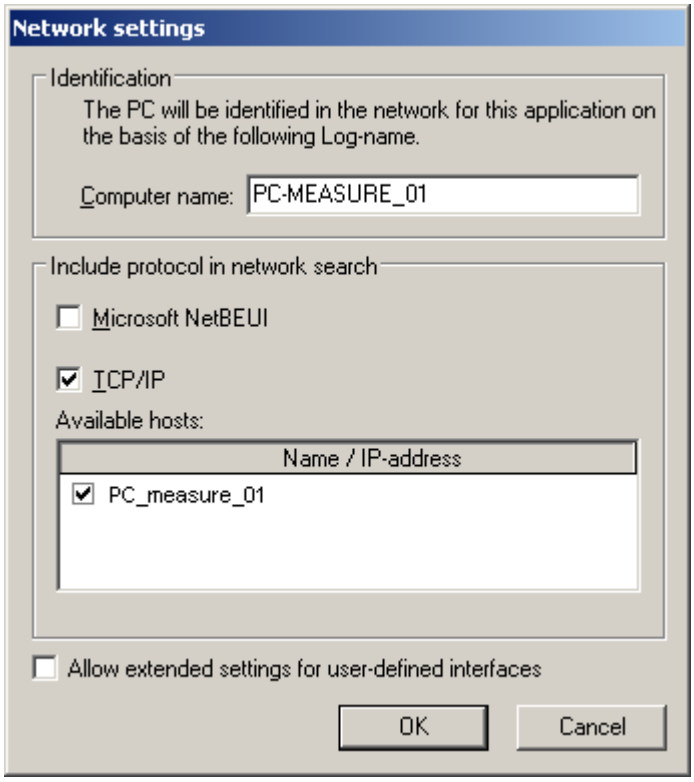
- Click on "*Devices*" under imc Systems to highlight that entry.
- Right-click the mouse nearby and select "*Add*" from the context menu which then appears.

The "*Add Device Interface*" dialog appears.



Add Device interface

If you have not yet added any device on this computer, then performing the network search automatically calls a dialog which can also be opened by the "*Network settings*" button.



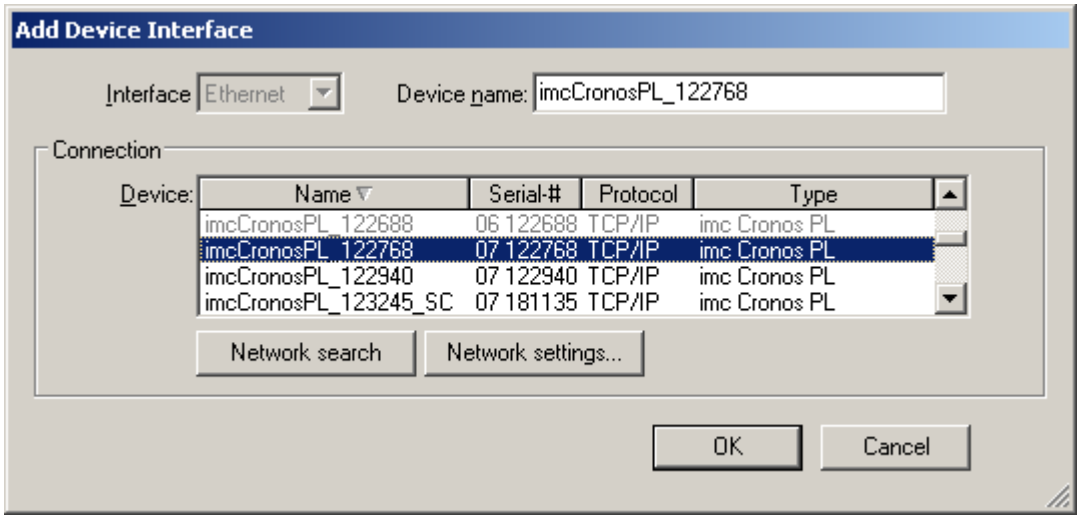
Selection of TCP/IP or NetBEUI devices

Activate the protocol which uses the measurement device and confirm your choice with "*OK*".

Note

imc STUDIO only supports devices over the TCP/IP protocols. Here you will also find devices over the NetBEUI protocol.

In the "*Add Device Interface*" dialog, a search is then performed for all devices in the PC's address space once you push the "*Network search*" button. Subsequently, the measurement devices appear in the list:



Available devices

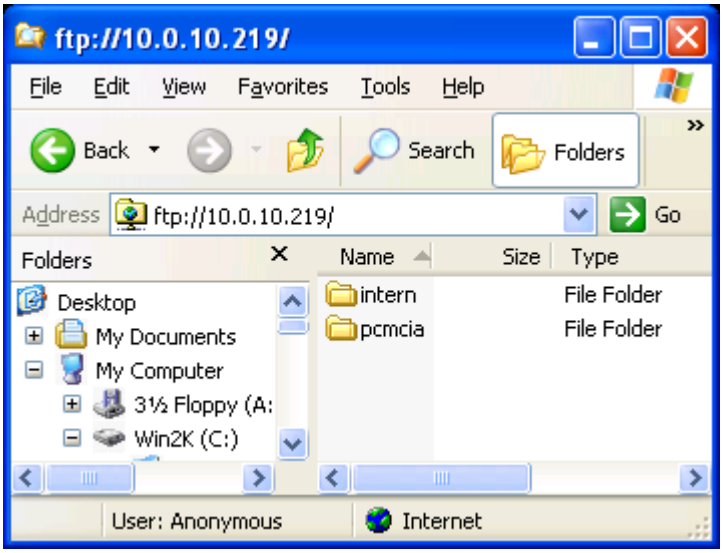
Select your measurement device and confirm your selection with "*OK*". The measurement device is then available.

7.4.2.2 FTP Access

For devices as of group 2 88 and higher:

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 113.

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).

7.4.3 Partition and File System

Fundamentals of FAT16/FAT32

Before using a data storage medium, it should be formatted and maybe also partitioned. The file systems FAT16 (maximum 2 GB) and FAT32 are available for selection.

- Devices belonging to Group 1 (not supported under imc STUDIO) only support storage media with the file system FAT16.
- Devices as of groups 2 and higher support both FAT16 and FAT32.

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

With Group 1 devices (not supported under imc STUDIO):

- If a storage medium is used alternatingly with a device form Group 1 and a different group's device, the medium must be reformatted accordingly (file system FAT16 <-> FAT32).
- If the file system FAT16 is used for all groups, formatting is still necessary. If the medium was used in a Group 1 device, it will subsequently not receive correct support by devices of Group 2 or higher.

Other devices of Groups 2 and higher: 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 626 and on avoiding data overflow 626.
- No limitations regarding the currently available data carrier volumes are known.
- The maximum filesize is 2 GB. In case a signal would exceed that limit, use interval saving 598.

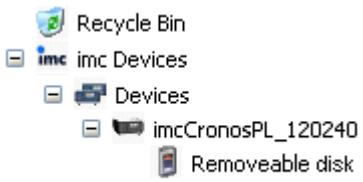
General limitations of the file systems

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.

7.4.3.1 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. imc recommends formatting in the device. In comparison to formatting by Windows, this provides higher data writing rates for high-speed channels

Formatting in the device (Recommended)



To perform formatting in the device, click on the icon of the measurement device of the Microsoft explorer at "imc Systems". Next right-click on the entry for the data carrier and go to the page Extra. Make sure that all data are already saved and click on the button for formatting.

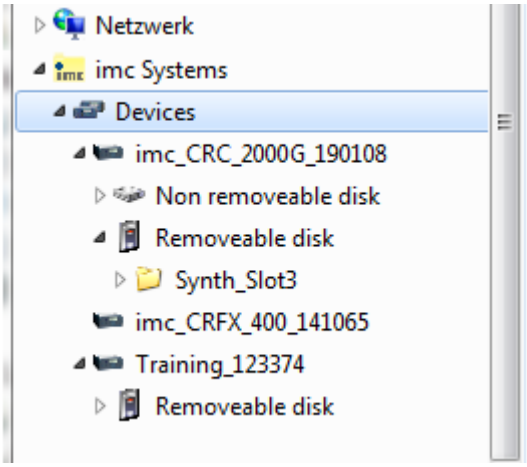
The device performs the formatting according to the following rule:

Drive Size	Cluster Size	File System
<= 128 MB	2 kB	FAT16
<= 256 MB	4 kB	FAT16
<= 4 GB	8 kB	FAT32
> 4 GB	16 kB	FAT32

Note

Formatting the internal data storage is not allowed if an experiment whose data are to be saved internally has just been prepared.

Formatting using the Windows-Explorer



For formatting using the Windows-Explorer's menu item *Format...*, right click the mouse over the data carrier's entry. Under Windows XP/Vista, the control *File system* offers the options "FAT32" or "FAT" (for "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.

7.4.3.2 Possible Problems

Due to the limitations discussed previously, the following problems can occur:

- It is possible that the increasing number of folders 628 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, 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, data overflow can occur if a high aggregate sampling rate is generated by very many low-speed channels.

Other limitations are presented here 628.

7.4.4 Notes on the use of the Storage Media

7.4.4.1 Data Overflow

In case of data overflow on the internal data medium, the position and amount of lost samples is recorded in the SysLog-file 605.

7.4.4.2 Avoidance of Data Overflow

Very many factors can contribute to a data overflow.

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 protocol channels

There is a concrete correlation to the following factors:

1. Data Rate, Channel Count and Formatting 627
2. Occupied Memory, Measurement Duration and Folder Count 628
3. Manufacturer and Age of the Storage Medium 628
4. RAM buffer duration of the channels 592

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 File Manager / the Windows Explorer shell. Otherwise, this additional burden could cause a data overflow.

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 (e.g. 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 400kSample/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 625 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 8kB/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 from 256 MB on are formatted with 8-kByte clusters and from 4 GB on with 16-kByte clusters. Cards of up to 1GB can alternatively be formatted by the PC with FAT16. With cards of 16GB 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 598 and trigger events in separate folders 604, 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.

7.4.4.3 ScanDisk

Check your storage media regularly for errors and correct these, if applicable, using ScanDisk under Windows or with a similar service program. Defective sectors are marked, and such defects reduce the amount of memory available on your storage medium.

7.4.4.4 Known Issues and Limitations

- Even when the above instructions are followed, in certain circumstances and with storage media above 2 GByte can lead to data overflows. This particularly affects devices belonging to groups 2 and 3 88. For this reason, to be safe always use the current software versions. These are available for you in the download area of the website www.imc-berlin.com.
- 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 the imc Hotline 13.

7.5 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.

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.

Reference

Please find differences of the processing speed between measurement amplifiers in the corresponding device manual, chapter: ... > *Synchronicity* > *Delay*.

7.5.1 What's the purpose of Synchronization?

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.

Note

- Synchronization to an external synchronization signal is achieved by setting the clock and subsequent long-term adjustment (closed-loop control) of the device time.
- 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. This also determines whether the measurement time is expressed in standard or daylight saving time.
- If there is no delay for synchronization when starting a measurement (the start time is already determined), the device clock time can no longer be set by a synchronization procedure until a new measurement is prepared! The frequencies of all internal clock rates, however, are synchronized to the external signal if there is one. In this case, the "Synchronization status" (clock status in the user interface software and imc Online FAMOS "`IsSynchronised()`") will still indicate "synchronized"! The frequency is synchronized, but the absolute time is incorrect!
  Upon the next measurement preparation procedure, the system tries again to apply the external clock's absolute time. For this purpose, the synchronization is temporarily interrupted and restarted later. The start of the next measurement should only take place after re-synchronization. Otherwise, this process will be repeated upon every new preparation procedure!
- 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).
- Synchronization signal interruption (only for devices of group 5 and higher 88):
  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.
- 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 applied (particularly if the device is synchronized), or
  - the device's DCF output is active, or
  - "NTP" or "PTP" is selected as the synchronization signal
- If a time zone is set for the device, then the PC time is converted to the device's local time when the clock is set.

7.5.2 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 or 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.

Synchronized measurement start (synchronized, but not simultaneous):

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,
- ...

Synchronous start (Simultaneous start of all devices):

When Synchronization is activated, simultaneous start 157 of all devices is possible. For this purpose, certain conditions must be met. Among others, an appropriate "Start options 154" 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.

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 configuration 114 (Ribbon *Setup-Configuration* > *Device properties*, section *Device time*, *synchronization*): *Time zone* and *Daylight saving time*.

Therms	Description
STD	STandarD time
DST	Daylight Saving Time
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 632	Synchronizable RealTimeClock of imc devices

7.5.3 Clock types

Internal clocks

Clock type	imc measurement device
SyncRTC 632	For all device types except imc µ-MUSYCS

External clocks

Clock type	imc measurement device
DCF77 633	For all device types (for imc µ-MUSYCS, only with additional DCF77 module)
IRIG-B 633	For devices of the group 5 and higher 88
GPS 634	All device types but not group 1 88
NTP 634	For devices of the group 5 and higher 88
PTP 635	For devices of the group 7 and higher 88 with the suffix "-GP"

7.5.3.1 Internal Clocks - Device Clock

Internal device clock: SyncRTC

For all device types except imc µ-MUSYCS.

Synchronizable Real Time Clock.

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. DCF- or GPS-clocks from Meinberg, Hopf, etc.).

Depending on the device type, additional synchronization mechanisms are also available. See: Clock types: External clocks 632.

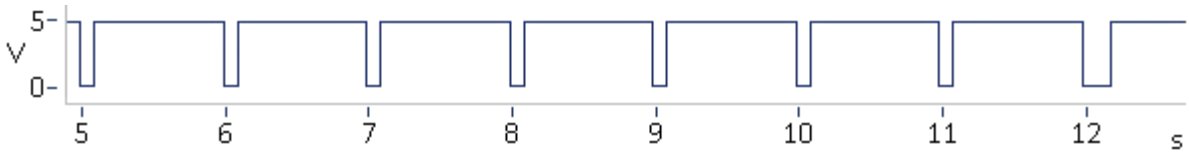
*Note: Certain terminals are not included with some device and housing types!

7.5.3.2 External Clocks: DCF77, IRIG-B, GPS, NTP

DCF77

For all device types.

The DCF77 signal is applied via the SYNC socket. The measurement device processes a LOW[1] 635 active 5V TTL signal (1Hz pulse). As well, inverted signals are recognized and evaluated by the hardware.



DCF77 TTL signal with inverted logic (Low active)

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!

IRIG-B

For devices of the group 5 and higher 88.

The IRIG-B signal is applied via the SYNC socket. The measurement device processes a LOW[1] 635 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 IRIG-B000, -B001, -B002 and -B003 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! This is important if a time zone was set for the device!
- During signal output only the BCD information will be applied, this means no CF and no SBS information will be applied.

GPS

All device types but not group 1 88.

The GPS (Global Positioning System) receiver is connected at the 9-pin DSUB terminal "GPS". 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.

NTP (Network Time Protocol)

For devices of the group 5 and higher 88.

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 113).

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. For setting NTP, the following parameters are needed:

- selection of the *time zone*
- observance of the *daylight saving's time* transitions
- IP of the *NTP server (1),* IP of the *NTP server (2)*

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 643.
- 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).

1: Due to the greater steepness of edges, LOW active signals achieve better precision than HIGH active signals.

7.5.3.3 External Clocks: PTP

For devices of the group 7 and higher with the suffix "-GP".

The Precision Time Protocol (PTP) serves synchronize clocks within distributed computer systems. Compared with NTP, its achievable precision is significantly higher.

Prerequisites and limitations: 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.

Duration of synchronization process: With the settings specified, the synchronization takes about 20-30s after initialization of the measurement. But depending on the wide variety of settings parameters, synchronization can also last significantly longer.

The procedure also takes longer if the measurement already before the device could adopt the time server's clock time. The time difference will then be compensated by modifying the device's internal clock (SyncRTC) adjusting over a long period of time.

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).

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.

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.

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

Experiment options (for all devices)				
Parameter	Min	Max	IEEE 1588 2008 (Default)	fos4X
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
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". 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.

Experiment options (for all devices)	
Parameter	Description
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.

7.5.4 Phase offset correction

For devices of the groups 1-4 88:

With devices of groups 1-4, 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 5 and higher 88:

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. .

7.5.5 Master/Slave setup

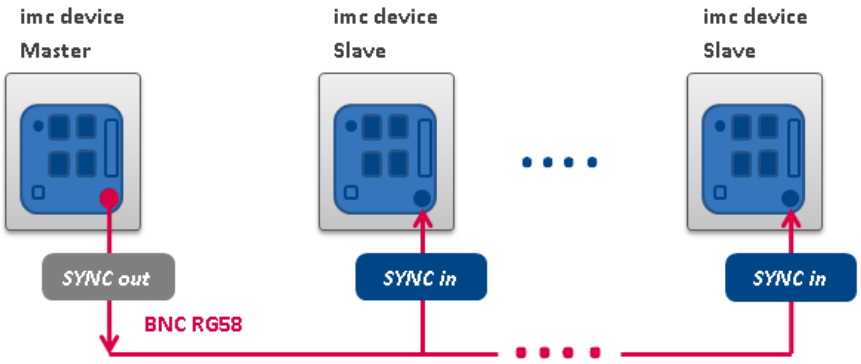
In the setups described here, all devices are synchronized to each other via SYNC. The clock rate is provided by a master device synchronized to an external clock. All devices are connected via the SYNC jack. No terminal resistor is needed.

Note

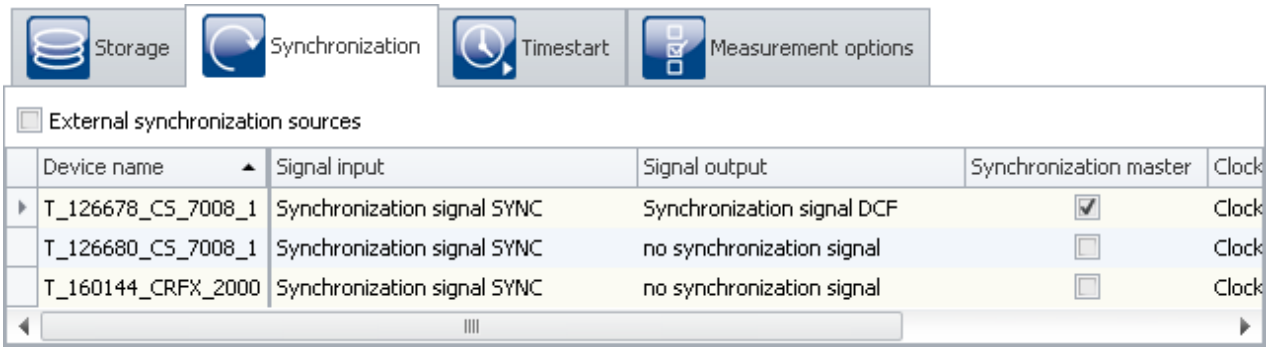
- If the synchronized devices are at different voltage levels, they should be compensated by means of a lead having the appropriate cross-section. If the SYNC terminal appears with yellow ring below it, it means the terminal is already isolated and shielded from voltage differentials (applicable to devices delivered as of Summer, 2012). Alternatively, it is possible to isolate the different voltage levels from each other using a adapter ISOSYNC 662.
  - See also technical specs in the corresponding device manual, chapter "Synchronization and time base"
- 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), the Master continues according to its internal clock. However, the devices remain synchronized to each other.

7.5.5.1 No external timer

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

- 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, and then only if the clock's mode was switched from slave to master.

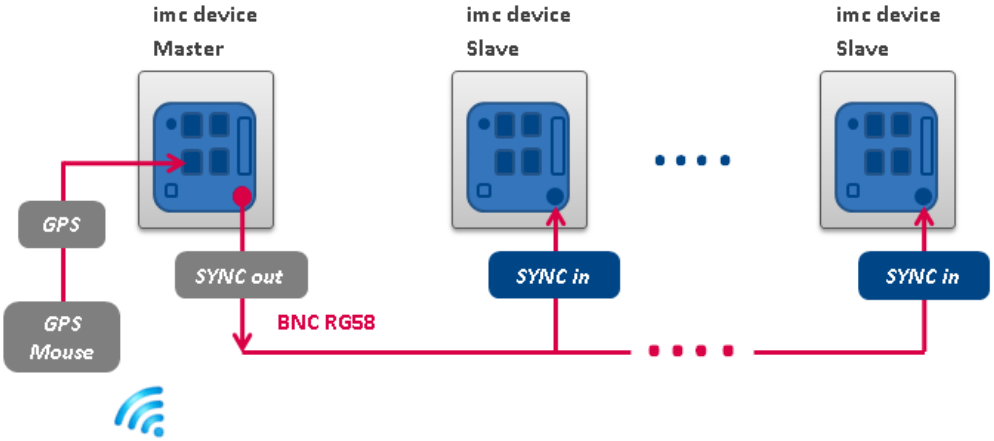
  This also happens, for instance, after activating the devices [1].

  The positive result of this is that when working with the devices, it is necessary to wait for the synchronization only once. The disadvantage is that the PC clock and the device clock drift apart during long-term measurements, but the devices remain synchronized to each other.
- 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.

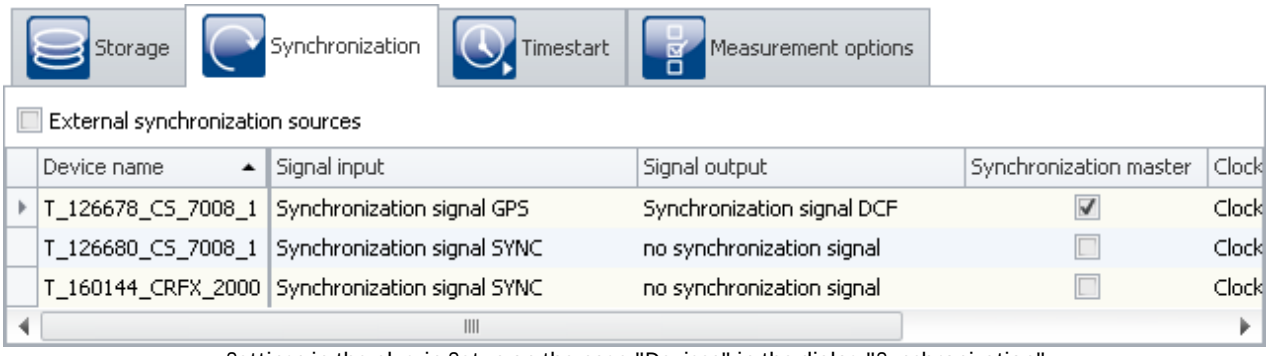
1: This doesn't apply if the the default synchronization setting is defined in the device properties. In that case, the time is only adjusted if the signal output is subsequently set to "No synchronization signal" on the Synchronizations-page. After the adjustment, run the initialization once and then set the desired synchronization.

7.5.5.2 GPS receiver as timer

The devices are synchronized with each other and synchronized to absolute GPS time.



Master device synchronized by GPS and slave devices



Settings in the plug-in Setup on the page "Devices" in the dialog "Synchronization"

Different response for GPS-synchronization + DCF/IRIG-B-output

- Behavior of group 2 to 4 88 devices: 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!
- Behavior of group 5 and higher 88 devices: DCF/IRIG-B-output begins upon the first synchronization with GPS. If there is an outage of the GPS-signal, the DCF/IRIG-B-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 640 . In the process, all connected DCF/IRIG-B Slaves are automatically adjusted to have the same phase.

7.5.5.3 GPS- and DCF/IRIG-B synchronization as time

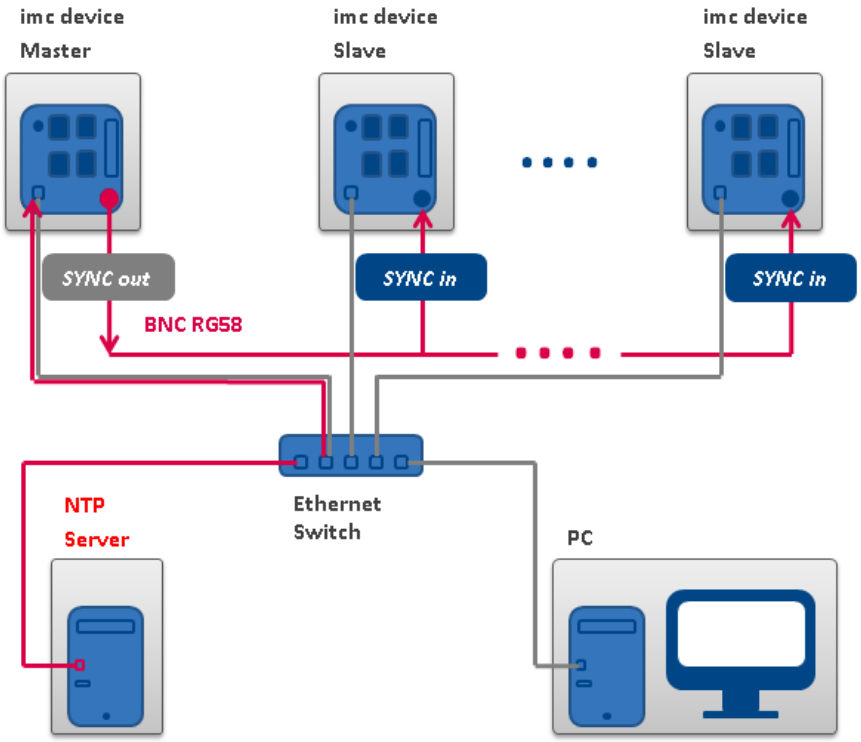
Different response for GPS- and DCF/IRIG-B synchronization + DCF/IRIG-B-output

- Behavior of group 2 to 4 88 devices: 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!

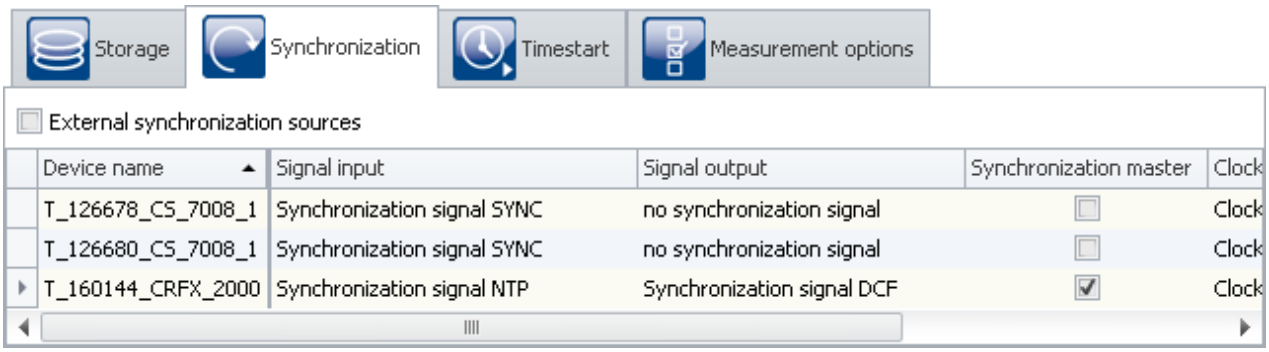
- Behavior of group 5 and higher 88 devices: DCF/IRIG-B-output begins immediately. Any GPS-signal applied always supersedes the device's own DCF/IRIG-B-signal as the synchronization source! If no GPS-signal is applied, the device synchronizes to its own DCF/IRIG-B-output.
  As soon as a GPS-signal is available, the device synchronizes to the GPS. The DCF/IRIG-B-output is briefly interrupted, so that all DCF/IRIG-B-Slaves adopt the GPS time. Following the first synchronization with GPS, the DCF/IRIG-B-output remains intact even if the GPS-signal fails!
  If the GPS-signal fails again, the synchronizes to its own DCF/IRIG-B-signal. If the external GPS-signal returns, the device synchronizes to the device by correcting the resulting phase offset 640. This automatically also adjusts all connected DCF/IRIG-B-Slaves to the phase.

7.5.5.4 NTP-time via LAN

The devices are synchronized with each other and synchronized to absolute NTP time.



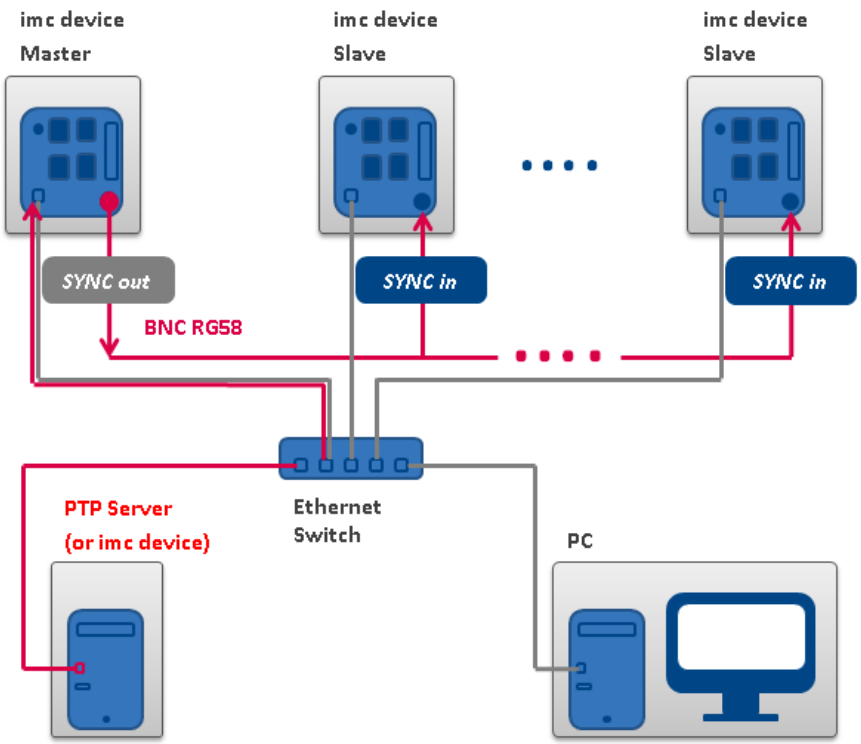
Master device synchronized by NTP and slave devices



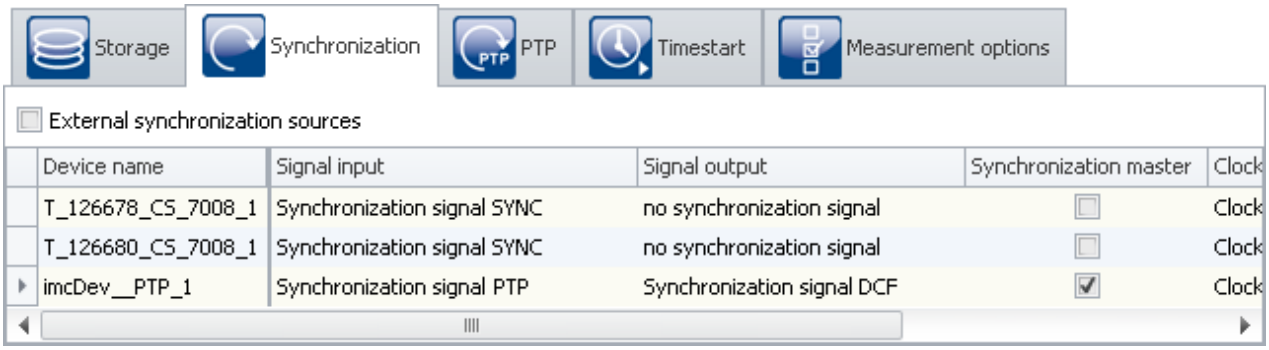
Settings in the plug-in Setup on the page "Devices" in the dialog "Synchronization"

7.5.5.5 PTP-time via LAN

The devices are synchronized with each other and synchronized to absolute PTP time.



Master device synchronized by PTP and slave devices



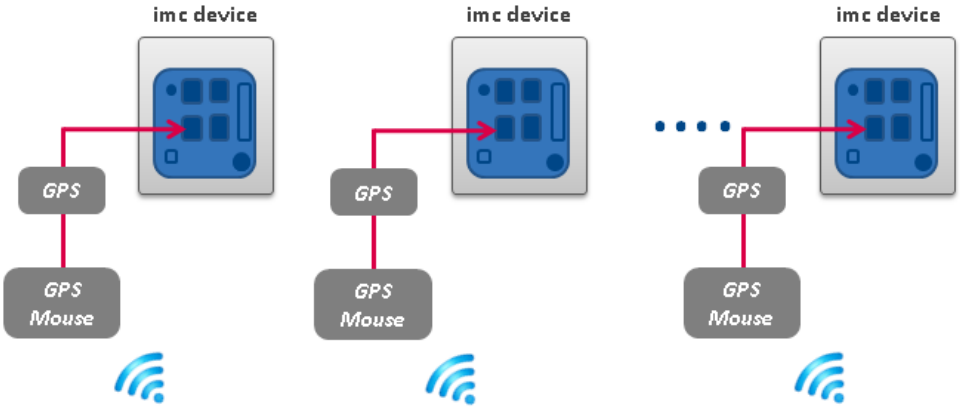
Settings in the plug-in Setup on the page "Devices" in the dialog "Synchronization"

7.5.6 Separate synchronization of all devices

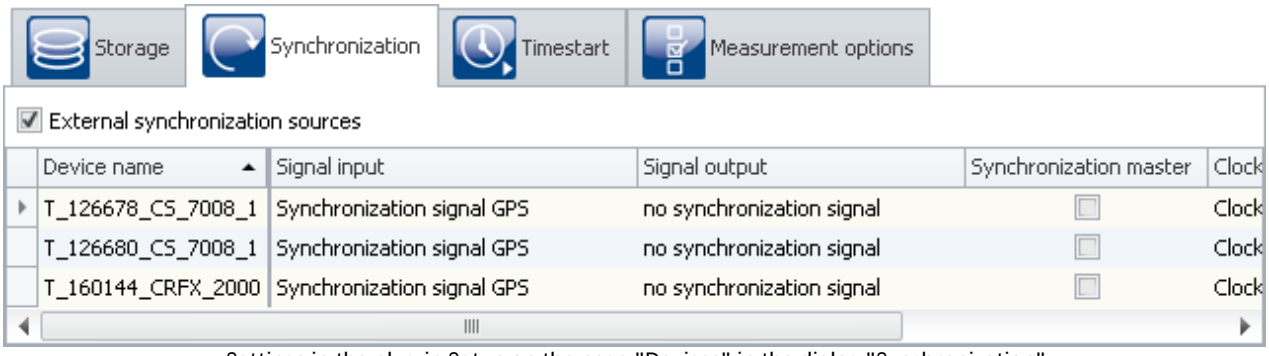
If connection between devices via a Sync line is not possible, then each device can be synchronized separately to an external clock.

7.5.6.1 GPS-time

Each device is synchronized to absolute GPS time and thus all devices are mutually synchronized.



Example of individual synchronization with GPS

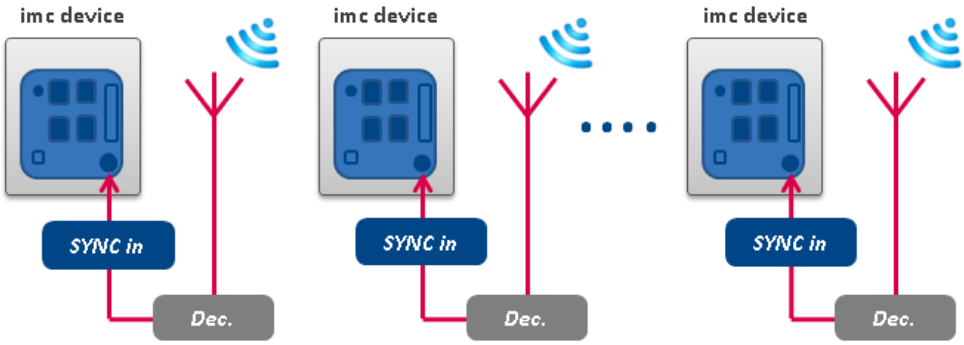


Settings in the plug-in Setup on the page "Devices" in the dialog "Synchronization"

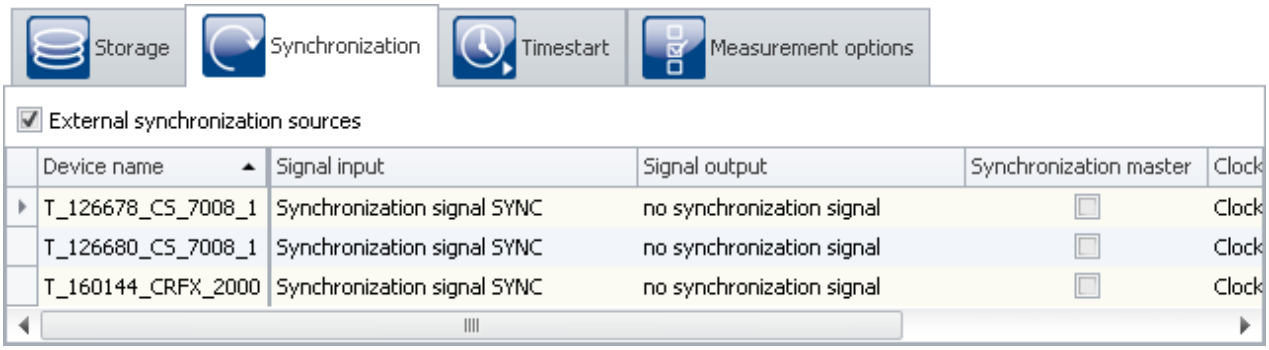
7.5.6.2 DCF77 time

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.



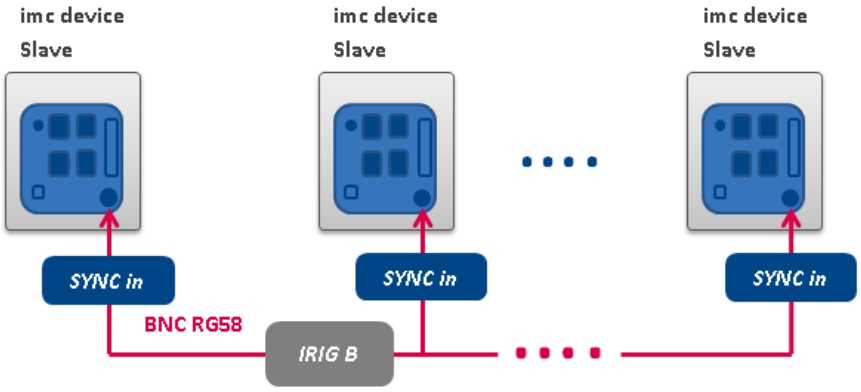
Example of individual synchronization with DCF77



Settings in the plug-in Setup on the page "Devices" in the dialog "Synchronization"

7.5.6.3 IRIG-B time

Each device is synchronized to absolute IRIG-B time and thus all devices are mutually synchronized.



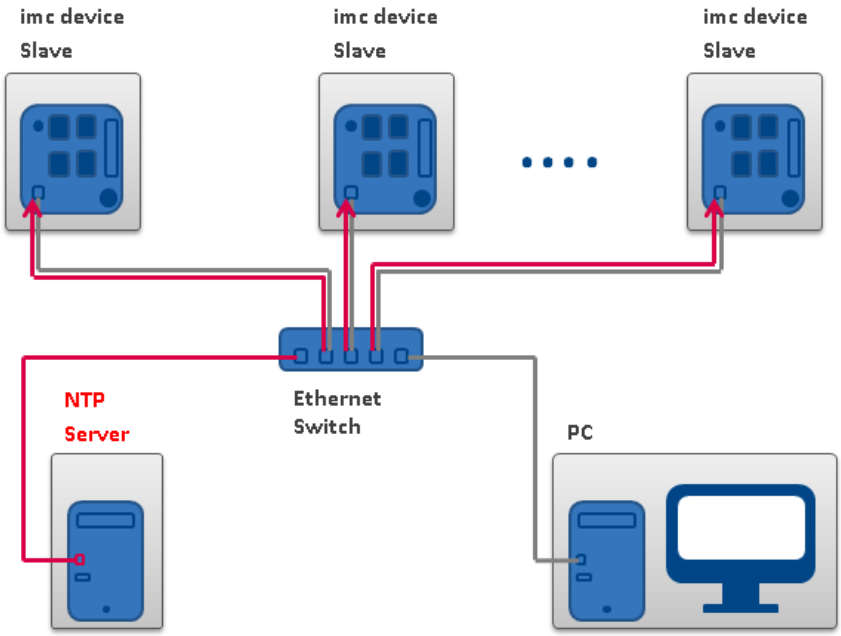
Example of individual synchronization with IRIG-B

Note

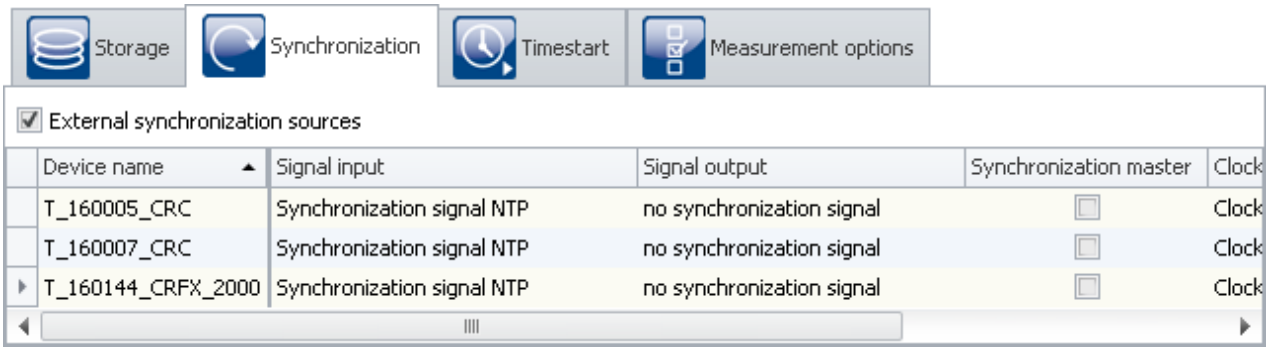
With IRIG-B, too, SYNC is selected as the signal input. The hardware recognizes independently that the signal is IRIG-B.

7.5.6.4 NTP-time via LAN

Each device is synchronized to absolute NTP time. All devices are thus mutually synchronized, although to a limited extent (see External clock: NTP 634).



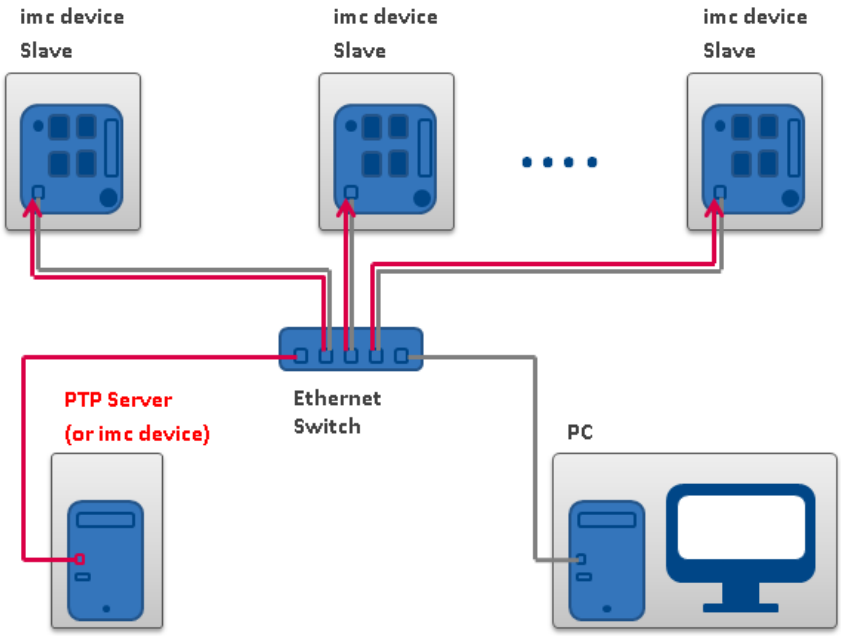
Example of individual synchronization with NTP



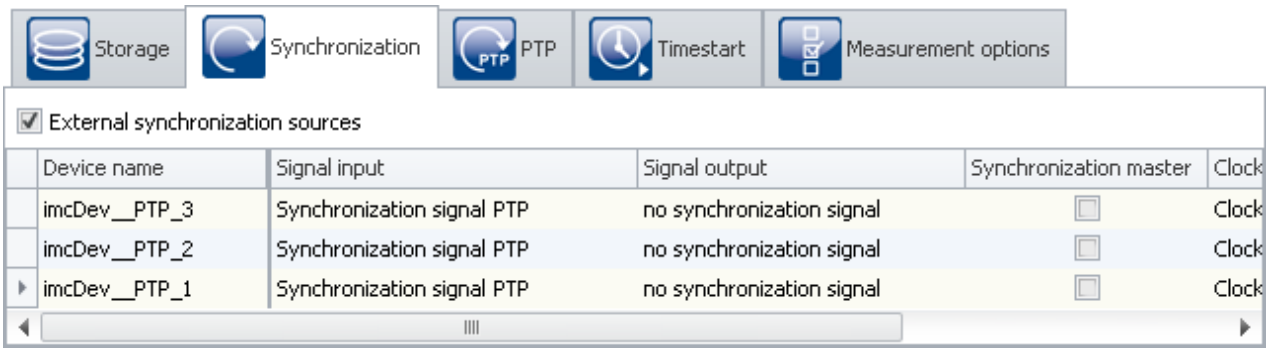
Settings in the plug-in Setup on the page "Devices" in the dialog "Synchronization"

7.5.6.5 PTP-time via LAN

Each device is synchronized to absolute PTP time. All devices are thus mutually synchronized, although to a limited extent (see External clock: PTP 635).



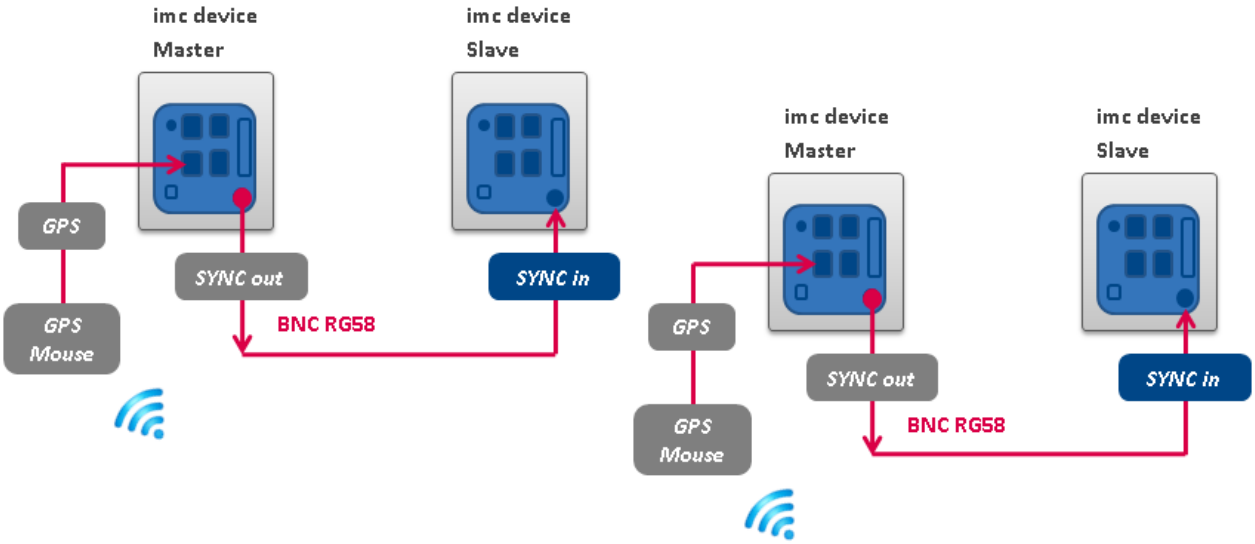
Example of individual synchronization with PTP



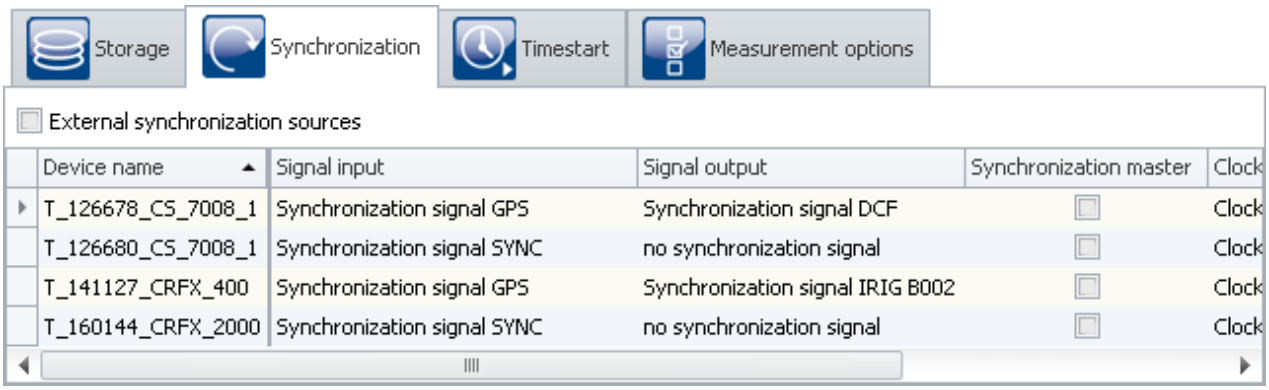
Settings in the plug-in Setup on the page "Devices" in the dialog "Synchronization"

7.5.7 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



Device name	Signal input	Signal output	Synchronization master	Clock
T_126678_CS_7008_1	Synchronization signal GPS	Synchronization signal DCF		Clock
T_126680_CS_7008_1	Synchronization signal SYNC	no synchronization signal		Clock
T_141127_CRFX_400	Synchronization signal GPS	Synchronization signal IRIG B002		Clock
T_160144_CRFX_2000	Synchronization signal SYNC	no synchronization signal		Clock

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.

7.5.8 Working with time zones

The abbreviations pertinent to the topic of time zones are presented in the section Definitions 631.

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?

A device assumes the UTC time if it is synchronized with a GPS-receiver. In this case, stating a time zone can provide the remedy. The device then uses the time zone set, irrespective of whether the clock was set by GPS or by a PC.

Users of NTP 634/PTP 635 must also specify a time zone because the device would otherwise assume the UTC time.

Note

- If no time zone is used, the devices apply the respective clock's time zone.
  - Use of GPS: device time = UTC
  - PC sets the device time: device time is local PC time.
  - No DST-switching
- This setting is compatible with devices of group 1 88 and can be used for mixed operation.
- All statements regarding the topic of time zones are only effective if a time zone has been set for the device.

The DST switch is activated in the Device properties 114 (Ribbon *Setup-Configuration* > *Device properties*, section *Device time, Synchronization*): *Time zone* and *Switching from/to daylight saving time*.

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 5 and higher 88 maintains its DST-state, but correctly adopts the source's time.

A device of groups 2 to 4 88 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 3 and 4 88, 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.

7.5.8.1 Possible scenarios

Example 1

- A device of groups 2 to 4 88 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 groups 2 to 4 88 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 groups 2 to 4 88 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 5 and higher 88 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 5 and higher 88 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.

7.5.8.2 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 groups 2 to 4 88:

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 groups 5 and higher 88:

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.

7.5.9 Application notes

7.5.9.1 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.

7.5.9.2 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 88:
  - With devices belonging to Groups 1-4, only the frequency error 640 is corrected.
  - Devices of groups 5 and higher attempt to compensate for the phase error 640.
- Deviation directly after loss of synchronization (1 ppm):

  $t_{delta} = t_{unsync} * 1 * 10^{-6}$; with t_{delta} = deviation in seconds

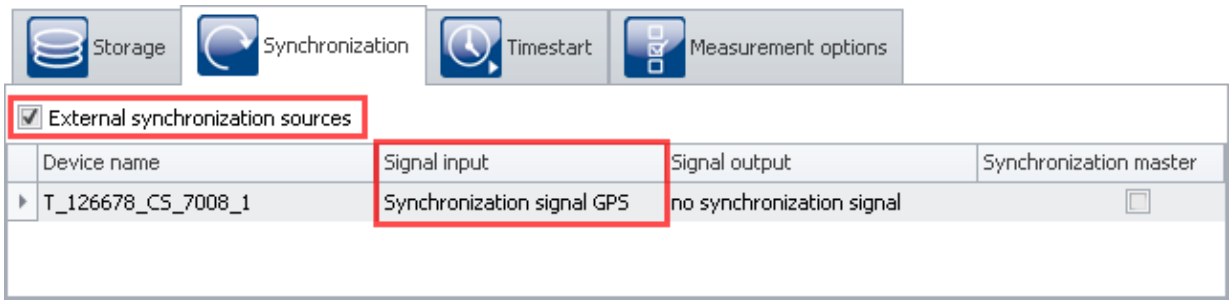
  t_{unsync} = Time without synchronization in seconds

  $1 * 10^{-6}$ = precision of the device time
- Deviation following long-term loss of synchronization (max. 50 ppm):

  $t_{delta} = t_{unsync} * 50 * 10^{-6}$

7.5.9.2.1 Measurement with one device

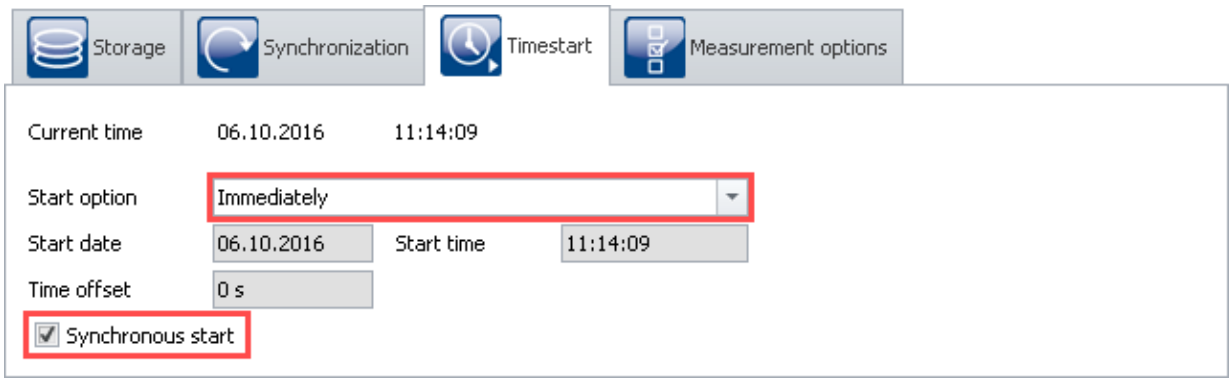
External synchronization via DCF77, IRIG-B, GPS receiver, NTP or PTP.



Settings in the Setup on the page "Devices" in the dialog "Synchronization"

Timestart option: "Immediately"

Synchronous start: activated



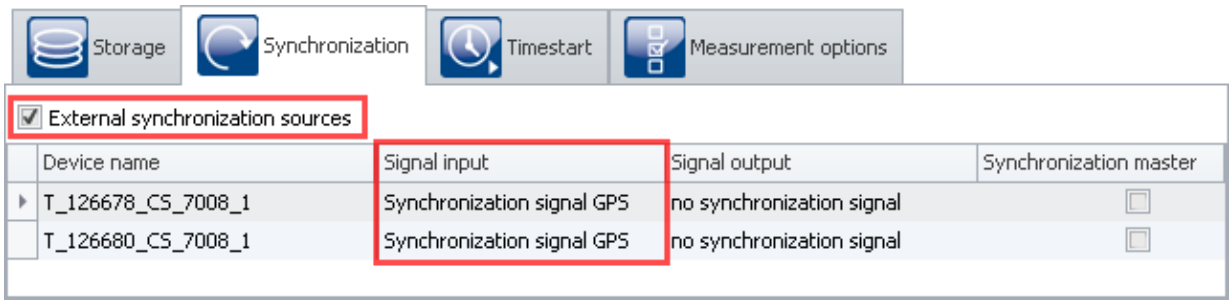
Settings in the Setup on the page "Devices" in the dialog "Timestart"

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) 632	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 99.	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 88: With devices belonging to Groups 1-4, only the frequency error 640 is corrected. Devices of groups 5 and higher attempt to compensate for the phase error 640.	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.

7.5.9.2.2 Measurement with multiple devices

Devices are externally synchronized separately. There is no connection via the SYNC terminals.

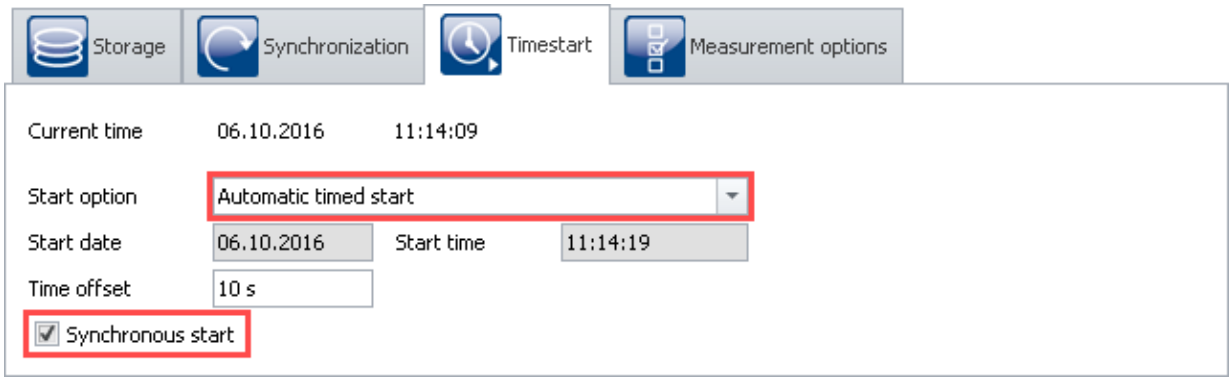


Settings in the Setup on the page "Devices" in the dialog "Timestart"

Synchronous start: activated

1) Timestart option: "Immediately": each unit behaves the same as for "Measurement with one device 656" mutually independently.

2) Timestart option: "Immediately" is not selected



Settings in the Setup on the page "Devices" in the dialog "Timestart"

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) 632	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 Timestart 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 88: With devices belonging to Groups 1-4, only the frequency error 640 is corrected. Devices of groups 5 and higher attempt to compensate for the phase error 640.	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.

7.5.9.3 Synchronization control

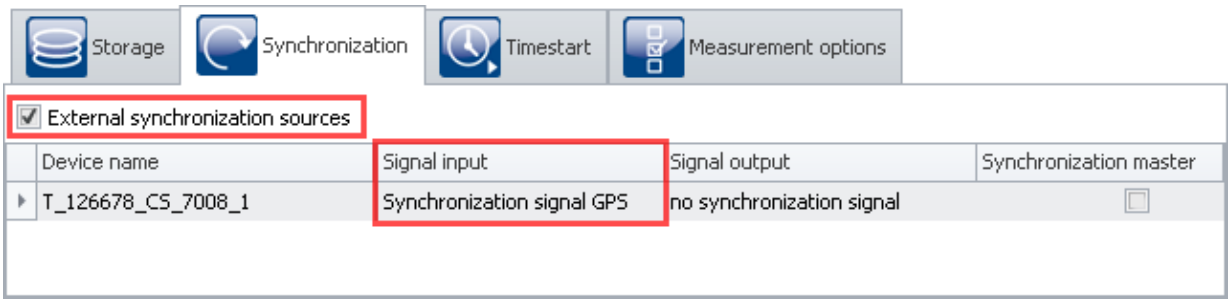
Normally, it's not necessary to stop recording of short measurements if the synchronization is lost. Use the formula presented in the section "What happens when?" 655 to calculate whether the resulting deviation is acceptable for your measurement.

The following functions and constraints must be taken into account for automated synchronization control:

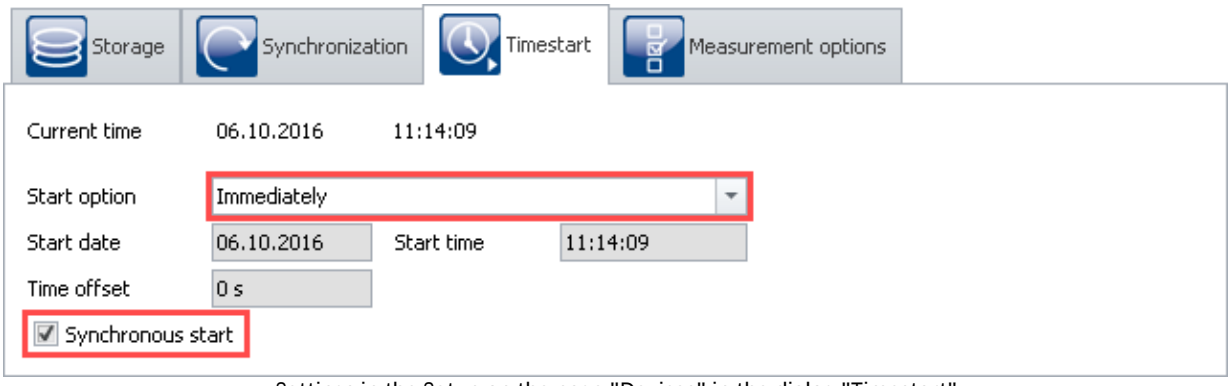
- If a device starts with a synchronized self-activated experiment, the measurement only starts once the synchronization has been established.
- For devices of the groups 1-4 88: After synchronization has been temporarily lost, the measurement must be re-prepared so that the external pulse clock can be received again.
- The synchronization status is requested using the imc Online FAMOS function `isSynchronized`.

7.5.9.3.1 Measurement controlled by PC

1. Synchronization settings



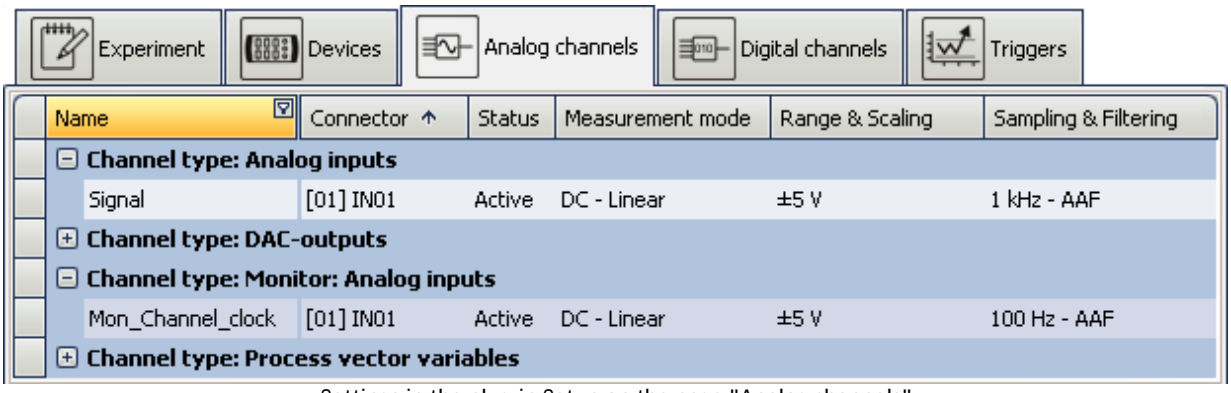
Settings in the Setup on the page "Devices" in the dialog "Synchronization"



Settings in the Setup on the page "Devices" in the dialog "Timestart"

2. Configuration

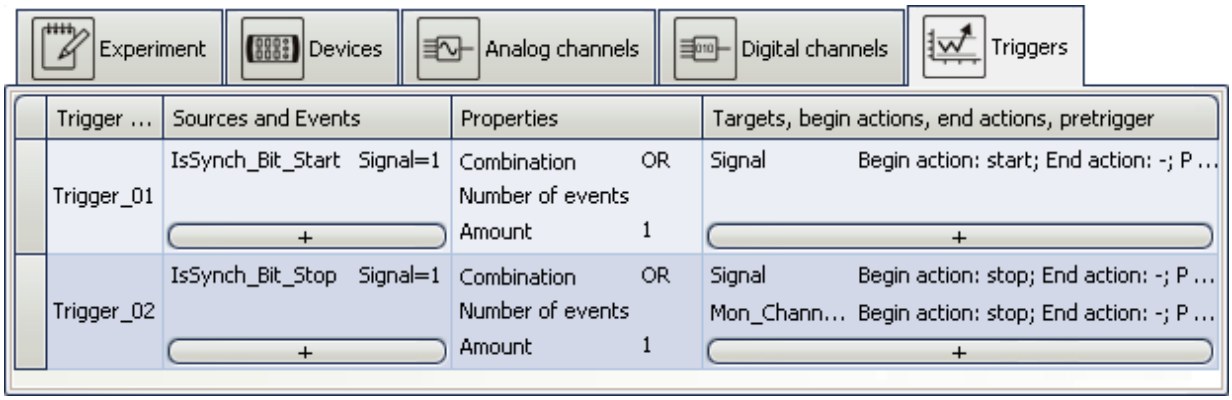
Activate along with the measurement signal to be captured a monitor channel with a sampling interval of *10ms* as the clock rate. Rename two virtual bits: *IsSync_Bit_Start* and *IsSync_Bit_Stop*.



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
```

7.5.9.3.2 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 660.
   - 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. This is the case with all imc CRONOS-PL/SL, imc BUSDAQ, and imc SPARTAN devices and some imc C-SERIES devices. 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 113. 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
```

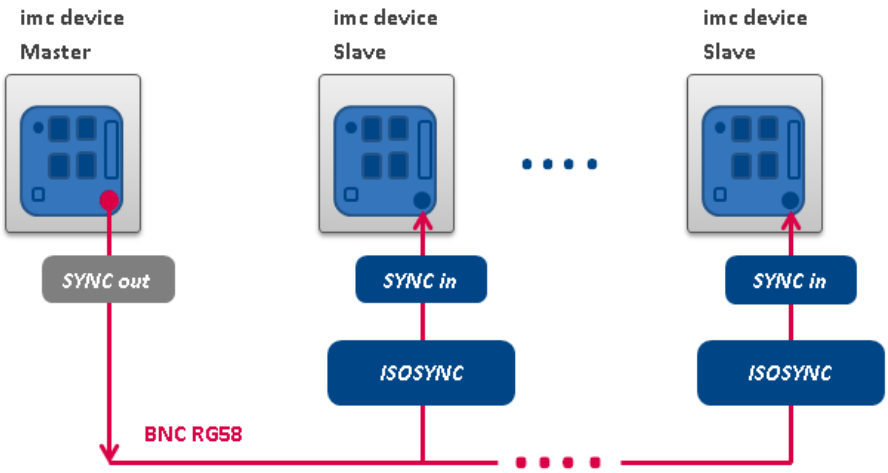
   This output must be used by an external control mechanism in order to connect to the Remote terminal or to temporarily switch off the supply voltage.

7.5.9.4 ISOSYNC isolated Synchronization

When using multiple devices connected via the Sync terminal for synchronization purposes, ensure that all devices are the same voltage level. Any potential differences among devices may have to be evened out using an additional line having adequate cross section. If the SYNC terminal appears with yellow ring below it, it means the terminal is already isolated and shielded from voltage differentials (applicable to devices delivered as of Summer, 2012).

Alternatively it is possible to isolate the devices by using the module *ISOSYNC:*

- For the purpose of isolating the synchronization signal, an ISOSYNC module is connected to the pertaining imc devices-SYNC-slaves. The ISOSYNC module has to be connected with SYNC terminal directly (a cable extension is not allowed)!
- The master device has no ISOSYNC module. That way it is determined what device is the master. The software must be configured according the hardware set-up. For a master device with internal isolation (yellow ring), the isolation is switched off automatically.



7.5.9.5 imc CRONOS (PC) with other devices

Synchronizing a stand-alone imc CRONOS unit (with built-in PC) with other imc devices is not possible!

7.5.9.6 Precision

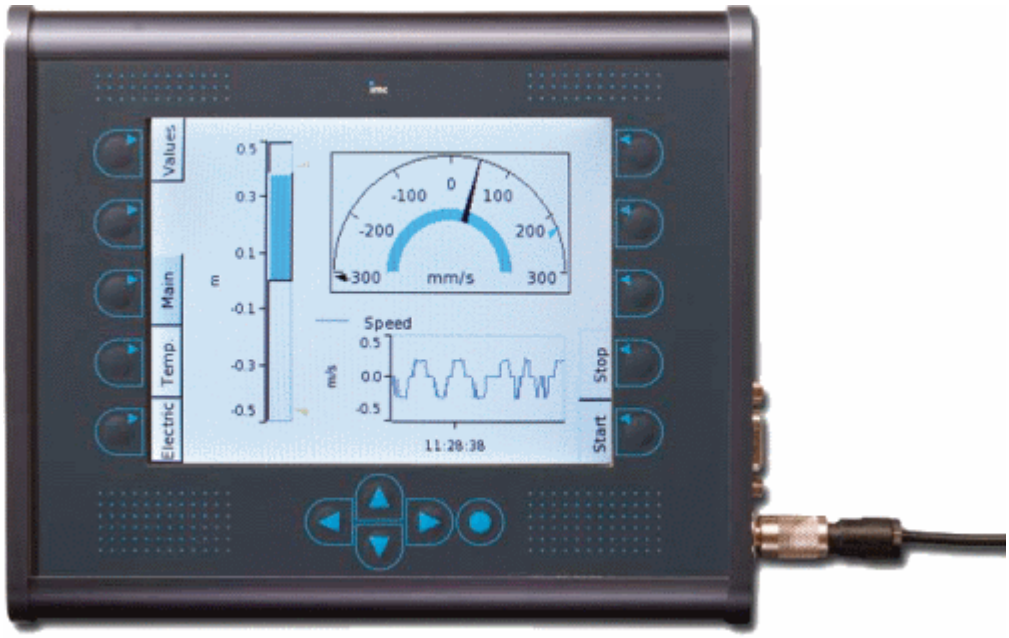
- The maximum time difference between data from different devices is 1 sampling interval, max 3ms (as long as nothing else is set, such as for instance with NTP).
- In a synchronized start, the devices are started simultaneously. However, the moment when the first sample value arrives can differ. This 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.

7.5.9.7 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. Voltage level differences: Ensure that all devices are at the same voltage level. Any potential differences among devices may have to be evened out using an additional line having adequate cross section. Alternatively, the devices can be synchronized with the module *ISOSYNC* while remaining electrically isolated. If the SYNC terminal appears with yellow ring below it, it means the terminal is already isolated and shielded from voltage differentials (applicable to devices delivered as of Summer, 2012).
4. Start delay: The time needed to arm the system is a multiple of the slowest sampling rate in the system. The minimum delay is approx. 3s. 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.
5. 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 593.

7.6 imc Display Editor

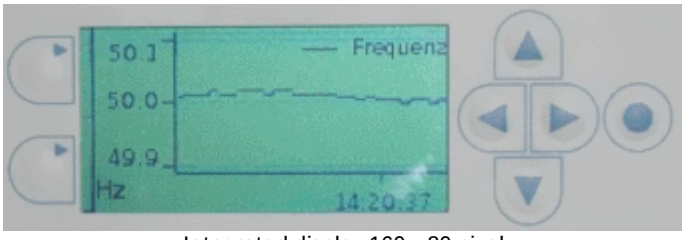
This chapter describes the operation of imc displays (external and integrated graphical display).



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.

Graphic Displays built into the device are monochrome with 160 x 80 pixels:



Integrated display 160 x 80 pixel

7.6.1 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.

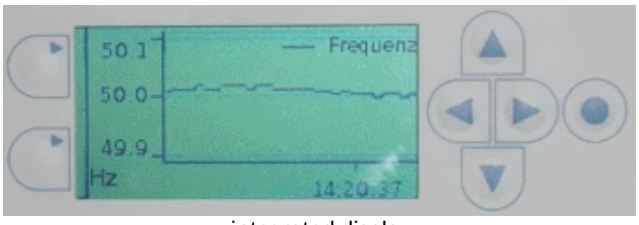
If the measurement device is prepared in such a way that it loads a certain configuration upon being activated, it's possible to run a measurement without any PC. The Display serves as a comfortable status indicator device and can replace or complement imc STUDIO when it comes to controlling the measurement. 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.

Graphic Displays - Hand-held terminals (external display) and built-in displays (integrated display)

external display



integrated display

Note

- Prerequisites:
  - Devices as of Group 2 88
- Devices which only support the alphanumeric Display can use the graphical Display in the compatibility mode 665

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: Ribbon *Devices-Configuration* > *Device Properties* 113)

7.6.2 Compatibility Mode

Devices which do not support the graphical Display can use the Display as an alphanumeric Display.

In the Display's System menu 667, select *Text mode* under *Display Mode*. Then the graphical Display can be configured and used as an alphanumeric Display.

When the Display is run in this mode, it is in the so-called compatibility mode with the alphanumeric Display. In that case, the Display can be run with every measurement device for which the Display Type 1 (40 characters x 16 rows) is entered in the device configuration.

For details, see System Menu 667.

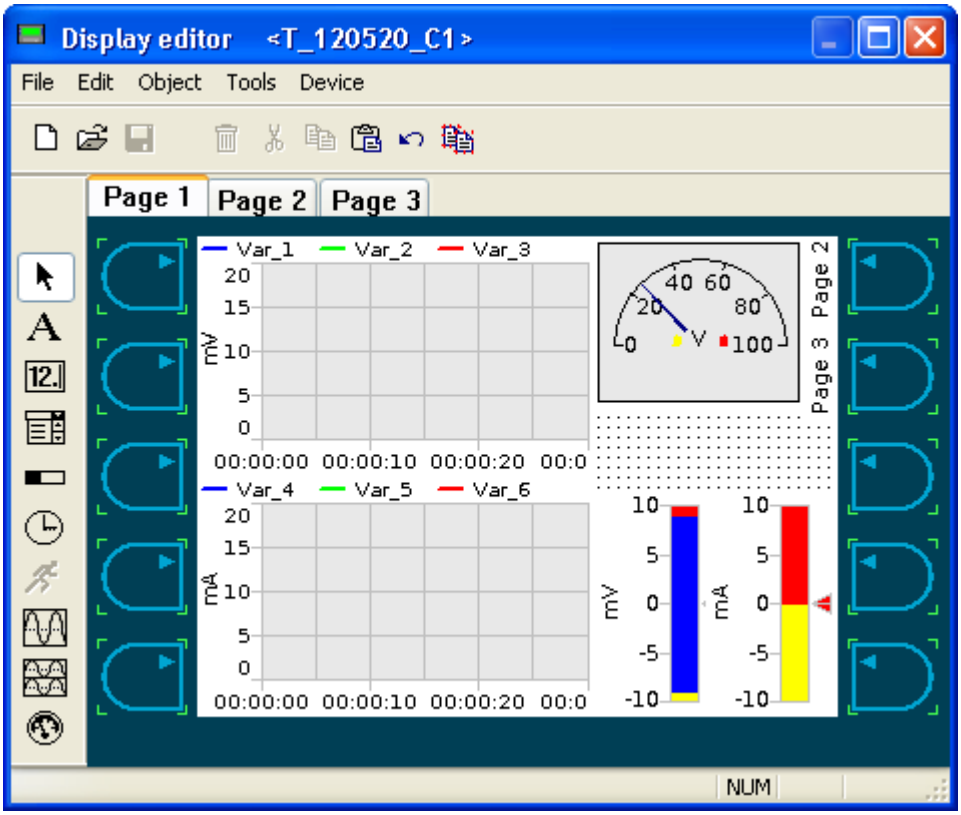
Note

When running the graphical display in text mode, connect the display after booting the measurement device. Otherwise the display would not work.

7.6.3 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 ether 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 676 or by remote control by means of the imc Online FAMOS function `DisplaySetPage` 1472.

7.6.3.1 Menu

File - Menu

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 - Menu

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.

Note

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 - Menu

The object menu contains the same items which are offered in the toolbar.

Extras - Menu

The item Firmware update 682 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 - Menu

Selection of the device whose Display is to be configured.

7.6.4 System Menu

If the Enter button is held down for about 3 seconds, the System menu appears. It contains the following items:

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

7.6.5 Keyboard

External (Handheld) display

Enter key (solid circle):

- Starts Input mode (sets the focus to an input element)
- Ends Input mode (Enter)

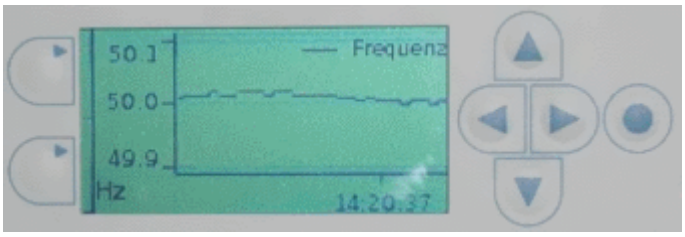
Arrow keys:

- 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

Integrated display



Using the vertical arrow buttons on the right side, it's possible to scroll through the various pages of the display. All pages can be reached in this way.

For quicker navigation, the upper of the buttons just left of the display screen can be used to avoid scrolling through every single page; instead, display skips to the overview windows

The horizontal arrow buttons on the right side serve as cursors and for scrolling through a list box if a line in the box is selected/highlighted.

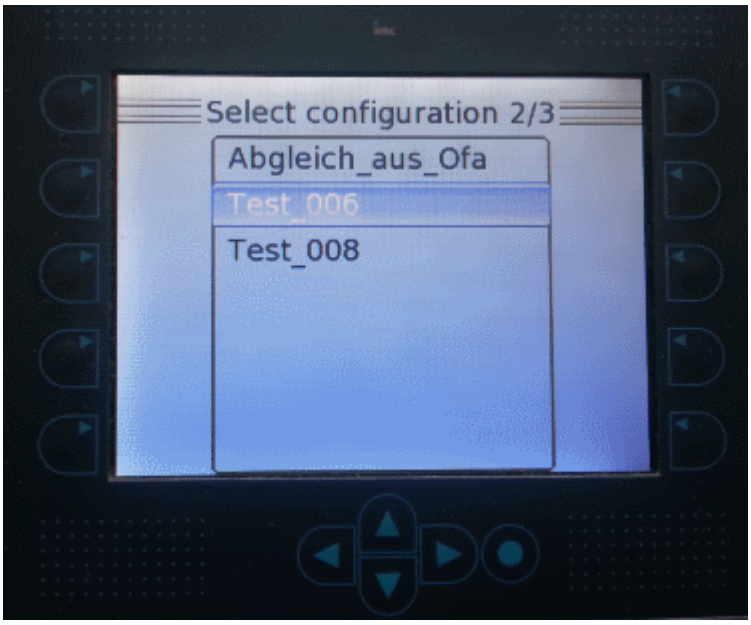
The round button is equivalent an ENTER key. This allows you to make the selection. A prolonged push of this button during measurement opens the selection menu, which also offers the choice of stopping measurement. A picture is provided with the description of the Diskstarts 669.

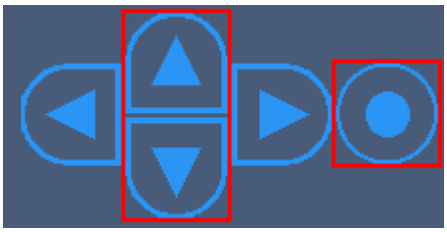
Using the lower of the buttons at the left of the display screen, it's possible to jump directly back to page 1. If page 1 is already currently selected, this button can be used to stop the running measurement.

7.6.6 Diskstart

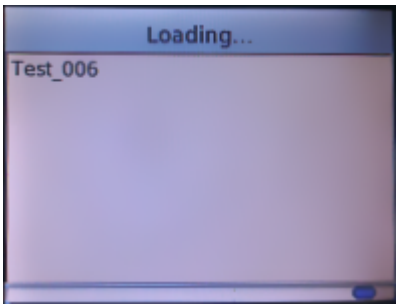
If the device memory contains *Diskstart configurations*, they can be selected in the display. If a Diskstart configuration was saved as an *Autostart* configuration, the selection procedure is skipped and the experiment starts immediately.

Once the device is activated, the experiment appear in a list.





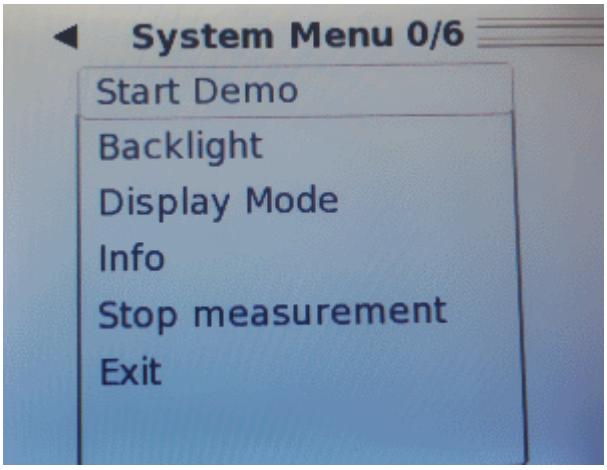
Using the UP/Down arrow keys, you can select the desired entry. Open the experiment using the round button.



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.



7.6.7 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.

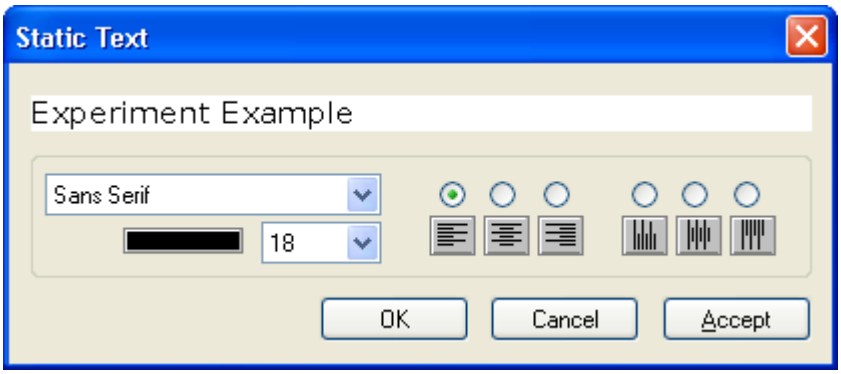
The types presented below are common to all dialogs.

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

7.6.7.1 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.



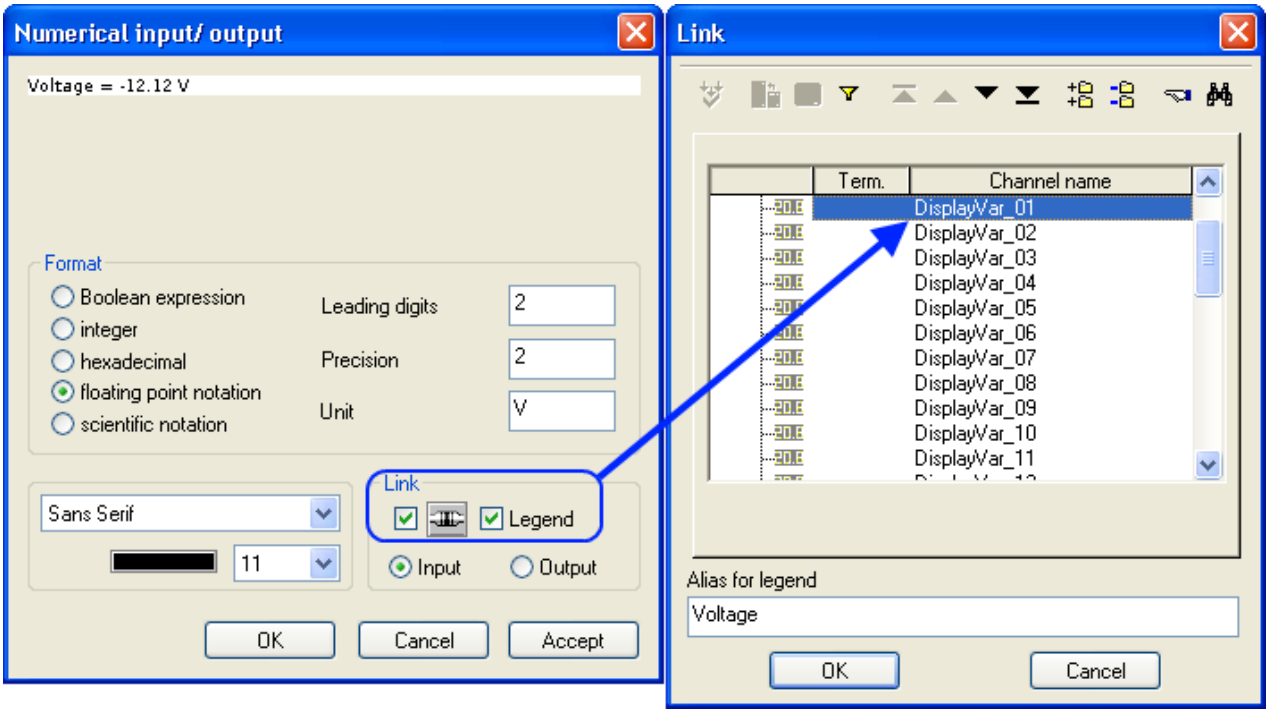
- horizontal text
- 90°-rotated text
- 3 different fonts (Serif, Sans-Serif, Sans-Serif-Mono)
- normal, cursive, bold, bold-cursive

7.6.7.2 Time

Display of the time with the selected font.

7.6.7.3 In-/Output (Numerical)

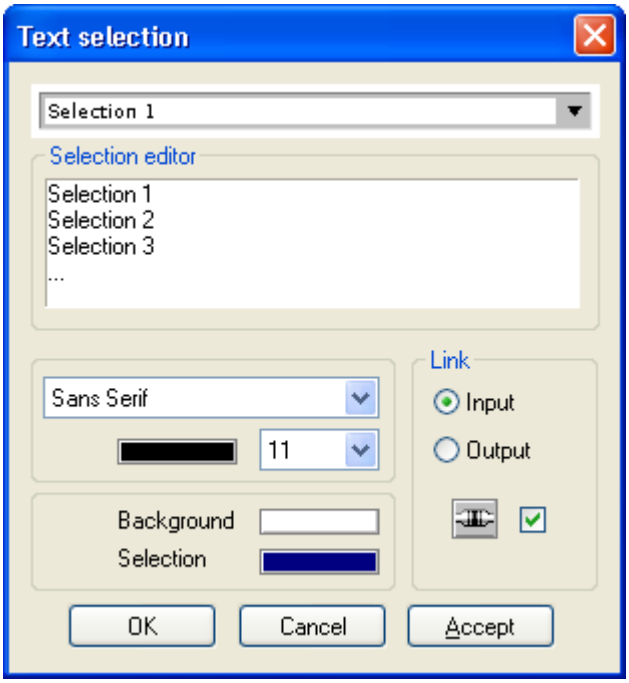
This object, which is linked to a variable, comes with a box for display of a unit to go with the numerical value is displays. The legend states the displayed quantity's name.



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.

7.6.7.4 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.



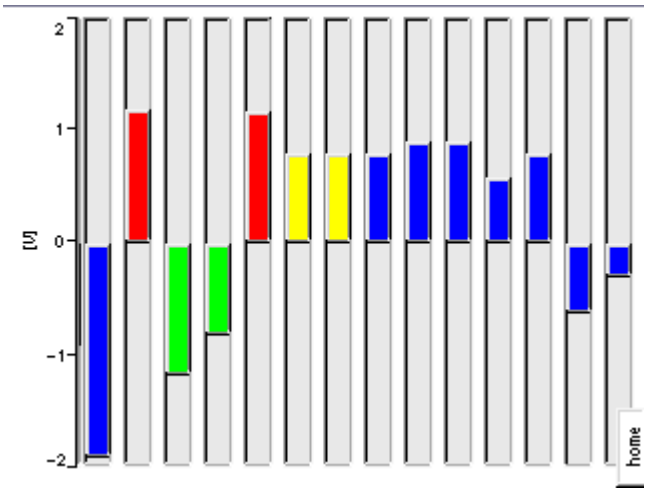
Enter one line of text for each expected state of the linked variables.

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..

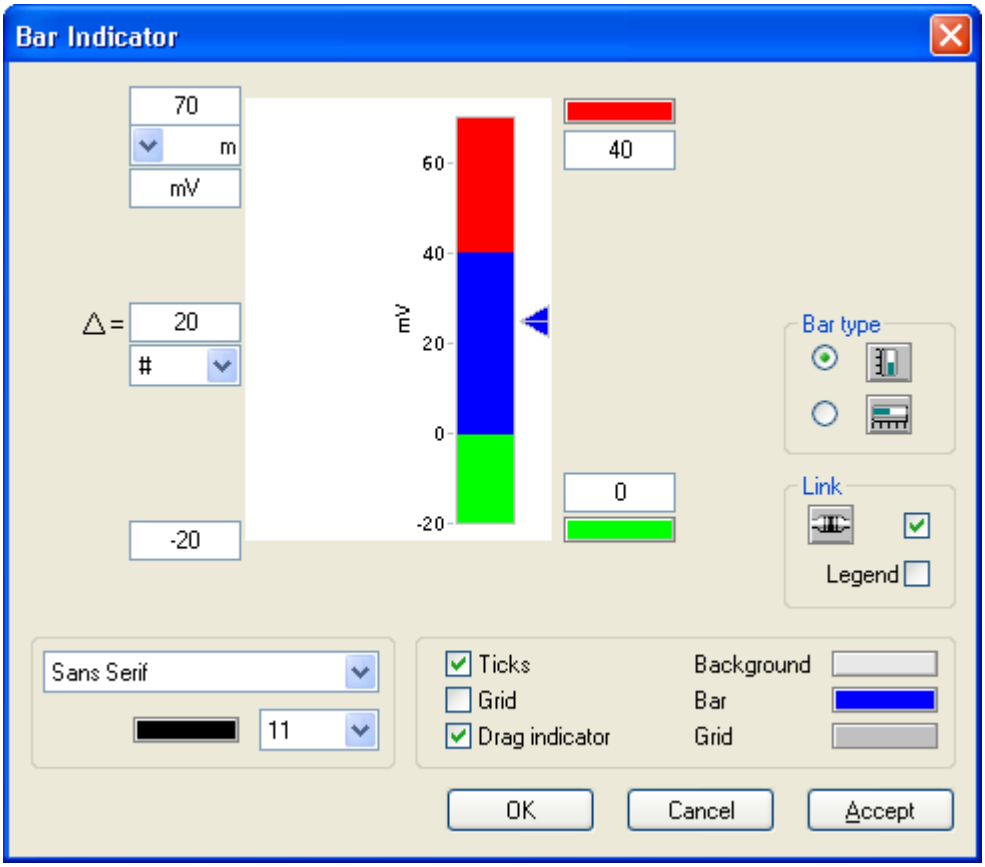
In the box Link, define the channel connection and designate the configuration either as an input or an output.

7.6.7.5 Bar Meter

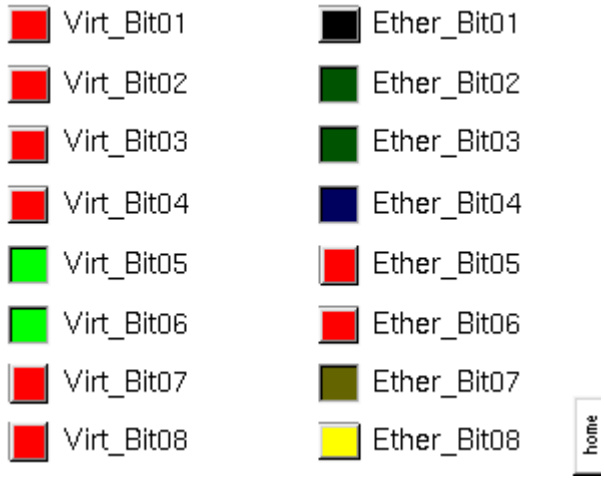
The bar meter is an output element which lets you graphically display a value which varies within a certain range (like a level indicator). The bar can run horizontally or vertically.



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.

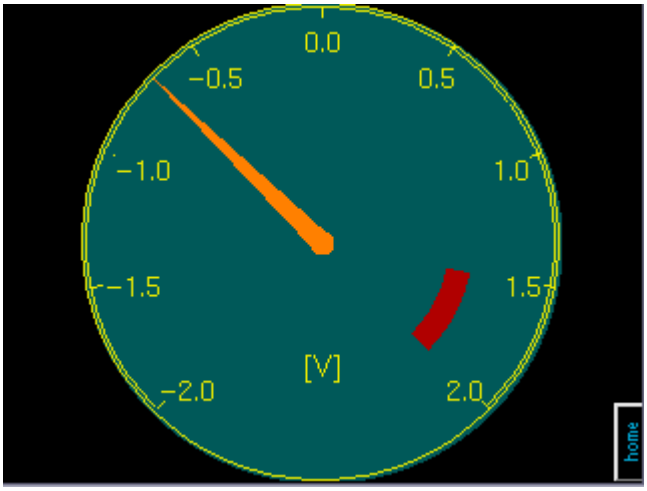


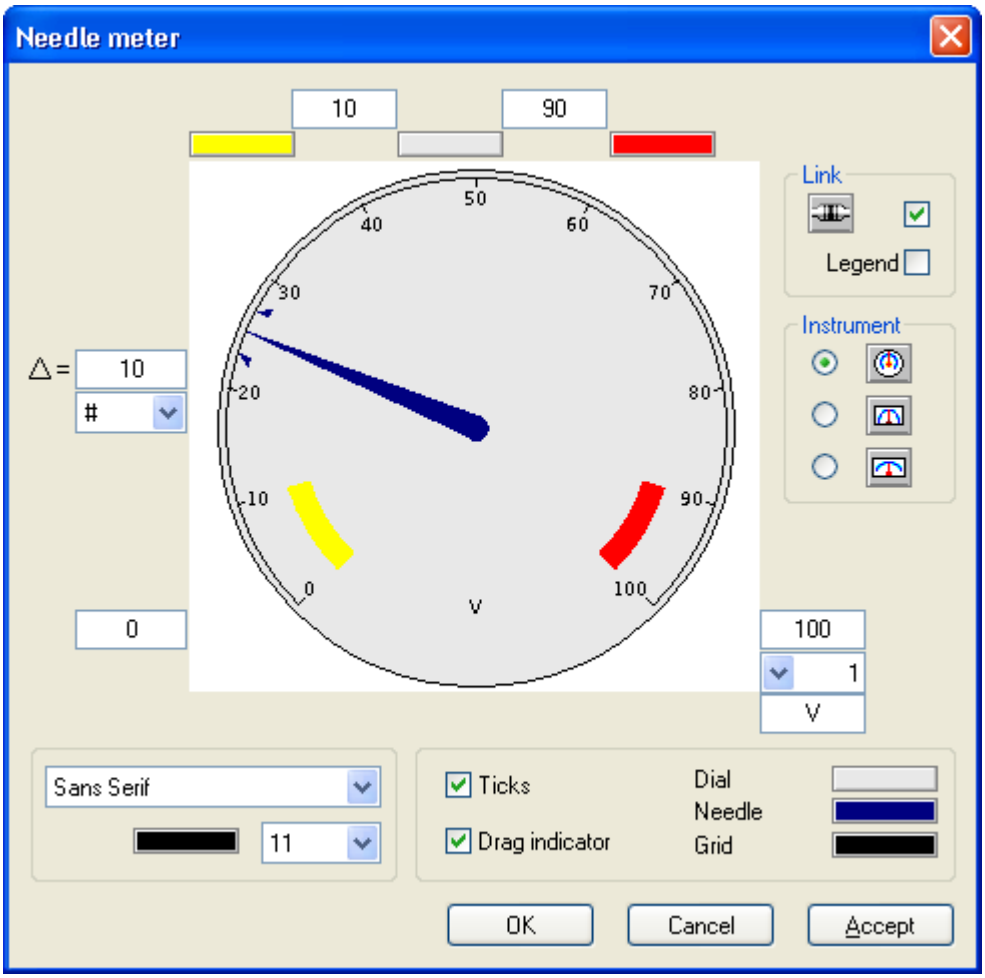
The bar meter can also be used to display digital states. In this case, there is simply no slave pointer or numerical values.



7.6.7.6 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.

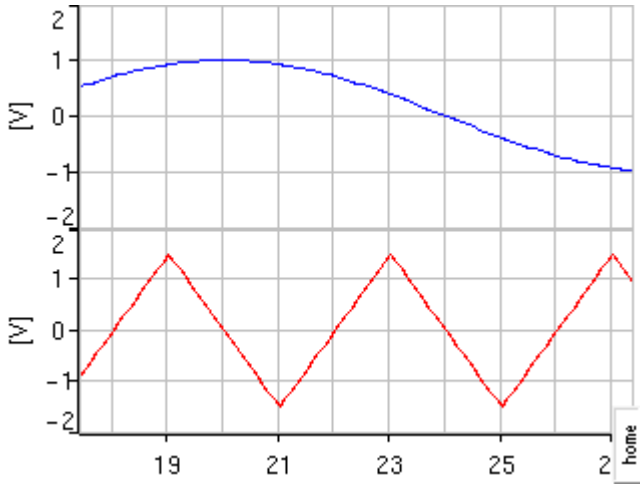




7.6.7.7 Curve Window/Line Chart

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.

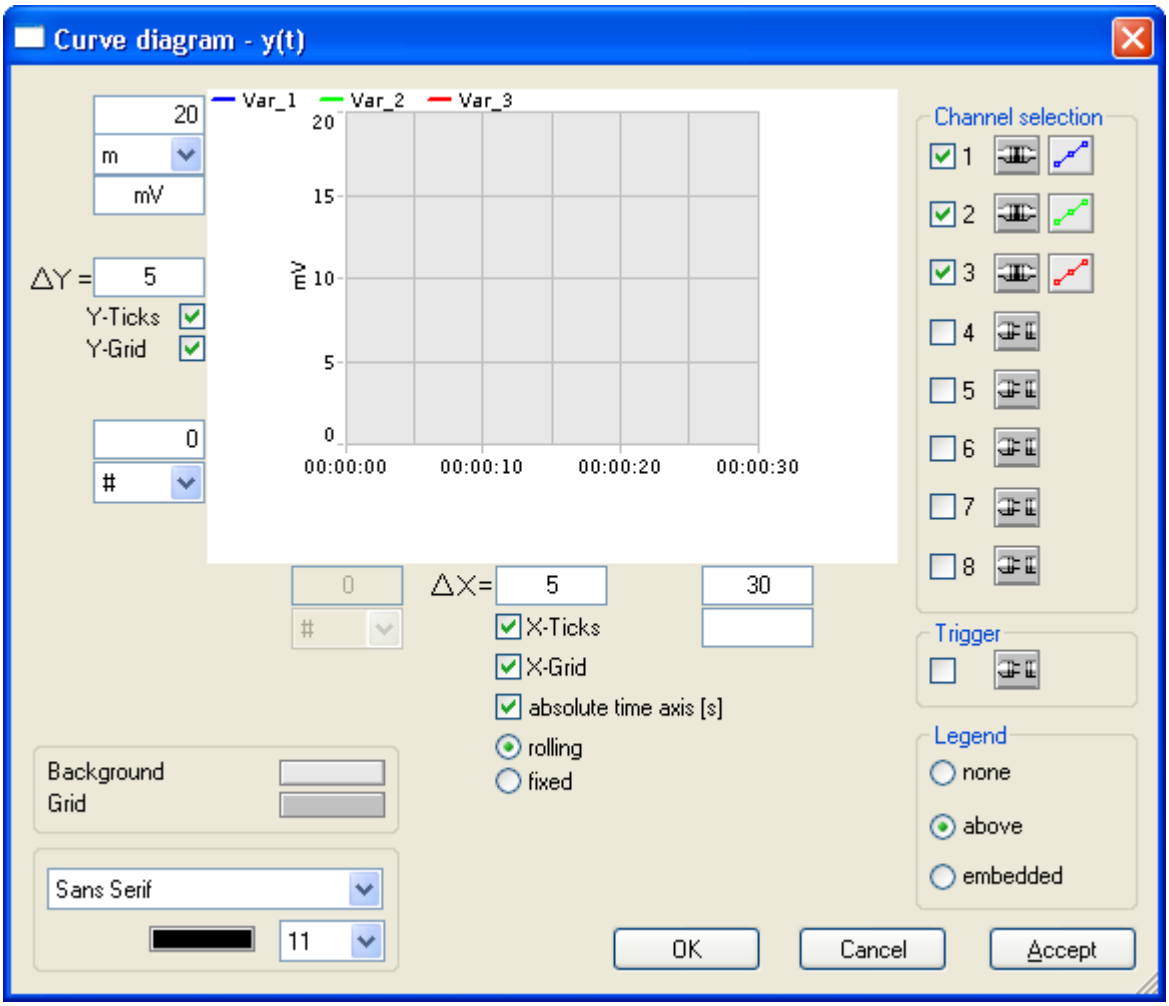


Each channel has a link to a Display variable, which can be activated or deactivated as desired. The dialog for making links has a control Legend, in which, for instance, the channel's name can be entered.

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.

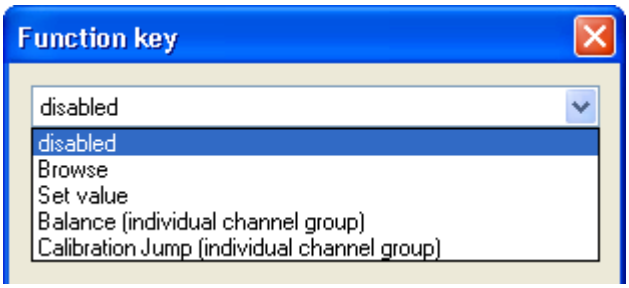
Both horizontal and vertical grid lines, either together or separately, can also be displayed.



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.

7.6.8 Function Keys on the Display



There are keys on the display's left and right edges which can be configured as desired.

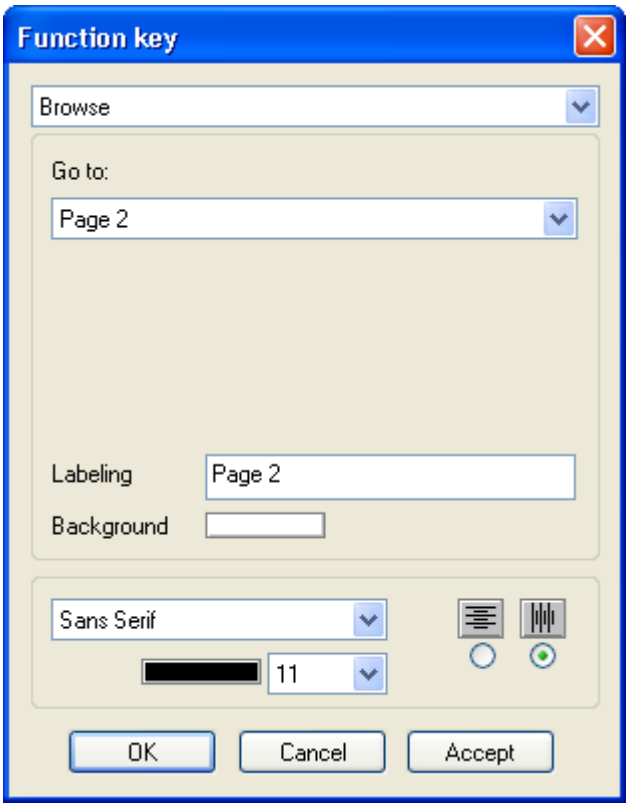
The keys are disabled per default. With the listbox you determine, whether the key:

- scrolls to a certain page
- sets the value of a variable
- or performs balancing and shunt calibration for dedicated channels.

Note

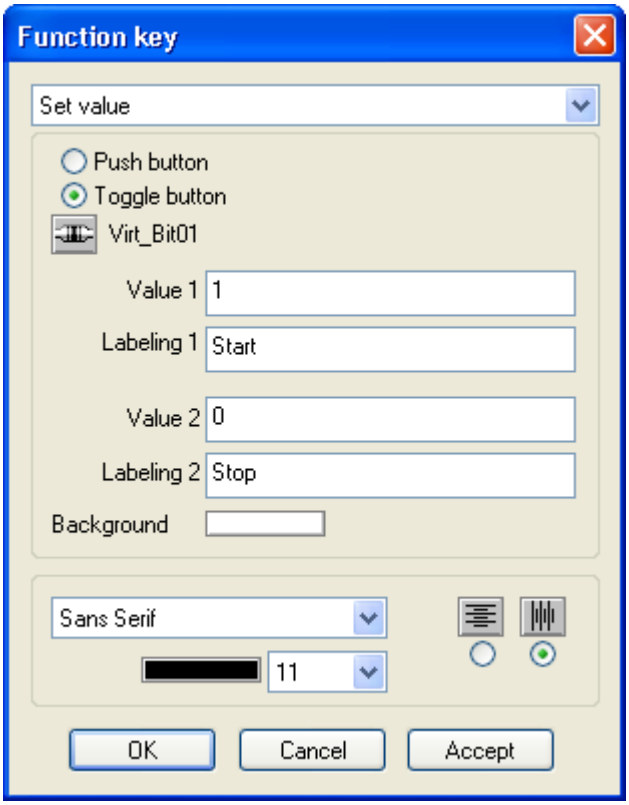
Using the imc Online FAMOS function `DisplaySetButton` 1472, the following functions can be executed by imc Online FAMOS as well.

7.6.8.1 Browse



If you have created several pages (Menu Edit 667), you can call those pages via function keys in mode *Browse*.

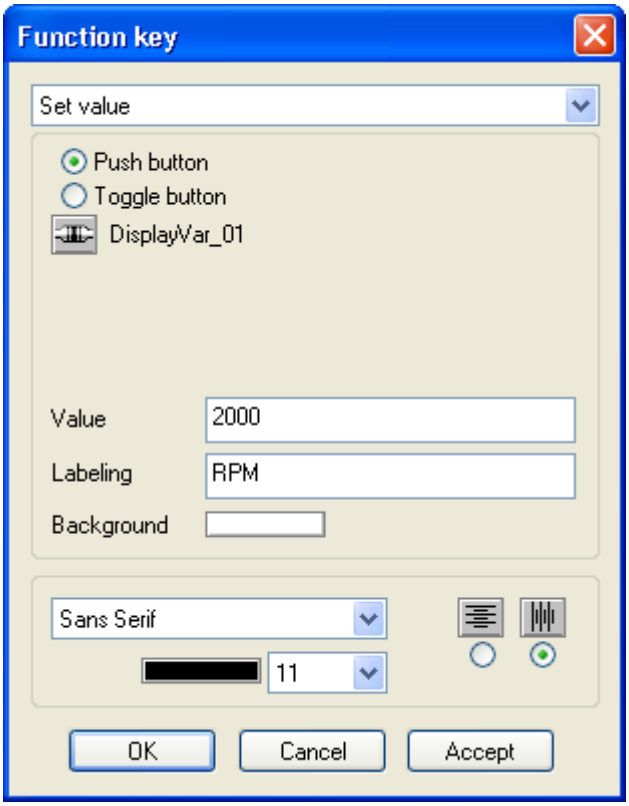
7.6.8.2 Set Values



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.

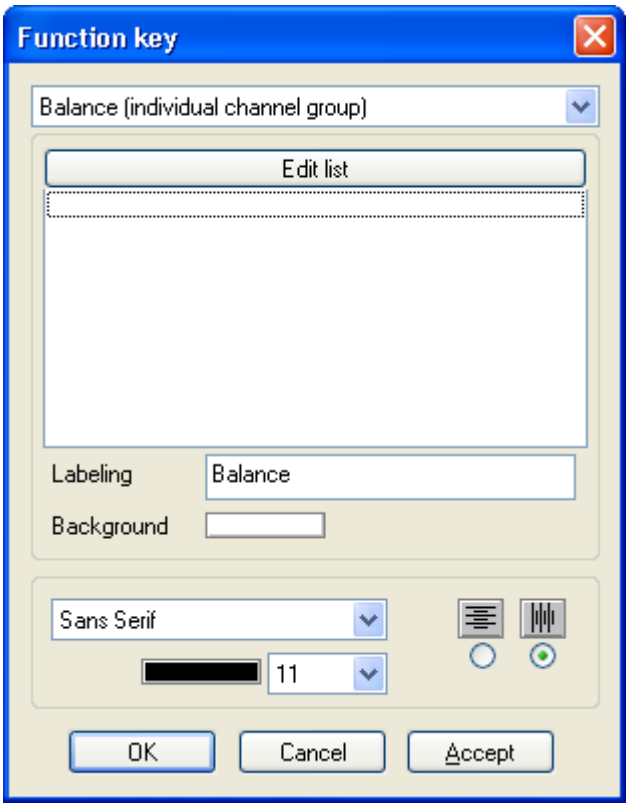


Push button:

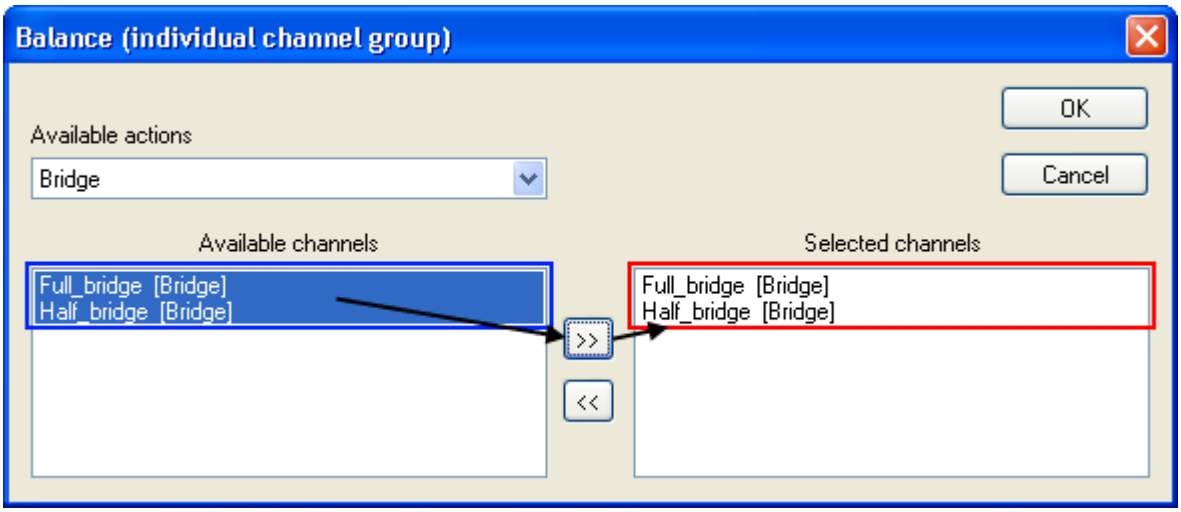
A key linked to a display variable, a certain value could be assigned to the variable that could control a process in imc Online FAMOS.

7.6.8.3 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.



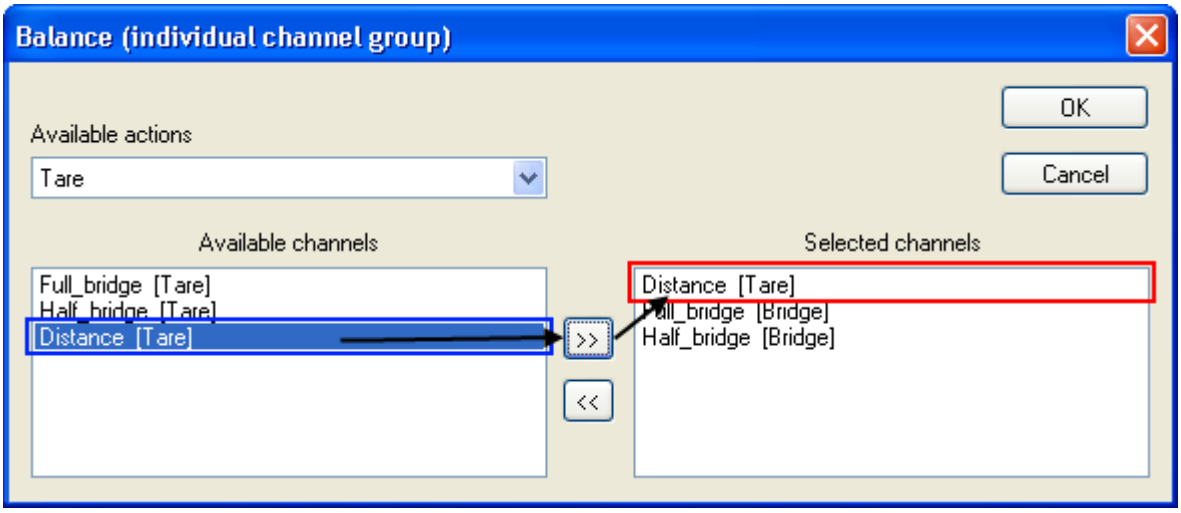
Clicking on the button *Edit list* opens a channel selection dialog.



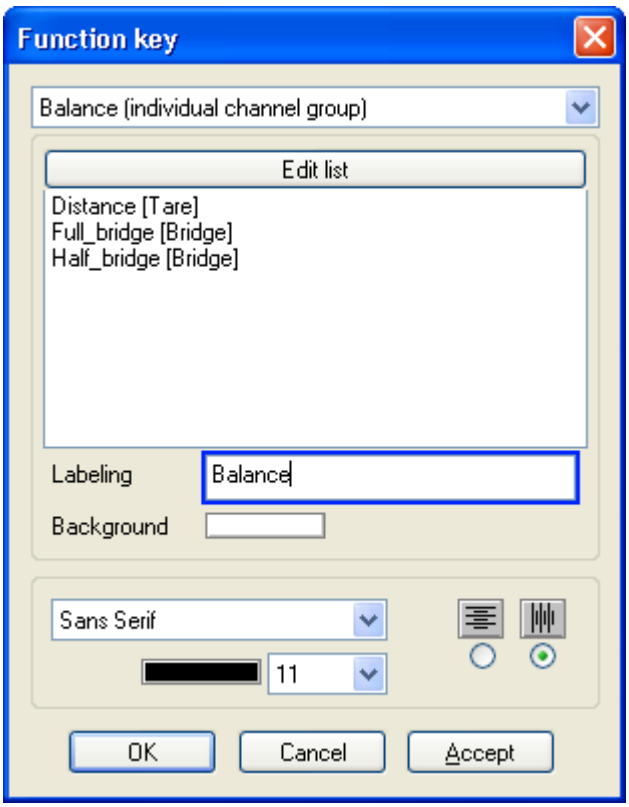
Balancing can also be carried out channel-by-channel. This selection dialog sets which channels are balanced when the Display function key is pressed.

Under *Available actions*, you have the choice between *Bridge* and *Tare*. Bridge channels can be either balanced or tared. For detailssee the manual: Plug-in: Setup > Channel balance 177.

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.



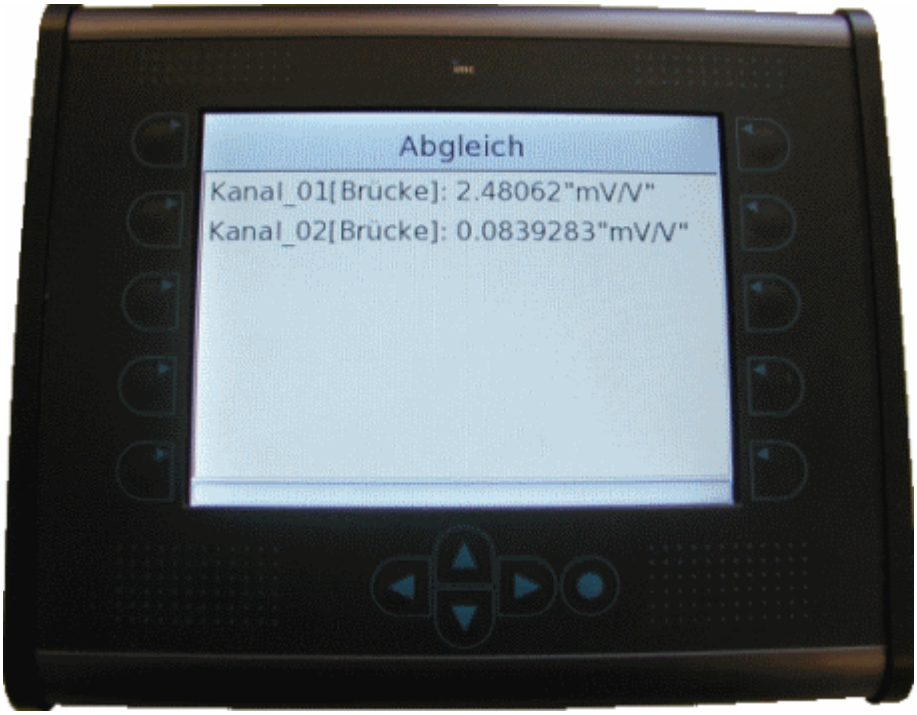
Under Available actions: *Tare*, an additional channel for capturing distance data appears.



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.

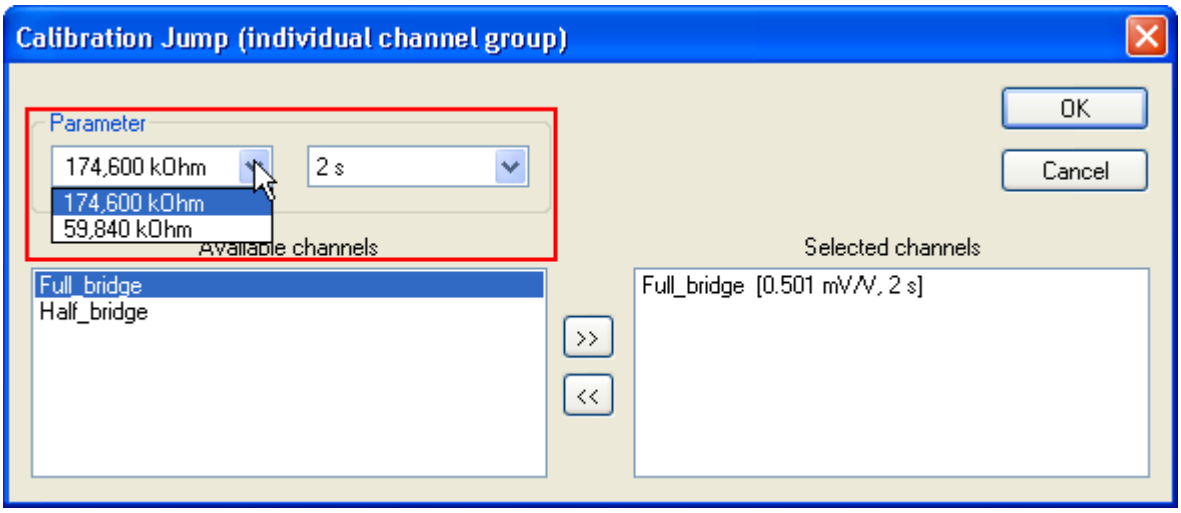
To carry out the action, the Display switches to balancing mode. Next, the compensated values are displayed:



Pressing the round button returns the screen to the normal view.

7.6.8.4 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 679. However, a shunt calibration 180 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.

7.6.9 Firmware Update

The display is running on its own operating system, that is updated with the firmware update 142. 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.

7.7 imc Messaging

This chapter describes the extensive features of imc measurement devices for sending text messages using different media (Email, FAX) and their operation using imc STUDIO.

7.7.1 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 network 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. The available message types include email and FAX in any combinations.

Along with obligatory text components, the message texts can also contain the current values of the network 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:

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 network 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 email when a threshold is exceeded or a similar event occurs.

The ability to place the current values of the network 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 network 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 network 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.

Important note

- Virtual bits and network bits should not change their value for durations under 50ms; better yet 200ms.
- Renamed virtual bits or network 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 686)!

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 105. 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 687.

System requirements for imc Messaging

Only devices having an appropriate interface are able to send messages. This includes all devices as of Group 2 88 and higher:

FAX — In order to send a *FAX*, a modem which supports G3-Fax Version 2 or 2.0 is needed. The fax machine must also be compatible with one of the two standards. This shouldn't present any problem for most current machines.

Email — *Email* can be sent either via *modem*, or via a *network interface*, if an appropriate server can be reached via the network.

If either a GSM-modem or a GSM mobile phone is used, then *it's not necessary to enable data transfer*.

There are no further requirements.

7.7.2 Message Types

The message types Email and FAX come with different strengths and weaknesses which can be considered when making a selection.

FAX

A fax is the most direct message form since it doesn't have an intermediary provider. The device dials the fax machine's number directly. There is also no time delay between the device's sending of the message and the arrival of the message at its destination. However, the fax machine would have to be permanently watched in order to use this advantage. Since a fax is transmitted as a bitmap graphic, the data volume transferred is much greater than that of an email. Therefore, transmitting a fax can take several minutes when using a GSM modem. A fax is only very conditionally suited to archiving, and re-processing is practically not possible at all. A fax message's size is limited to one page in imc Messaging. A modem is absolutely necessary for sending a FAX.

For the constraints on the rows and columns, see the technical specs.

Email

An email can contain almost any amount of text, with the size of the configuration file, however, setting certain limits. Transmission of email by the provider usually occurs after a few seconds, though there is no guarantee for this. By contrast, the email-client may be set to retrieve messages only at longer intervals. In terms of archiving and re-processing, emails are best suited. In addition, sending an email doesn't require a modem if a mail server is accessible via the LAN interface. In this way, the slow process of dialing via modem is also omitted, so that the emails are sent much more quickly.

If, for instance, the modem is occupied by a PC dialing in, it is still possible to send emails via the LAN. On the other hand, all messages which rely on the modem are detained until the modem is available again.

However, to send an email, 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 16xxx or 19xxx support *TLS/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 at the end of this chapter: e-mail via LAN 706 and e-mail via Modem 707

7.7.3 Reliability

7.7.3.1 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 686).

If the modem is busy because a PC has dialed in, then all messages which require that modem will be delayed until it's free. 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.

7.7.3.2 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, for instance due to a PC dialing in, all messages to be sent via modem must be held back until it's once again available. Therefore, an occupied modem won't lead to failed delivery!

Emails which can be sent via the LAN are delivered even if the modem is busy. If appropriate, they can be bumped up in the sending queue. Thus, the messages aren't necessarily sent in the order they were created!

If a message can't be delivered due to configuration errors such as a wrong telephone number or 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!

7.7.3.3 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 email or a FAX.

Note

If a folder has been created by a firmware version 2.5R1SP9 or older, the device is not able to overwrite the folder with new information!

7.7.4 Configuration

The configuration of imc Messaging is accomplished by means of one ASCII-file per device, each has to be imported as supplemental file 105. 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 704 lists the necessary entries for a variety of providers, making it possible to perform the configuration through simple cutting and pasting.

7.7.4.1 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, FAX
Message destination	`Destination`	Telephone number or Email 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`" or a "`FAX`" is to be sent. Depending on the message type selected, various additional specifications are needed.

Example: `MessageType=FAX`

The specification of the message recipient is made under the keyword "`Destination`". For a faxed message, the value in question is the fax machine's number. The number's digits must be entered without spaces. The recipient of an email may be one or more email addresses, separated by commas.

Example: `Destination=030123456`

or

`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 695).

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!)

7.7.4.2 Specifics of the different message types

7.7.4.2.1 Email

Emails 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 email.

On the other hand, if the email is to be sent via modem, this requires entering a whole series of parameters in addition to the specifications already described in chapter Templates 704.

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 TLS: 465, STARTTLS: 587
Connection encryption [1]	`SMTPServerSecurity*`	TLS or STARTTLS
Sender address	`SenderAddress`	An email address which is accepted by the mail server as a return address
Email's subject line	`Subject*`	Any text with variables

* optional; [1] only for devices with serial number 16xxx and higher

Note

Since imc DEVICES version 2.8R5SP9 the new keyword "*SMTPLoginMethod*" replaced "*SMTPAuthentication*".

For the "*SMTPAuthentication*" parameter, "*cram-md5*" and "*login*" were mixed up. This has been corrected with "*SMTPLoginMethod*".

For compatibility "*SMTPAuthentication*" is as previously supported. For example, "*SMTPAuthentication=cram-md5*" still activates "*login*". Thus, an existing configuration is still functional after software update . But in the future "*SMTPLoginMethod*" should be used instead.

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 email 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 email 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 email. 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

TLS: 465

STARTTLS: 587

SMTPServerSecurity

Selection of the connection encryption. Only for devices of serial number 19xxx and higher. The following encryption settings are possible:

TLS: encrypt entire communication.

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 email, a return address must always be given, otherwise delivery is denied. Some servers also check whether the email 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 email 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 email 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 email 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 695)

Example:

```
Subject=Limit violation in test Rig No.2
```

Destination

With emails, the recipient's address is given with "`Destination`". In contrast to all other messages, emails 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 email 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 email 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 email 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

# Email 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 email generated by an imc measurement device.
```

Example: A configuration having all specifications

```
# Send an email 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>

# Email 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 email generated by an imc measurement device.
```

Example: A configuration encrypted with STARTTLS

```
# Send an email if Virtual_Bit01 changes to 1.
# email 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!
```

Example: A configuration encrypted with TTLS

```
# Send an email if Virtual_Bit02 changes to 1.
# email without dialing in
[Virt_Bit02]
Event=TransitionToOne
MessageType=EMAIL

#SMTPServerPort=465
SMTPLoginMethod=cram-md5
SMTPServerSecurity=TLS
SMTPLoginName=MyName@freenet.de
SMTPPassword=MyPassword
SenderAddress=Mein Geraet <MyName@freenet.de>
Destination=phantom@company.de
Subject=Message from test bench {DeviceName}!
Message={DeviceName} {DateTime, %Y-%m-%d %H:%M:%S}
Message=Hallo!
Message=Keep smiling!
```

7.7.4.2.2 FAX

For a FAX, only the specifications described 684 are necessary. The fax machine's telephone number, alone, conveys all that is necessary for sending the message. You must only be sure to include any digits necessary to obtain a dialing tone, or any area codes.

Example

```
# sends a fax if Ether_Bit04 changes to 1.
[Ether_Bit04]
Event=TransitionToOne
MessageType=FAX

# 0 for dialing tone, area code 030 for Berlin, phone number 4567890
# All without spaces!
Destination=00304567890

# An example of a message text.
Message=Hello!
Message=
Message=This is a fax generated by an imc measurement device.
```

Note that both your modem and your fax machine must support the standard G3-Fax Version 2 or 2.0. However, this is usually the case anyway.

7.7.4.3 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 Email- and FAX-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 707.

7.7.4.4 Variables

The message texts and the subject lines of an email 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 682 and UDP Status Monitoring 700:

Variable	Description
`AllMessageCounter`	Total count of messages generate. Format numerical 697.
`AllMessageLosses`	Total count of messages which were discarded because undeliverable or due to lack of resources. Format numerical 697.
`DateTime`	Date and time in any format 698.
`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 697.
`DisplayVar_01-32`	the display variable's current contents Format numerical 697.
`Ether_Bit01-32`	the network bit's current state Format numerical 697.
`EventOverruns`	total number of events which couldn't be processed because the came too quickly in succession. Format numerical 697.
`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 697.
`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 697.
`ThisMessageLosses`	Placeholder in message text for the number of times a generated message of this kind was lost. Format numerical 697.
`TimeOfFirstStart`	Contains the starting time of the first measurement following the last preparation procedure in any format 698.
`Virt_Bit01-32`	the virtual bit's current state. Format numerical 697.

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`.

7.7.4.5 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)

Formatting specification	Definition
%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

7.7.5 Special characteristics of using network bits

When multiple devices are used jointly in a network, the transitions of the network bits affect all of the devices. This can be exploited to cause messages to be triggered by devices for which no modem is available.

For this purpose, the device having the modem is thus configured that it responds to a change of network bits by sending a message. The bits can then be controlled by any device in the network. Thus, only a single file containing all message configurations is necessary for the configuration.

Note that message preparation must be completed for all devices before the network bits can function as expected. Also be sure to consider that all devices, in fact, affect the network bits, even ones tasked with a completely other measurement.

In particular, preparation of a measurement resets all the network bits. In other words, if someone else prepares a measurement, this can cause your devices to already begin sending messages!

7.7.6 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 105 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 or a DNS-name 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 If-Config, 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 695 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 700.

See also: How does configuration via FTP work? 709

7.7.6.1 Syntax / example /Limitations

`[UDP Status Monitoring]`	Keyword for section UDP Status Monitoring
`DestinationIP`	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]
DestinationIP=192.168.0.1
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}
```

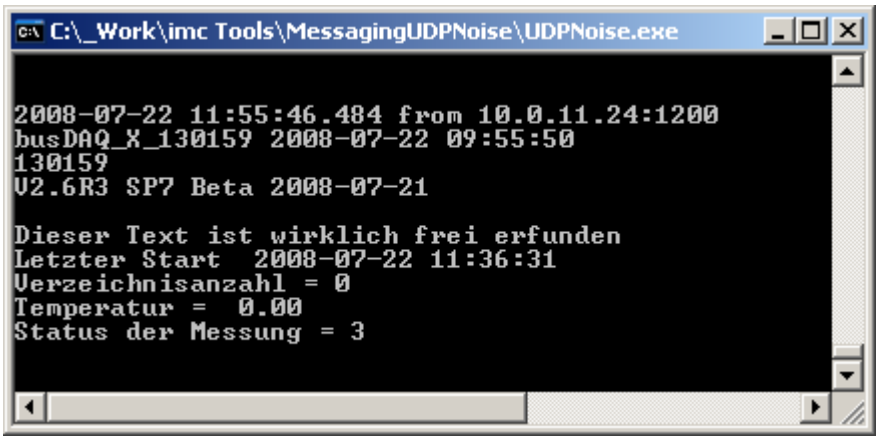
Find the variables and their format in chapter Variables 695.

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.

7.7.6.2 Test of UDP Status Monitoring

For testing the UDP Status Monitoring, the program *UDP Status Monitoring.exe* is provided (included in the DVD).



Messages arriving via UDP

7.7.7 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 687. 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.

7.7.8 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 email 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 device sends every fax numerous times, spread out over a large time frame.

Cause: Your fax machine isn't totally compatible with the modem used, so that it seems to the device that the fax was lost. Thus, the device repeats the message for such a long time that it finally gives up.

Remedy: Replace the modem or fax machine!

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: When using imc Messaging in conjunction with automatic callback, it sometimes takes unusually long for the device to call back.

Cause: The device may send a message between your call and the return call. Due to this, the return call may be delayed by as much as approx. 5min.

Remedy: None.

Problem: Having configured messages for network bits, these messages are now constantly triggered although the bits can't have changed.

Cause: The network bits are affected by every device in the network. It may happen that a device in use by someone else has changed some network bit(s).

Remedy: Construct a separate network, or use the virtual bits.

Problem: When working with multiple devices, the first messages triggered by the network 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 network bits before concluding preparations for all of the devices.

7.7.9 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
Email sending duration	maximum permitted: 80s
Number of characters in an email	limited only by size of configuration file
FAX sending duration	maximum permitted: 280s
Line length of a FAX	maximum: 75 characters
Number of lines in a FAX	maximum: 75 lines
Input ranges	
Bit variables Counter variables Display-variables	0 or 1 32 bits unsigned (0 to 4294967295) 32 bits floating point (-3.4e+38 to +3.4e+38)

7.7.10 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 email address with a provider, in order to obtain a login name. This can be done via the provider's website.

GMAIL

```
MessageType=EMAIL
SMTPServer=smtp.gmail.com
SMTPLoginMethod=login
SMTPServerSecurity=TLS
SMTPLoginName="your login name with this provider"
SMTPPassword="your password with this provider"
SenderAddress="your email address with this provider"
Destination=name@provider.de
Subject=My test message
Message=Hello!
#all sent emails are saved automatically!
```

GMX

```
SMTPServer=pop.gmx.net
SMTPAuthentication=cram-md5
SMTPLoginName=your login name with this provider)
SMTPServerSecurity=StartTLS
SMTPPassword=(your password with this provider)
SenderAddress=(your email address with this provider)
```

WEB.DE

```
MessageType=EMAIL
SMTPServer=smtp.web.de
SMTPLoginMethod=login
SMTPServerSecurity=StartTLS
SMTPLoginName=user # without @web.de
SMTPPassword="your password with this provider"
SenderAddress="your email address with this provider"
```

SMARTMOBIL (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 email address with this provider"
```

Freenet

```
SMTPServer=mx.freenet.de
SMTPAuthentication=cram-md5
SMTPLoginName= (your login name with this provider)
SMTPPassword= (your password with this provider)
SenderAddress= (your email address with this provider)
```

7.7.11 Step by Step

7.7.11.1 Setup Connection

Modem

1. Prepare modem with SIM Card.
2. Choose initialization script with IF-Config and enter the correct PIN, see Manual: *Setup - Connect the device > Special options for connecting to the device > Modem connection (TCP/IP with PPP) > Configuring devices as a PPP-Server* 324.
3. Prepare the device and connect the modem. Wait until the modem starts blinking with about 1 Hz.

WLAN

See Manual: *Setup - Connect the device > Special options for connecting to the device > Device connection via WLAN* 313.

LAN

See Manual: *Setup - Connect the device > Special options for connecting to the device > LAN-Configuring the imc units* 307.

7.7.11.2 Sending an Email

An Email 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 105 into the experiment.

Sending an Email via LAN

For this example, the device must be connected to the internet via LAN.

Example

```
# Send an Email, if Virtual_Bit01 is set to 1
[Virt_Bit01]
Event=TransitionToOne
MessageType=EMAIL

# IP-address of mailserver
SMTPServer=mx.freenet.de
SMTPAuthentication=login
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>

# Email addresse of recipient
Destination=john.smith@imc-berlin.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
```

Sending an Email via modem

For this example, a modem 706 is necessary.

Example

```
# Send an Email, if Virtual_Bit01 is set to 1
[Virt_Bit01]
Event=TransitionToOne
MessageType=EMAIL

# Data for Internet access
# 0 for dial tone
ProviderPhoneNumber=22243
ProviderLoginName=freenet
ProviderPassword=mobil

# Data for mail server access
SMTPServer=mx.freenet.de
SMTPAuthentication=login
SMTPServerSecurity=StartTLS
SMTPLoginName=TestMail@freenet.de
SMTPPassword=MyPassword
SenderAddress=My device <TestMail@freenet.de>

# Email addresse of recipient
Destination=john.smith@imc-berlin.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
```

7.8 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 (for devices as of group 2 88 and higher). The command is called by transmitting a text file 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 device

The autostart configuration will be started. Without autostart configuration, the device will be started but without recording.

2. Reconfiguring and starting the device 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 device 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.

Procedure

All file actions on the device are located directly in the root folder.

1. User:

- If necessary, delete from the device the files "xmlrpccmd.call" and "xmlrpccmd.call.valid" .
- If necessary, delete from the device the files "xmlrpccmd.response" and "xmlrpccmd.response.valid".
- If necessary, copy the new configuration file to the internal data storage medium.
- Copy "xmlrpccmd.call". This file must contain a valid command description (see examples supplied).
- Copy "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.

7.8.1 How does configuration via FTP work?

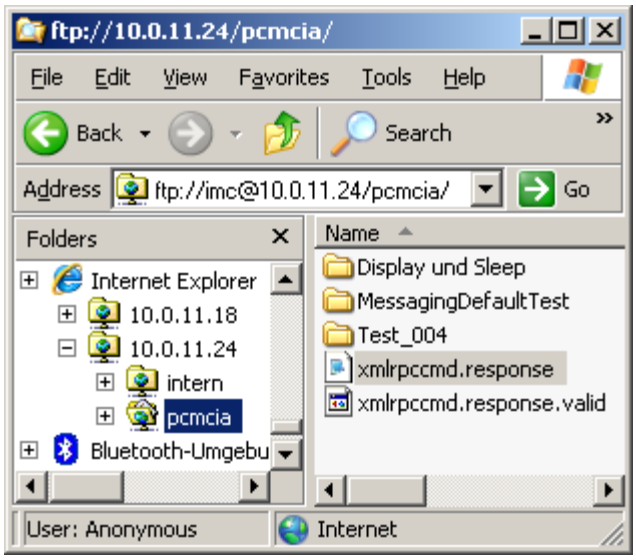
The following example shows how an imcMessaging default configuration is activated. At the same time, the device is instructed to perform cyclical transmission of information via UDP Status Monitoring.

1. First create a small self-activated experiment with a device as of group 4 88 and higher.

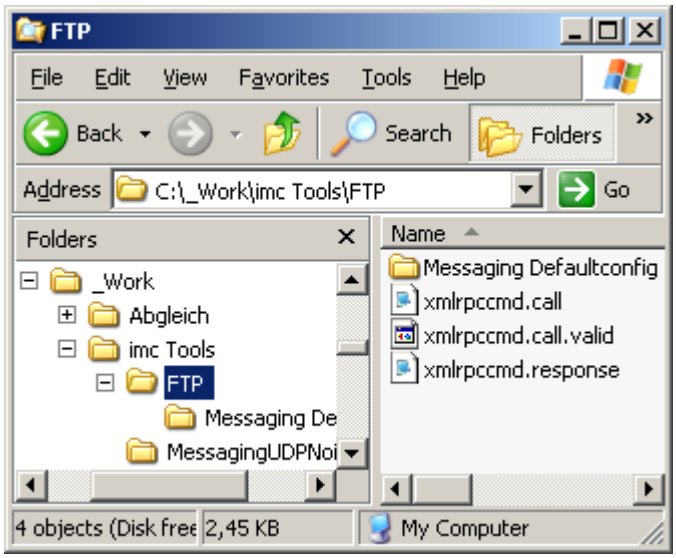
   While the experiment is running, Virtual Bit 1 must be set. The performance of this event can be automated with imc Online FAMOS, for example, using a timer or a ramp function. Save this experiment to the device as a self-activation configuration.

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

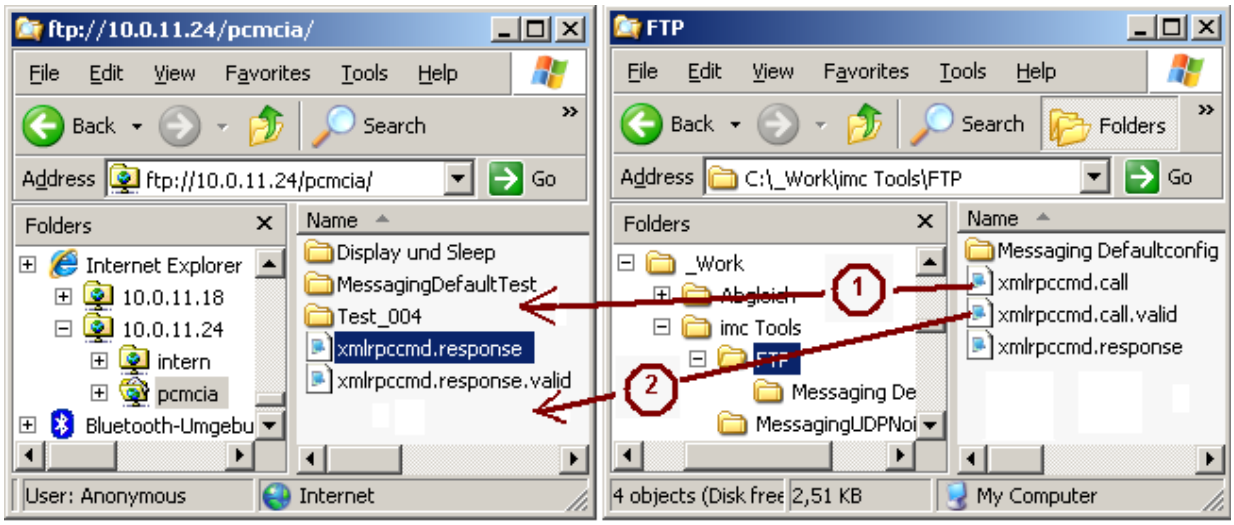
3. Create a text file "xmlrpccmd.call" and copy the following sample text to this file:

   The segments of code appearing below in boldface must be adapted. The comments, appearing in green, are to be deleted subsequently.

```
"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-berlin.de
; e-mail address to which the device sends the message generated.
Destination=Max.Mustermann@imc-berlin.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>

"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].

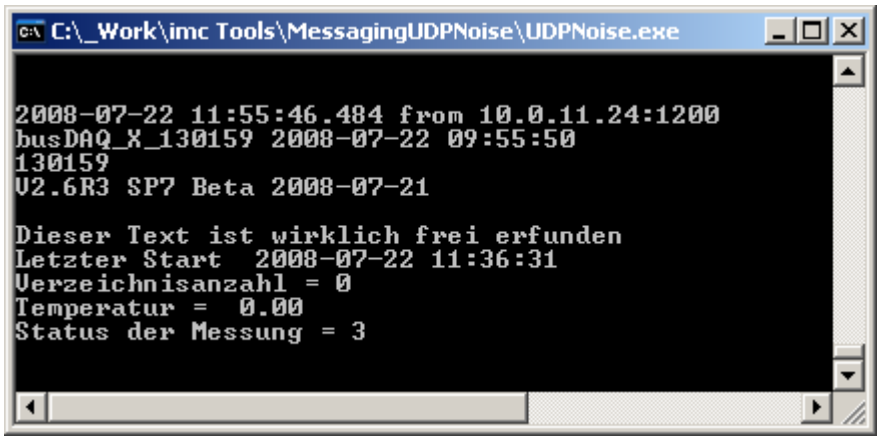
   Now the copied files disappear and xmlrpccmd.response and xmlrpccmd.response.valid appear. The response file can be read with a text editor, and its return value is 0 if everything worked. Otherwise, an error code is returned. In this case, check your xmlrpccmd.call file again. If you don't find any errors, please contact our hotline.

```
...
- <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

7.8.2 Syntax of command call and response

The following commands are implemented:

XmlRpcCmdReconfigure Reconfiguring and starting the device.

The following parameters can be specified in the command call (file: "xmlrpccmd.call"):

SourceDrive:

"Removable" -> PCMCIA or CF-Card
"Nonremovable" -> IDE or CF-Adapter

SourceFile:

"devXXX.ums.zip" or "devXXX.ume.zip", XXX refers to the device number.

DestinationDrive:

"Removable" -> PCMCIA or 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:

- *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 694

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!

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:

- *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

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:

- *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.

7.8.2.1 Reconfiguring and starting the device

```
"xmlrpccmd.call":
<?xml version="1.0"?>
<methodCall>
   <methodName>XmlRpcCmdReconfigure</methodName>
</methodCall>

"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>
```

7.8.2.2 Reconfiguring and starting the device with the specified configuration

```
"xmlrpccmd.call":
<?xml version="1.0"?>
<methodCall>
   <methodName>XmlRpcCmdReconfigure</methodName>
   <params>
     <param>
       <value>
         <struct>
           <member>
             <name>RequestID</name>
             <value>2007-10-18 12:34:38</value>
           </member>
           <member>
             <name>DestinationDrive</name>
             <value>Removable</value>
           </member>
           <member>
             <name>Experiment</name>
             <value>Test_001</value>
           </member>
         </struct>
       </value>
     </param>
   </params>
</methodCall>

"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>
```

7.8.2.3 Reconfiguring and starting the device with the transmitted configuration.

```
"xmlrpccmd.call":
<?xml version="1.0"?>
<methodCall>
   <methodName>XmlRpcCmdReconfigure</methodName>
   <params>
     <param>
       <value>
         <struct>
           <member>
             <name>RequestID</name>
             <value>2007-10-18 15:49:06</value>
           </member>
           <member>
             <name>SourceDrive</name>
             <value>Removable</value>
           </member>
           <member>
             <name>SourceFile</name>
             <value>dev002.ume.zip</value>
           </member>
           <member>
             <name>DestinationDrive</name>
             <value>Removable</value>
           </member>
           <member>
             <name>Experiment</name>
             <value>Test_001</value>
           </member>
         </struct>
       </value>
     </param>
   </params>
</methodCall>

"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>
```

7.8.2.4 Get version with GetSoftwareVersion

```
"xmlrpccmd.call":
<?xml version="1.0"?>
<methodCall>
   <methodName>GetSoftwareVersion</methodName>
   <params>
     <param>
       <value>
         <struct>
           <member>
             <name>RequestID</name>
             <value>2008-06-02 14:23:56</value>
           </member>
         </struct>
       </value>
     </param>
   </params>
</methodCall>

"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</name>
             <value>V2.6R3 SP7 Beta (31.5.2008)</value>
           </member>
         </struct>
       </value>
     </param>
   </params>
</methodResponse>
```

7.8.2.5 imc Messaging default configuration

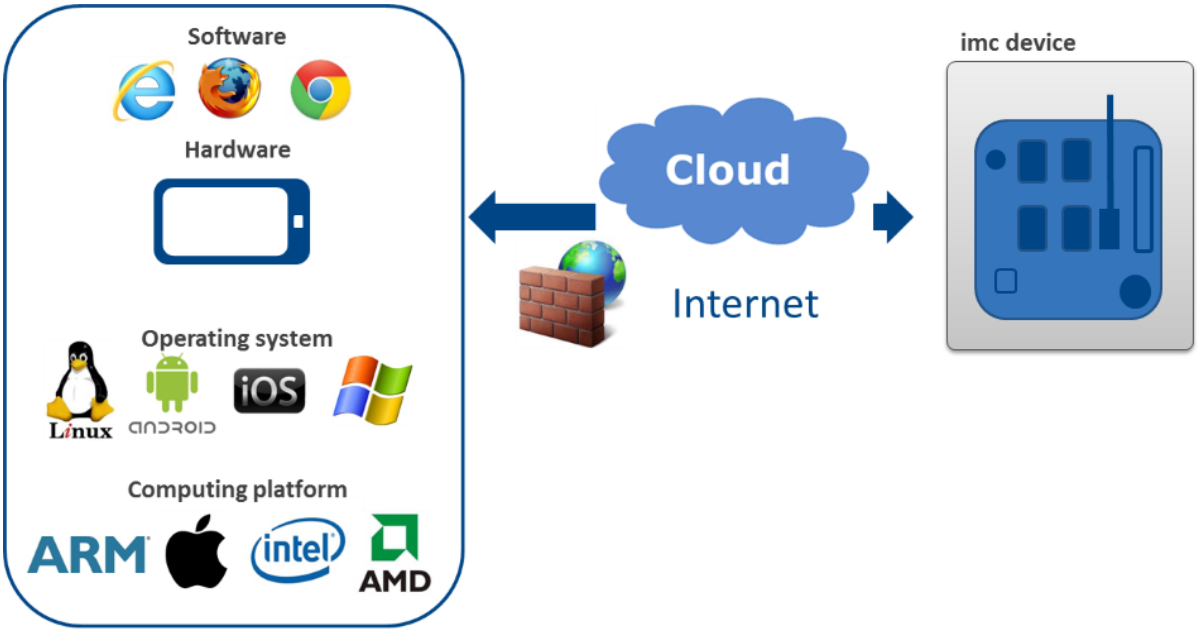
7.8.2.5.1 Delete default configuration

```
"xmlrpccmd.call":
<?xml version="1.0"?>
<methodCall>
  <methodName>SetimcMessagingDefaultConfig</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>RequestID</name>
            <value>2008-06-02 13:25:12</value>
          </member>
          <member>
            <name>Configuration</name>
            <value></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>

"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>
```

7.8.2.5.2 Set default configuration

```
"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=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>

"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>
```

7.9 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 DEVICES or 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 (Ether bits)
- Altering of the system configuration by selection among Auto- and Diskstart experiments pre-loaded to the measurement device
- Downloading of measured data (files)

7.9.1 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

Internet Browser	Device software
• Firefox Version 16 • Internet-Explorer Version 9 • Google Chrome 26.0 • Opera 12.02 • Safari 5.1.7	• imc DEVICES Version 2.9R2 and higher • imc STUDIO Version 5.0 and higher • Option imc REMOTE with imc REMOTE WebServer (including LinkSecure)=3 117 (one license per device required)

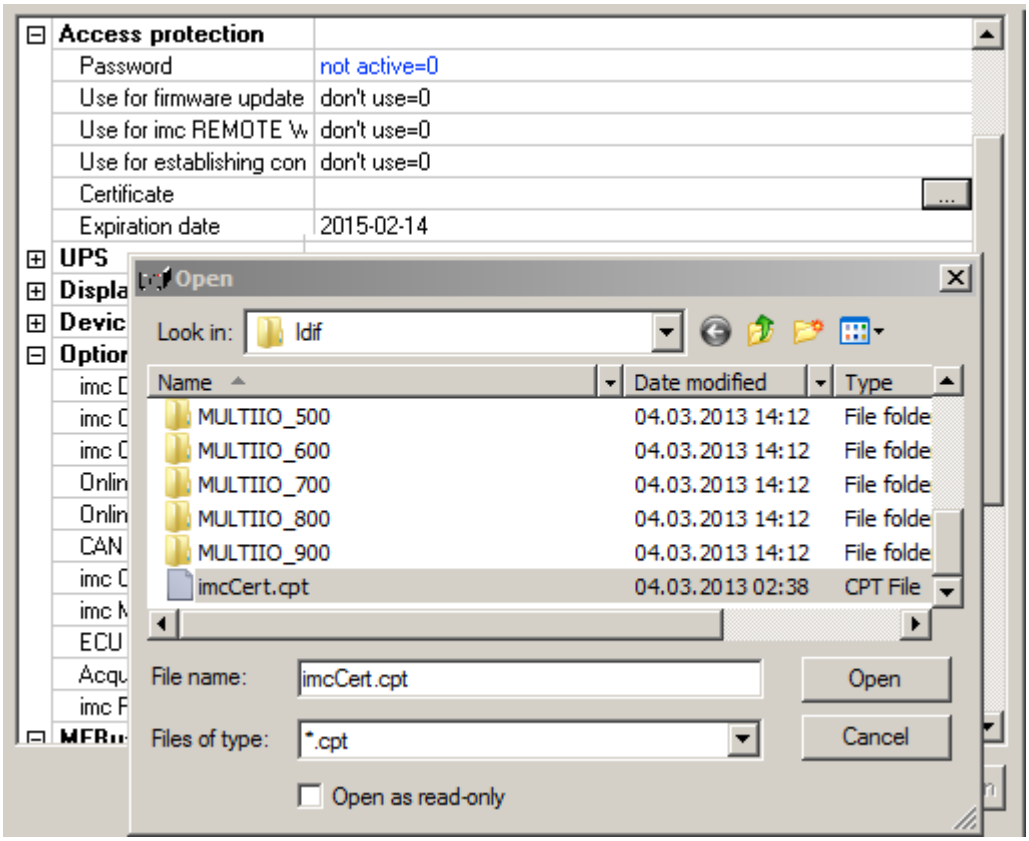
Measurement devices for imc REMOTE WebServer	Measurement devices for imc REMOTE WebDesigner
Devices from group 5 on are supported: • imc CRONOS*compact* • imc CRONOS*flex* • imc BUSDAQ*flex* • imc SPARTAN-R • imc C-SERIES-N • miniPOLARES	Devices from group 7 on are supported: • imc CRONOS*compact* • imc CRONOS*flex*

7.9.2 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.

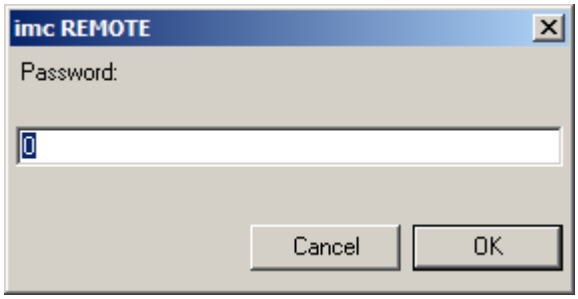
1. Open the dialog Device Properties (Ribbon *Setup-Configuration* > *Device Properties*.

2. Load the certificate *imcCert.cpt*, which is required for secure access via https. The software points directly to the correct folder.



3. Activate imc REMOTE : imc REMOTE WebServer (including LinkSecure)=3

4. Enter the activation code which you obtained by purchasing the option.



5. If access is provided via Internet, we recommend that in the branch *Access protection*, you set the setting *Password* to "*active = 1*."

6. Close the dialog *Device Configuration* by clicking on *OK.*

7. The imc REMOTE WebServer is activated. Remember the password for the purpose of subsequent calls of WebServer.

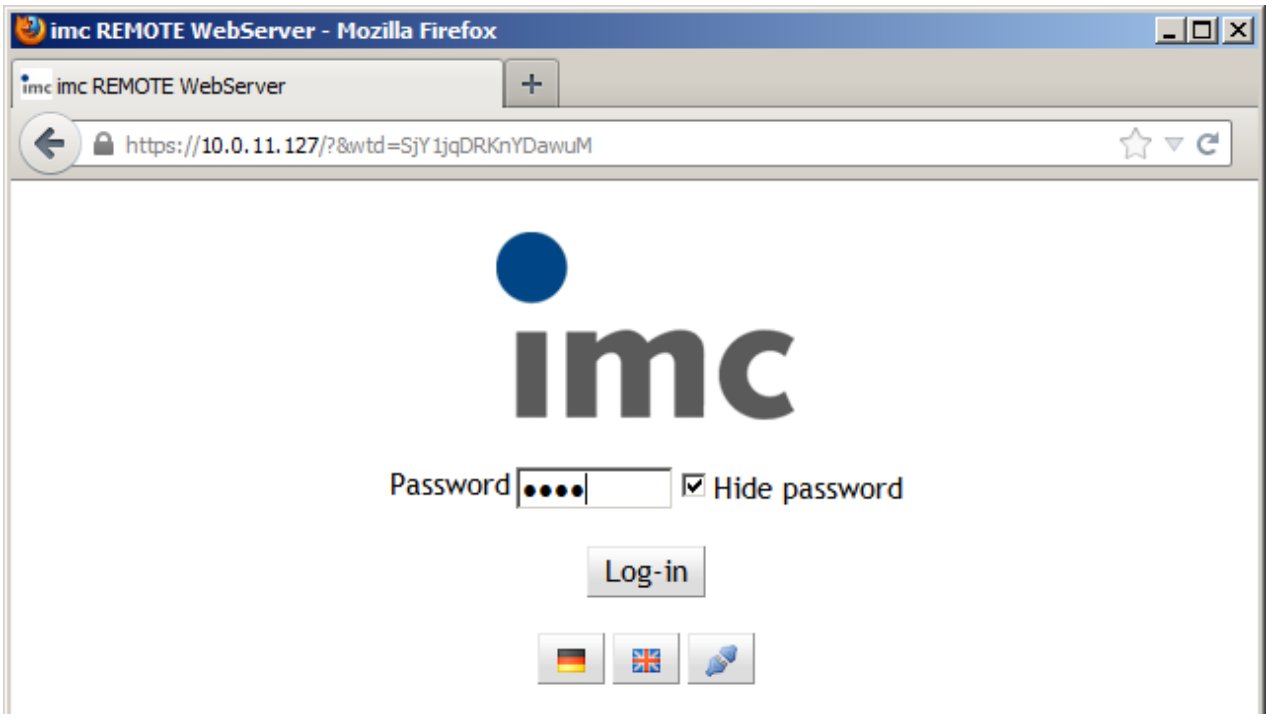
7.9.3 Calling from the Browser

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.

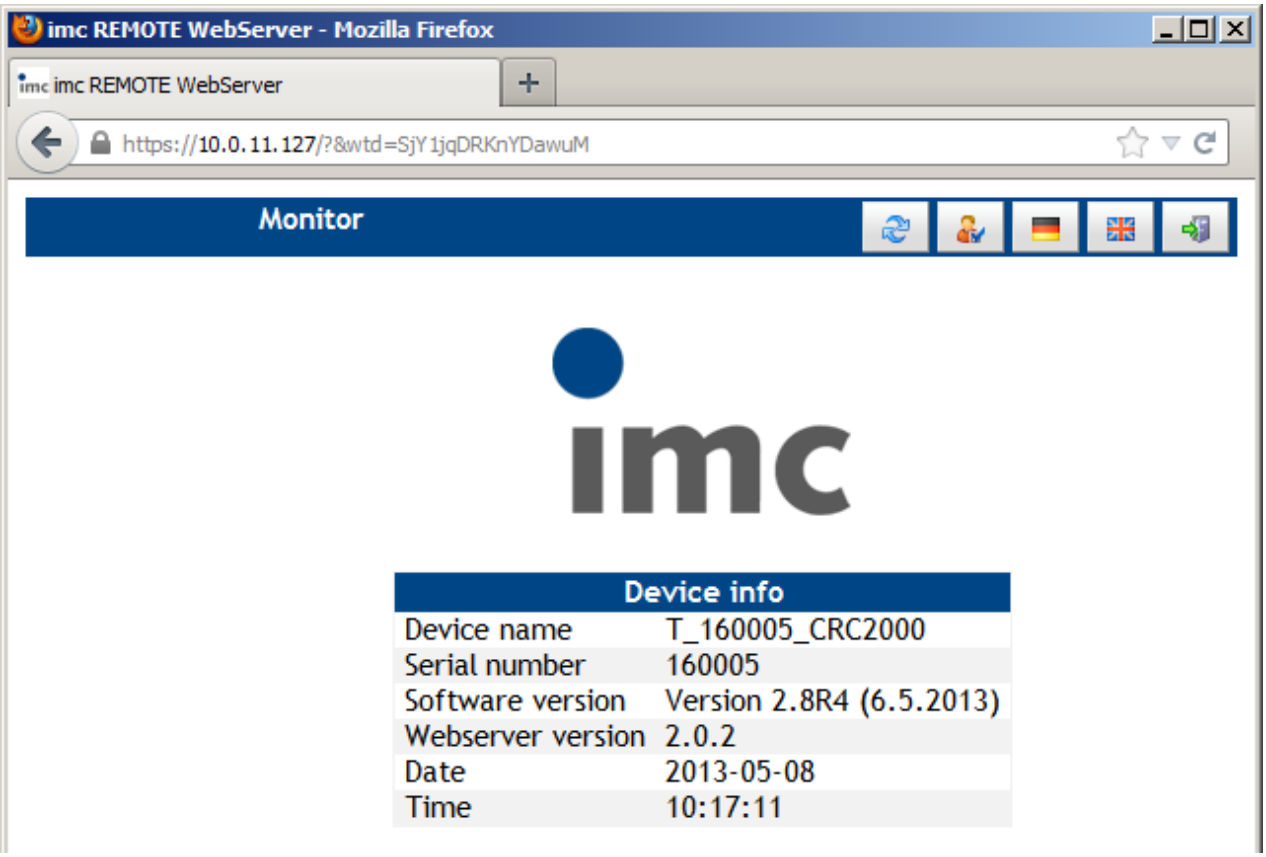
The IP-address for the device can be viewed and set using the dialog Device interface configuration 307 (Ribbon *Setup-Configuration* > *Device interfaces*).

The input mask for the password you previously entered for the user-defined device appears.



Access to the device and Log-in

After successfully entering the password, the device information appears in the user role 725 "Monitor".



Main page of WebServer

7.9.3.1 Language Selection

The language displayed can be changed at any time:

7.9.3.2 Closing a session

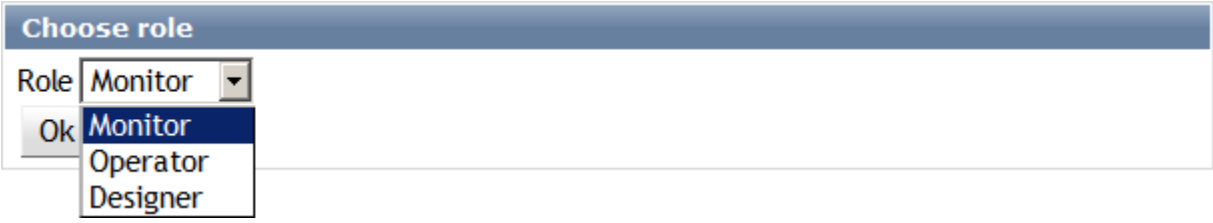
Always close a session by clicking on the Close symbol at the upper right corner.

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.

7.9.4 User Role

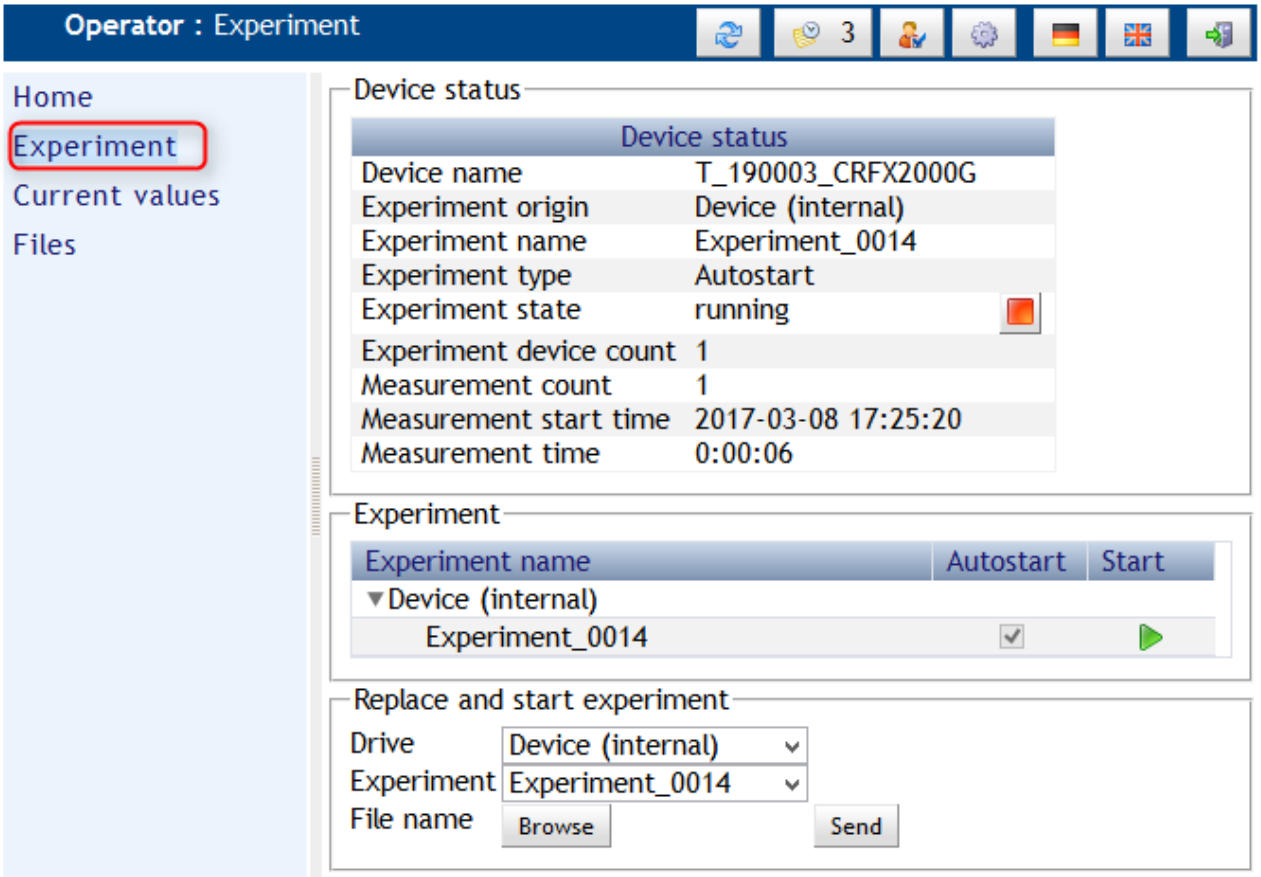
Version 2.0 and higher: There are three different user roles. By clicking on this icon, it is possible to change the selection of the user role.



User Role	Description
Monitor *(no password)*	• Observation of the system and measurement status • Querying of the system software-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 DEVICES or 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, Ether 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.

7.9.5 Experiment

Experiments already present on the system can be replaced via the WebServer. This is only possible in Operator mode.



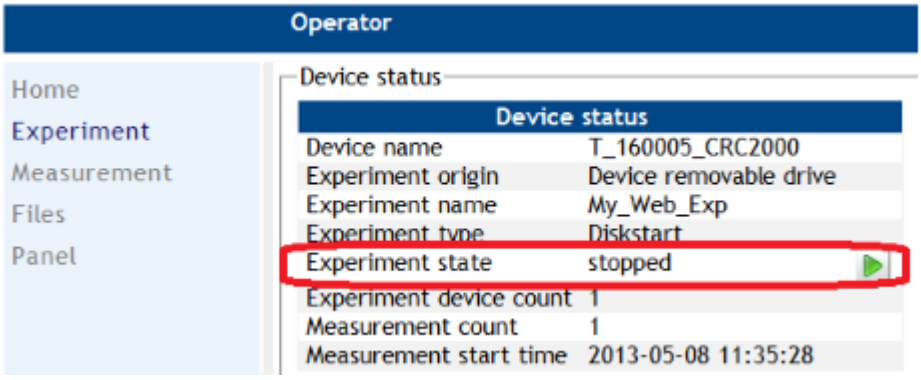
Step by step:

1. Create an Autostart- or Diskstart experiment 99 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.

7.9.5.1 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*.

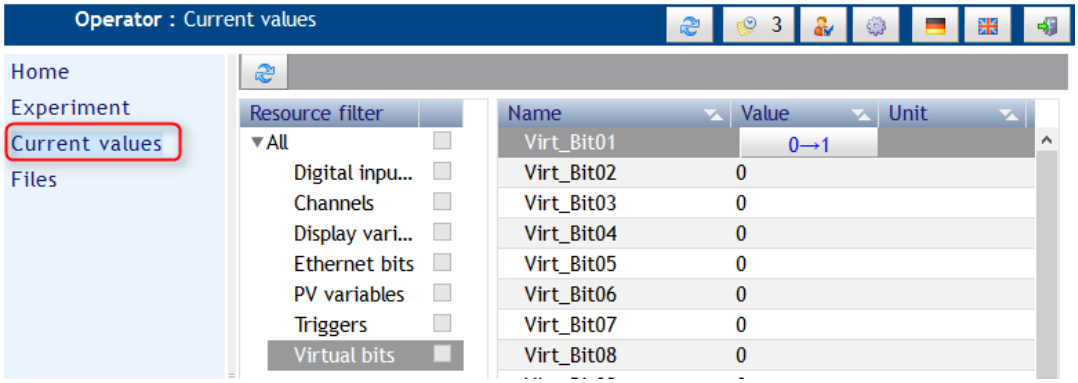


Note

If the measurement device is additionally connected with imc STUDIO or imc DEVICES, the function Start/Stop with imc Remote WebServer is deactivated. In this case, the Start/Stop button is hidden.

7.9.6 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.

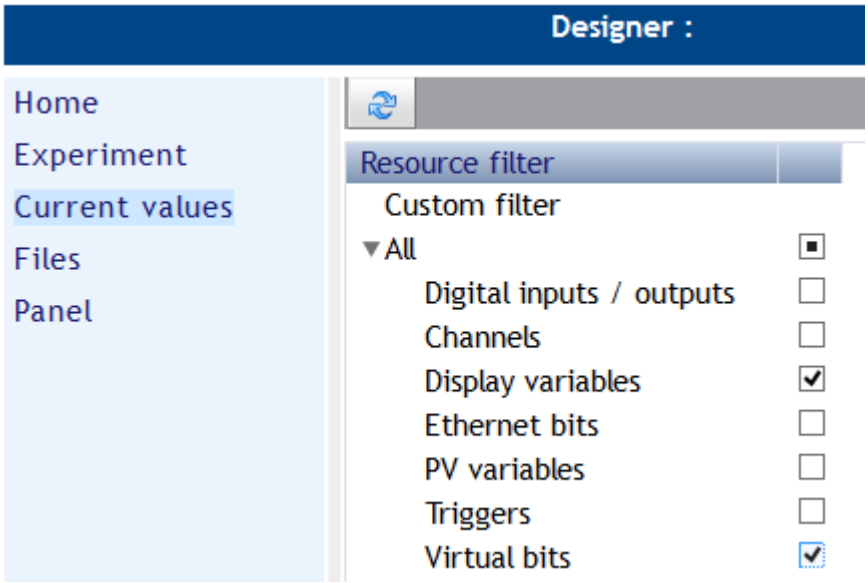
You can also set the value of the Display variable.

The channels' current values are displayed by the PV-variables. 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* 736 or *Numerical input/output* 737.

It is not possible to write values triggers.

7.9.6.1 Adapting the design

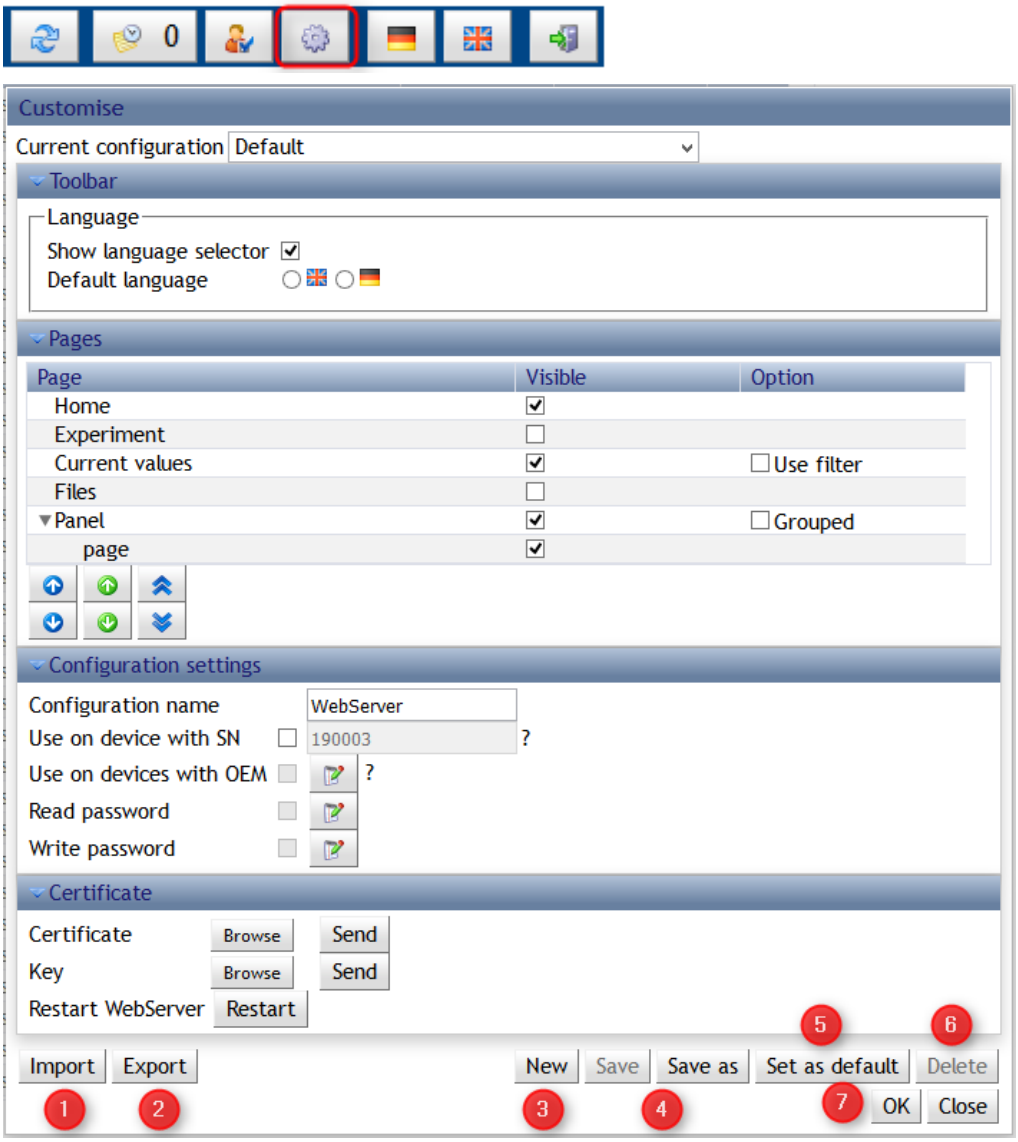
In the Designer mode, you set which resources are visible to the Operator.



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.

To apply the settings, call the settings dialog via the associated button, which is only visible in Designer mode:



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*.

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*.

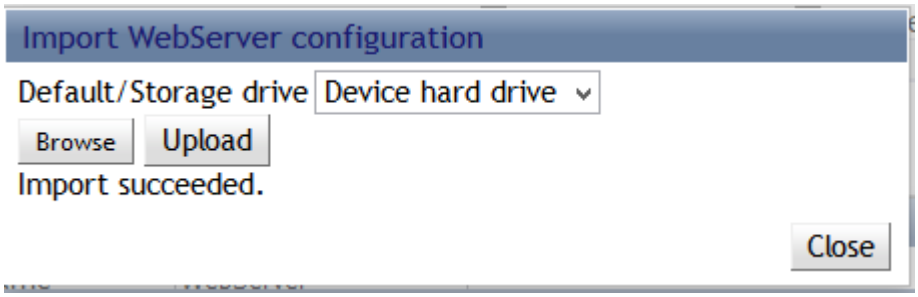
By means of *Save* and *Save as (4)*, the current configuration can be saved in the device's memory.

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.

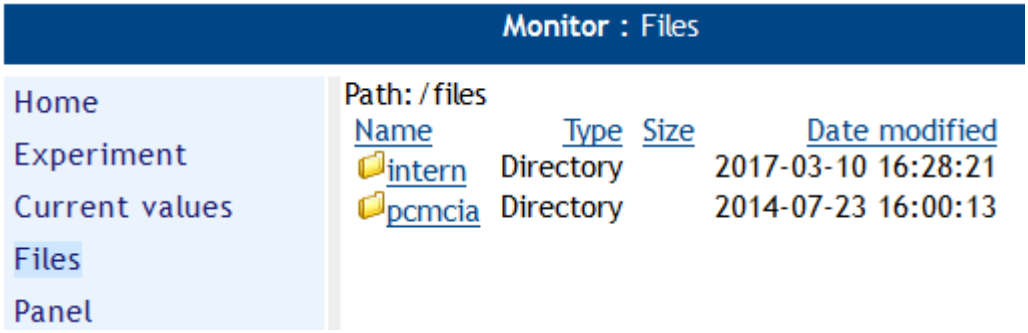
Import (1) also opens a dialog for selecting *WebServer.imcwsc.* Import is performed when you click on the button *"Upload"*.



When you copy the file *WebServer.imcwsc* to the experiment's folder on the PC, the settings are applied upon the next upload.

7.9.7 Files

The item Files provides access to the internal and to the removable data storage.



The single files can be downloaded.

7.9.8 Panel

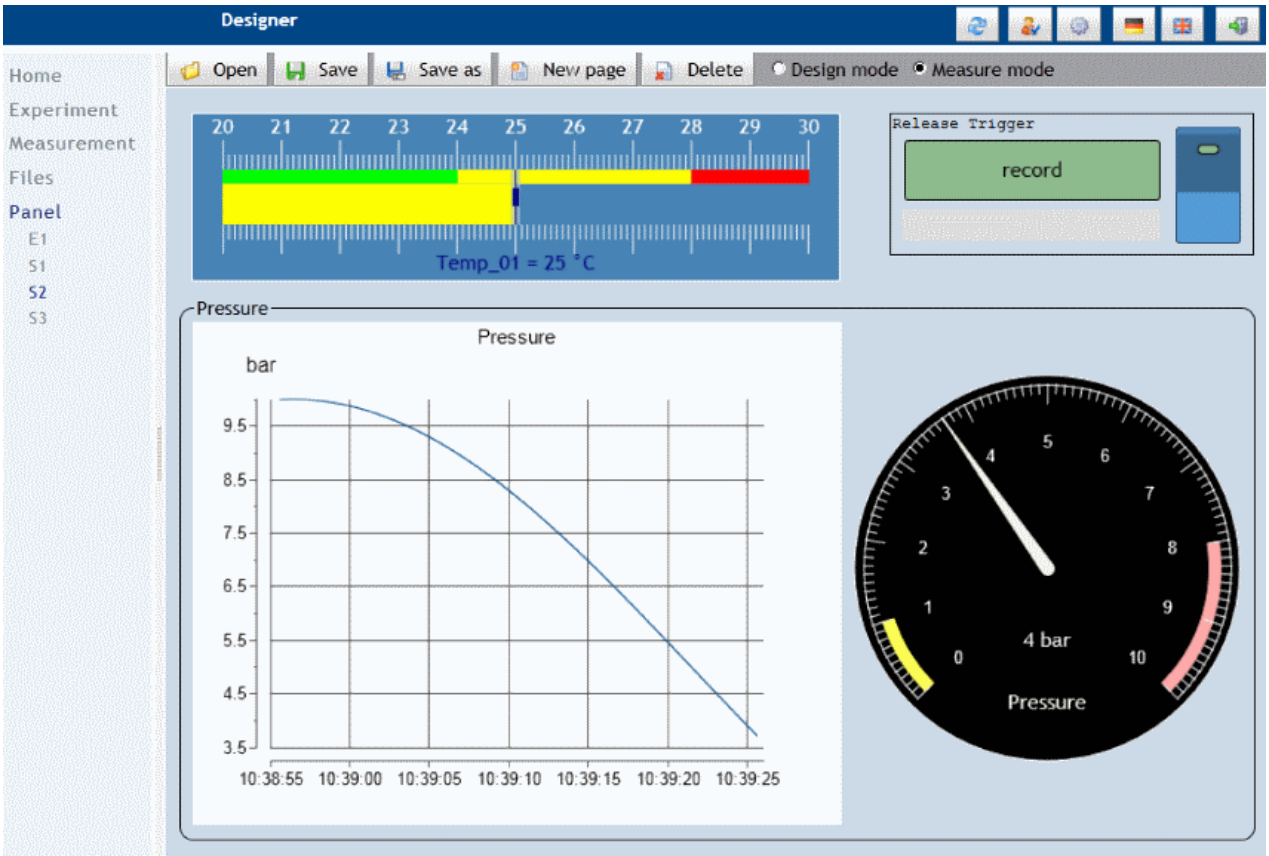
The web server from version 2.3 allows the use of panels. In the panel pane, it is possible to create multiple pages, by means of which the measurement's status can conveniently be displayed.

Important Notes

We recommend stopping measurement while designing the web-pages. This can prevent possible performance bottlenecks, which, especially on smaller devices (CRC/CRFX-400, BUSDAQ), 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* 725, 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".

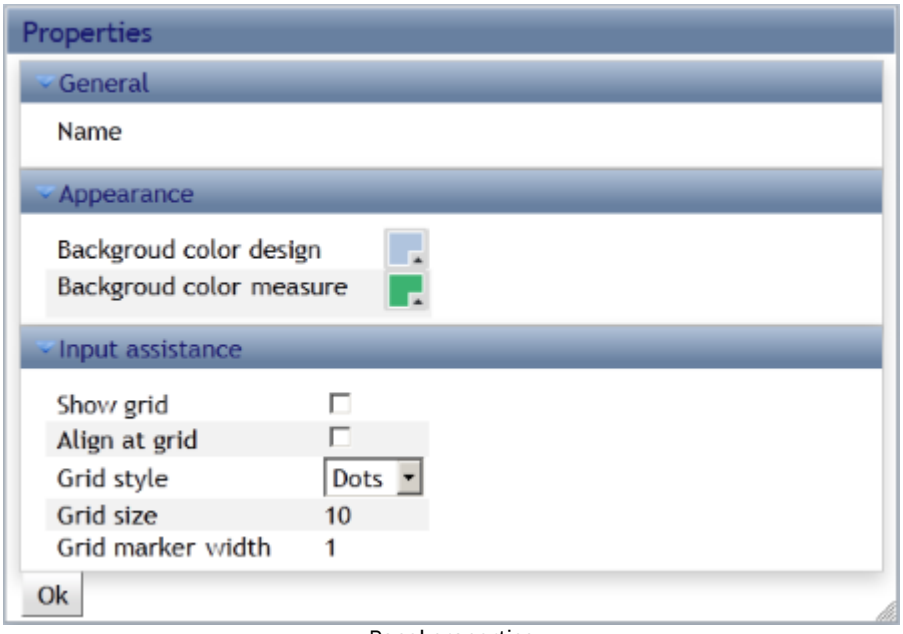
Note

- 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 727.

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.

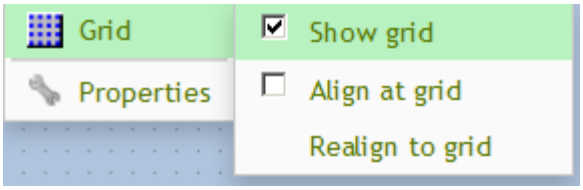
By right-clicking on the Panel page, the Properties become available for editing:



Panel properties

Along with the background color, the grid is also defined here.

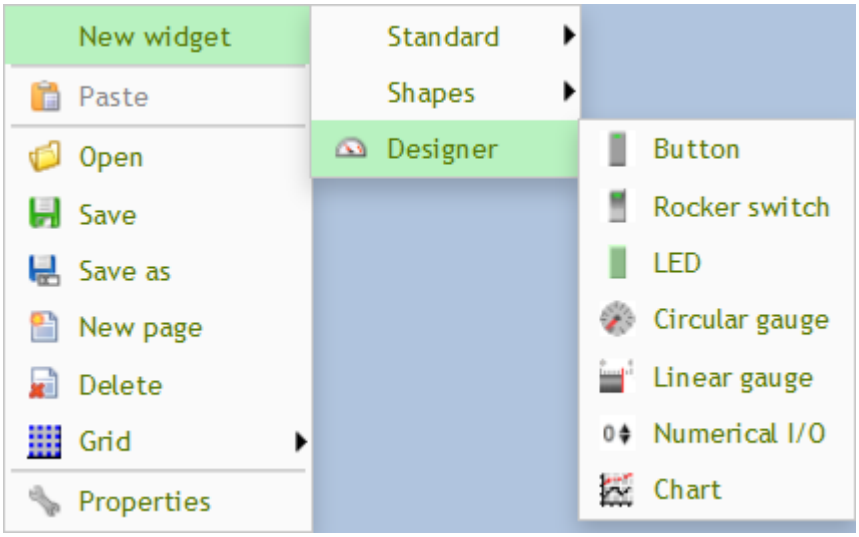
However, the grid can also be used by means of the menu item Grid:

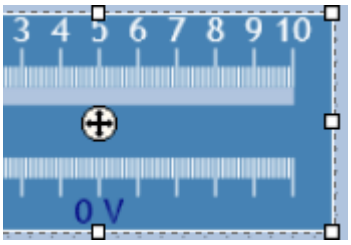


Grid context menu

7.9.8.1 Widgets

Widgets are created by right-clicking the mouse over the Panel.



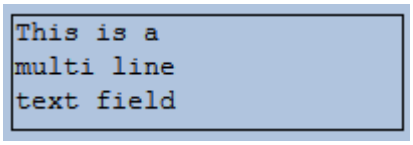


To call the widget's properties, right-click the mouse on the crosshairs in the center.

Note

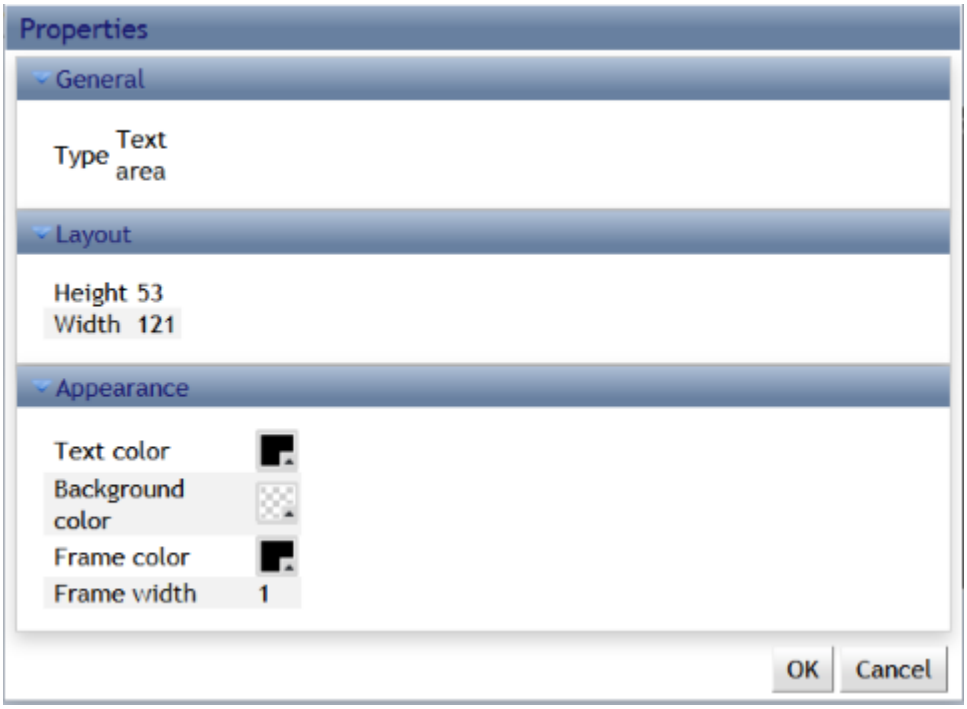
- Note that the Web-Browser only receives single values from the device. Other channel parameters such as the sampling rate, unit or input range are not available. For this reason, these parameters must be entered manually in the widgets' properties.
- Although the Panel closely resembles a program's 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



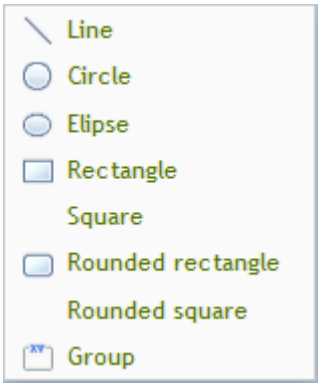
Widget for displaying text

It is possible to set the colors for the font, 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. The font style and size can not be altered.

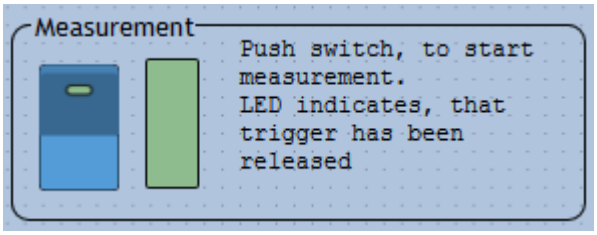


Shapes

As graphics elements, a variety of shapes are available.



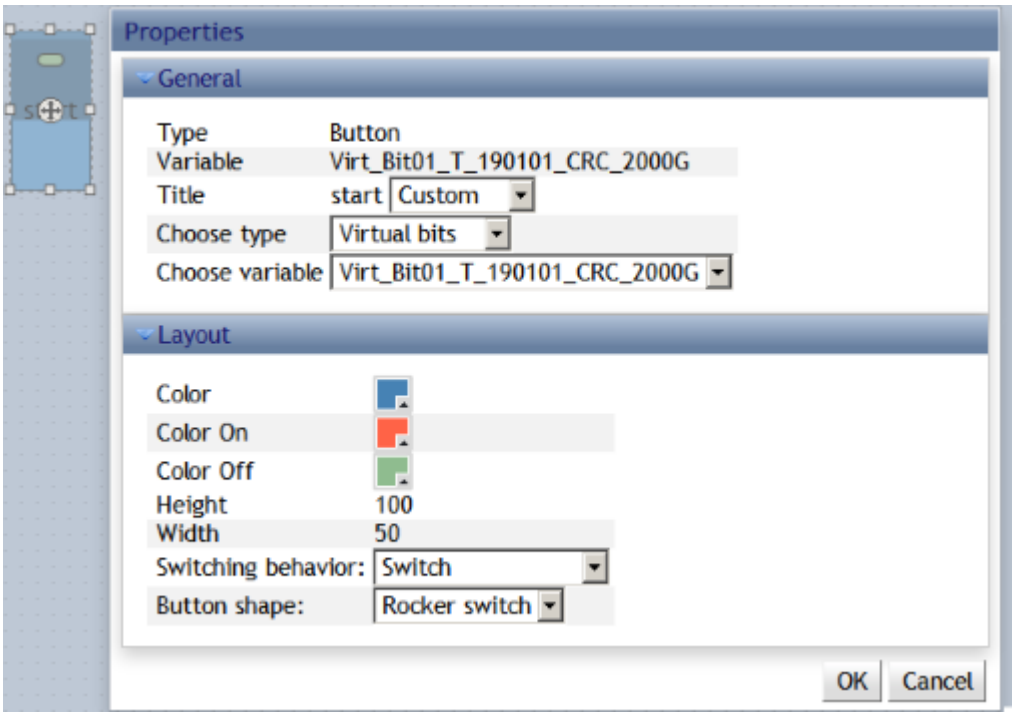
The group box can additionally be supplied with text.



Designer - Press key/toggle switch

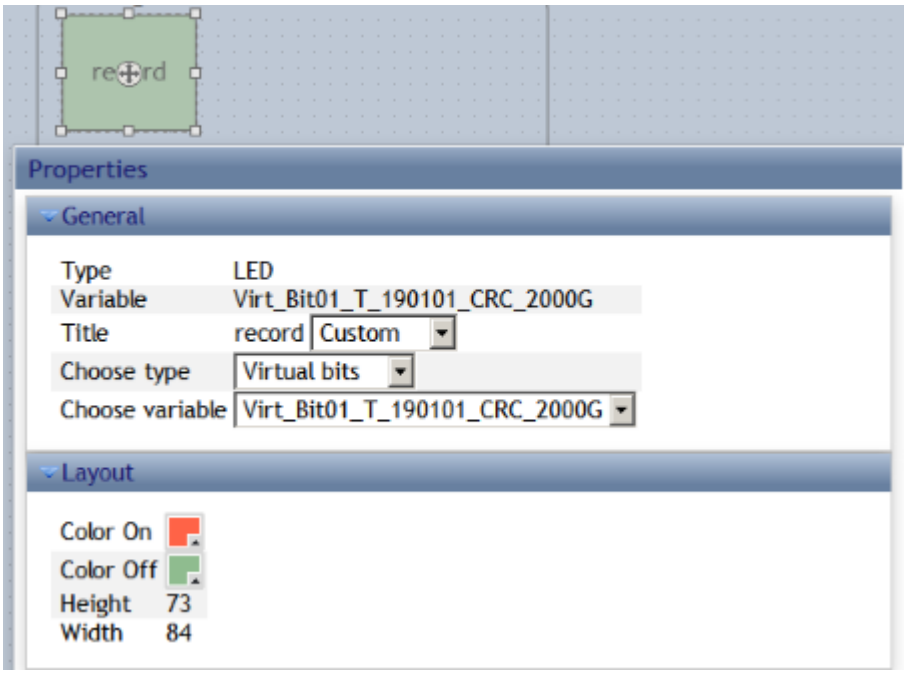
Switches/Pushbuttons can be linked with any digital variable (Virtual_Bits, Etherbits, digital inputs)

Their behavior and appearance can be set in the properties: Switch/Pushbutton and Rocker switch/Button.



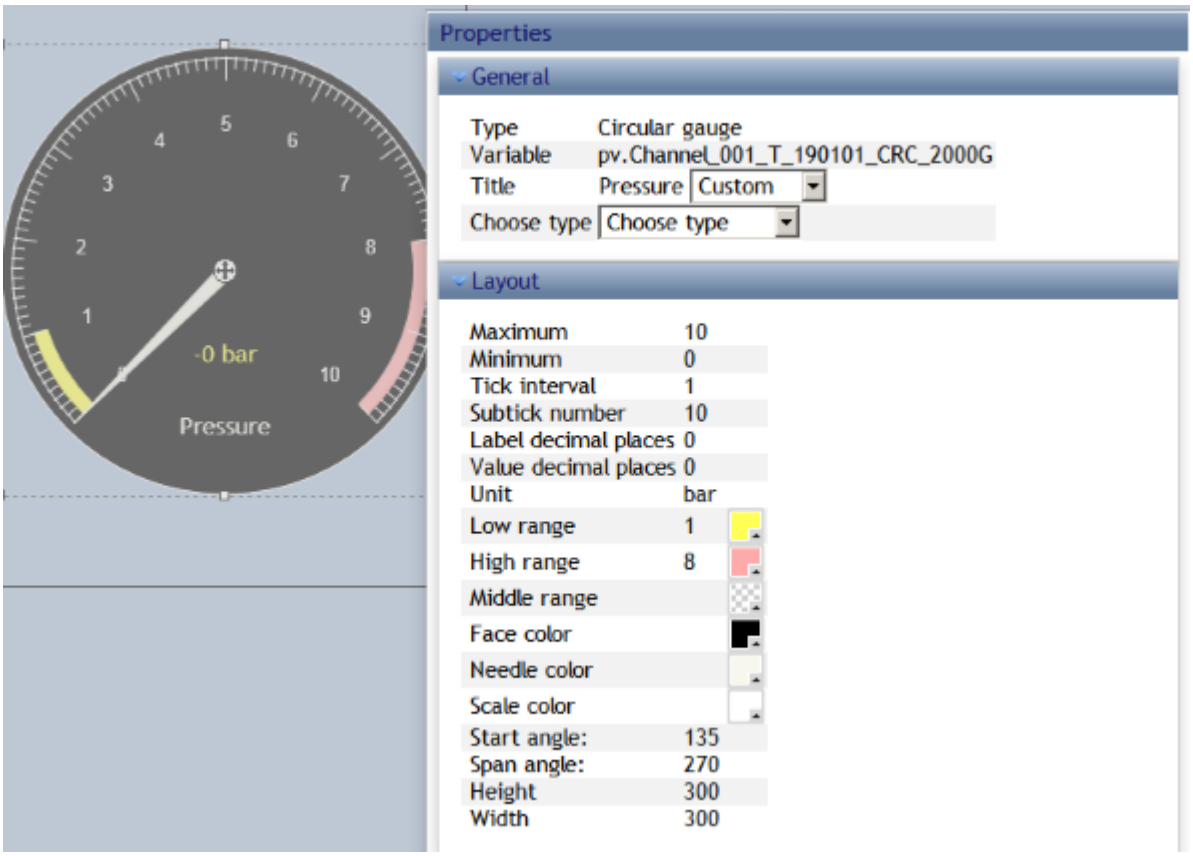
Designer - LED

LEDs can be linked with any digital variables (Virtual_Bits, Etherbits, digital inputs)



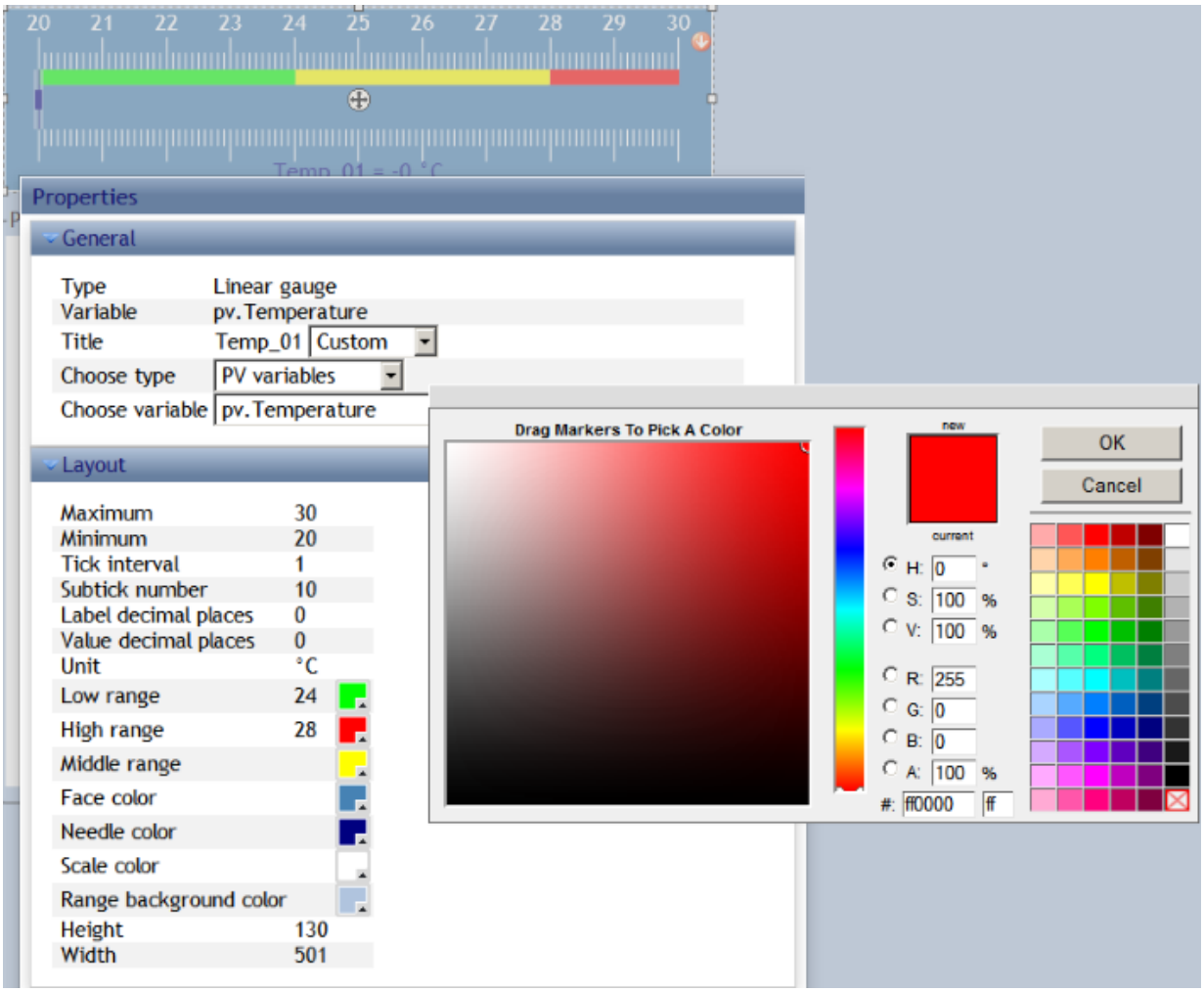
Designer - Pointer instrument

The available settings for the pointer instrument correspond to those of the bar meter.



Designer - Bar meter

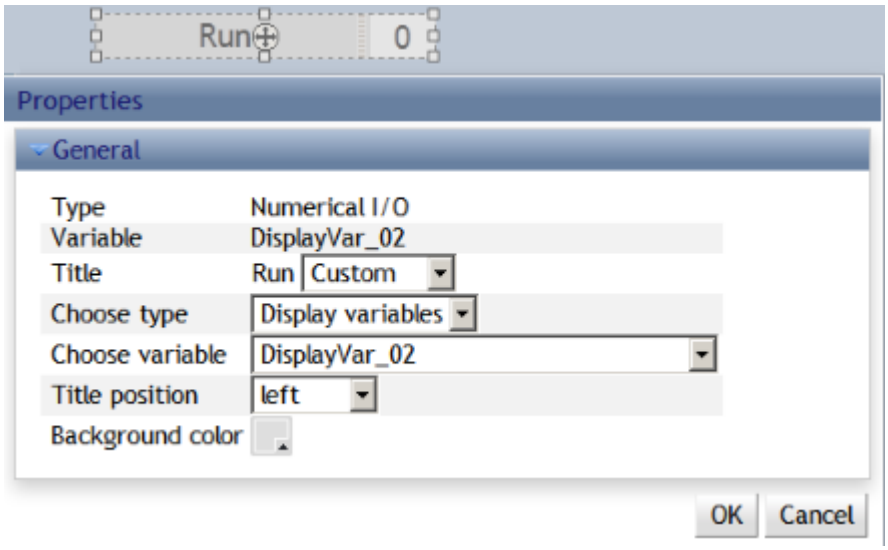
Display of the instantaneous value by a bar meter



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.

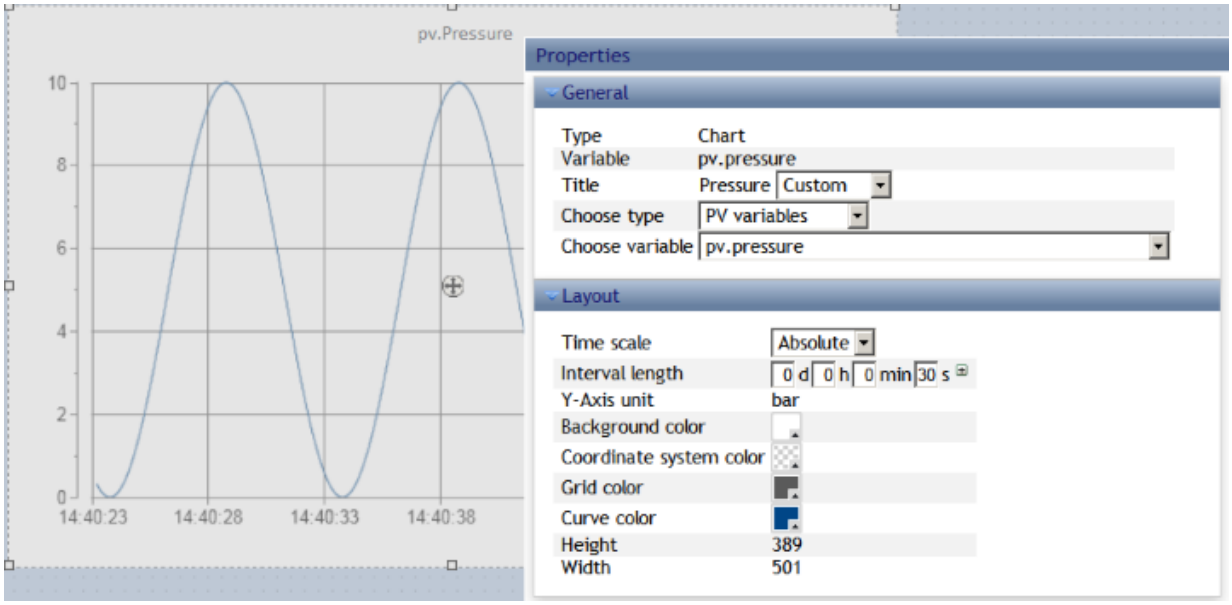
Designer - Numerical input/output

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, Etherbits, digital inputs)



The scaling is set automatically along with the maximum and minimum values of the data measured.

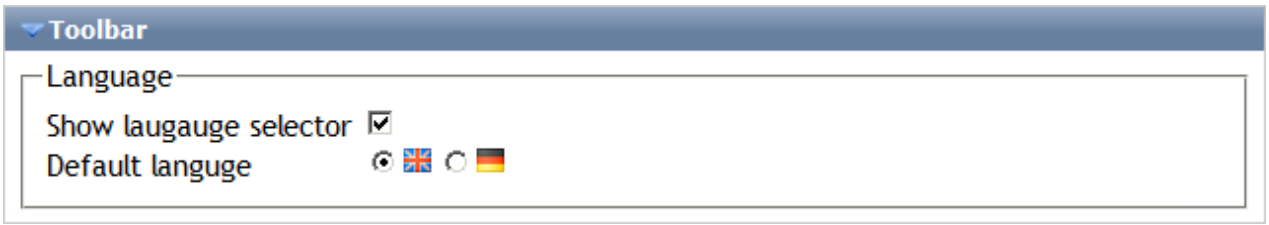
The x-axis can be set to relative or absolute.

7.9.9 Miscellaneous

To apply miscellaneous settings, call the settings dialog via the associated button, which is only visible in *Designer mode*:

Toolbar

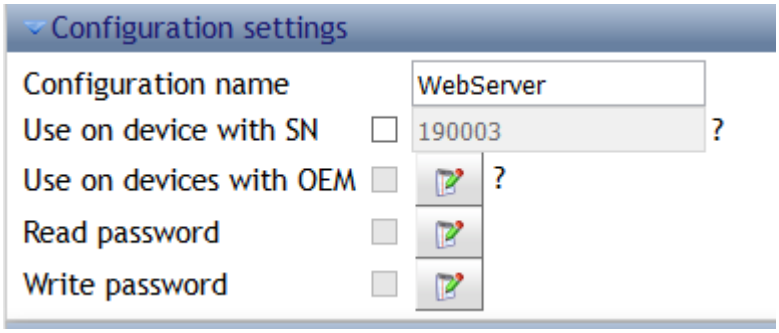
Set the display language in which the WebServer starts up.



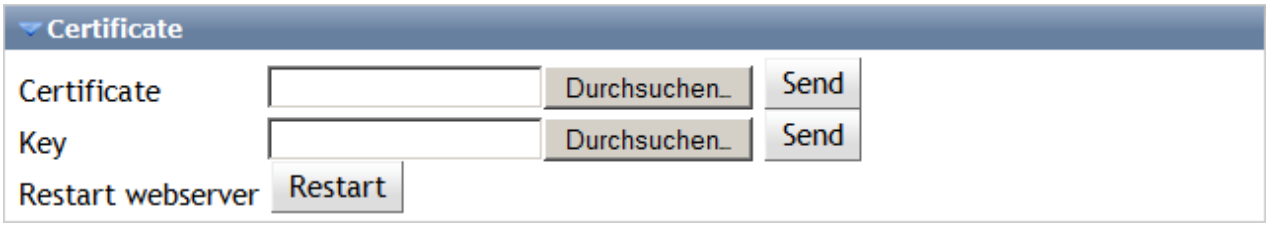
Configuration

Protect the settings by assigning passwords.

The OEM identifier is required for custom devices and applications in order to enable fixed Panel configurations (e.g. imc POLARES, imc WAVE etc.). 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.

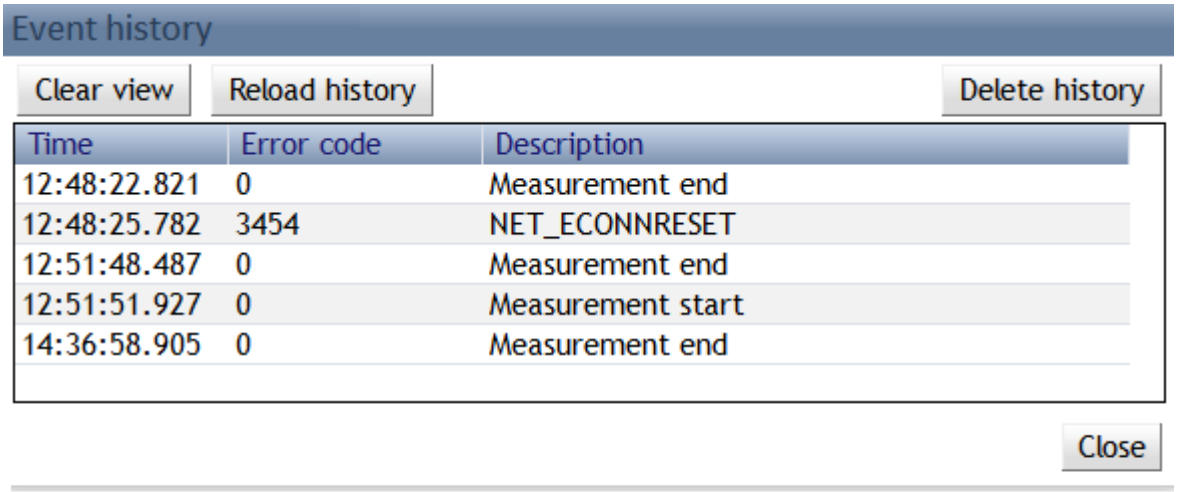
In the uploading process, the Webserver verifies whether the certificate and the key match each other. The validity can only be checked the next time the Webserver is started. if the certificate is not valid, it is rejected and the old one remains in force.

This can only be performed in either *Operator* or *Designer mode*.

7.9.10 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:



7.10 imc Online FAMOS

imc Online FAMOS is an expansion for imc measurement devices, offering a large number of real-time functions for pre-processing of data. The pre-processing is performed by a digital signal processor (DSP) in the device. This means that the functions are carried out quickly, independently of the PC.

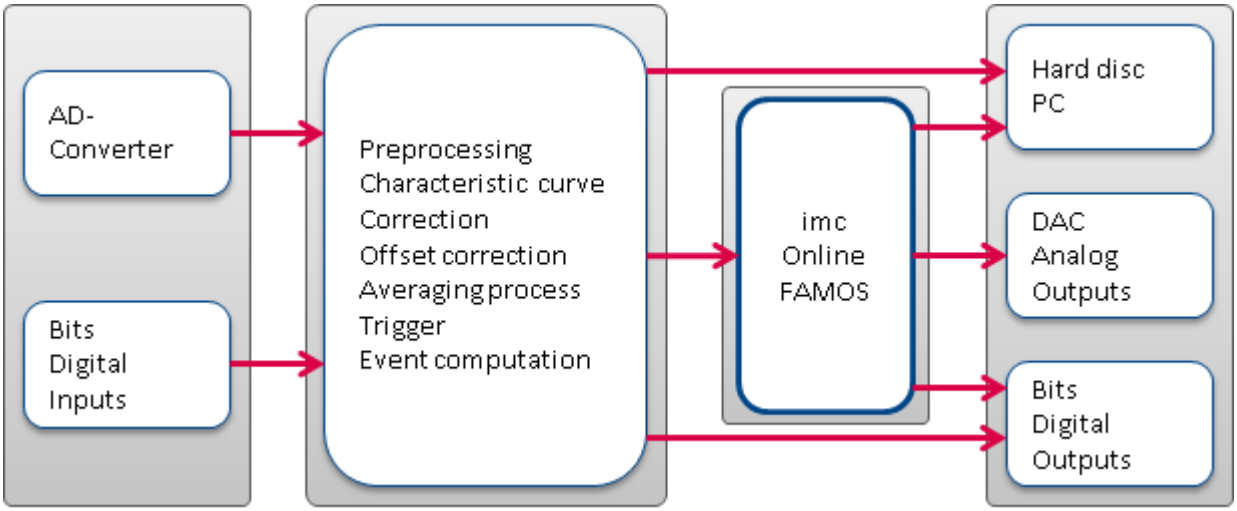
The data volume between the device and the PC can be substantially reduced by the pre-processing.

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.

7.10.1 imc Online FAMOS - Overview

imc Online FAMOS 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`**.

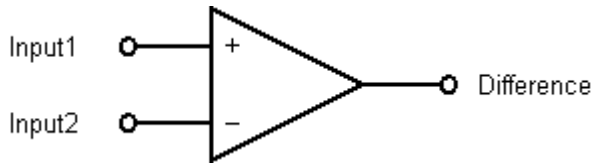
When measurement is started, the device performs the following procedure:

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.

imc Online FAMOS can compute continuously without interruption. Because measurement data are instantly processed and the computed results are immediately forwarded, imc Online FAMOS does not require particularly extensive memory.

Note: LED 6 flashes during measurement

As of imc DEVICES Version 2.9R2 and imc STUDIO 5.0R6, the LED number 6 automatically flashes in a 1-second rhythm during running measurement when imc Online FAMOS is used.

LED 6 does not flash

- if it is used in the source code
- if flashing is deactivated in the Options 773.

7.10.1.1 System Requirements

With the exception of imc BUSLOG, all devices which support imc STUDIO are also able to run imc Online FAMOS.

To use imc Online FAMOS Professional the system has to comply with the following requirements:

- Devices as of group 2 88 and higher

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.

7.10.1.2 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 imc Hotline 13 for more details.

7.10.1.3 What can imc Online FAMOS 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 10kHz, the Synchronous Task 746 can be used. For this purpose, imc Online FAMOS Professional is necessary.
- imc Online FAMOS 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 in which the loop can be processed. If your application requires definite response times, then these can be guaranteed with an Synchronous Task 746 of imc Online FAMOS Professional.
- Fieldbus channels recorded in time stamping mode, can't be proceeded in imc Online FAMOS!
- Unsigned 32-bit integer values (32-bit UINT) are not calculated correctly in imc Online FAMOS.
  - 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 imc Online FAMOS, the bit count must be set to 31 bits following import of the DBC file.

7.10.1.4 imc Online FAMOS Configuration and Device Equipment

The possible configurations for imc Online FAMOS depend on how the device is equipped:

When imc Online FAMOS configurations are copied to other devices, their behavior may well be altered. For instance, an assignment `LED_01 = Greater(Kana_01, 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.

7.10.1.5 Comparison with (PC) imc FAMOS

This section is intended for those of you who are already familiar with imc FAMOS (imc signal processing software, functioning off-line). We explain here some of the main differences between imc Online FAMOS and imc FAMOS.

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 imc Online FAMOS, 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 which reason imc Online FAMOS does not require extensive memory.

For all applications having online functionality requirements, it is not adequate to wait until the end of the measurement.

7.10.1.6 What can imc Online FAMOS do?

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 device µ-Disk, to DACs and to digital output bits.

If, instead of digital processing by imc Online FAMOS, 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 imc Online FAMOS functions, then this is comparable to an analog circuit. Thus, imc Online FAMOS can be viewed as a substitute for a freely configurable analog pre-processing circuit.

7.10.1.7 What happens when imc Online FAMOS Overflows?

General notes on data overflow in software

imc Online FAMOS operates without synchronization to the data capture. This means that imc Online FAMOS works through it program in an uninterrupted loop.

imc Online FAMOS working without overflow

If the data rate is low enough, and there are no new samples available for calculations, then there is nothing for imc Online FAMOS to do.

If the online calculation is so complex that new samples accumulate by the time the calculation is done, then imc Online FAMOS enters the new data into the next calculation iteration. If the average number of new samples remains equal, there is no overload.

imc Online FAMOS 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, imc Online FAMOS 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. For brief measurements, of course, this is tolerable if the RAM-buffer duration 592 is adequate. But if the accumulated data exceeds the RAM buffer depth, the software will announce a data overflow!

Special case: Overflow in the synchronous task

When using the Synchronous Task 746, 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 virtual channel.

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
    Overload_Sync10ms = SyncOverload( 10 ) + Channel_002 * 0
End
```

7.10.1.8 imc Online FAMOS with Control Command

Conventional imc Online FAMOS without control commands mainly provided the ability to define virtual channels and subject them to calculations. An imc Online FAMOS program is executed line-by-line; all instructions entered are carried out in the order they appear in the program.

In imc Online FAMOS with control commands, 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". This makes it possible to follow different threads of instructions within an imc Online FAMOS program. Thus, different procedures can be defined in imc Online FAMOS to depend on values of variables. Also, operations can be directly assigned to various states within the course of a measurement (e.g. start, the end or during the measurement).

An example shall illustrate the difference between imc Online FAMOS with and without control commands:

Example

```
OnTriggerStart( Trigger_48 )    ; to be carried out at start of measurement
    Reset = 1                   ; Reset is a digital output
End

OnTriggerMeasure( Trigger_48 )  ; 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 756
.

7.10.1.8.1 Transforming to imc Online FAMOS with/without Control Commands

By default, imc Online FAMOS starts without control commands. To change to imc Online FAMOS with control commands, select the menu item *Extra > imc Online FAMOS with 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 to imc Online FAMOS is made, these operations are automatically assigned to the control commands `OnAlways` and `OnTriggerMeasure`. Thus, an error-free imc Online FAMOS program can be implemented without any modifications after switching to imc Online FAMOS with control commands.

The switch to imc Online FAMOS with control commands can also be made via the *Options* menu. Select the item *imc Online FAMOS without control commands*. Only such operations are directly transferred which can also be used in imc Online FAMOS without control commands, while all other operations still appear, but commented out. After conversion, then, it may be that only portions of the program can be run, and generally, modification of the source text becomes necessary, often imposing serious limitations on the imc Online FAMOS program. As a rule, it is inadvisable to port a program created with control commands to imc Online FAMOS without control commands.

Trigger note with LED:

Suppose the measurement duration is set differently for different channels, and the LED is activated in response to `On_Trigger_START`, and the LED is set to go off in response to `On_Trigger_END`. Then the LED's flashing duration corresponds to the measurement duration of one of the various channels, but which channel cannot be determined.

7.10.1.8.2 Additional Groups in the Function List

If imc Online FAMOS with control commands is activated, 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 by imc Online FAMOS. 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 which can only be used in imc Online FAMOS with control commands; 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 1440.

7.10.1.9 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 746
- Synchronous Tasks 746 including accessories
- Direct access to the Process Vector 763
- PID-Controller 785

7.10.1.9.1 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. The accuracy is not reduced by that feature. The increased speed is achieved by as many functions as possible using the signal processor'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.

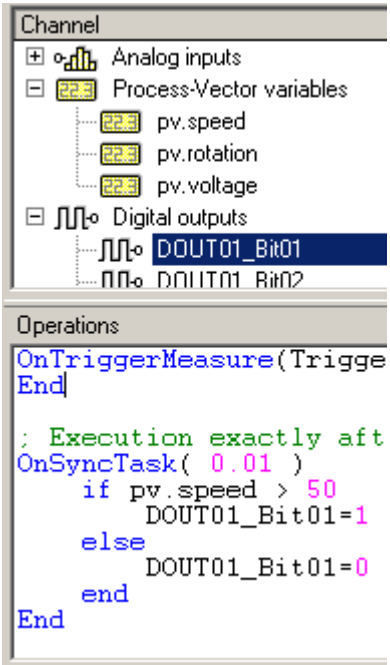
In general all functions are considered. But memory intensive functions will be preferred, because they speed up most.

Examples

- FFT: 1,5 –2,5x
- Basic calculations: 2x
- Digital filters: 3-4x

7.10.1.9.2 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.



- 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`.

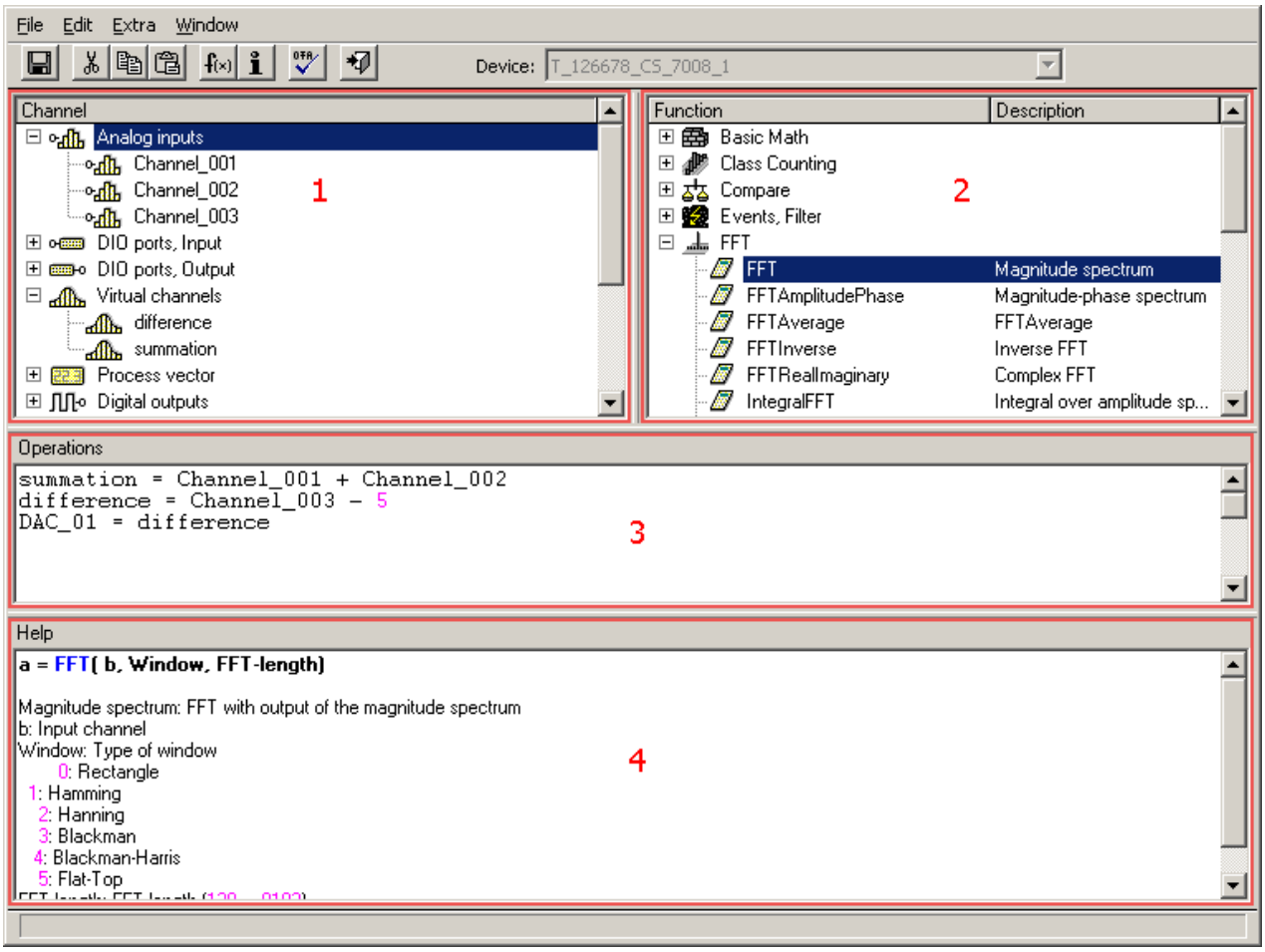
7.10.2 Starting imc Online FAMOS

Select imc Online FAMOS from the Ribbon *Home* (or *Setup-Configuration*) to start imc Online FAMOS.

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.

7.10.3 Main Window Elements



The window is subdivided into four regions:

1. Channel List
2. Function List
3. Editor (Text box) for the calculation operations
4. Help

Region 1: Channel List

Here is the list of all available device variables (e.g. all active input channels)

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.

	Description		Description
	active analog input channels and incremental encoder channels		digital output bits (of all DIO-ports)
	DIO ports, input		analog outputs (DACs)
	DIO ports, output		housing LED's
	calculated virtual channels		virtual bits
	calculated local channels		ethernet bits
	calculated local single value variables		display variables
	process vector variables 763		beeper (miniature speaker)
	digital input bits (of all DIO-ports set for bit-input)		trigger which starts an analog channel

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 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 channel 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. For a detailed description of the various ways of calling the Formula-Assistant, see the chapter Formula-Assistant 775.
- by calling the Functions dialog: Using this dialog, it is possible to set multiple function calls for a function, each with different channel parameters and settings, and move them to the editor. For a detailed discussion, see the chapter Functions dialog 777.

Functions are entered into the editor along with parenthesis.

Region 4: Help

Here you find an explicit description of the functions and of the channels' properties.

To open the help text for a channel or a function, left click the mouse to select the desired item. The editor remains unchanged.

7.10.4 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 formulas using standard mathematical notation to the editor. 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 to 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.

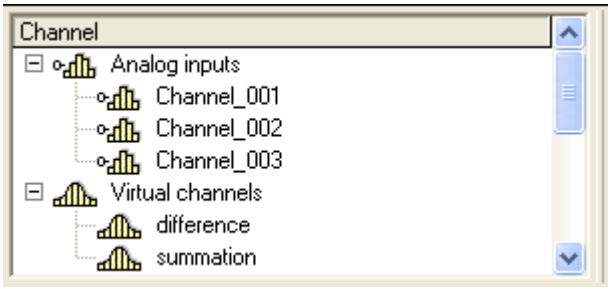
Adding a channel name or function

- To add a channel to the editor, drag it from the left part of the *Channels list* and drop it in 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.

The various formulas must be defined row-wise.

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.

Save

To save a result, click on the symbol in the toolbar. Alternatively, you can also use the F3 key or select *File > Save to Configuration*. However, this is not absolutely necessary, since these entries are also saved with the experiment.

Close imc Online FAMOS now.

The newly defined virtual channels should now appear in the plug-in Setup

7.10.5 Variables and Syntax

7.10.5.1 Virtual Channels and Local Variables

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_01 + Channel_02
CorrectedOffset = Channel_01 + 5
Stretch = 2 * Channel_03
```

You can also use parentheses and minus signs:

```
Complicated = -( 3 + 4 * ( Channel_01 + 1 ) )
```

Function arguments must be written in parentheses:

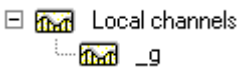
```
SquareRoot = sqrt ( Channel_01 )
```

Formulas which are not permitted:

Incorrect	Explanation
`a = 1`	This is not allowed because 1 is a constant and cannot represent a data stream. The virtual channel a requires a sampling time just as a physical channel does. If you want to set a virtual channel to a fixed value, use a formula similar to the following: `a = Channel_01 * 0 + 1` Attention: If you use *imc Online FAMOS with control command*, the formula `a = 1` is allowed 756. In this case, **a** is a Local variable.
`b = DAC1`	This is not allowed because an analog output (DAC) can only be written to but not read from.

Local Channels and Variables

If you need intermediate results to compute a virtual channel, it is possible to keep them local in the DSP 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 Channel 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 imc Online FAMOS program 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_01 * 3
Virtual1 = _LocalChannel + 1
Virtual2 = _LocalChannel + 2
Virtual3 = _LocalChannel + 3
```

Creating local variables or single values

Local single values 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.

Example

```
_Constant1 = 3 + 4 * sqrt ( 5 )
Virtual1 = Channel_01 * _Constant1
Virtual2 = Channel_02 * _Constant1
Virtual3 = Channel_03 * _Constant1
```

Note for Single Values

Note that single values must be prefixed with an underline, as long as imc Online FAMOS without Control Commands is used!

For single values, the underline is not required, as long as imc Online FAMOS with Control Commands is used. Because single values are not transferred to PC.

7.10.5.2 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 DSP 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_01 * 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".

7.10.5.3 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 Channel List):

- DAC
- Digital outputs
- Beeper
- LED
- Virtual bit and Ethernet bits
- Display variables
- Triggers

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 for the DAC channels is taken into account.

Definition:

`DAC1 = 5` `DAC2 = Channel_01 / 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:

`Virtuel1 = 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_01, -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:

`Virtuel1 = 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_01, -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.

Definition:

`LED1 = 1`
`LED1 = 0`
`LED1 = STRI ( Channel_01, -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_01, -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_01, -5, 5 )`
	Assignment of a number

Triggers

A trigger can be released not only in response to events as defined in the plug-in Setup, but also by imc Online FAMOS.

As soon as an assignment of 1 is made to an armed trigger, this trigger is released.

imc Online FAMOS does not affect the arming of the trigger. A trigger can initiate recording of data. imc Online FAMOS releases the trigger just as if a different event had released the trigger.

Passive triggers and 1-Triggers can not be released.

Definition:

`Trigger_01 = 1`
`Trigger_01 = 0`
`Trigger_01 = STRI ( Channel_01, -5, 5 )`
	By assignment of a 1 (TRUE), the trigger is released.

7.10.5.4 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)
```

7.10.5.5 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_01 + Channel_02 ; Computes the sum
; Computation ends here
```

Multiline comments / Block comments

Several lines can be commented using (* .....*).

Example

```
(* the following lines are commented
Sum1 = Channel_01 + Channel_02 ; computes the sum
Sum2 = Channel_03 + Channel_04 ; computes the sum
Sum3 = Channel_05 + Channel_06 ; computes the sum
this is the end of the commented block *)
```

7.10.5.6 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 becomes necessary to force the input channels into a common sampling rate, this can be done in imc Online FAMOS using various functions. Two good possibilities are `ReSample` and `Mean`.

7.10.6 Variables and Syntax with Control Commands

By default, imc Online FAMOS starts without control commands. The following descriptions apply to imc Online FAMOS with Control Commands.

In order to be able to use the advanced functions and options offered by the control commands, activate them. In the menu, select *Extra > imc Online FAMOS with Control Commands*.

More information is presented in the section: imc Online FAMOS with Control Command 744.

7.10.6.1 Creating a Variable

Using imc Online FAMOS with 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 and can be created in either Integer or Float format.

They are created in the `OnInitAll` block (see: Types of Variables 761).

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) with 24-bit precision

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( Trigger_48, 4)
   int VarInt   = 1
   VarFloat     = 0.0
   int pv.Var1 = 0
   pv.Var2      = 0
   VChanReal    = SingleValueChannel( Trigger_48, 1000 )
End

; run upon start of the measurement
OnTriggerStart(Trigger_48)
   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(Trigger_48)
   ;...
   if Virt_Bit01
      Virt_Bit01  = 0
      VChanReal   = pv.Channel_001
   end
End

; Executed at the end of every measurement
OnTriggerEnd(Trigger_48)
   DisplayVar_01 = pv.Channel_001 - v[1] ; difference end - start
   DisplayVar_02 = pv.Channel_002 - v[2] ; difference end - start
End
```

7.10.6.2 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.

```
IsGreater = a > b
```

Greater-operator. Is the first operand greater than the second operand? IsGreater = 1, if the first operand is greater, otherwise it is 0.

a: 1. Operand

b: 2. Operand

IsGreater: result

Example

```
OnInitAll
    Value = 0
    VrtBit_01 = 0
End
OnTriggerMeasure( Trigger_48 )
    Value = CurrentValue( Channel_01, 0, 0.0 )
    If Value > 5
        VrtBit_01 = 1
    End
End
```

Combinations of different conditions can be implemented using the operators `AND` or `OR`, for instance, `IF VirtBit_01 > 0 AND VirtBit_02 = 0`.

7.10.6.3 Defining Virtual Channels with Conditions

When channel data are processed in *imc Online FAMOS without control commands*, virtual channels are always filled with values (according to the clock rate of the parameter channels, etc.). In *imc Online FAMOS with 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_01* is filled with values, otherwise it remains empty:

```
If VrtBit_01 > 0
    VrtChannel_01 = Channel_01 + 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_01 = Channel_01 + 10
Else
    VrtChannel_01 = Channel_01 + 20
End
```

If a virtual channel in an imc Online FAMOS program 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_01* wouldn't be defined for `VrtBit_01 = 0`.

```
If VrtBit_01 > 0
    VrtChannel_01 = Channel_01 + 10
Else
    VrtChannel_01 = Channel_01 + 20
End
VrtChannel_02 = 2*VirtChannel_01 + Channel_01
```

7.10.6.4 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, imc Online FAMOS 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_01 > 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 imc Online FAMOS	imc Online FAMOS automatically transforms this to

`LED_01 = 1`
`If Channel_01 > 0`
`.....`
`If Channel _01 < 0`
`.....`
	`LED_01 = 1`
	`_cv = CurrentValue(Channel_01, 0, 0.0)`
	`If _cv > 0`
	`.....`
	`If _cv < 0`
	`.....`

If an expression is queried, e.g. `Channel_01 - 1 > 0`, the current value of the expression `Channel_01 – 1` is automatically generated by imc Online FAMOS and placed before the query, and the single value generated is then queried:

Coded in imc Online FAMOS	imc Online FAMOS automatically transforms this to

`LED_01 = 1`
`If Channel_01 + 1 > 0`
`.....`
`If Channel_01 - 1 < 0`
`.....`
	`LED_01 = 1`
	`_cv = CurrentValue(Channel_01 + 1, 0, 0.0)`
	`If _cv > 0`
	`.....`
	`_cv = CurrentValue(Channel_01 - 1, 0, 0.0)`
	`If _cv < 0`
	`.....`

If virtual channels are queried, e.g. `If VrtChan_01 > 0`, then the current value in the channel is determined and noted after each definition (in imc Online FAMOS with 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 imc Online FAMOS	imc Online FAMOS automatically transforms this to

`If VrtBit_01 > 0`
`  VrtChannel_01= Ch01+3`
`Else`
`  VrtChannel_01= Ch01+5`
`End`
`LED_01 = 1`
`If VrtChannel_01 > 0`
`.....`
`If VrtChannel_01 < 0`
`.....`
	`If VrtBit_01 > 0`
	`  VrtChannel_01 = Channel_01 + 3`
	`  _cv = CurrentValue(VrtChannel_01, 0, 0.0)`
	`Else`
	`  VrtChannel_01 = Channel_01 + 5`
	`  _cv = CurrentValue(VrtChannel_01, 0, 0.0)`
	`End`
	`LED_01 = 1`
	`If _cv > 0`
	`.....`
	`If _cv < 0`
	`.....`

Note on CurrentValue-funktion

The `CurrentValue`-functions automatically generated by imc Online FAMOS 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 imc Online FAMOS program; it isn't added automatically.

7.10.6.5 Example - Simplified Motor Simulation

The use of various control commands is illustrated 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 ( Trigger_48 )
    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 ( Trigger_48 )
    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 (Trigger_48 )
    Voltage = channel_01 * 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
```

7.10.7 Types of Variables

The following is a list of the usual variable types which you can define in imc Online FAMOS. 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 (Trigger_48, 10)`	Array = VectorStatic (Trigger#, Size)	OnInitAll
Data array dynamic	`vsChan = SingleValueChannel (Trigger_48, 1000)`	Array = SingleValueChannel (Trigger#, DataRate) various additional functions	OnInitAll
Vector series dynamic	`vChan = VectorChannel (Trigger_48, 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 variables (single value;)

Description	Example	Syntax	Definition
Single value as Float	`_Lokal_Var1 = 4` `_Lokal_Var2 = 0xA5 ; hex`	_Vaiable = 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`
		VaiableA = 0
		float VaiableB = 0
			OnInitAll

pv-variables (single value;)

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.VaiableA = 0
		float pv.VaiableB = 0
			OnInitAll

Local texts ( )

Description	Example	Syntax	Definition
Text	`Text = "Hello"`	various functions	

7.10.7.1 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.

7.10.7.2 Process Vector Variables

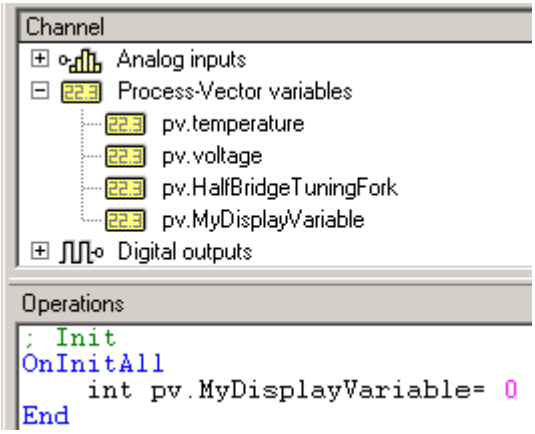
The *Process Vector* is a collection of single-value variables (process vector variables or pv-variables). For *all measurement channels* of the device, these single-value variables are already pre-defined. Updating of the device process vector variables occurs immediately after preparing (Download) the measurement device.

The read-access to the variables is possible with imc Online FAMOS. However, creating user-defined pv-variables and writing to these is only possible with imc Online FAMOS Professional 746. Additionally, in order to create pv-variables 756, imc Online FAMOS with Control Commands 744 must be run.

Together, these single-values (device process vector variables and user-defined process vector variables) can be arranged as a logical structure which mirrors the process under measurement.

Properties of a process vector variable:

- Current measured value directly from the digitizer (data acquisition channels: e.g. analog channels)
- Trigger independence
- User-extendable
- Can be accessed from the CAN- and LIN-bus
- GPS information 767 (with GPS interface)
- State of the external power: The pv-variable "*pv.State.ExternalPower*" 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")
- Deviation of system time "*pv.State.SyncTimeDeviviation*" (only with NTP):
  - Current time deviation of devices from the NTP-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.
- Creating a user-defined pv variable:



Creating a user-defined pv variable

Once the system is prepared (Download), the current measurement values are entered directly to the Process Vector. This takes place independent of the trigger, in other words, prior to its release and also beyond the measurement's end. imc Online FAMOS also works independent of the trigger, and by means of the Process Vector can now access all current measurement values. These can then be used, in conjunction with the *Synchronous Tasks*, for monitoring and control purposes.

Deactivation of process vector variables for data acquisition channels

By default, process vector 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 is only available if the process vector variables are deactivated. Otherwise you receive messages indicating that the aggregate sampling rate has been exceeded, although tis is not true mathematically.

How to deactivate the Process vector variables for data acquisition channels is described in the manual for the plug-in: Setup - Measurement options 157.

Support of process vector variables of the type IEEE-Float

Process vector variables of the type IEEE-Float (e.g. process vector variables for CAN-channels of the type Float) can also be used in imc Online FAMOS. Corresponding CAN-channels then appear in the Channel table, as well as the associated process vector variables.

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 also is applied to the pre-defined pv-variables.

Example

Device name "imcDev__05123456"

pv.Channel_01_imcDev__05123456" for pv-variable of a data capture channel "Channel_01"

"pv.GPS.longitude_imcDev__05123456" pre-defined pv-variable for the GPS position:

The variable type of pv-variables can be controlled by imc Online FAMOS

pv-variables can be generated at various points. In most cases, the type (Int, Float) can only be set in imc Online FAMOS when it is created. Using imc Online FAMOS,you can modify the typeof pv-variables (e.g. belonging to the CAN-Assistant), which are not themselves created in imc Online FAMOS.

- int: a pure numerical value (disregarding factor and offset); with 32-bit precision
- float: a scaled numerical value (factor and offset taken into account); with 24-bit precision

Example

```
OnInitAll
    Int pv.CAN_001
    ; pure numerical value, 32-bit resolution
    ; Factor and offset disregarded
    ; If required, add the factor and offset to
    ; the imc Online FAMOS source code.

    Float pv.CAN_002
    ; Scaled numerical value, Float-resolution
    ; Factor and offset already taken into
End
```

7.10.7.2.1 Saving/Restoring pv-Variables

When a measurement is concluded or the device is deactivated, the current values of process vector 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 process vector variable is restored with the same name depends on the restoration strategy used.
- A fully functional UPS is required.

Declaration

To save and restore process vector 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 process vector 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 process vector 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.

7.10.7.2.2 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 process vector variables (PVV) of imc CRONOS*flex* modules are updated at a maximum rate of 5 kHz. The process vector variables are thus subject to the following constraints:

Process vector 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

7.10.7.2.3 GPS

At the nine-pin GPS socket it is possible to connect a GPS-receiver of the type Garmin GPS18LVC, GPS18-5Hz etc.. 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 imc Online FAMOS.

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)
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)`

GPS information	Description
pv.GPS.vdop	Dilution of precision for vertical see e.g. www.iota-es.de/federspiel/gps_artikel.html (German)

For internal use only:

- pv.GPS.counter
- pv.GPS.test
- pv.GPS.time.rel
- pv.GPS.time.usec

Notes

pv.GPS.latitude and pv.GPS.longitude are scaled as integer 32 with 1E-7. They must be proceeded as integer channels, otherwise precession will be lost. If the virtual channel is created by a addition with a channel, the result must be multiplied by 10^{-7}:

latitude = Channel_001*0+pv.GPS.latitude *1E-7

7.10.7.2.4 pv-Variables as Monitor Channels

In the plug-in Panel, it is possible to link Widgets with the process vector variables. Since these are already updated following the preparation (Download), in other words before release of a trigger, this technique is a simple alternative to monitor channels.

7.10.8 Menu

File - Menu

	Menu item	Hot key	Description
	Import Text File	(Shift + F2)	Loads a text file.
	Export as Text File	(Shift + F3)	The contents of the editor are saved in a text file. This does not replace the Save to Configuration command!
	Save to Configuration	(F3)	This saves the entries to the editor in the experiment.
	New	-	This clears the contents of the editor and deletes the virtual channels from the Channel List.
	End	-	Ends imc Online FAMOS.

Edit - Menu

	Menu item	Hot key	Description
	Undo	(Ctrl + Z)	This command can be used to reverse any editing done with the Copy, Paste, Cut or Delete functions.
	Find and Replace	-	Finds and replaces texts in the imc Online FAMOS editor.
	Copy	(Ctrl + C)	Copies the selected area in the editor to the clipboard.
	Paste	(Ctrl + V)	This inserts the contents of the clipboard to the selected region in the editor.
	Cut	(Ctrl + X)	This cuts the selected area from the editor and copies it to the Clipboard.
	Delete	(Ctrl + Del)	This deletes the selected area from the editor.
	Properties 770	(F5)	This is used to redefine any of the properties of a virtual channel.
	Formula Assistant	(Shift + F1)	This starts the Formula Assistant. The Formula Assistant helps you to select and parameterize a mathematical function (see Formula Assistant 775).
	Function Dialog	(ALT + F1)	This starts the Function Dialog, which is an aid in setting the parameters of functions having multiple channel parameters (see Function Dialog 777).
	Show Help	(Ctrl + F1)	This calls the help function for the selected object in the editor or in the Function List.
	Start External Editor 772	(F4)	An external editor is used for editing entries. What editor is used is specified in the menu item *Extra > Options*.
	Perform syntax-check	(F6)	A syntax-check of the expression entered in the editor is carried out. If an error is located, it is visually marked. Otherwise, the message "No error found" appears in the status bar.

Extras - Menu

	Menu item	Description
	Options	Editing of an imc Online FAMOS-program is also possible by means of an external editor. (See Using an external program editor 772) It is possible to deactivate automatic LED-flashing. (See LED-flashing during measurement 773)
	Directories	This calls a dialog for specifying the default working directory. See Defining default directories / paths 772.
	imc Online FAMOS with control commands	Activate control commands 744.
	imc Online FAMOS without control commands	Deactivate control commands 744.

Window - Menu

This menu provides you with various options for configuring the main window:

	Menu item	Description
	Standard	This resets the window to its default configuration. The Function List, editor and Help boxes and the Channel List are all shown.
	Channels	This enlarges the Channel List to provide you with an overview of the available channels. The Help box is also displayed.
	Functions	This enlarges the Function List to provide you with an overview of the available functions. The Help box is also displayed.
	Entries	This enlarges the editor to provide you with an overview of the operations defined. The Help box is also displayed.
	Help	This enlarges the Help box. Use this function to read larger help files.

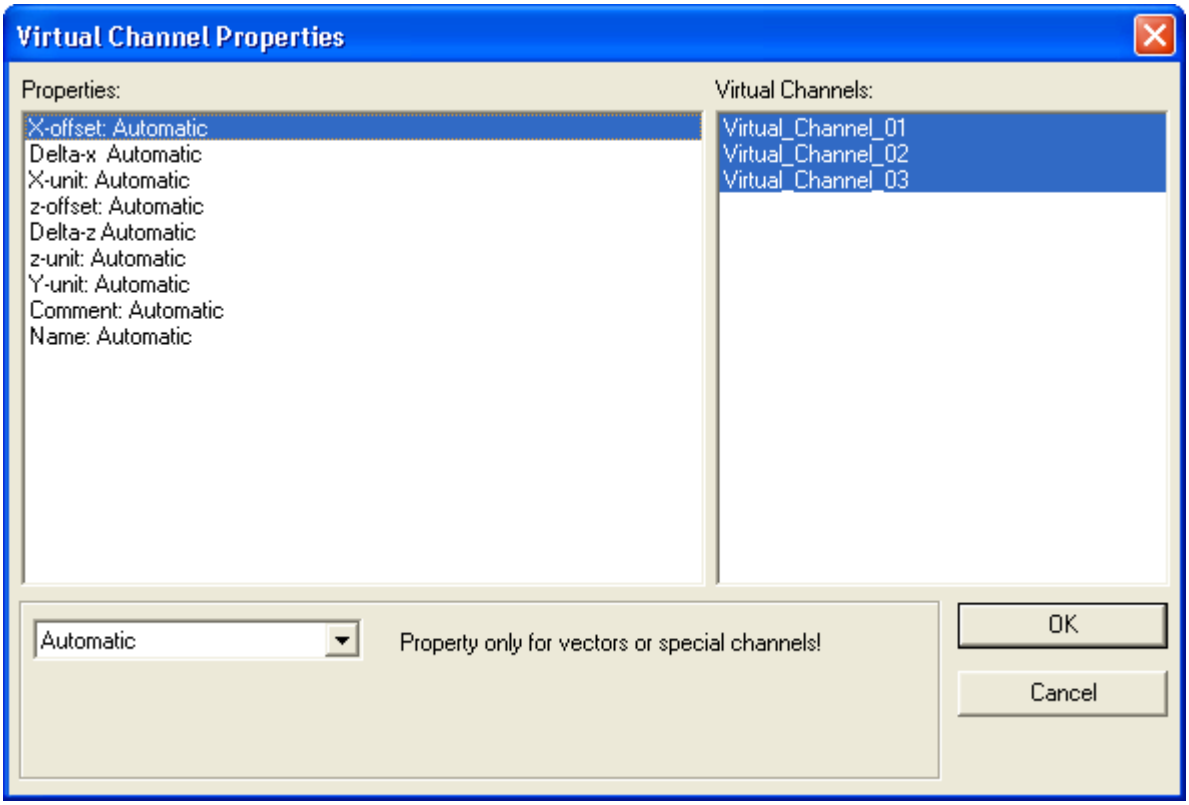
7.10.8.1 Properties of Virtual Channels

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:

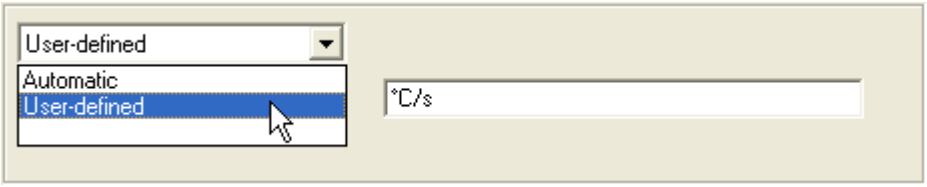
- by means of the menu item *Edit* > *Properties* or
- by clicking on the symbol in the toolbar.



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 by imc Online FAMOS. This property can only be set for vectors.
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 by imc Online FAMOS. 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	The distance between two sample values in the z-direction can be set to a fixed value for purposes of display in the curve window. Otherwise, the value is automatically determined by imc Online FAMOS. 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	You can also assign an alias name to a virtual channel, in addition to its system name. This option is only useful when you wish to keep the system name short while specifying the full name here. Please note that all system names have to be unique! The same restrictions on normal input channel names also apply here.

7.10.8.2 Defining Default Directories / Paths

The dialog for setting the default directories is called via imc Online FAMOS menu item *Extra > Directories* 769.

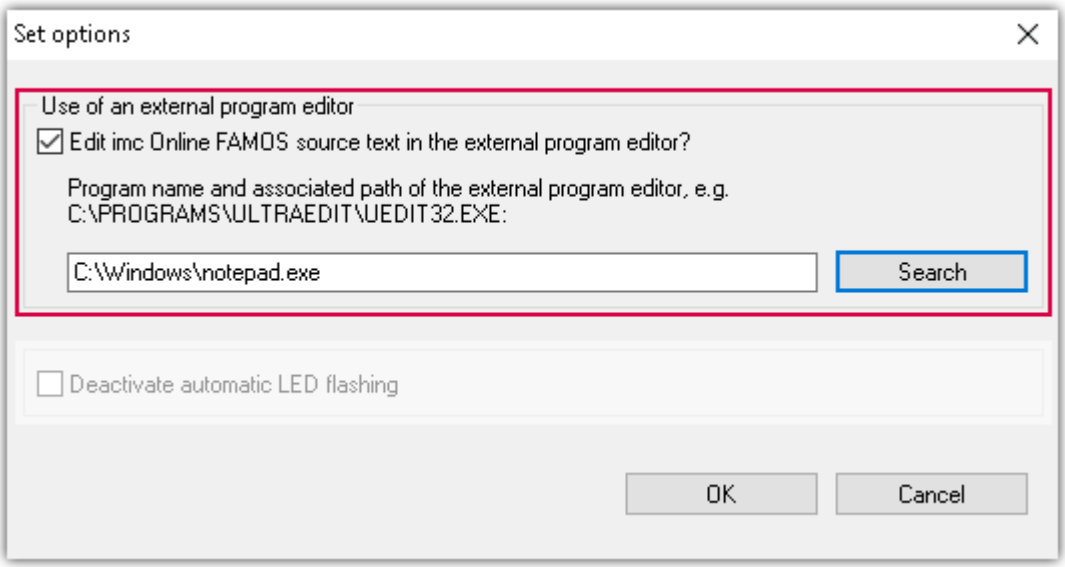
Note

- In imc STUDIO, the default folder can not be changed.
- Supplemental files such as characteristic curves must be imported to the experiment.
  See the description: *Setup > Ribbon > Configuration > Supplemental Files* 105.

7.10.8.3 Using an External Program Editor

It's possible to enter the imc Online FAMOS source text using an external program editor (e.g. UltraEdit, ...).

- 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 the temp file "DevTemp.ofa". This file is deleted as soon as the external editor is closed.

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.

7.10.8.4 LED-flashing during measurement

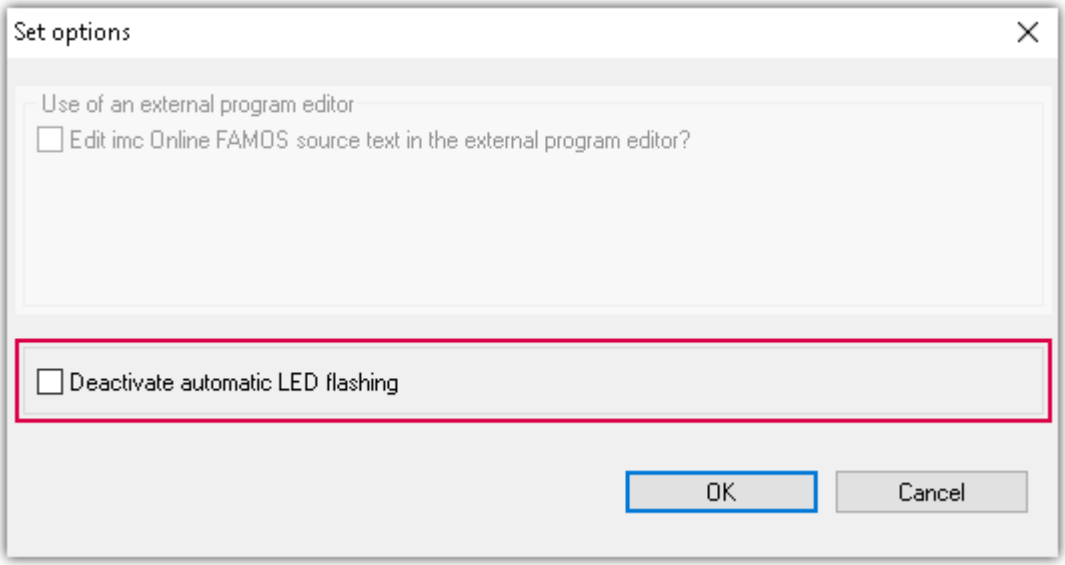
During a running measurement, LED 6 flashes at a 1-second rhythm when imc Online FAMOS is being used. This provides a simple visual indication of whether the measurement is running with the calculation.

LED 6 does not flash,

- if it is used in the source code
- if flashing is deactivated in the Options.

Deactivating flashing in the Options

- Open the imc Online FAMOS options: Menu *Extra* > *Options*
- Activate the option: *Deactivate automatic LED flashing*



7.10.9 Context Menu

Channel List

Right-clicking in the Channel List calls the following context menu:

	Menu item	Description
	Tree view	The channels are displayed under headings corresponding to the various channel types.
	List view	A simple list of all channels, ordered by their types, is displayed. There are no explicit headings indicating the channel types. But each channel's type can be recognized by the symbol beside the channel's entry in the list.
	Collapse tree	Collapses the entries for all channels on the level of the channel types. Only effective in the directory tree view.
	Expand tree	Expands the channel type entry to show all associated channel entries.

Function List

Right-clicking in the Function List calls the following context menu:

	Menu item	Description
	Tree view	All functions are listed alphabetically with regard to function type grouping.
	List view	All functions are listed alphabetically without regard to function type grouping.
	Collapse tree	Collapses the entries for all channels on the level of the channel types. Only effective in the directory tree view.
	Expand tree	Expands the channel type entry to show all associated channel entries.
	Sort by function name	Sorts according to the function name appearing in the list view.
	Sort by description	Sorts according to the description appearing in the list view.
	Formula Assistant	The Formula Assistant 775 is called for the functions selected in the Function List.

Editor

Right-clicking in the editor calls the following context menu:

	Menu item	Description
	Undo, Copy, Paste, Cut, Delete	See the section on the Edit 769 menu.
	Colors	The colors dialog lets you set the colors for the text and for the background in the editor. It is also possible to set separate colors by which to denote various syntax elements.

Help

Right-clicking in the Help box the following context menu:

	Menu item	Description
	Copy	See the section on the Edit 769 menu.
	Colors	The colors dialog lets you set the colors for the text and for the background in the editor. It is also possible to set separate colors by which to denote various syntax elements.

7.10.10 Formula Assistant

Formula-Assistant helps you supplying parameters for a mathematical function.

Starting the Formula Assistant

The Formula-Assistant creates a function at the cursor's position in the editor. Therefore, set the cursor's position accordingly beforehand.

Any one of the following methods can be used to start the Formula Assistant:

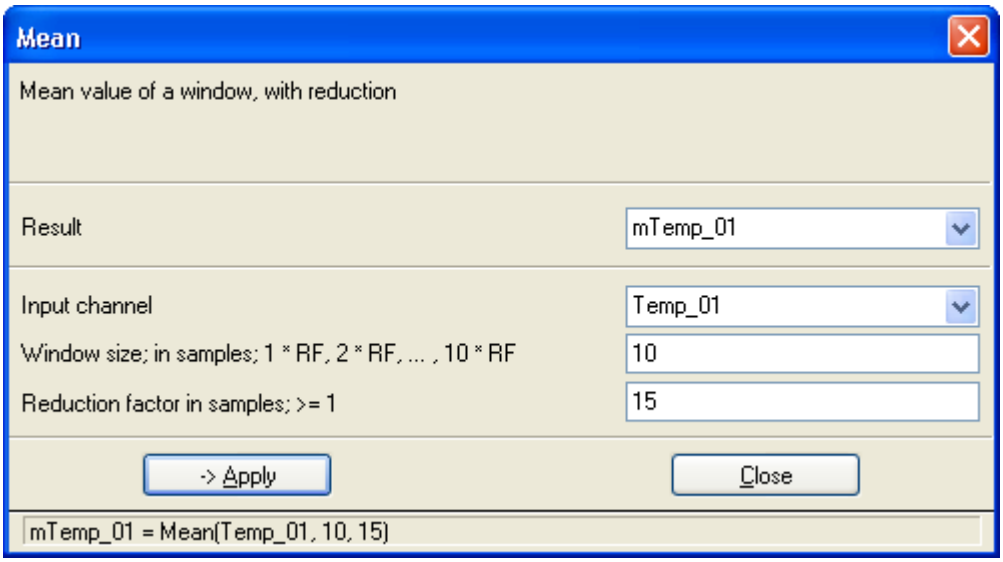
1. Click on its icon in the tool bar: f(x)
2. Choose *Formula Assistant...* from the *Edit* menu.
3. *SHIFT + F1*.
4. Use Drag&Drop to drop a channel listing from the Channel List onto a function's listing in the Function List.
5. Use Drag&Drop to drop a function listing from the Function List onto a channel's listing in the Channel List.
6. Select *Formula-Assistant...* in the context menu in the Function List.

For 1, 2 or 3, the editor has priority: The Formula Assistant is opened for the function selected in the editor.

Otherwise, the Formula-Assistant starts for the function selected in the Function List.

7.10.10.1 Using the Formula Assistant

A dialog comes up for making settings for the math function.



Formula Assistant

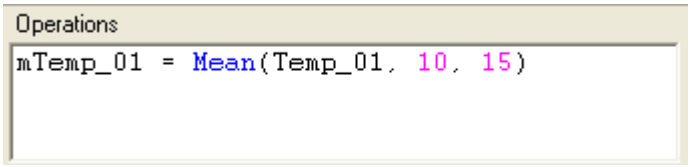
What parameters are needed depends on the math function selected. For instance, for the Rainflow-function, there are 10 parameters to set. Procedures, in contrast to functions, don't have result values.

For the result, a variable name can be entered manually or an output channel can be selected from the pop-down list. For channel parameters, the variable name can also be entered manually or an input channel can be selected from the pop-down list; some parameters are not channels and their pop-down lists offer other selections.

The status bar displays the resulting formula:

Status bar

Click on *Apply* to transfer the finished formula into the editor.



Editor

Viewing and changing a formula

- To change the parameters of a function in the editor,
- place the cursor on the function and press *Shift* + *F1*.

This calls up the Formula Assistant with the function and displays the current selections.

- When the newly formulated expression is transferred to the editor, the current selection in the *Operations* box is replaced by the new formula text.

Therefore, only change the current selection in the editor if you wish to insert the formula at a different location in the editor!

Writing multiple formulas

It isn't necessary to close the Formula Assistant in order to enter multiple calculation operations.

When you place the cursor at the end of the text in the editor, a line-break is automatically added when the formula is transferred from the Assistant.

Creating nested expressions

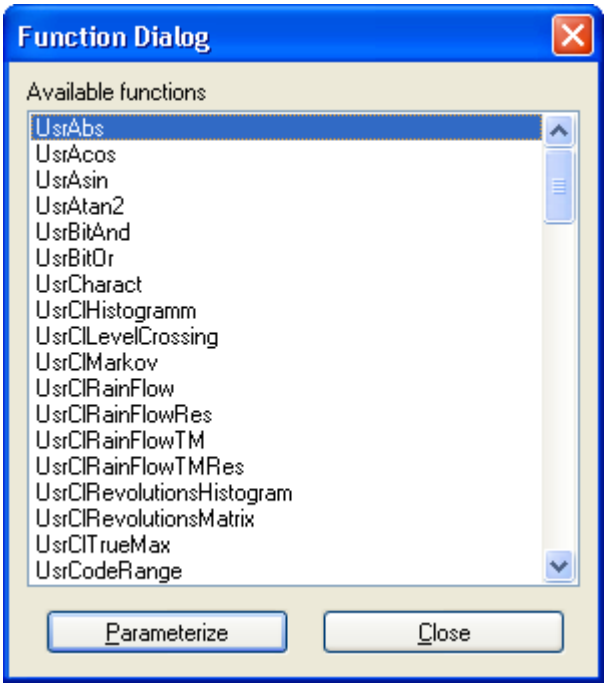
If you leave the input box for the *Result* empty in the Assistant, the formula doesn't take the form of an equation, but of a simple expression. This makes it possible to transfer the formula from the Assistant into the position for the argument of another (a higher-level) function in the editor.

7.10.11 Function Dialog

The following discussion is address to imc Online FAMOS experts and describes the use of input aids. These come in handy for making nested expressions, or for quickly entering the same formula for multiple channels.

Call the *Function Dialog* either via its menu item *Edit* > *Function Dialog* or by pressing *ALT+F1*.

A list box offering all available functions for selection appears:



By double-clicking on a list entry or clicking on the *Parameterize* button with a list entry already selected, the parameterization dialog 775 is called.

Every function listed in the selection dialog is defined in the file ONLFAMOS407.INI (German) and ONLINEFAMOS.409.INI (English) along with how it handles its function parameters. They are located in the installation directory.

In the file ONLFAMOS.INI, you are able to specify which function parameters are fixed and which ones are variable, and how these parameters are to be edited in the dialog (see below).

Using the *Function Dialog*, multiple entries of a function can be made in the editor. A call of the function is formulated for each channel selected in the Channel List, and entered in the editor as an expression using the corresponding channel name. The expressions are transferred to imc Online FAMOS as a block of text whose beginning is denoted by `FUNCDLG User function name` and whose end by `FUNCDLG End`.

The block's definition line also contains the imc Online FAMOS function call followed by all pertinent channel names. When the *Function Dialog* is called from within the editor, the function's parameters are transferred along into the dialog. For this purpose, the beginning of the current selection must be located in the block's definitions line. *ALT+F1* starts the *Functions Dialog*. It is also possible to change the function's parameters in the definitions line. If the changes are compatible with the codes in the ONLFAMOS.INI file, the *Functions Dialog* will reflect the changes upon being called from within the editor.

To change the function call of the entire block, start the Function Dialog from within the editor.

If the parameters of a function in the Function Dialog's initialization line are changed in the editor, and the Function Dialog is called from within the editor, the changed parameters are reflected in the Function Dialog. If the changed parameters are not suited to the code words in the dialog, a warning message is posted, and the dialog is started according to the settings in the INI-file. Example: if the INI-file specifies a 1 for the code word FIX, and the corresponding value in the editor is 0, then 1 is implemented and a warning message appears.

7.10.11.1 Using the Function Dialog

1. Select those channels from the Channel List, to which the function is to be applied.
2. a) Call the *Function Dialog* and select the desired function from the list of user's function names.
   b) Call the parameterizing dialog directly from the editor.
3. Transfer the block into the editor using -> *OpBox*.
4. After changes to the selection in the Channel List, the currently selected channels are indicated in the *Function Dialog*

The topmost (not editable) box shows all channels which will appear along with the function in formulas in the editor.

Otherwise, operation of the dialog is analogous to that of the Formula Assistant.

What is displayed in the Function Dialog and which function parameters can be changed are subject to settings in the file ONLFAMOS.INI.

What are the minimum required contents of ONLFAMOS.INI?

Code	What appears in the Function Dialog
`[FUNCDLG...]`	Specification of a distinctive section name: a section name for the Function Dialog consists of the word "FUNCDLG" together with a number which is unique for the file ONLFAMOS.INI (e.g. 'FUNCDLG3'). These numbers which unequivocally characterize the sections may be spaced apart by a maximum of 10, so that the sections can still be recognized by the Function Dialog (for instance, 1, 3, 8, 17 are all allowed; by contrast, information on a section of the number 36 will be disregarded by the Function Dialog, if only section names with 1, 3, 8 and 17 are otherwise present). If many sections are removed from the ONLFAMOS.INI-file, it may be necessary to revise the numbers of the sections still left.
`CAPTION=...`	Specification of a distinctive user's-function name. In the Function Dialog all user's function names appear in the list of available functions. On the basis of this name it should be possible to recognize which imc Online FAMOS-function is meant (e.g. 'UsrFFT').
`FUNCTION=...`	Specification of the name of the imc Online FAMOS function (e.g. 'FFT').
`COMMENT=...`	Comment on the function (e.g. 'Determines the amplitude-spectrum'). The text is entered without quotation marks.
`RESULT1_DEFINE=...`	Specification of a filter for the result name '*' (e.g. 'Res_*'). '*' is replaced by the appropriate channel name in the lines which are transferred to the editor.
`RESULT1_TEXT=...`	Description of the result (e.g. 'Amplitude-spectrum'). The text is entered without quotation marks.
`PAR1_DEFINE=...`	Specification of a code, followed by information on the code (see below under 'Codes'), e.g. 'EDIT 128'
`PAR1_TEXT=...`	Description of the parameter (e.g. 'FFT-length (128 ... 8192)'). The text is entered without quotation marks.
`PAR1_DEFINE= EDIT ..`	A "2" is shown in the Function Dialog. The number in the dialog can be changed. A "2" is shown in the formula line.
`PAR1_DEFINE= LIST ...`	The dialog shows a drop-down list box. The texts appear after colons on the lines of the list. As the default, the first line in the list appears selected in the dialog. If a D or d (for Default) appears after a number, the corresponding line in the list appears selected. The divider between two texts in ONLFAMOS.INI is " || ". Only "||" is not recognized as a divider! In the formula line, the number is entered in front of the colon.

Code	What appears in the Function Dialog
PAR1_DEFINE= COMBO ...	Properties as for the code word LIST. Additionally, text can be written in the corresponding input box. If the text on a line matches a line in the drop-down list, the corresponding number is entered in the formula line, otherwise the text itself (e.g. a number) is entered.
PAR1_DEFINE= CHANLIST	The dialog shows a drop-down list box of the channels. You can select a channel from the list or write a channel name in the entry box. The text in the entry box is copied into the formula line.
PAR1_DEFINE= SELCHANS	The function is called for all channels selected in the Channel List (these are also indicated in the non-editable). A '*' appears in the formula line.
PAR1_DEFINE= FIX ...	The parameter is not indicated in the dialog. The text after FIX (e.g. a number, but also strings such as ' FILT001.DAT' are allowed) is entered into the formula line. For instance, the number 2.
PAR1_DEFINE= DEFAULT	The parameter is represented just as in the Formula Assistant, and is indicated in the dialog. The entire line PAR1_DEFINE DEFAULT can also be omitted. The Formula Assistant default settings are also indicated in like manner. But in general, it is best to specify a code.
PAR1_DEFINE= RANGEMAX1 ...	Input range for the specified channel, the upper limit is updated when the Functions dialog is opened. The parameter does not appear in the dialog. The input range can only be determined by analog input channels and incremental encoders. (If no input range can be determined for a channel, or if an error has occurred, the minimum and maximum are then both set to 0.0. The syntax-check then posts an error-message.) The '1' at the end of the code word indicates that the maximum is determined from the signal denoted by the function's first parameter. Results of procedures are not counted as parameters.
PAR1_DEFINE= RANGEMIN1	Lower limit of the input range for the first function parameter.
PAR1_DEFINE= RANGEMAX2	Upper limit of the corresponding channel's input range, analogous to the code word RANGEMAX1. However, in this case the maximum of the signal given by the second function parameter is determined. Results of procedures are not counted as parameters.
PAR1_DEFINE= RANGEMIN2	Lower limit of the input range for the second function parameter.
PAR1_DEFINE= PARi	The parameter does not appear in the Functions dialog. The same text as for the referenced parameter is entered in the formula line (also the same number, for example). If the referenced parameter, for instance EDIT, is stated in the dialog as taking the value 5, then the value 5 is also entered in the formula line for the parameter to which 'PARi' is assigned. Then, if the referenced value is changed in the input box, the referencing parameter is changed correspondingly. The referenced parameter must have been previously defined. To assign PAR7, for instance, to the 5th parameter would not be possible.

If no code was specified for a parameter, the default setting from the Formula Assistant is indicated. But in general, a code should be entered.

Note

The file ONLFAMOS.INI is overwritten when imc STUDIO is installed. Before you install a new version of imc STUDIO, first save the ONLFAMOS.INI file in a different folder or under a different name if you wish to retain changes previously made to this file!

7.10.11.2 Examples

Example 1

Determining the amplitude spectrum of two channels, Channel_01 and Channel_02. Starting with both channels selected in the Channel List, call the *Functions Dialog* for the FFT and transfer the function call into the editor of imc Online FAMOS using "OpBox". With the settings in ONLFAMOS.INI shown below, the text at bottom is then transferred into the editor:

ONLFAMOS.INI:

```
[FUNCDLG1]
CAPTION=MyFFT
FUNCTION=FFT
COMMENT= Determining amplitude-spectrum
RESULT1_DEFINE=Res_FFT_*
RESULT1_TEXT=Spectrum
PAR1_DEFINE=SELCHANS
PAR2_DEFINE=COMBO 0:Rechteck || 3d:Blackman
PAR2_TEXT=Type of window function:
PAR3_DEFINE=EDIT 128
PAR3_TEXT=FFT-Length (128...8192):
```

Editor of imc Online FAMOS:

```
; FUNCDLG UsrFFT: Res_FFT* = FFT(*, 3, 128) || Channel_01 || Channel_02
Res_Channel_01 = FFT(Channel_01, 3, 128)
Res_Channel_02 = FFT(Channel_02, 3, 128)
; FUNCDLG End
```

Example 2

Class-counting of the three channels CL_01, CL_02 and CL_03, with the Cross-Histogram. Starting with the three channels selected in the Channel List, call the *Functions Dialog* for ClCrossHistogram, select the channel CL_04 in the Channel-Combobox in the *Functions Dialog* and transfer the function call into the editor of imc Online FAMOS using "OpBox". With the settings in ONLFAMOS.INI shown below, the text at bottom is then transferred into the editor:

ONLFAMOS.INI:

```
[FUNCDLG2]
CAPTION=MyCrossHistogram
FUNCTION=ClCrossHistogram
COMMENT=Class-counting with the Cross-Histogram
RESULT1_DEFINE=Res_CH_*
RESULT1_TEXT=Result-Matrix
PAR1_DEFINE=SELCHANS
PAR2_DEFINE=CHANLIST
PAR3_DEFINE=RANGEMIN1
PAR4_DEFINE=RANGEMAX1
PAR5_DEFINE=EDIT 10
PAR5_TEXT=Number of classes (bins)
PAR6_DEFINE=RANGEMIN2
PAR7_DEFINE=RANGEMAX2
PAR8_DEFINE=PAR5
PAR9_DEFINE=EDIT 1
PAR9_TEXT=Res-units analog input channel
PAR10_DEFINE=FIX 1
```

Editor of imc Online FAMOS:

```
; FUNCDLG UsrClCrossHistogram: Res_* = ClCrossHistogram(*, CL_04, RANGEMIN1, RANGEMAX1,
                        10, RANGEMIN2, RANGEMAX2, 10, 1, 1) || CL_01 || CL_02 || CL_03
Res_CL_01 = ClCrossHistogram(CL_01, CL_04, RangeMin(CL_01), RangeMax(CL_01), 10,
                        RangeMin(CL_04), RangeMax(CL_04), 10, 1, 1)
Res_CL_02 = ClCrossHistogram(CL_02, CL_04, RangeMin(CL_02), RangeMax(CL_02), 10,
                        RangeMin(CL_04), RangeMax(CL_04), 10, 1, 1)
Res_CL_03 = ClCrossHistogram(CL_03, CL_04, RangeMin(CL_03), RangeMax(CL_03), 10,
                        RangeMin(CL_04), RangeMax(CL_04), 10, 1, 1)
; FUNCDLG End
```

7.10.12 Information and Tips

7.10.12.1 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 imc Online FAMOS, 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)
```

7.10.12.2 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

$$U_ = \sqrt{\frac{1}{T}\int_{t=T}^{T}\left[u(t) - Meanvalue(u(t))\right]^2 dt}$$

directly, T being the time over which the data are collected, an error message will be generated when the imc Online FAMOS-program 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 $u_$ 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)
```

7.10.12.3 Determining one or more Frequencies' Contributions to a Signal

Purpose:

The 150Hz 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 1kHz and N=1024 are specified, the frequency spacing is 0.9765625 Hz. This means that it is impossible to determine the 150Hz 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 1.024kHz, and the 150th spectral line is located exactly at 150Hz.

Example

Declare `Current` as the measurement channel to be analyzed, which is sampled at a rate of 1kHz and whose 150Hz spectral line is to be charted over time. One appropriate imc Online FAMOS program 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.

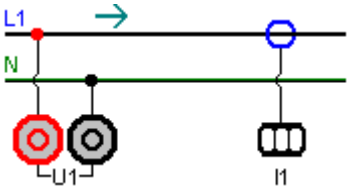
The assigned name, `_I_Channel`, begins with an underline, identifying it as an internal variable. This isn't strictly necessary, but it helps to conserve memory.

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 150Hz, since the frequency spacing is 1Hz, 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.

7.10.12.4 Power Measurements

Here follows the description for the Single-phase, Two-phase- and Three-phase-power measurement.

7.10.12.4.1 Single-phase Measurement Power1()



Single-phase power measurement

Effective value (root-mean-square)

The effective value is the square root of the mean-square of the input signal.

$$y = \sqrt{\frac{1}{T}\int_T x^2 dt}$$

Instantaneous power

The operation Instantaneous Power supplies the product of two respective sampling values.

$$p = u \bullet i$$

Active power

The active power is the average of the instantaneous powers during an averaging interval.

$$P = \frac{1}{T}\int_T (u \bullet i)\ dt$$

This describes the actual power recorded by the user.

Apparent power

The apparent power is the product of the root-mean-squares for current and voltage.

$$P_S = U \bullet I$$

The root-mean-squares are calculated according to the algorithm described under the operation effective value (root-mean-square).

Reactive Power

The reactive power is the geometric difference between apparent and active power.

$$Q = \sqrt{S^2 - P^2}$$

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

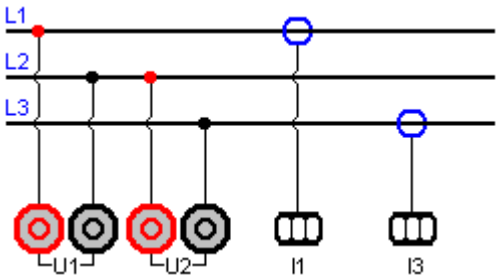
The power factor is the ratio between active power and apparent power. Active power and apparent power are calculated according to the above algorithm.

$$\cos(\varphi) = \frac{P}{S}$$

The power factor corresponds to the cosine of the phase shift between current and voltage.

7.10.12.4.2 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.

$$p = u_1 \bullet i_1 + u_2 \bullet i_2$$

Active power

The active power is the average of the instantaneous powers during an averaging interval.

$$P = \frac{1}{T}\int (u_1 \cdot i_1 + u_2 \cdot i_2)\, dt$$

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.

$$S = \frac{\sqrt{3}}{2}(U_{eff1} \bullet I_{eff1} + U_{eff2} \bullet I_{eff2})$$

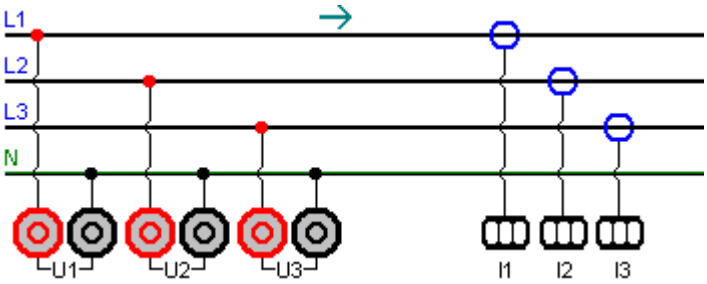
It is required that the symmetry of the supply circuit (three equal voltages with phase shifts of 120°) be stable. The root-mean-squares are calculated according to the algorithm under the operation effective value (root-mean-square).

Reactive power and Power Factor

See Single-phase power measurement 783

7.10.12.4.3 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 784.



with N (Y-connection)

Instantaneous power

The operation instantaneous power supplies the sum of the products of two respective sampling values.

$$p = u_1 \bullet i_1 + u_2 \bullet i_2 + u_3 \bullet i_3$$

Active power

The active power is the average of the instantaneous powers during an averaging time.

$$P = \frac{1}{T} \int \left(u_1 \bullet i_1 + u_2 \bullet i_2 + u_3 \bullet i_3 \right) \, dt$$

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.

$$S = U_{eff1} \bullet I_{eff1} + U_{eff2} \bullet I_{eff2} + U_{eff3} \bullet I_{eff3}$$

The root-mean-squares are calculated according to the algorithm under the operation effective value (root-mean-square).

Reactive power and Power Factor

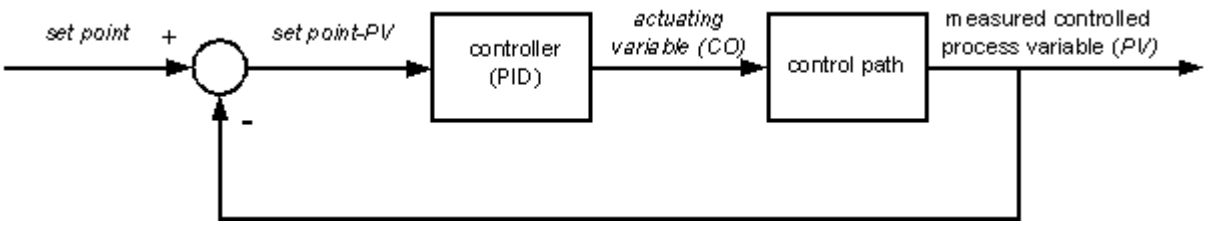
See Single-phase power measurement 783

7.10.12.5 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:

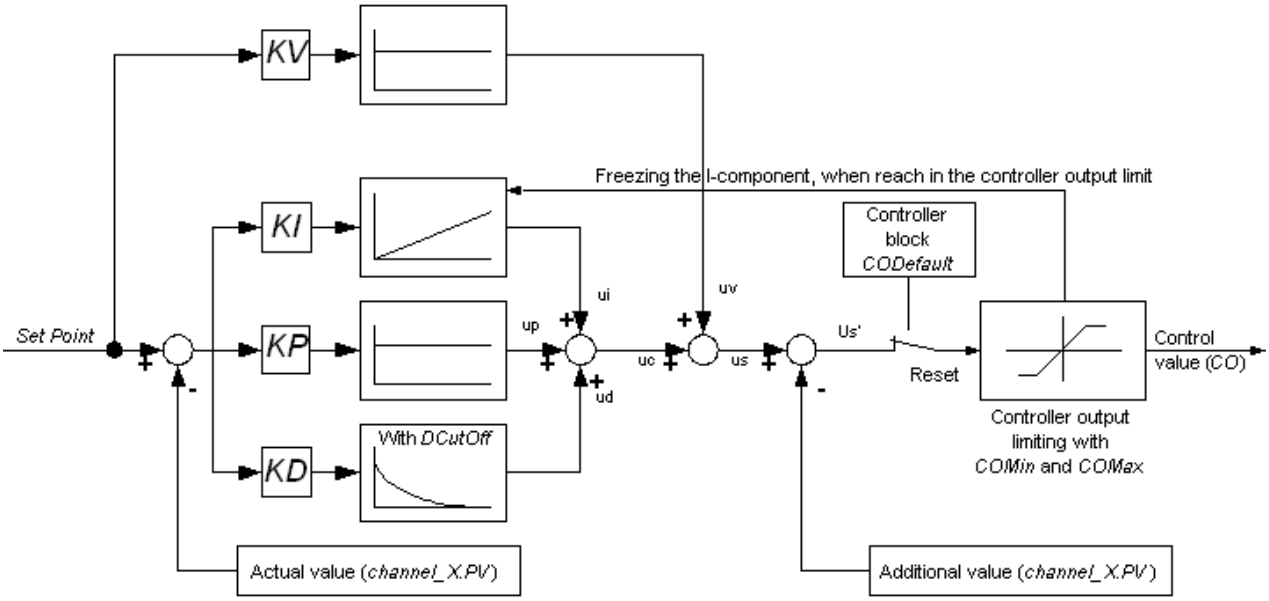
$$CO = KP*(SetPoint-PV) + KI*\int(SetPoint-PV) + KD*(SetPoint-PV)'$$

7.10.12.5.1 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.

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.
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

End
OnSyncTask( 0.1 )
    ; controller output scaled for DAC
    DAC_VoltageEngine = EngineController.Calc( pv.Speed )*0.05

    If Virt_Bit01 = 1 ; reset controller
        EngineController.Reset = 1
        Virt_Bit01 = 0
    End
    If Virt_Bit02 = 1 ; take new set point from display variable
        EngineController.SetPoint = DisplayVar_01
        Virt_Bit02 = 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.

7.10.12.5.2 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.

7.10.12.6 Comments for Class-Counting Functions

To apply class-counting functions in imc Online FAMOS, 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 imc Online FAMOS), 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>.

7.10.12.7 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!

7.10.13 Function Reference

Reference: Function Reference

The functions are described in the chapter imc Online/Inline FAMOS Function Reference 1440.

7.11 Miscellaneous

This chapter contains tips and tricks, frequently asked questions (FAQ), hints for debugging and more.

- Tuning, Tips and Tricks 792
- FAQ 797
- Troubleshooting 807
- Glossary 809
- File Extensions 811

7.11.1 Tuning, Tips and Tricks

Find some tips to optimize the use of imc measurement devices.

- Curve window updating 792
- Jiggling display and heavy burdening of the processor 793
- Long-term measurement 793
- Data overflow (Tips) 794
- Avoid large sampling times 795
- Multiple-triggered signals 795
- imc Online FAMOS - Computation uncertainty in summations 795
- imc Online FAMOS - Insufficient global RAM 796
- imc Online FAMOS - Digital Filters 797
- Pulse spikes in the signal 797

7.11.1.1 Curve Window Updating

Many curve updates per second during a running measurement make it easy to view the data plotting.

→ Tip: Fast interface

The Ethernet connection needs to be as free of interference as possible.

→ Tip: 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.

→ Tip: 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.

→ Tip: 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.

7.11.1.2 Jiggling Display and Heavy Burdening of the Processor

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.

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.

- Add the program imc.Studio.exe to the list of low-risk programs
  The default path is: C:\Program Files (x86)\imc\imc STUDIO...
- Add the program imcDevices.exe to the list of low-risk programs
  The default path is: C:\Program Files (x86)\imc\imc DEVICES...
- Add the database path along with all subfolders to the list of low-risk folders.
  The default path is: C:\Users\Public\Documents\DB
- Ensure that its data are not searched for viruses during reading or writing

7.11.1.3 Long-term Measurement

Say, for instance, you wish to make a really long measurement. You start a new experiment. You set the channel sampling interval to 1ms, and the measurement duration to *undefined*. 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: Ring buffer memory

- If the purpose is only to view data, than a circular memory buffer (ring buffer) should be set on the PC for each channel (Storage dialog). The ring buffer memory has a certain capacity, 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* 1035 (right mouse-click in curve window, context menu, Communicator). 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.

Warning

A high number of folders on the internal storage medium could cause loss of data!
See chapter *Avoidance of Data Overflow* > *Data rate* 627.

7.11.1.4 Data Overflow - 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

Use the controls on the page: Analog/Digital channels in the dialog: Data transfer to set a greater buffer time for each channel. "Auto" is only 10s; 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; 2s are usually enough.

→ 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, ...

Warning

A high number of folders on the internal storage medium could cause loss of data!
See chapter *Avoidance of Data Overflow* > *Data rate* 627.

7.11.1.5 Avoid Large Sampling Times

Say, for instance, a temperature is to be measured, and that 1 sampling per 10s is sufficient resolution. If a sampling time of 10s is set for the temperature channel in the configuration dialog, then there are 3 unfortunate consequences:

1. 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.
2. Since two data points must be sampled before a measurement can start by trigger, it takes 20s for the temperature channel, and even other, faster sampling channels to begin measuring.
3. At least one more sample must be made before the measurement of a channel can be stopped. Therefore, the system's response to pressing the "Stop" button is considerably delayed.

→ Tip: Mean ()

- Set a higher sampling rate in the Configuration dialog. Especially for temperature measurement, sampling rates of at least 200Hz 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.

7.11.1.6 Multiple-triggered Signals

Signals which cause multiple trigger releases generate Events which can accumulate to a data volume which overloads the system.

→ Tip: 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.

7.11.1.7 imc Online FAMOS - Computation Uncertainty in Summations

Goal: An incremental counter is to be used to find a cumulative displacement. An incremental counter 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
```

7.11.1.8 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.

7.11.1.9 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.

7.11.1.10 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.

7.11.2 FAQ

- Arithmetic mean 798
- CAN 798
- Data overflow (FAQ) 800
- Display / Display-Editor 800
- Device hard drive 801
- Incremental counters 801
- Network 801
- imc Online FAMOS 802
- Parameterizing bridges 802
- Scaling and input range 803
- Synchronization 805
- Triggers 806
- Working with two monitors 806
- Timed start 806

7.11.2.1 Arithmetic Mean

Question: Why does it take so long to start or stop measurement?

Answer: If one of the data capture channels is using the function *Arith. Mean* on the page: Analog/Digital channels in the dialog: Sampling & Preprocessing, then even for all other channels the data are displayed in the curve window only after the following delay:

t_1 = Number of samples to be averaged * sample time.

E.g. 10000 * 1ms = 10s.

Furthermore, the system needs up to two samples to settle. Thus the complete start process takes:

$t_{start} = t_1 * 2 = 20s$

7.11.2.2 CAN

Q: 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.

Q: Where are the monitor channels of the field bus channels?

Answer: There are no monitor channels, but with a trick 427 (CAN Monitor Channels from CAN Assistant) they can be created.

Q: How to change the baudrate of a GPS mouse?

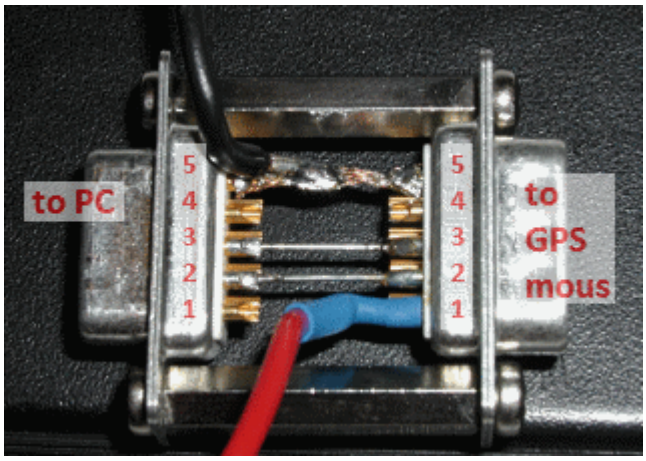
Answer: You have connected a GPS-mouse to a imc CRONOS-PL. That GPS receiver should be used with a imc CANSER-GPS module afterwards. The imc CRONOS-PL's firmware reconfigures the mouse to 19200 Baud. The imc CANSER module expects a baudrate of 4800 and is not able to reconfigure the mouse.

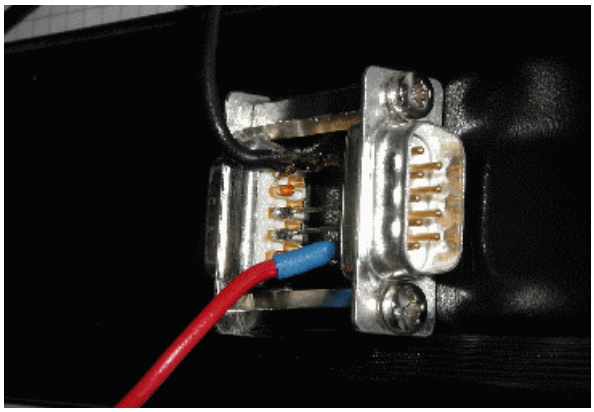
This can only be canceled using a PC and special software. For the mouse's power supply, an RS 232 to RS 232 adapter must be constructed.

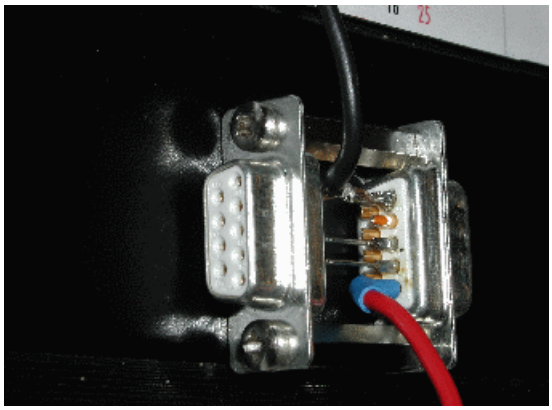
Adapter:

- Connector to the mouse: DSUB 9pol male
- Connector to the PC: DSUB 9pol female

Pin	Connection	Power supply connection at mouse terminal
1		Pin1: The mouse's 5V contact
2	Pin 2 - Pin 2	
3	Pin 3 - Pin 3	
4		
5	Pin 5 - Pin 5	Pin 5: Ground (0V), also connected to the PC-terminal's Pin 5 as reference ground.





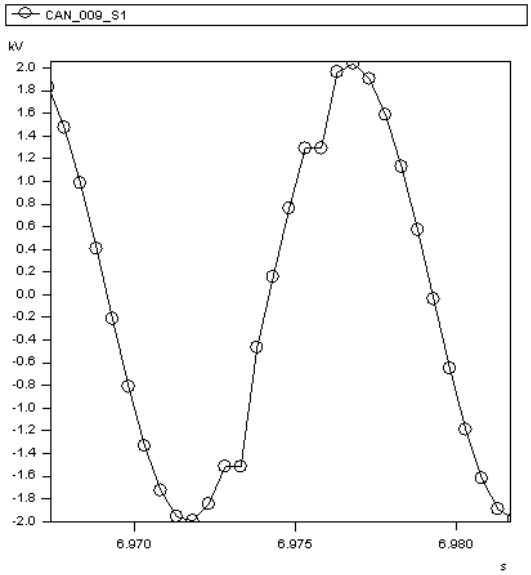


Reprogramming procedure:

Software: Contact the imc hotline and get the *GARMIN.zip* and unzip it.

1. start GPSCFG.exe.
2. Cnnect mouse with adapter at COM 1 or COM 2.
3. Set up connection.
4. Change the Baud rate to 4800 Baud and export to mouse ("Upload configuration").
5. Disconnect.
6. Set up connection again (verify that the connection now has 4800 Baud).
7. Change the configuration of the "NMEA Sentence Selection"; GPGGA and GPRMC must be active.
8. Export changes to the mouse ("Upload configuration").
9. Then the mouse should be recognized by imc CANSER.

Q: 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.

7.11.2.3 Data Overflow - FAQ

Q: Why does data overflow occur after a certain duration of measurement, although the connection is OK?

Answer: If the buffer duration is too small, data overflow can result, in which case measured data are lost. The data captured subsequently are then generally false or missing.

See also Data overflow (tips) 794

7.11.2.4 Display / Display-Editor

Q: A change made to the single-line list on the Display doesn't take effect.

Answer: Any change must be confirmed by pressing Enter.

Q: 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 using the Ribbon *Setup-Configuration* > *Device properties* 113
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 update when preparing the measurement.

7.11.2.5 Device Hard Drive

Q: The system won't recognize the data storage drive.

Answer 1: Drives with FAT12 or NTFS are not supported! Devices support FAT16 and for devices of the group 2 88 and higher FAT32.

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.

Q: My 16MB card doesn't work.

Answer: Flash drives up to 16MB must first be erased before it can work.

Note

Important Note

- Setting up more than 1000 folders is to be avoided, since it slows down the file system substantially.

7.11.2.6 Incremental Counters

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.

7.11.2.7 Network

Q: Why does it take so long to start the imc STUDIO- Software on my laptop?

Answer: A laptop normally operated within a company network is usually set to DNS. Working with imc STUDIO in stand alone mode can lead to substantial delay until Windows gives up trying to find the DNS in the non-existent network.

Solution: switch off DNS

Q: Why does IF-Config not allow setting the Address/Subnet, for example as 192.168.0.167 / 255.255.252.0?

Answer: The imc measurement devices exchange information via Subnet broadcasts consisting of Etherbits, for example. The Subnet broadcast address is composed of the IP-address and the subnet mask. The combination described would result in an impermissible value. For this reason, IF-Config prevents the user from setting this combination of values.

Solution: Set subnet to 255.255.255.0.

7.11.2.8 imc Online FAMOS

Q: 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.

Q: 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.

Q: The field bus channels are not available in imc Online FAMOS, but they are listed in the main dialog.

Answer: Field bus channels recorded in time stamping mode, can't be proceeded in imc Online FAMOS.

Q: 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.

Q: How many channels @100kHz sampling rate can be proceeded online by the function SoundPressureLevel simultaneously?

Answer: Online FAMOS can proceed up to 3 channels (independently from the main board).

7.11.2.9 Parameterizing Bridges

Q: 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.

7.11.2.10 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 803
- Scaling factor with offset 804
- Calculated two-point scaling 804
- Two-point scaling by means of the amplifier adjustment dialog 805

7.11.2.10.1 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 35mV/V at a scaling factor of 356.65 N per mV/V.

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 35mV/V, the appropriate input range is 50mV/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.

7.11.2.10.2 Scaling Factor with Offset

The example presents a displacement sensor which delivers 0V at -11.55mm. The scaling is 0.2597V/mm. The maximum value is 25mm.

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= 36.55mm. This means that the sensor delivers a max. value of 36.55mmx0.2597V/mm = 9.492V. If necessary, set the coupling to DC and then set an input range of 10V.

2. Scaling factor

   Go to the dialog: Range & Scaling. The scaling factor must be converted to physical units per volts:

   Y-factor= 1/ 0.2597V/mm= 3.8506mm/V

   Note that the value must be expressed in the SI-unit "m" and not "mm".

3. Offset

   Enter the offset if -11.55mm.

4. Input range

   The offset entered causes the input range to shift.

7.11.2.10.3 Caluculated Two-point Scaling

The example describes a pressure sensor delivering a current of 4 to 20mA. The spec sheet indicates that 4mA represents applied pressure of 0bar and 17.5mA represents 10bar.

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 20mA.

2. Range & Scaling

   Go to the dialog: Range & Scaling.

3. enter the values

   Next enter the two value pairs from the spec sheet.

7.11.2.10.4 Two-point Scaling by means of the Amplifier Adjustment 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 100kg is available. The sensor delivers a maximum of 8V.

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 10V.

2. Scaling factor

   Go in the dialog: Range & Scaling . Enter 1N /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 / Compensation

4. Channel balance: Point 2

   Apply a load to the crane with the standard weight. The force in Newtons is 100kgx9.861m/s^2= 986.1N. 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.

7.11.2.11 Synchronization

Q: 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 593 size could result in sample shift.

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 chapter on synchronization 655.

7.11.2.12 Triggers

Q: 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.

Answer: 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.

Problem:

- 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 *Trigger_48*.

7.11.2.13 Working with two Monitors

Q: How does one move dialogs which are positioned on disconnected monitors?

Answer: Dialogs can be moved by means of Windows key combinations:

1. Select the imc software from the Windows taskbar.
2. Press key combination ALT+Space (to open the dialog's system menu)
3. Press key *m* for *moving*.

Now the cursor is located on the invisible dialog title bar. Move the window with the four cursor keys.

7.11.2.14 Timed Start

Q: Although I started measurement by timer, the first samples are only read a few seconds later.

Answer: Same effect as described above (in the chapter Arithmetic mean 798). If you start your measurement with "Timed start" at, for example, 5:00 PM or on any full hour, there may be a delay before measurement actually begins, due to causes within the system and dependent upon the particular sampling rate specified. An inauspicious choice of the sampling rate may lead to a delay of several seconds.

7.11.3 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 the imc Hotline 13.

[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 807
- Error message when starting measurement 807
- Device not found or measurement data is not displayed 807
- Making a measurement 808
- Hardware defects 808
- Problems saving to internal drive 808
- Error description 809

7.11.3.1 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.

7.11.3.2 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 Setup - Connect the device 295).
- Next, check that the correct connection type has been specified. (i.e. Ethernet, TCP/IP, PPP - 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.

7.11.3.3 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 Setup - Connect the device 295.

7.11.3.4 Making a Measurement

There are several steps involved in performing a measurement. With imc STUDIO, it is important they be carried out as listed below.

Errors can occur if, for example, an experiment is loaded before the device has been connected with the PC both physically via interface cable and in the software.

1. Create new experiment or open an existing one (Ribbon *Home* > *New experiment* or *Open experiment*)
2. Select devices
3. Ribbon *Home* (or *Setup-Control*) > *Connect*
4. Create a signal configuration (on the page: Analog/Digital Channels)
5. Selection and configuration of the data storage (on the pages: Devices in the dialog: Storage and Analog/Digital Channels in the dialog: Data transfer)
6. Prepare measurement (Ribbon *Home* (or *Setup-Control*) > *Download*)
7. Connect signals
8. Adjust amplifiers (on the page: Channel balance)
9. Save experiment (Ribbon *Home* > *Save experiment*)
10. Start measurement (Ribbon *Home* (or *Setup-Control*) > *Start*)

7.11.3.5 Hardware Defects

If measurements still 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*) > *Download*
8. Start measurement (measurement duration 10s)
9. Select channel and display as curve
10. Restart measurement (measurement duration 10s)

7.11.3.6 Problems Saving to Internal Drive

For detailed explanations see chapter Device Hard Disk, removable drive 616

An error will occur if the device is prepared for an Autostart measurement involving the device harddisk and no µHD is in the device when it is switched on.

7.11.3.7 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 imc Hotline 13:

- Device serial number
- Software version used, export of the version information (About)
- Windows operating system version
- Experiment file from the database: <Experiment>.imcStudio

7.11.4 Glossary

Term	Description
µ-Disk	PCMCIA-, CF-, ExpressCard, USB-data storage or device hard drive exclusively for data acquisition under imc STUDIO.
action	Functions which carry out specific activities, such as starting the measurement, bridge balancing, or file operations, are designated here as actions.
Adjustment	Adjustment means tuning an instrument's readout deviation to within defined tolerances. A calibration is performed before and after adjustment.
aggregate sampling rate	Sum of the sampling rates of all active channels, which the device is able to process. See Technical Specs.
application	A program which runs under MS-Windows.
arming	Arming the triggers means preparing it to be released. Before the trigger is armed, it makes no difference whether or not the trigger condition is met.
Calibration	Investigation of an instrument's actual state based on specified target values.
Curve Manager	The Curve Manager is included with imc applications FAMOS, imc STUDIO, SEARCH and imc-DEVICES. It enables you to display your data in freely configurable "curve windows". These can contain any number of channels or computed variables, with individual or common axes. For complete information, see the manual Panel > Curve Window 854.
Data Manager	The Data Manager is a component of the Curve Manager, see the manual: Panel > Curve Window 854.
data set	A group of related numerical values, e.g. a series of temperature measurements in sequence.
digital I/O	Each device is equipped with binary input channels with TTL or 24V level or free relay contacts (alarm outputs).
display variables	The interaction between imc STUDIO and the optional display is conducted through the use of virtual display variables or bits, which the user either can have evaluated for the purpose of display; or can modify for the purpose of influencing the measurement process.
DSP	Abbreviation for Digital Signal Processor. The processor inside the device for online calculations, data management and data storage.
events	Defined state of or course within a measurement signal. (See the chapter Sources and events 187).
File format	The file format for the measurement files is identical to the one used by imc FAMOS. It is based on the DSF-Standard format.
Filter Design	A graphics-based filter designing program from imc.
Hot-Plug	Switching the device hard-disk while a measurement is running. (See Hot-Plug 618)
imc FAMOS	Program for subsequent signal analysis (Fast Analysis and Monitoring of Signals).

Term	Description
imc FAMOS Reader	Program for visualization, graphical analysis and measurement documentation, able to work with any data format.
imc Online FAMOS	With imc Online FAMOS lit is possible to calculate result channel signals. The functions are performed in real-time as part of the capacity of the DSP. (See chater imc Online FAMOS 739)
imc Online FRAME	imc Online FRAME is a program package for measurement engineering consisting of two user interfaces, a compiler and a mathematics library. It can be considered an extension of imc Online FAMOS. Its capabilities include open- and closed-loop control tasks as well as evaluation and intermediate processing of measurement results during the actual measurement.
measurement	The process beginning when the Start button is pressed (or when the device is activated if it is running in Autostart mode). The device stops measurement independently one all of the measurement's parts are finished. All captured data which are saved to the hard drive as belonging to a measurement are placed in a common folder.
measurement duration	The duration of a measurement can be determined by multiplying the number of samples with the of sampling interval.
monitor-scanner-inputs monitor-digital I/O	Monitor channels provide you with a means of observing data without recording it. They also make it possible to capture a channel at two sampling rates, one for recording and one for monitoring. It's also possible to make a triggered temperature measurement and have the respective most recent data point displayed as a number by means of a monitor channel.
multi-machine mode	Multi-machine mode makes it possible to conduct a measurement of a slowly-changing quantity in the background, while in the foreground several trigger machines initiate the recording of faster processes.
multi-monitoring	During a measurement it may be of interest to observe the data acquired by one device on multiple other PCs. Not to be confused with the Monitor channels in the standard software. One PC is designated the Master and it exports the configuration to a device and performs the measurement. A special program is needed for the purpose of viewing the data on the passive PCs. Contact imc if interested.
Multiple triggering	One trigger is released multiple times to capture portions of a measurement of limited duration. (See the chapter Multiple triggering (Multi-shot operation) 195)
net bits	Network-wide virtual bits which can be evaluated by the devices' trigger machines.
Official Calibration	In contrast to regular calibration this is an inspection of whether an instrument's indicated value lies within legally stipulated tolerances. Such official calibration can only be performed by government officials. The procedure includes inspection and certification. During the official calibration's validity period, the specified operational error limits must not be exceeded.
Pretrigger	The duration of data acquisition prior to a trigger event.
sample	Digital data point belonging to a measurement signal.
ScanDisk	Program for fixing hard drive errors. (See the chapter ScanDisk 628)
scanner inputs	The scanner channels are the analog input channels of the device. Since the acquisition of data is not really analog, but rather the active measurement channels are scanned at a certain rate, one speaks of scanning the input channels.
synchronous operation	Synchronized measurements on multiple devices.
time base	The device's internal clock. The time bases of multiple devices may drift apart over time during operation. The drift is in the range of 10-5 .. 10-4 , which can lead to deviations of several seconds per day. A remedy to this problem are radio clocks such as the DCF77 or GPS (both options). See also synchronous start.

Term	Description
Transitional Recording	Signal-dependent data reduction procedure (see the chapter imc Online FAMOS 739)
trigger	Event-dependent starting or stopping of data recording
trigger machine	Up to 48 mutually independent triggers can be defined. (See chapter Trigger Machine 185)
Unzoom	Reverses the zoom function, displaying the entire picture.
virtual bits	32 digital bits of information (to be regarded as a 32-bit register) available in the internal memory for control purposes. Virtual bits can be set or read by imc Online FAMOS or in response to events.
virtual channel	Results signal channel computed from the input channels by imc Online FAMOS
Zoom	Command to enlarge a portion of a window; the opposite is Unzoom.

7.11.5 File Extensions

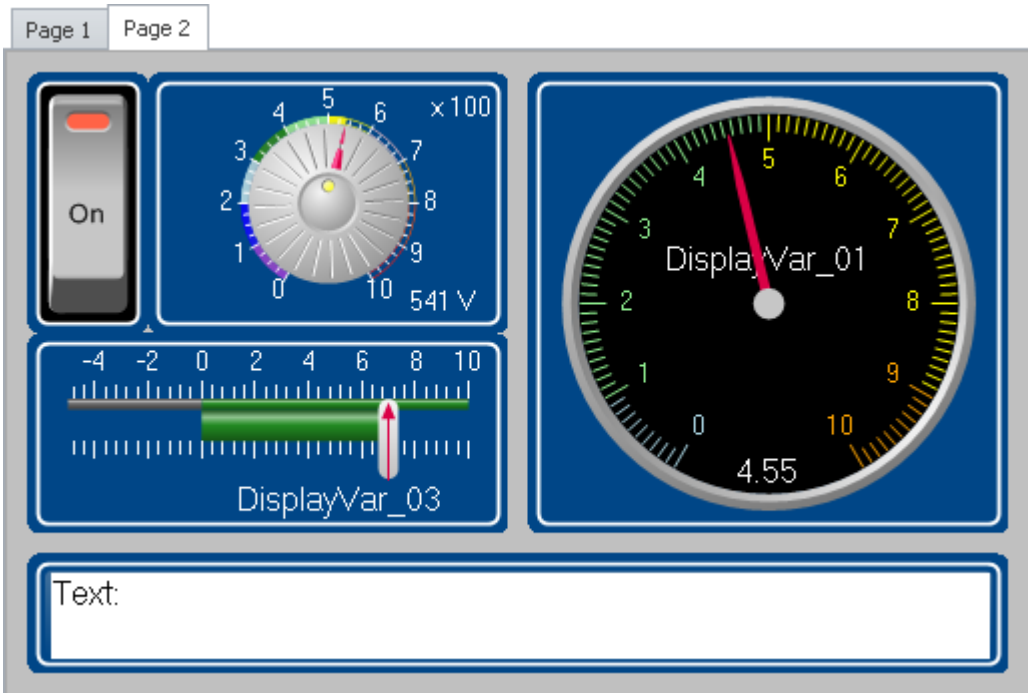
Extension	Definition
ASC	imc FAMOS ASCII file
BAK	backup file
BIN	imc FAMOS binary file
BMP	bitmap
CCV	curve configuration
CFG	imc STUDIO configuration
CLB	settings: clipboard
DAT	imc FAMOS waveform (computed data)
DEF	DLL header
DLL	Program library (dynamically linked library)
DOC	Microsoft Word document
DRB	Report Generator (German version)
DSP	Signal processor program
EXE	Executable program
FAS	File Assistant: file format description
HLP	WINHELP help file
INI	Initial settings file
OFA	imc Online FAMOS program list
OFC	imc Online FAMOS configuration file
PMU	imc DEVICES configuration
PRT	Report Generator (English)
RAW	imc waveform ("raw data")
RTF	Protocol file readable by WINWORD or Wordpad
SEQ	imc FAMOS-sequence
SET	Curve Manager: settings
SNS	Sensor characteristic
TMP	Temporary file
TXT	Text file
UMC	imc DEVICES-Device-configuration file, containing all experiment settings
UME, UMF, UMS	imc DEVICES-Device file, created upon preparation for Autostart or Disk start.

Extension	Definition
UMI	Device-configuration file for imc STUDIO, updated upon each time software connected to the PC
UMT	imc DEVICES-Device file, created upon disconnection of the PC.

8 Panel

imc STUDIO Panel is the imc STUDIO plug-in 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 1079), 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 1079 optimized for on-screen display
- Report pages 1079 optimized for printouts

Widgets

Various Widgets 817 are available as instruments and controls. The amount of Widgets varies by the product configuration. Even the curve window 854, 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) 852 to Widgets such as switches and buttons.

Operation

For the positioning of Widgets 817 and the assignment of channels and variables 1089, a variety of operation techniques can be used, such as Drag&Drop or context menus.

The Widgets can be operated by keyboard or mouse.

8.1 Ribbon

8.1.1 Control

Device control

Here you find the menu actions for device control.

For a detailed description, see the manual:
Setup - Manual > *Ribbon* > *Control* 90.

View

	Menu item	Description
	Panel Fullscreen Mode	

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	
	Print	

8.1.2 Design

New

	Menu item	Description
	Insert Page	Selection window for creating a new Dialog- or Report-page is opened. (see: Insert - Dialog / Report 1079)
	Standard Dialog	Creates a new Standard dialog page.

Edit

	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 1084).

View

	Menu item	Description
	Panel Fullscreen Mode	

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)

8.2 Tool Windows

There are four tool windows which belong to the Panel:

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 process vectors in a hierarchical tree diagram. With this tool window you can also set the variable linkage 1089 for a Widget.
Properties 816	The Properties window changes its contents in response to the object selected. If no Widget is selected, the page properties 1081 are displayed.
Page templates 828	Ti create new pages there are templates for dialogs (optimized for screen display) and reports (optimized for printout).

8.2.1 Properties (general)

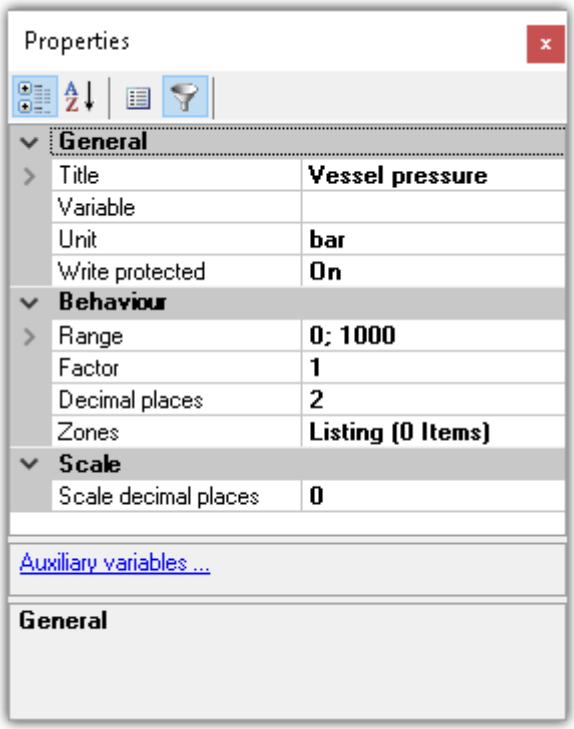
In the tool window Properties (or via the context menu of the selected Widget, (see context menus of the Widgets 834), the individual Widgets or the Panel pages can be adapted. The content of the Properties window depends on the respective element.

There are two display modes:

	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.



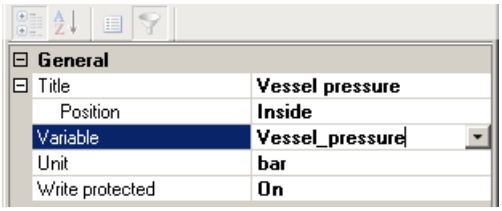
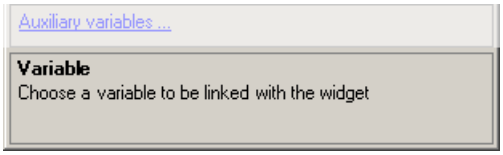
In the window *Properties*, these properties can be adapted to your wishes. Here, for instance, the title *Vessel pressure* has been added, the unit set to *bar*, and the display range set to 0 through 1000.

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.

8.2.2 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 36 or Technical Data Sheet 1678)

Widgets are placed (834 inserted 834) 834 on the Panel page.

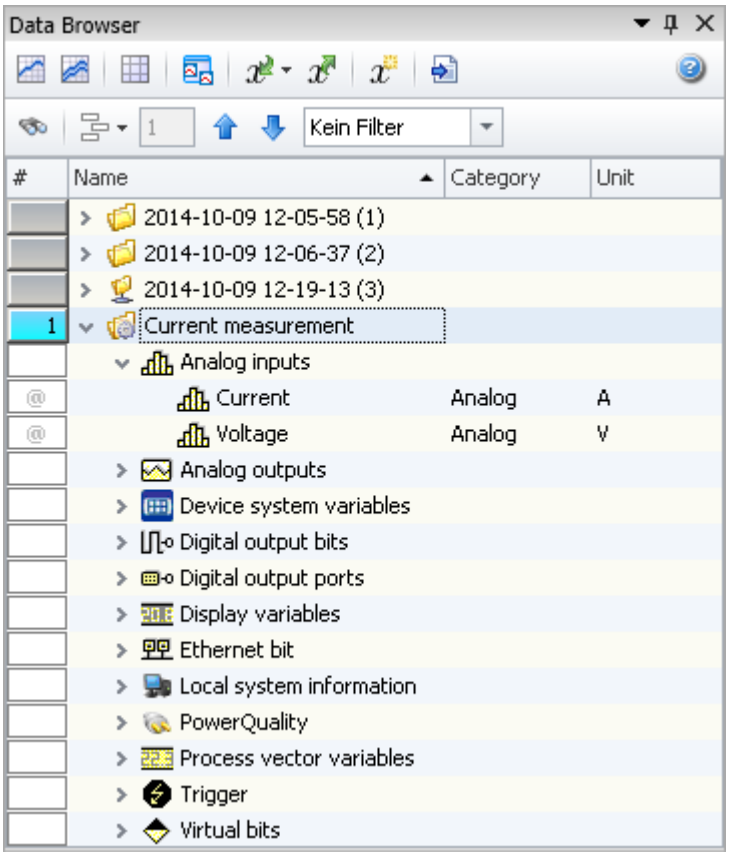
- Widgets - Operation and Properties 833: Select this link to go to further information on operation of the Widgets.
- Special Widgets 1074: The descriptions of certain special Widgets are presented here.

8.2.3 Data Browser

The tool window Data Browser shows the available data (called variables) such as measurement channels, Display variables or process vectors in a hierarchical tree diagram. From this tool window, you can move the variables by means of Drag & Drop 1090 to a Panel page and link them with Widgets. You can also navigate through measurements 822 and assign measurement numbers 821.

FAQ: Why are the current channels not displayed?

The tool window Data Browser only shows changed device configuration (e.g. channels) once the menu action Process configuration has been selected. On this topic, see the chapter Process configuration 139 in the documentation on the plug-in imc STUDIO Setup.



Data Browser in the Panel

Structure

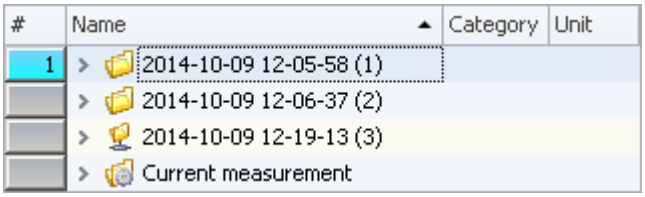
The Data Browser has two toolbars (Variable 824 and Search/Navigation 824) and a Table with multiple columns. Columns can be hidden or shown, according to the user's choice.

The Table presents the measurements and measured data in a tree diagram.

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 821)
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 variable in a certain measurement contains multiple events, it is possible to select the desired event here. The associated Widgets then show the event selected.
Comment	Displays the content of the variable's property: *Comment*. With device variables, this can be defined by means of the plug-in Setup.
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 plug-in Setup.
Metadata columns (optional)	If metadata about the measurement are exported, they can be added by means of additional columns in the Data Browser (see Metadata-Assistant).

Measurements and measurement names

The Data Browser, along with the entry for the Current measurement ( ), displays all measurements saved with the experiment.



Current measurement ( )

The variables under *Current measurement* comprise the device variables (i.e. analog channels, Display-variables, virtual channels, ...), as well as all calculated variables such as virtual channels or imc FAMOS results variables or any user-defined variables created.

These variables 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 (x)*".

(year-month-day hour-minute-second); x: measurement number

Notes regarding measurement

- The measurement name can be arranged to be defined by the user. On this topic, see the settings for the Measurement storage area 63. 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: *Project Management > General options > Load measurements*.
  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 that you may only use symbols which are allowed. The exception is the Backslash (\). If you use a Backslash (\) in the name, a subfolder is created. E.g. "Motor_X\Measurement_1". This creates a folder : "Motor_X", which contains an additional folder: "Measurement_1". The measured data are saved in this latter folder.

Operation - Loading/unloading messages

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 path to their storage location, but the data located there are not in the software's own memory.

However, only measurements which are already loaded can be displayed or processed.

Note

It is best to only load measurements which are currently needed. Loaded measurements make demands on the memory. If any measurement is no longer needed, it is best to unload it.

To load measurements, there are multiple methods:

- open the measurement's context menu and click on the corresponding item,
- select the measurement by double-clicking or click on the symbol in front of the measurement's folder,
- or use the arrow keys and Enter key.

To unload measurements

- 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 would need to be loaded, for its contents to be displayed.
	Saved measurement status: loading	This stored measurement is in the process of being loaded. This is a temporary condition which will last until all the measurement's data have been loaded to the program's memory. Subsequently the status changes automatically to "loaded".
	Saved measurement status: loaded	This is a stored measurement which has been loaded. Loaded measurements can be viewed and processed.

Note: Automatic loading on demand

If a Widget is linked with a variable belonging to a saved measurement, this measurement is loaded automatically when the option: *Automatic loading on demand* 59 is activated.

Note: Loaded measurement

When a measurement is loaded, its associated files are 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.

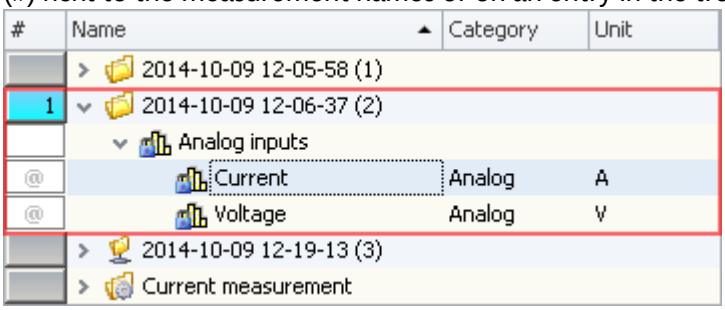
By means of the measurement number, it is possible to access measurements without using or even knowing their respective assigned names.

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 reference measurement. The reference 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.

Navigating

The assigned numbers (measurement number) can be transported through the list of measurements.

This means that by using the (cursor) arrow keys, or by means of the buttons in the toolbar ( ), you can make the respective numbers appearing in the "#" column move up and down through the list of measurements. Doing this associates the assigned number(s) with the measurement(s) corresponding to the current list position(s) (see Operation - Control and Navigation 822).

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 or the arrows in the toolbar 824 ( ).

Navigation using the mouse button

When a measurement or variable is selected by mouse click, it receives the current measurement number (see: Operation - Assigning/Locking measurement numbers 821).

Navigation by arrow key/button (cursor or toolbar)

Using the cursor keys or by means of the toolbar, you can navigate according to the Navigation mode set. All assigned measurement numbers are relocated upward or downward by the step size specified.

In the drop-down list in the toolbar ( ), you can select the navigation mode:

	Navigation mode	Description
	Default	The selection skips from cell to adjacent cell.
	Over measurements	The measurement number is relocated from one measurement to the next.
	Over events	The measurement number remains with the measurement selected. If the measurement consists of multiple events, the focus navigates through the events.
	Over measurements and events	The measurement number is relocated from one measurement to the next. If a measurement consists of multiple events, the focus navigates through the events first, before the focus moves to the next measurement.

Navigation step size

In the input box (1) in the toolbar, you can specify a step size (≥ 1) for the navigation.

Operation - Checkout and Check-in

Saved measurements are available for subsequent processing. You can save the results of imc FAMOS analyses along with the measured data, or also process the measured data and save the returned results back on the hard drive.

When a measurement is loaded, the file is imported once from the hard drive. The variables belonging to the same loaded measurement can be processed in imc STUDIO. However, the changed are automatically discarded when the measurement is unloaded.

You can also write changes back to the file. To do this, proceed as follows:

- Load 820 the measurement, e.g. by means of the context menu item: *Load measurement(s)* 825 ( )
- Obtain writing authorization via Checkout, e.g. via the context menu item: *Checkout* 825 ( )
- Edit the variable
- Write the changes to the file via Check-in, e.g. via the context menu item: *Check-in* 825 ( )
- Unload 820 the measurement, e.g. via the context menu item: *Unload measurement* 825 ( )

Variable symbol - State

The extra character in the variable's symbol in front of the variable's name reflects the variable's state:

	State	Description
	Loaded variable (protected)	This is a loaded variable (e.g. analog input), which can not be processed. Any changes will be discarded. To perform editing, select the context menu item: *Checkout* 825 ( ).
	Loaded variable (with write-permission)	This is a loaded variable (e.g. analog input), which can be processed. Changes can be saved using the context menu item: *Check-in* 825 ( ).

8.2.3.1 Toolbar

Toolbar: Variable

	Menu item	Description
	Show individually	All selected channel variables (e.g. analog inputs) are opened in separate "free-floating" curve windows 1077. All other selected variables (e.g. Display variables) are added to the window Current values 826.
	Show together	All selected channel variables (e.g. analog inputs) are opened in a shared "free-floating" curve window 1077. All other selected variables (e.g. Display variables) are added to the window Current values 826.
	Show current values	Opens the dialog Current values 826, 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/Import variables	Load: Creates variables with values from a file. Import: Imports values from a file to existing variables. (More info is provided in the chapter *Shared Components* > *Command Reference* > *Load/Import Variable* 1647)
	Export variables	Exports variables' values to a file.
	Create user defined variable	Creates user-defined variables, including single value- and text-variables, or report channels. (More info is provided in the chapter *Shared Components* > *User-Defined Variables* 1671)
	Load reference measurement (shows comparison measurement)	Allows viewing and comparison with measurements from other experiments. Selected measurements from other experiments can be displayed in the current experiment's Data Browser. This makes comparisons possible.
	Help	Basic information on the Data Browser

Toolbar: Search/Navigation

	Menu item	Description
	Find	Displays a search box for the purpose of locating elements in the Data Browser. Enter the text to be located. When you click on the <ENTER> key or the button ( ) at right, the search starts. The first element is selected. To select the next element, click again on the Enter key or on the button ( ) at right.
	Navigation Mode	Selection of Navigation mode (see Operation - Control and Navigation 822)
1	Navigation step size	Step size for the Navigation (see Operation - Control and Navigation 822)
	Step up/down	Navigate one step up/down (see Operation - Control and Navigation 822)

8.2.3.2 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
	Update Data Browser	
	Filter	Opens the Editor for creating a filter. The Data Browser then shows only the appropriate content according to the filtering.
	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, or report channels. (More info is provided in the chapter *Shared Components* > *User-Defined Variables* 1671)
	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 chapter *Shared Components* > *User-defined Variables* 1671)
	Show individually	All selected channel variables (e.g. analog inputs) are opened in separate "free-floating" curve windows 1077. All other selected variables (e.g. Display variables) are added to the window Current values 826.
	Show together	All selected channel variables (e.g. analog inputs) are opened in a shared "free-floating" curve window 1077. All other selected variables (e.g. Display variables) are added to the window Current values 826.
	Find	Displays the Search box. See the matching description for the toolbar 824.
	Navigation mode	Selection of Navigation mode (see Operation - Control and Navigation 822)
	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 820)
	Unload measurement(s)	Unloads loaded, saved measurement(s) (see Operation - Loading/unloading measurements 820)
	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 820)
	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* 62)
	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 819).
	CopyName	Copies the name of the measurement to the Clipboard.

Table context menu - Saved variable (type)

Right-clicking the mouse over the table on saved variables or their category causes additional context menu items to appear:

	Menu item	Description
	Checkout	Obtains writing rights to saved variables, so that they can be processed (see Operation - Checking in/out 823)
	Check-in	Writes changes to variables, which have been checked out, to the file. Subsequently, the writing rights are deleted (see Operation - Checking in/out 823)
	Undo	Discards changes to variables which have been checked out.

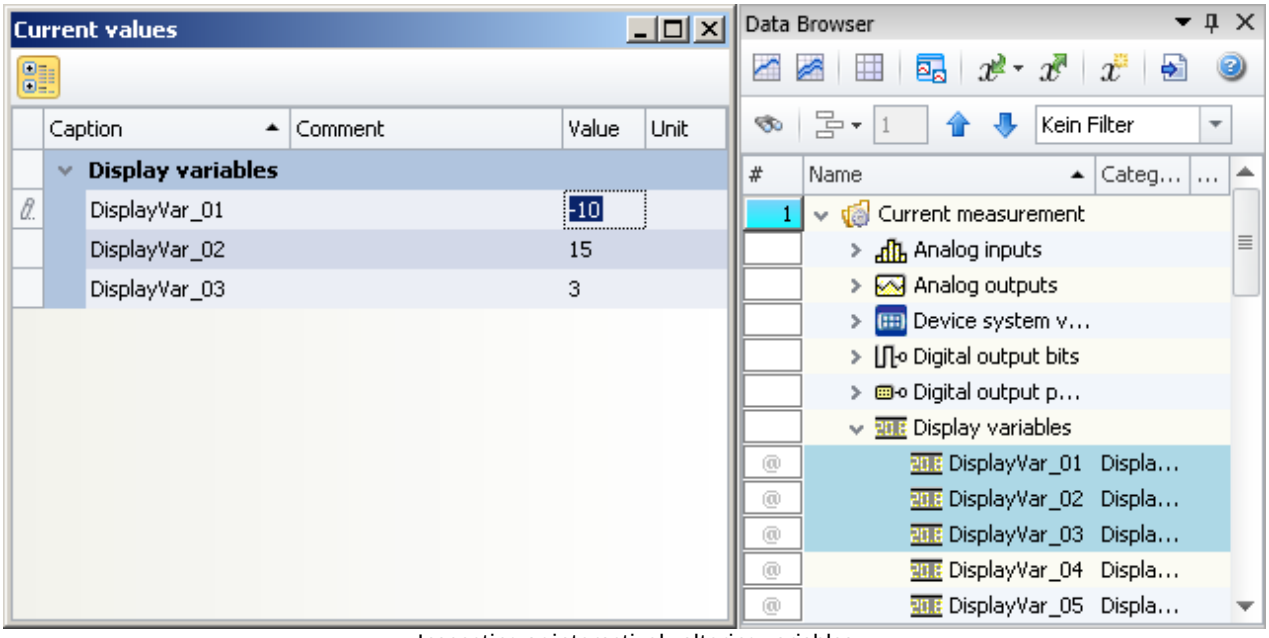
8.2.3.3 Current values - Inspecting/changing variables

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* 824-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

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

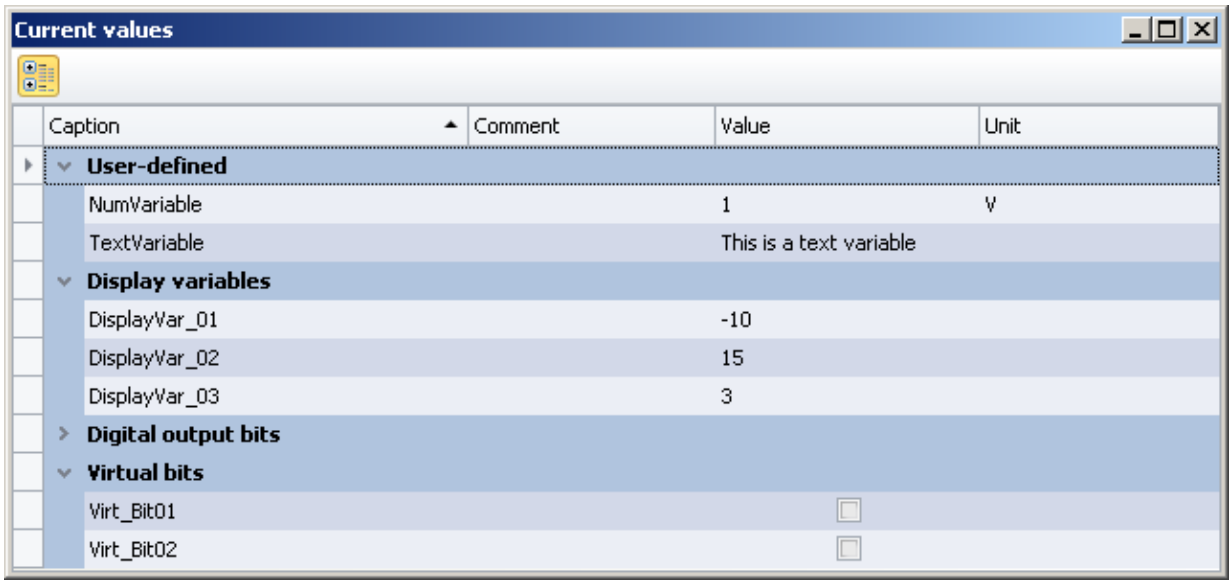
Structure and configuration

The dialog current values shows the desired variables in a table with multiple columns. The columns can be configured 133, as described in the plug-in Setup.

The variables are organized in groups. To activate or terminate the grouping, click on the Grouping button ( ).

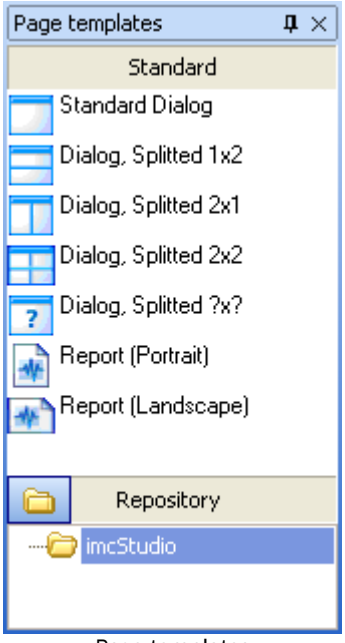
The table contains, among other things, the variable names and their current value. The value is displayed in the Edit box as a number or text. For Ethernet bits there is a checkbox.

The dialog settings are saved with the experiment and displayed along with it too, if it was open.



Dialog: Current Values
Sample variables

8.2.4 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 1079 a new page.

8.2.5 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: %HOMEPATH%\Documents\imc\imc STUDIO\Widgets
- Page repository: %HOMEPATH%\Documents\imc\imc STUDIO\PanelPages

Note: The Repository is user-dependent

The default path is in My Documents. Consequently, it is user-dependent. If all users are to have access to the same repository, relocate it accordingly.

Suggestion:

- %PUBLIC%\Documents\imc\imc STUDIO\...

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.

8.3 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 1079)

	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).
	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 828).
	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 1085).
	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)
		Access Rights to the Page	The authorization to edit and display specific pages can be linked to specific user roles.
		Lock Page	Lock/Unlock the current page. Individual pages can be protected against changes (see Lock and unlock page 1084).
	Copy		Copies the selected page to the Clipboard.
	Paste		Inserts a page previously copied to the Clipboard. If the page's designation is identical to that of an already existing page, the imported page is inserted with a new page title (serial number).
	Page		Deletes the page. To restore an accidentally deleted page, click on the menu ribbon item *Edit* > *Undo* ( ).
	Rename		Changes the page's title in the page tab.

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 (does not apply to Widget-settings).
	Redo	Restores a previously reversed change.

	Menu item		Description
	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.
		<x> skin	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 1085).
	Adapt Page Size		See Adapt/automatically adapting the size of pages and Widgets 1083.
	Properties		Opens the Properties window of the Panel page 1081.

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 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 840)
		To Background	Moves the selected Widget all the way to the background, beneath all other Widgets (see: Widget-levels - Foreground/Background 840)
		Align	Aligns all selected Widgets to a line. (see: Align and Cascade 840)
		Cascade	Widgets are in stacked at the upper left corner of the Panel page according to their respective Widget-levels. (see: Align and Cascade 841)
	Group		Groups Widgets together (see: Grouping Widgets 843)
	Dock		The selected Widget is docked at the respective edge (see: Docking 842)

	Menu item	Description
	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 852)
	Into the Repository	Saves the selection or the selected Widgets in the repository (Tool Window: Widgets 817).
	Properties	Opens the Properties window of the Widget 843.

8.4 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 ( ).

Or at the associated symbol next to the Page name.

By default, this mode is deactivated following the first program start. When you use Drag&Drop to place a Widget on the Panel, the Design Mode is activated automatically. To change the status directly, click on the symbol shown above. The mode is a property of the whole Panel and affects all its pages.

To completely protect a selected page against being changed (independently of whether Design Mode is on), use the function Lock page 1084.

Note: Delete 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 plug-in, but it only applies to the pages present.

8.5 Widgets - Operation and Properties

Widgets are placed (834 inserted 834) 834 on the Panel page.

They offer a variety of context menus 834 and properties 843

In general, there are multiple ways to operate the Widgets:

- Move and change size 835
- Select 836
- Editing of Widget-elements 837
- Move to background/foreground (Widget-levels) 840
- Align and Cascade 840
- Docking 842
- Grouping 843

Some Widgets have special functions, such as:

- link to variables (see the chapter: Variable linkage 1089)
- Extra variables (multiple scales/pointers and calculations of variables, e.g. Difference)
- Zones 847
- Linking commands with switches 852
- Use of placeholders 1655

The structure

Each Widget consists of different *elements* 837:

- Boxes displaying the title or the current value
- Scales
- Numbers
- Meters, bar meters, ...

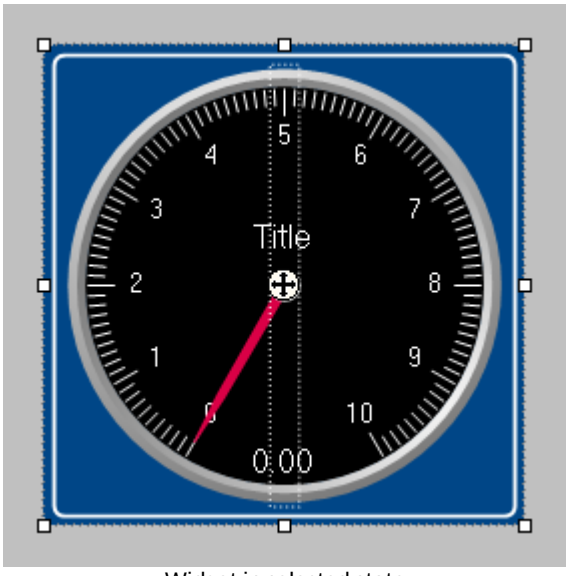
Some of these element have their own context menu 834 and can be edited 837. Among other things, they can be moved, or their size and color can be changed. Some of these changes can also be made through the use of the Widget's properties 843.

8.5.1 Insert Widget

There are various ways to create a Widget on the page:

- Use Drag&Drop to move the Widget from the toll 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 830 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 818 to Panel page and select the desired Widget (see: Variable linkage by Drag&Drop 1090)

Once you are finished positioning, the Widget appears as selected, as shown in the image below:



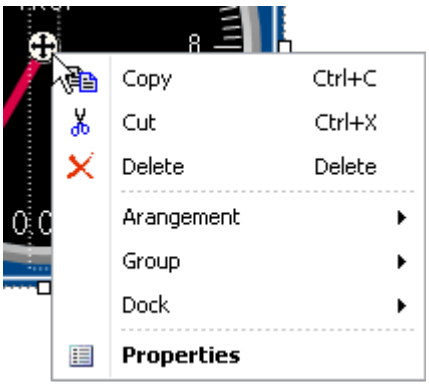
Widget in selected state
(example)

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 1085).

8.5.2 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.

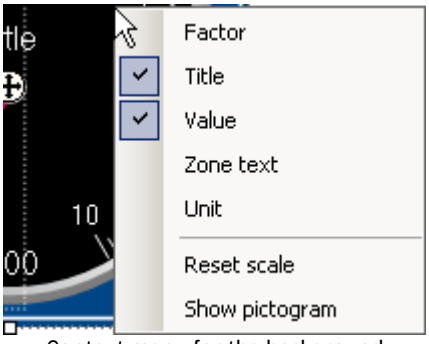
By this means, you can also open the Properties 843 without using the tool window.

Other context menus

Some Widgets also have additional context menus:

- to show or hide Widget-elements 837, such as boxes, and for resetting some properties
- to edit selected Widget-elements 837

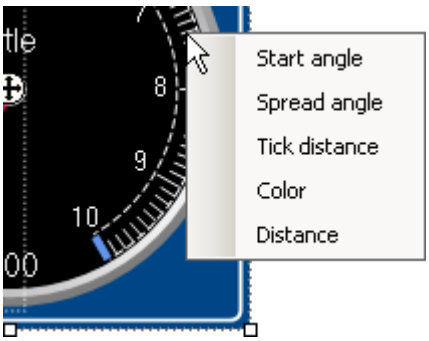
Here, using the example of a *standard meter*.



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 837, for example.



Context menu for the Widget element

To open this context menu, select the Widget-elements 837 and right-click over the background or the control.

A context menu appears, using which the Widget-elements 837 can be edited.

8.5.3 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.

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 1085).

Selection

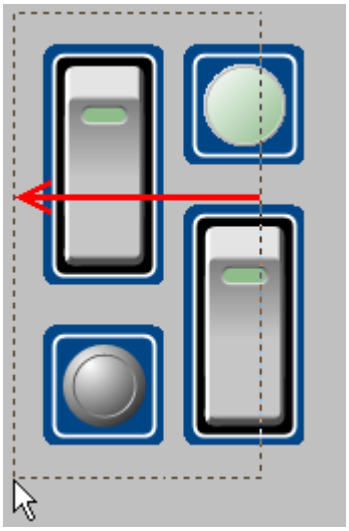
On a single page, you can select either one or multiple Widgets. The operations below can be performed as desired in any order.

- 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.

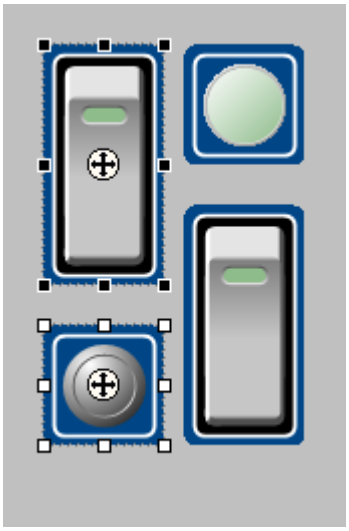
Selecting multiple Widgets

- 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:

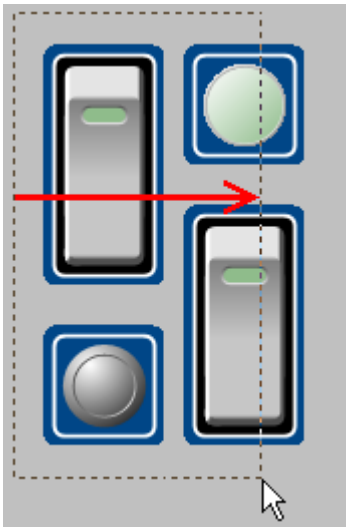


Dragging the frame from right to left

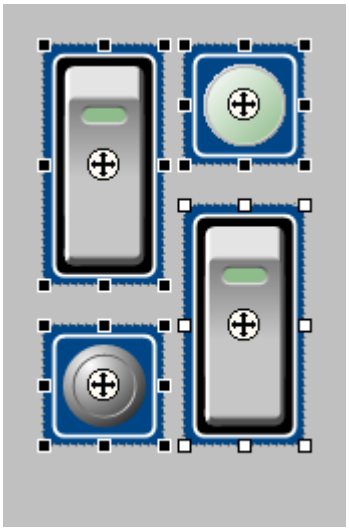


Completely framed Widgets are selected

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

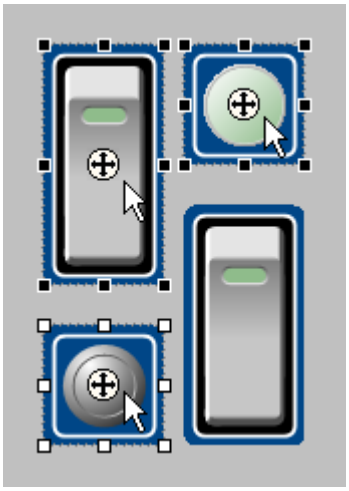


Any Widgets touched are selected

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:

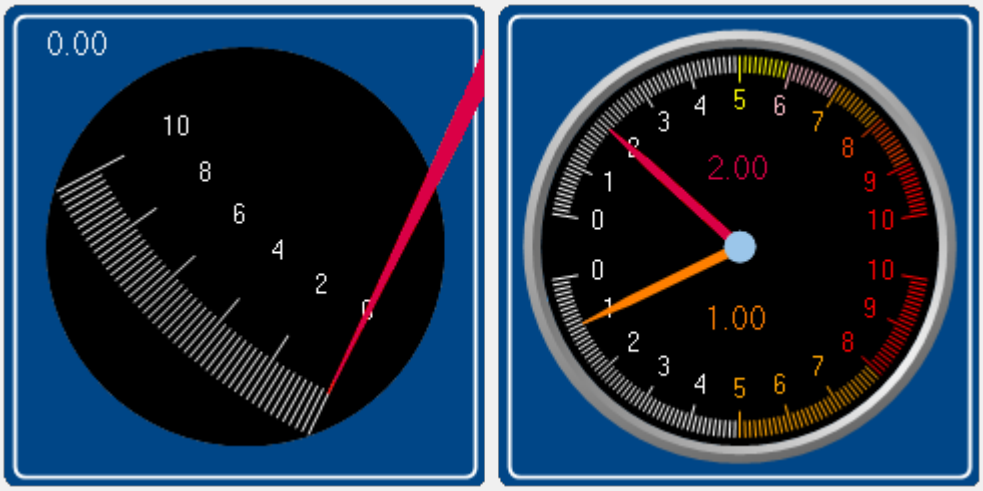
- Boxes indicating the title or the current value
- Scales
- Numbers
- Meters, bar meters, ...

Editing elements

The various Widgets offer many ways to vary the appearance without having to set the Properties 843. You can:

- move and enlarge boxes
- edit various Widgets elements

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.

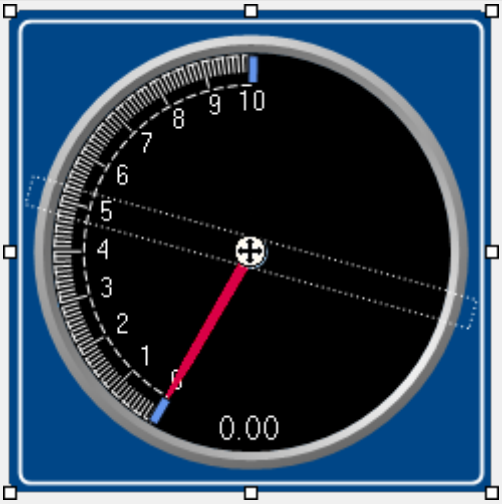
To illustrate, the images below use the Widget *Standard meter* as an example:

Example 1: Move and change the size of boxes

Move and change the size of boxes

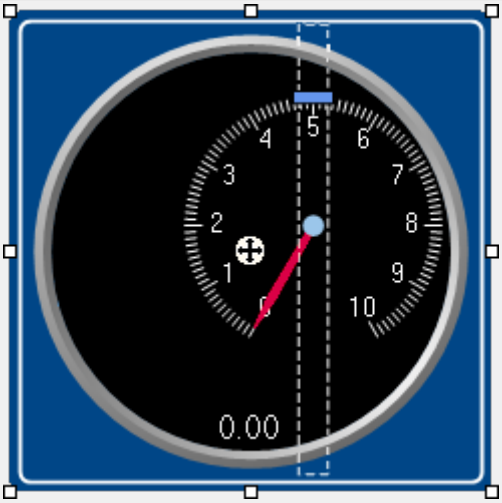
Example 2: Changing the scale angel and the sweep angle

(see also Properties 843)



Example 3: Changing scale position and radius

(see also Properties 843)



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 842.

By means of the level, you determine whether a Widget appears above or behind another one.

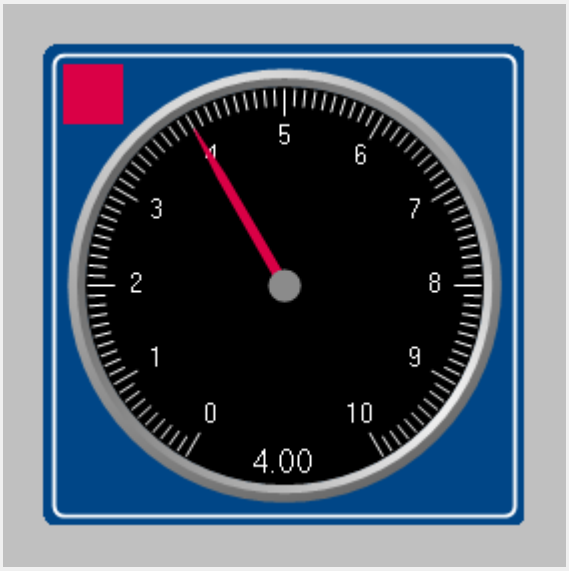
To do this, open the context menu of the"crosshairs" (⊕) within the selected Widget.

- Select *Arrangement*:

	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.

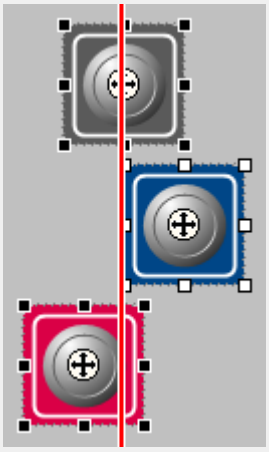
To do this, open the context menu of the "crosshairs" (⊕) within the selected Widget.

- Select *Arrangement*:

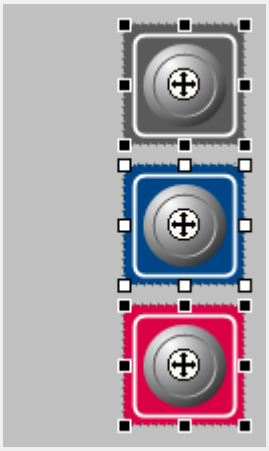
	Align top	Alignment to the upper edge of the last Widget selected
	Align left	Alignment to the left edge of the last Widget selected
	Align Right	Alignment to the right edge of the last Widget selected
	Align Bottom	Alignment to the bottom edge of the last Widget selected

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.

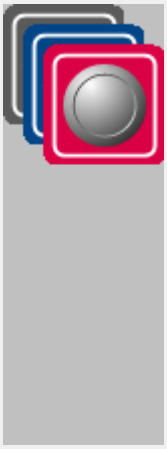
- Select *Arrangement*:

	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.

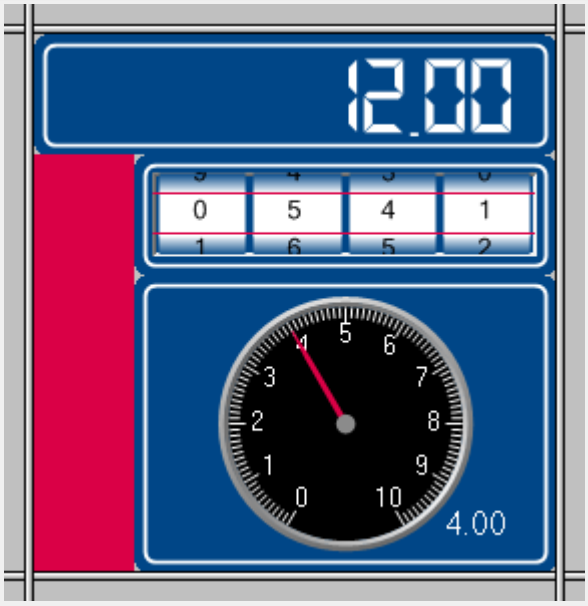
To do this, open the context menu of the "crosshairs" (⊕) within the selected Widget.

- Select *Dock*:

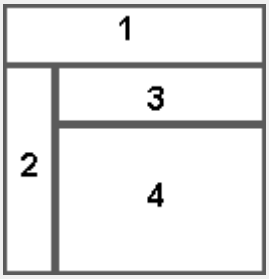
	Free	The Widget is not permanently docked to any edge, but can be freely moved. The size can be varied toward any side.
	Top	The Widget is permanently docked to the upper edge. The size can only be varied toward downward.
	Left	The Widget is permanently docked to the left edge. The size can only be varied 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 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*:

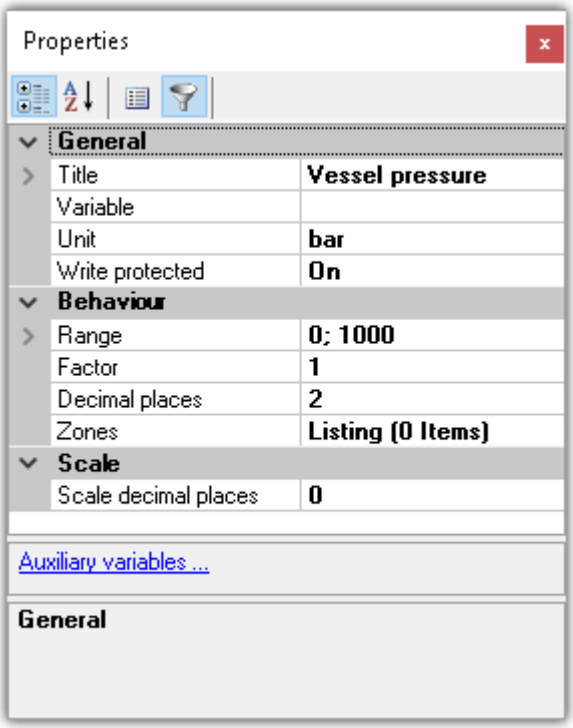
	Create Group	Selected Widgets are joined to a group.
	Dissolve Group	Dissolves the grouping. The Widgets can then each be edited separately.
	Enter Group	Opens the existing group. The properties of the individual Widgets can be edited separately.
	Leave Group	Closes an open group.

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.

8.5.4 Properties - Widget

If you have selected a Widget, its properties can be adjusted in the tool window 815 Properties (or via the context menu of the selected Widget, see context menus of the Widgets 834).

The content of the Properties window depends on the respective Widget selected. For info on the window structure, see Properties (general) 816.



Below, certain commonly occurring properties are listed.

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 62", 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.
	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.
Variable		How to link a Widget with a variable is described in the chapter Variable linkage 1089.

Property	Description
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 842.
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).

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)
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 847.
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)
Bitmask	Displaying variables' individual bits Selected Widgets offer the option of only displaying individual bits in a variable. Example: A Field-bus 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.
Decimal places	Number of decimal places for display of the value. (See also "Scale decimal places")
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 852 which are carried out at state transitions (called events).
Increment	Increment size for Widgets with buttons for up and down.
Range	The display range, or the range of valid entries. E.g., for a standard meter, the min- and max-values for the scale. If the range is to be indefinite, you can enter "Infinite" or "-Infinite".
From the Variable	The range can be made to depend on the variable, if the variable has its own range (e.g. Analog Channels). The range could always match the measurement range, for example. By default, the channels do not return any range. If the measurement range is to be used, please change the setting on the following page: Setup page "Analog channels"; Dialog: "Curve properties")
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 1655.
Zones	Displays existing zones. See Zone dialog 847 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; ...
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
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.

8.5.4.1 Zone dialog

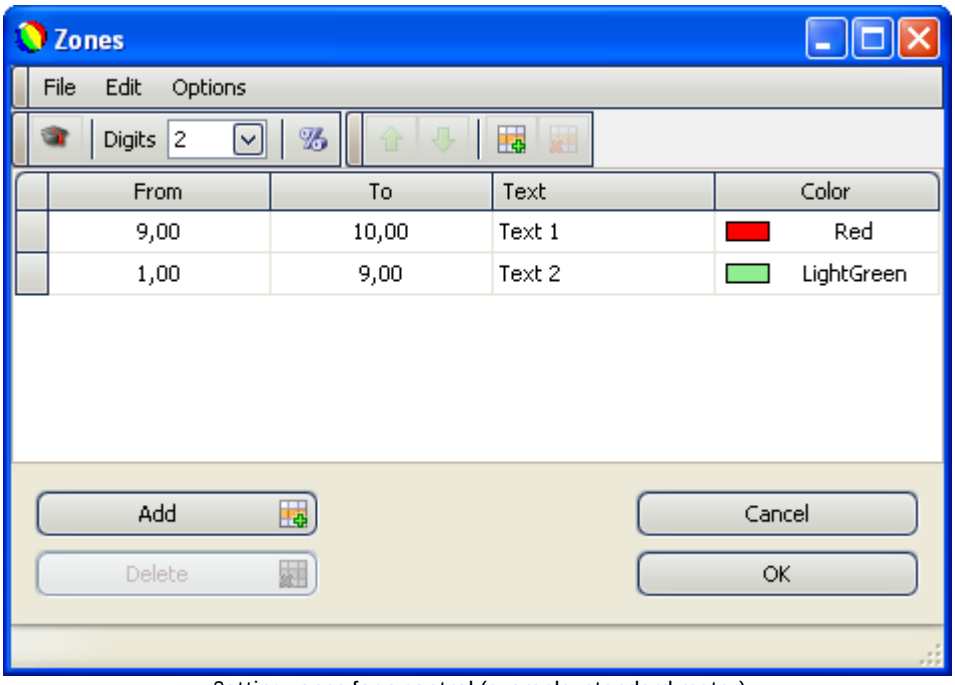
In this dialog, it is possible to, for example, define for a *Listbox* the values which are set by an entry, or the value ranges of various Widgets can be assigned colors, as displayed below:

Example of a standard meter

To open the dialog, proceed as follows:

- Open the properties of the Widget.
- Click on the button ... in the box *Zones*

The dialog Zones is called, which appears as shown below:



Setting zones for a control (example: standard meter)

The dialog's structure and the displayed columns 850 depend on:

- the Widget
- the Widget property Write protected 843
- the option Advanced display 851

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 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 851 of the range.

Adding or deleting a zone

To create the zone, there are three options: select the corresponding column and

- click on one of the two buttons
- open the Edit menu and select the command Add row or Delete row
- open the row's context menu and select the command Add/Delete

Moving the zones

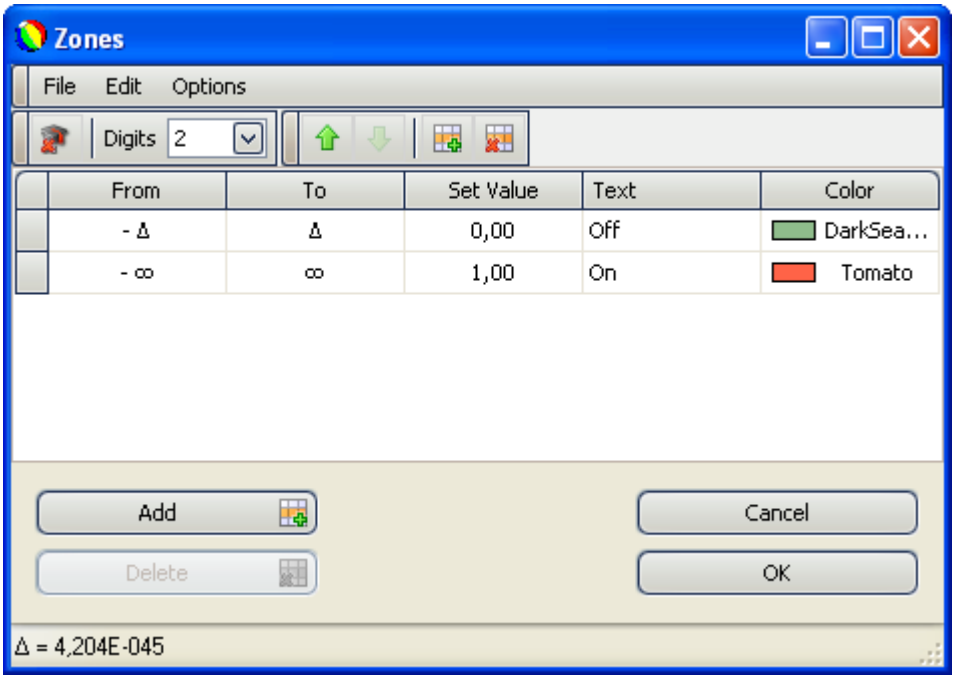
The top zone has top priority 851.

To change the order, there are three methods: Select the column and

- click on one of the two buttons
- open the Edit menu and select the command Move row up/Move row down
- open the context menu of the row and select the command Move row up/Move row down

Infinity and zero

In special cases, symbols are used for *Zero* and *Infinity*. Below, the zones for a switch are shown with Advanced display 851.



Zones of a switch with Advanced display

The switch should be *Off* when near *Zero* and for all other values *On*.

8.5.4.1.1 The columns and their interdependencies

Column *Set Value* or *From/To*

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 851 is activated, the columns Set Value, From and To are displayed if present.

Interdependencies

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.

Automatic adaptation

- If you have activated Advanced display 851, this automatic adaptation is deactivated. This means you can also set values outside of the range.
- Make note that because of the above fact, the Widget may get into an inoperable state.

Example: *Listbox*:

The Widget has four zones:

Set Value	Text	From (not displayed)	To (not displayed)
0	Item_1	-0,5	0,5
1	Item_2	0,5	1,5
2	Item_3	1,5	2,5
3	Item_4	2,5	3,5

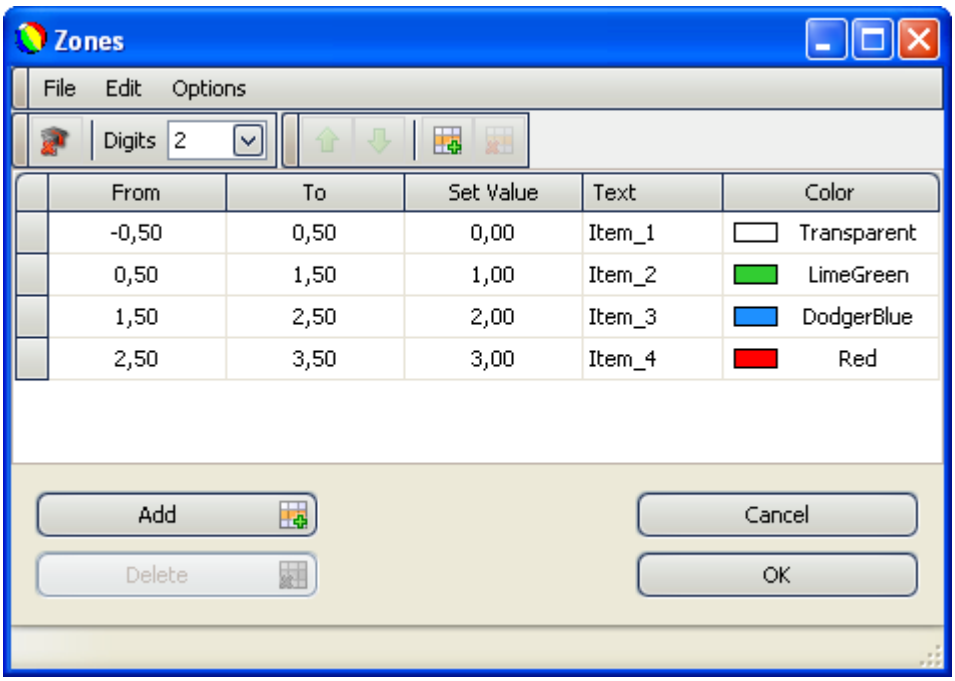
If you change the value sin the column Set Value and have not activated the Advanced display, the other values adjust themselves.

Set Value	Text	From (not displayed)	To (not displayed)
0 > 1	Item_1	-0,5 > 0,5	0,5 > 1,5
1 > 2	Item_2	0,5 > 1,5	1,5 > 2,5
2 > 3	Item_3	1,5 > 2,5	2,5 > 3,5
3 > 4	Item_4	2,5 > 3,5	3,5 > 4,5

8.5.4.1.2 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*.

8.5.4.1.3 Menu options

Advanced display

By default the Zones dialog has a simple display. The structure is adapted to the normal operating style for the Widget. This dialog varies according to the Widget used.

The Advanced display is very valuable for complex use of the Widgets.

- All columns are displayed. Complex use of Widgets is possible, e.g.:
  - the *Listbox*, which is to be used as input and output control,
  - switch with more than two zones
- Zones depend on the values of variables
  - The values of the columns *From*, *To* and *Set value* can be set in response to a variable value

To activate the Extended Display, the following ways are possible:

- Click on the button
- Open the Options menu and select the command Advanced display
- Open the context menu in empty space in the table, and select the command Advanced display

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. There are two ways to activate this Percentage entry:

- Click on the button %
- Select the item Percentage in the Options menu

Numeric Zone / Textual Zone

Some Widgets, for instance *Tables* belonging to the group *Automotive*, *Industrial* and *Designer* can display texts or numerical values.

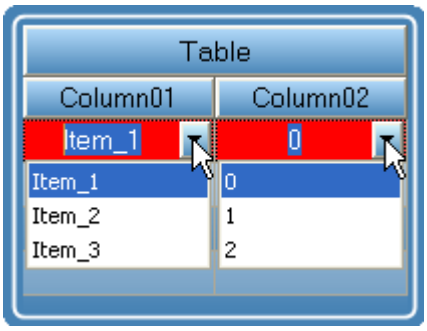


Table with numerical and textual zones

In order to activate the Numeric or Textual zones display, two ways are available:

- Click one of the two buttons 1 2 3 A B C
- Select the item Numeric Zone/Textual Zone in the Options menu

Digits

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*.

This setting only refers to the display of numbers in this dialog and has no effect on the Widget.

To change the number of decimal places, there are three methods:

- Click the button for the box Digits 2 and either select an entry or enter a value.
- Open the menu Options and select the command Digits, click on the box's button and either select an entry or enter a value.
- Open the table's context menu, select the item Digits and select one of the entries.

The following values can be entered: 0, 1, 2, ..., E0, E1, E2, ...

8.5.4.2 Linking commands with switches

Certain Widgets can be linked with *commands* which are carried out at state transitions (called *events*).

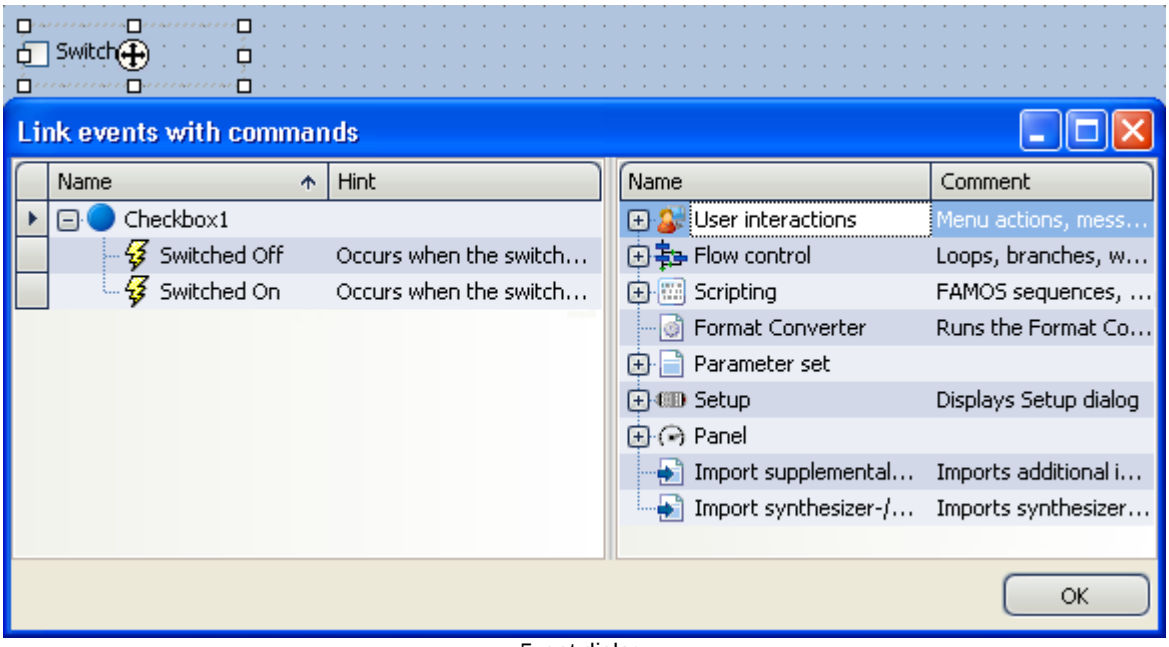
As an example, the two events ( ) belonging to a *Switch* from the group *Standard*:

- *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 843 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 Command Sequence).

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* 1702)
- the Panel page *Report* opens

8.6 Curve Window

8.6.1 Introduction

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 possesses the conventional characteristics of windows in Windows applications.

The curve window has an extensive menu offering a multitude of options for display and graphical evaluation of waveforms. The virtual multi-tasking operation in Windows allows the curve window to always display the most recent data. Curve windows can be operated with the mouse, the keyboard, or a combination of both.

The following offers a short overview of the main features of the curve window:

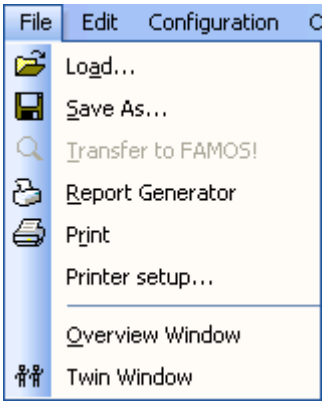
	Description
scaling	Automatic or manual scaling of the x- and y-axes
Zoom	allows detailed viewing of a curve section
Unzoom	returns to entire curve window display
Measurement values	Two independent measurement cursors are available, which display the x- and y-values of the curves, as well as value differences 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.
Report Generator	Enables the user to design measurement reports: text, curve graphs and other graphics can be combined and positioned as desired. The entire page can be printed as a data sheet.
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.

	Description
	Various options are available for waterfall display, e.g. the angle of view.
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.

Note

Some screen shots appearing in the text were adopted from older imc FAMOS versions.
Therefore, the appearance of the user interface as depicted in this manual may be slightly different from the actual appearance of the user interface of your system.

8.6.2 "File" menu



	Menu item	Description
	Load 856	Loads a curve window configuration from a file.
	Save as 858	The curve window configuration (e.g. the axis scaling) is saved in a file.
	Transfer to imc FAMOS! 859	The waveform in the curve window is transferred to imc FAMOS.
	Report Generator 859	Calls the Report Generator. The report's layout can have your own personal, creative design.
	Print 859	Prints out the curve window.
	Printer setup 861	Sets up the printer used for printing out the curve window's contents.
	Overview Window 862	Displays an overview window.
	Twin Window 863	Generates a new curve window with the same appearance.

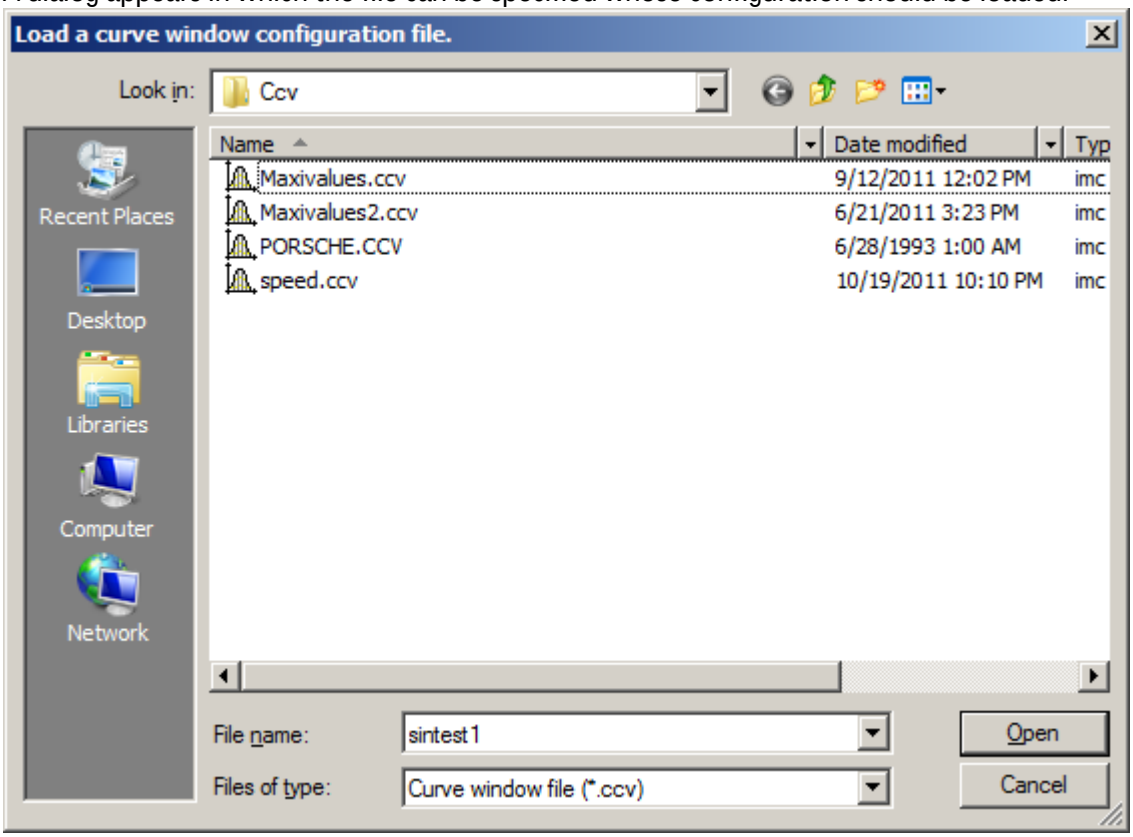
8.6.2.1 Load Configuration

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.

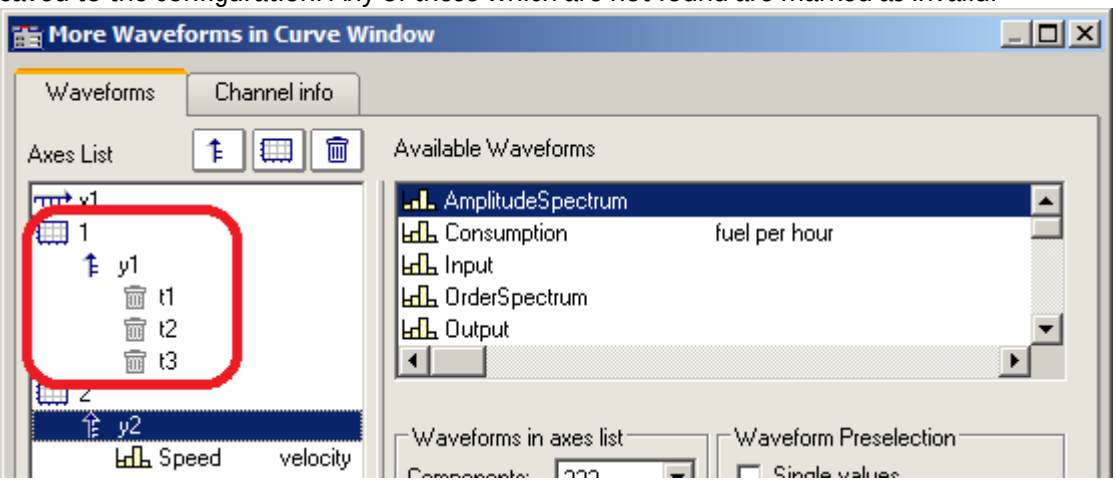


- Select file format "CCV".
- Select the desired file and end the dialog by clicking on *Open*.

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 1072, you can directly transfer a configuration from one curve window to another.

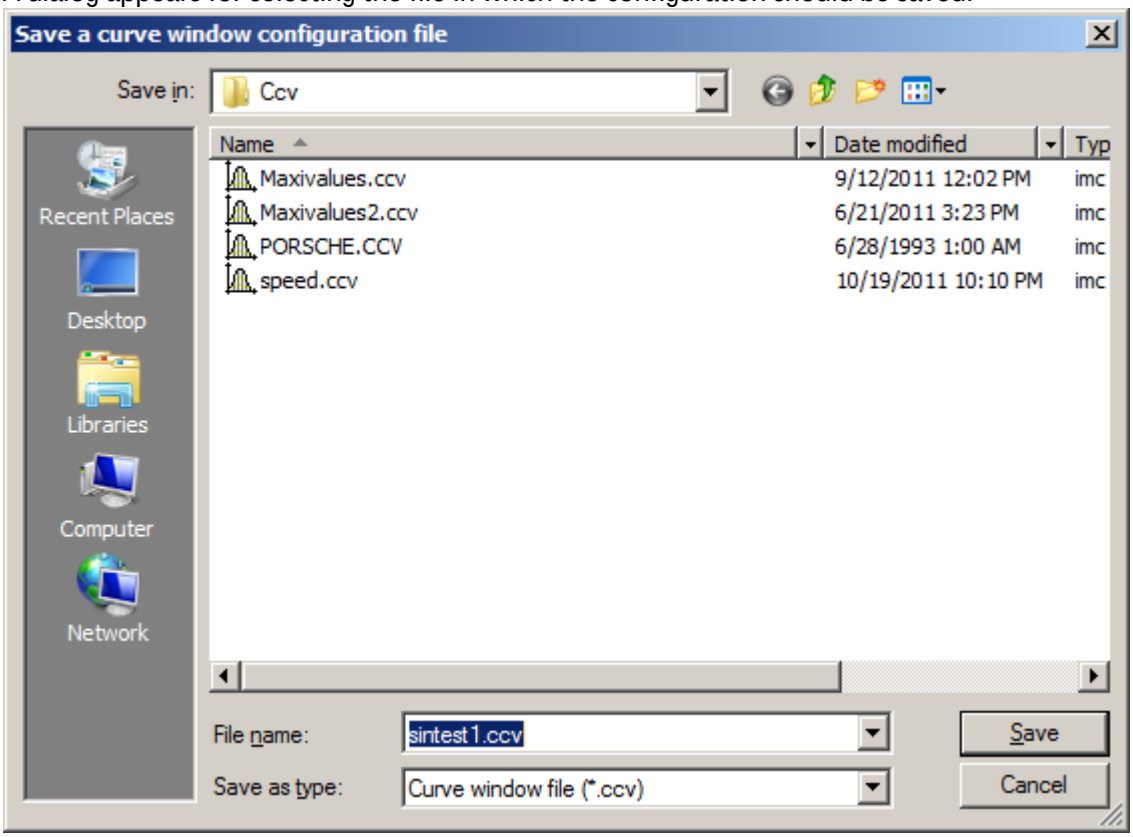
8.6.2.2 Save Configuration

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 1072 you can directly transfer a configuration from one curve window to another.

8.6.2.3 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* 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.

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 870" for more information.
- Waveforms can also be transferred to imc imc FAMOS using Drag and Drop 1072.

8.6.2.4 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.

8.6.2.5 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 902".
- 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 859" 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 1044", "Printer Setup 861"

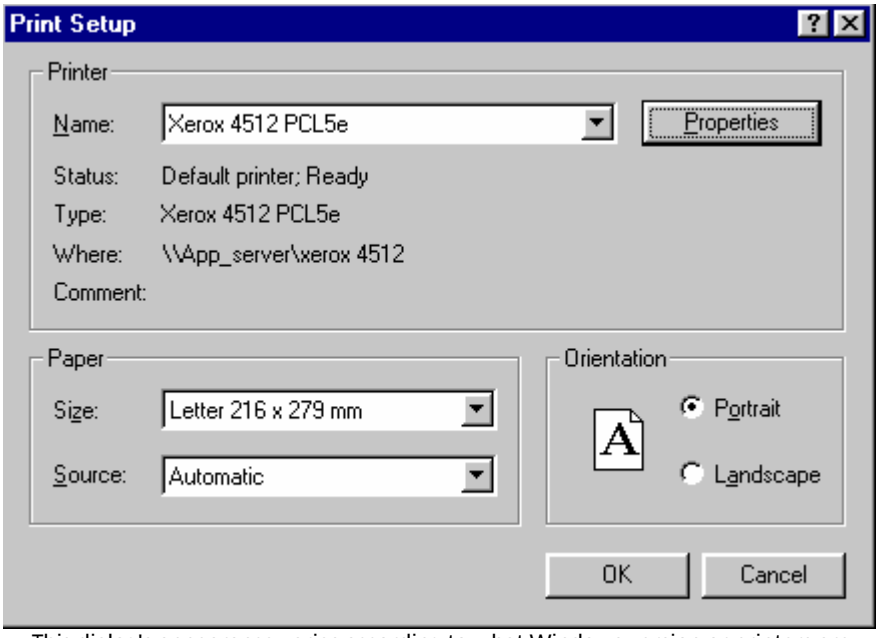
8.6.2.6 Printer Setup

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.

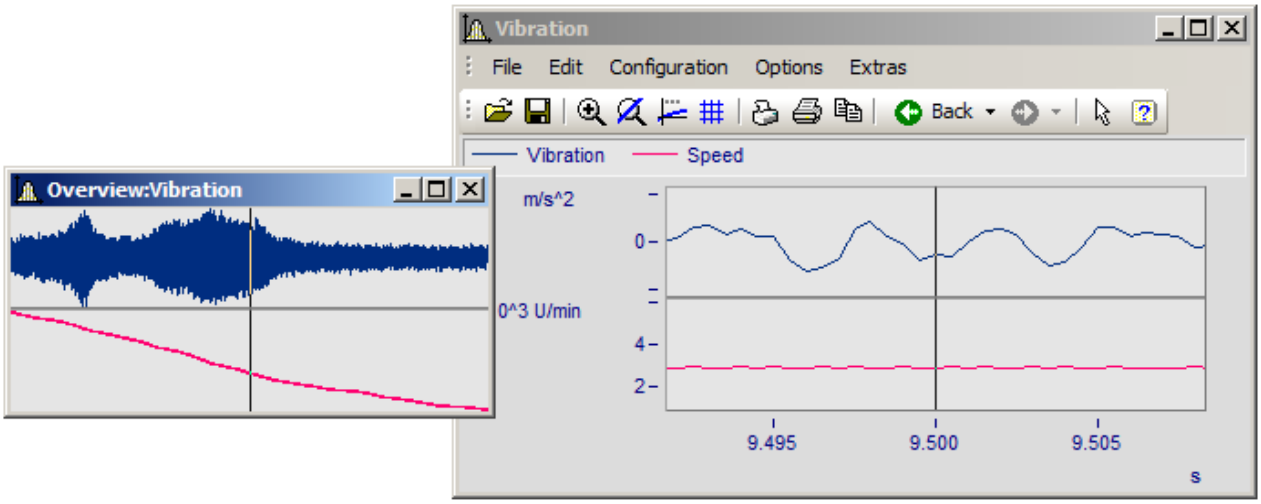
The settings apply to all of the computer's imc curve window's and remains operative upon the program's next start.

8.6.2.7 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.

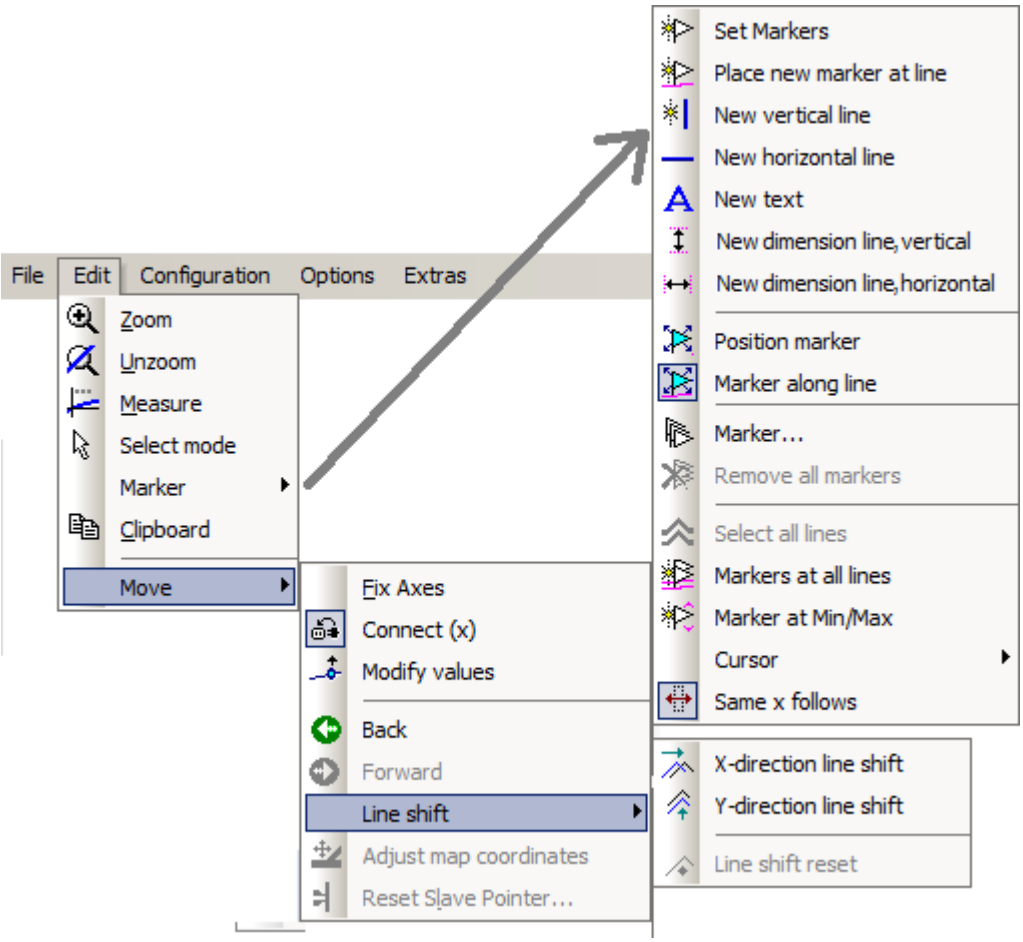
8.6.2.8 Twin Window

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 927) darstellen.
- Display of a Waterfall graph in 3D, or a color map.
- Simultaneous display of the exact same data both in an overview and zoomed.

8.6.3 "Edit" menu



	Menu item	Description
	Zoom 865	Enlarges a section of the curve window.
	Unzoom 866	Displays the entire curve.
	Measure 867	A measurement value window and measurement cursors for measuring curves are displayed.
	Select-mode 881	In this mode, it is possible to select and edit legends, coordinate systems, axes, curves and markers.
	Marker 882	Proceeds to the list of marker functions.
	Clipboard 902	Copies the graphics in the curve window into the Windows clipboard.
	Move 904	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' 1073.

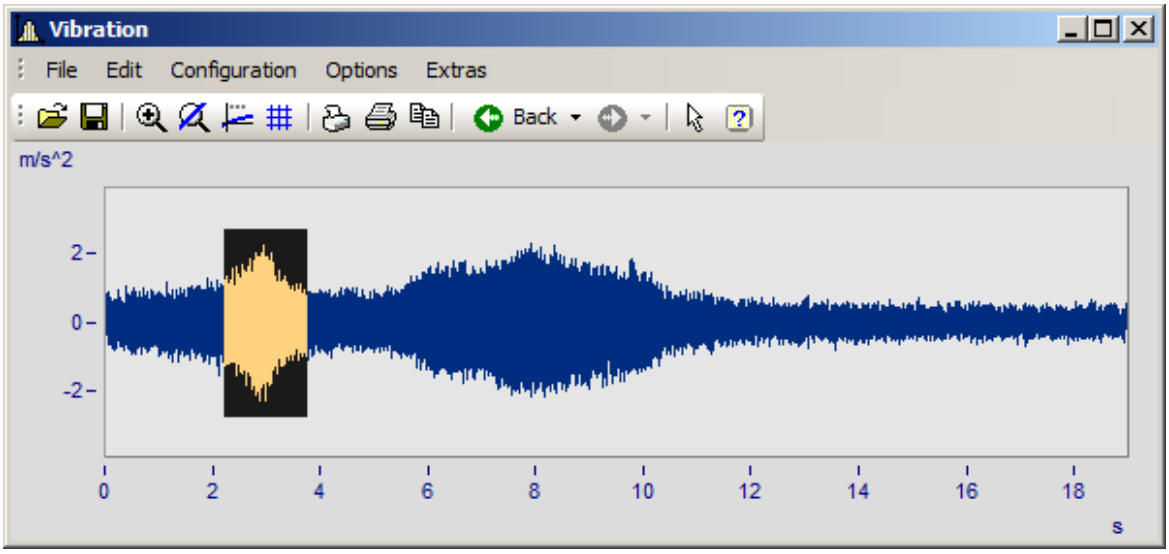
8.6.3.1 Zoom

Function

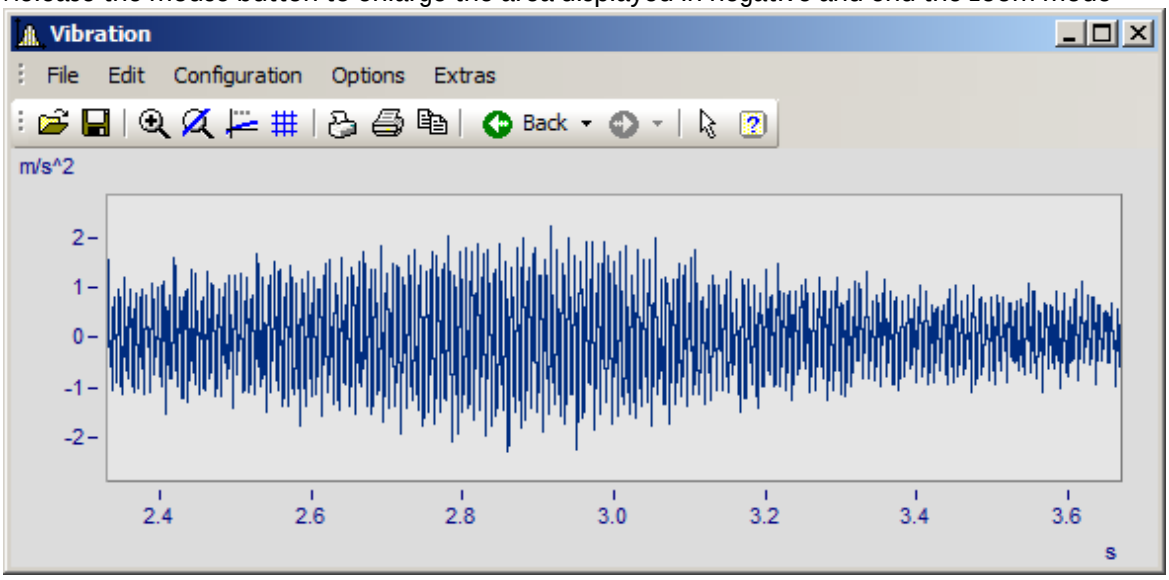
Any region of a displayed curve can be enlarged for viewing (zoomed). Zooming can be done in both x- and y-directions at the same time. The zoomed region is a rectangular area within the coordinate system, with size and position determined by the user.

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* 1053 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* 1005). 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* 910 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.

8.6.3.2 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.

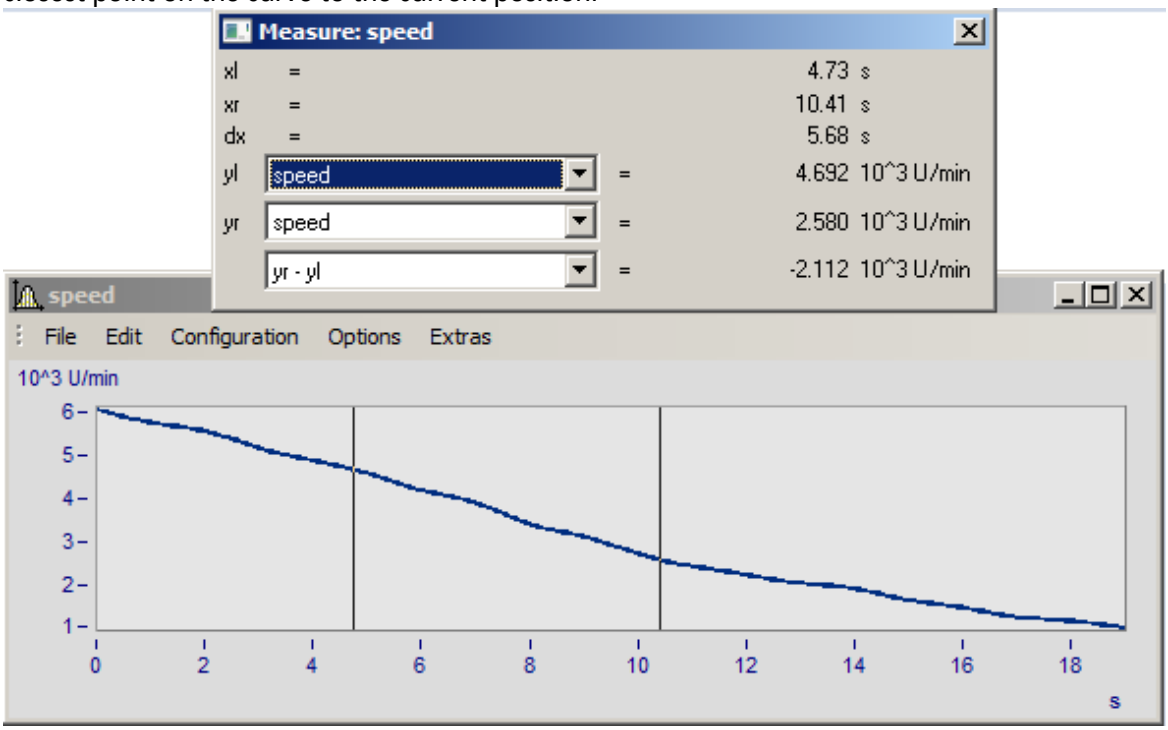
8.6.3.3 Measure

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

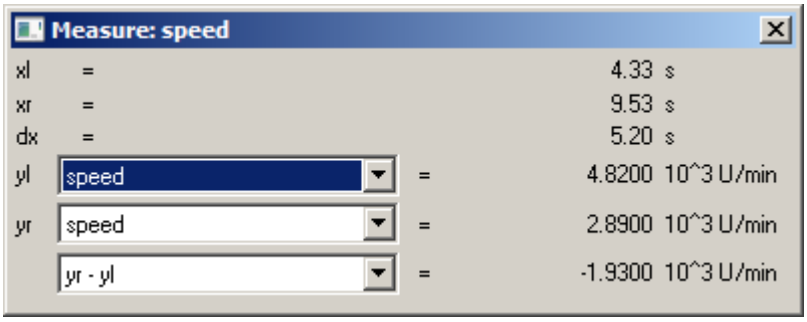
- The crosshairs may cross each other.
- In normal time displays, the horizontal line of a measurement cursor is visible only as long as the curve lies within the coordinate system and is defined for the x-coordinate of the measurement cursor.
- Zooming in XY or locus plots may result in the measurement 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* 1018 menu.
- The horizontal measurement cursor can be switched off in the *Options/ Presettings* 1053 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.

8.6.3.3.1 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.

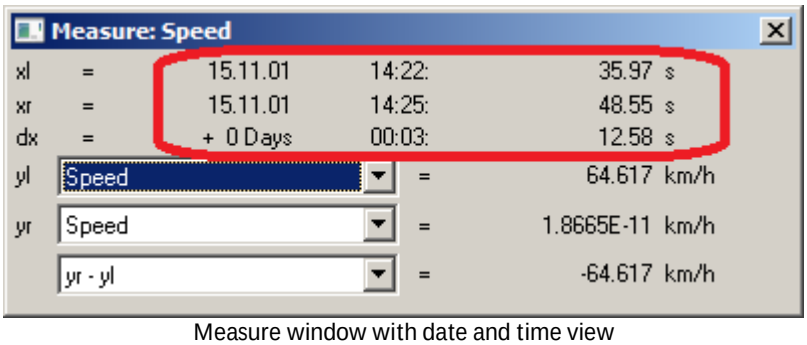


In the selection lists *yl* and *yr*", a waveform displayed in the curve window can be selected for measurement, which is accomplished using the measurement cursors. The list box at the bottom of the dialog contains the following entries as available choices of calculated values:

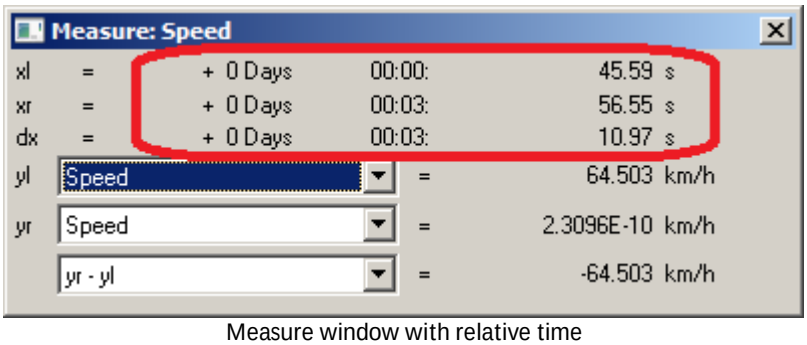
Symbol	Definition
xl	x-coordinate of left measurement cursor
xr	x-coordinate of right measurement cursor
dx	Difference between the x-coordinates, xr-xl
xr/xl	Quotient of the x-coordinates of the measurement cursors
yl	y-coordinate of the specified curve at the intersection with the left measurement cursor
yr	y-coordinate of the specified curve at the intersection with the right measurement cursor

Calculations	Definition
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

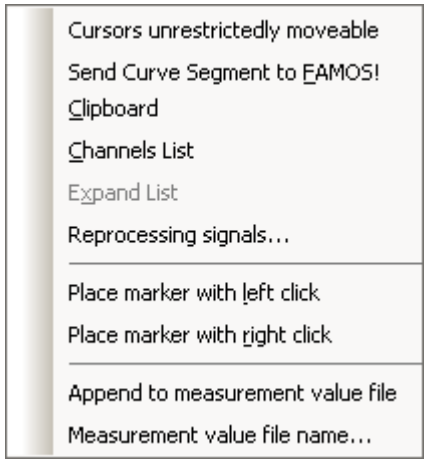


Measure window with relative time

A negative sign before data in *days/hours/minutes* and *seconds* pertains to the entire time declaration, not only to the days.

8.6.3.3.2 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. 1057.. 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

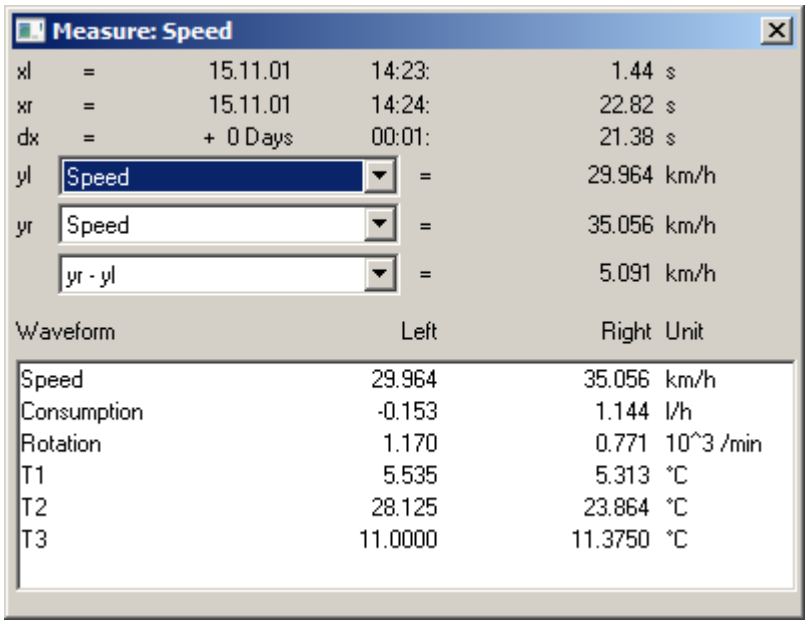
More about Transfer options 1057

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.

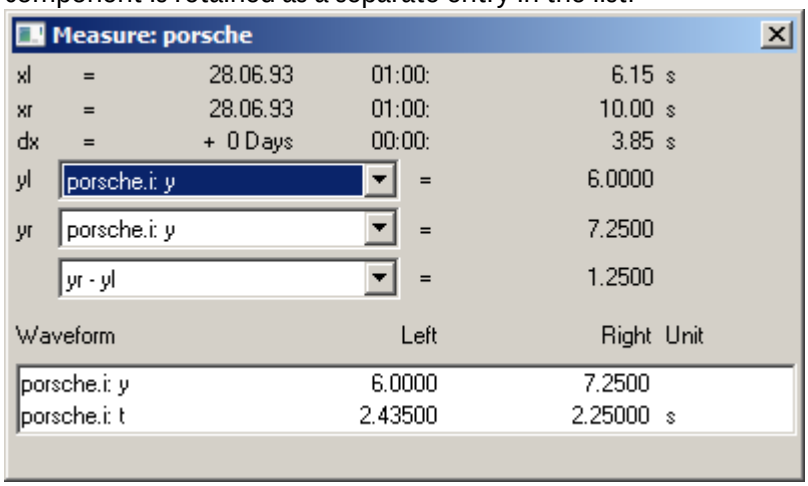
Remarks

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 874.

Place marker with left click; Place marker with right click

Set a marker at the measurement cursor positions. Subsequently, a dialog for marker settings 892 appears.

Reference

More about "Markers" 882 and "Marker Definition" 892

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.

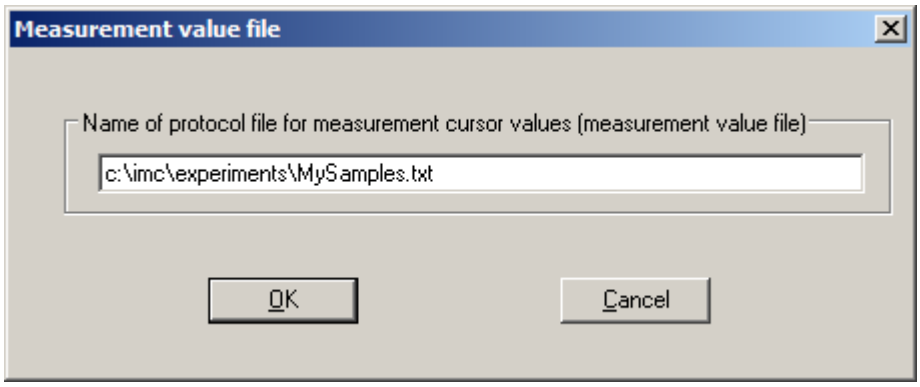
By means of an entry in the Registry, it can be made visible again:

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

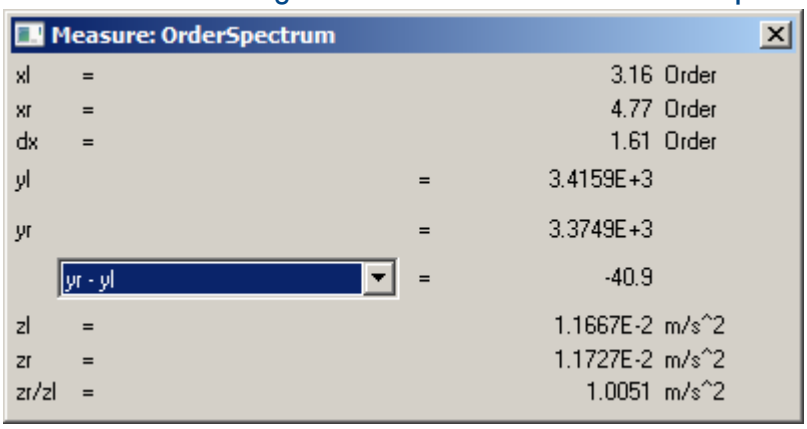
This menu item calls a little dialog in which the name of the measurement data file can be edited. The setting for this filename is valid globally for all measurement value windows.



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.

8.6.3.3.3 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.

8.6.3.3.4 Reprocessing signals

Function

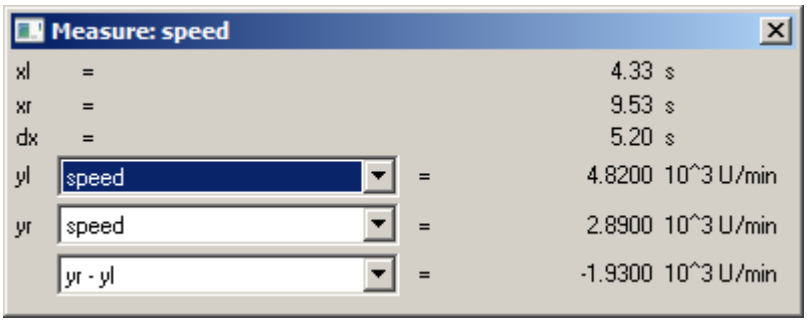
Here it is possible to edit, for instance, to smooth clip peaks from, simple signals. Thus, invalid readings can be removed from a signal and replaced with plausible ones, or distorted readings can be corrected, for example, by eliminating offsets or drifts.

These functionalities are only available off-line (in imc FAMOS).

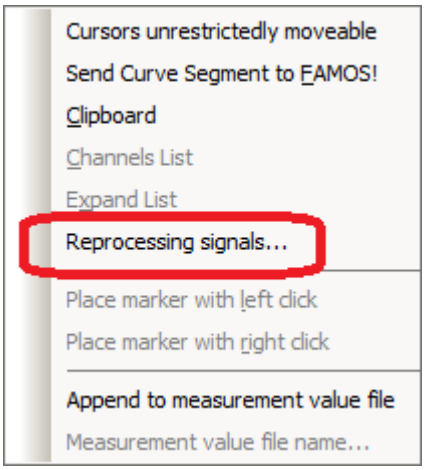
Mouse Operation

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.

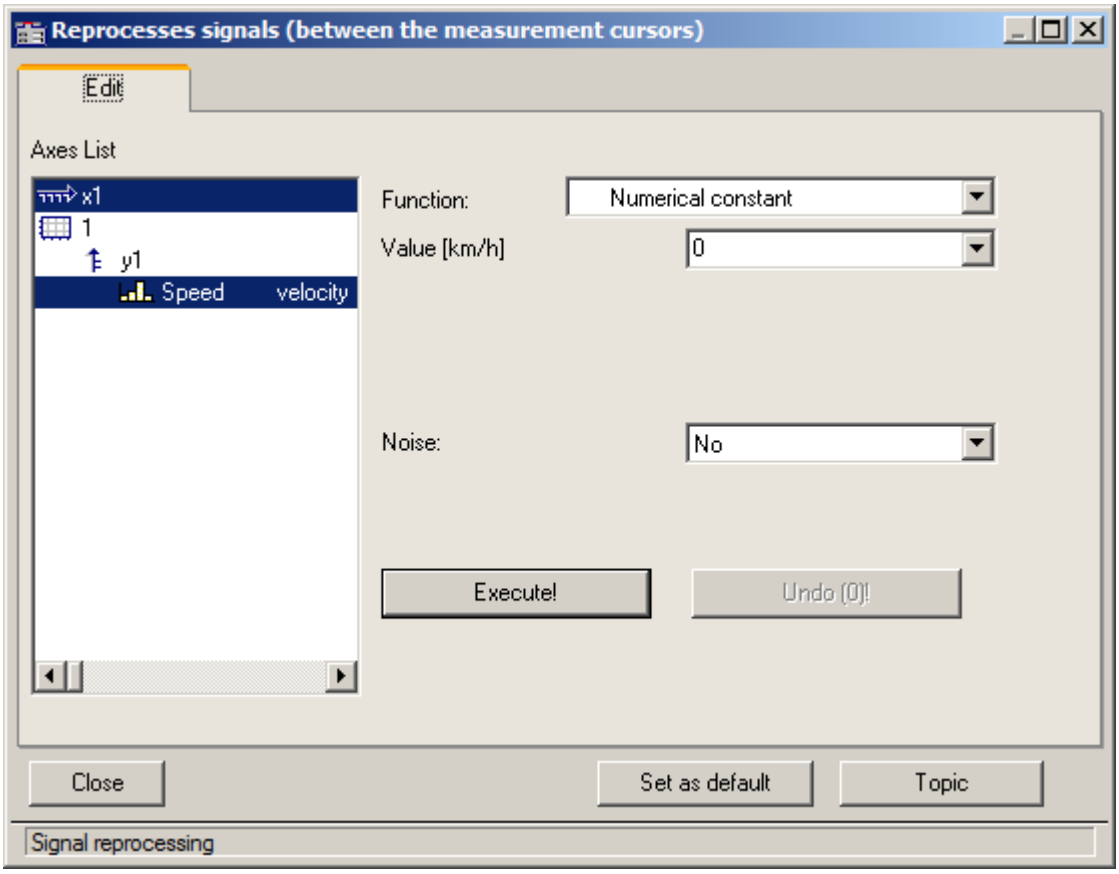
First, the Measurement value window is opened:



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!

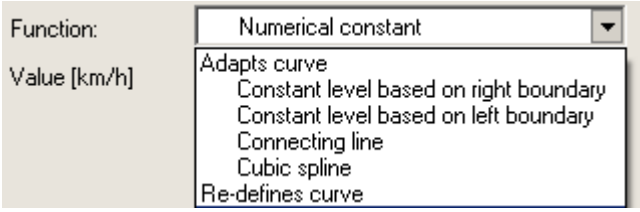
- The dialog for reprocessing signals can only be opened for one curve window at a time. So if you wish to reprocess another curve window's signal also, first close the *Reprocessing signal* dialog in the first window.
- The dialog for reprocessing signals has no *Cancel* button. If the dialog is exited, 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 1043" 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.

8.6.3.3.4.1 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* 878 and *Boundary interval* 879.

Adapt curve

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

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

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

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

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

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.

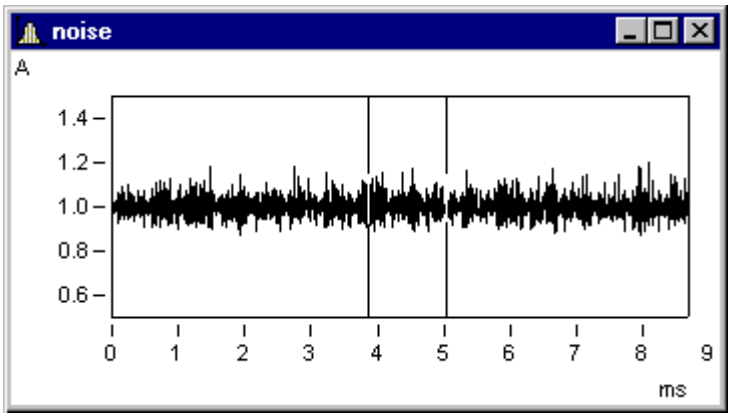
Add Gaussian-distributed noise: A noise signal with Gaussian distribution (normal distribution), having a specified RMS-value, is added to the readings within the delineated region.

8.6.3.3.4.2 Boundary interval

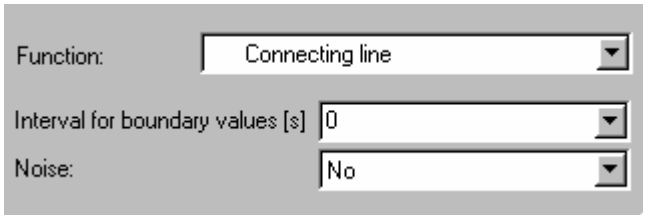
Many functions use the boundary values at the edges of the delineated region; for instance, all values within the delineated region are bridged by a connecting line or spline.

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.

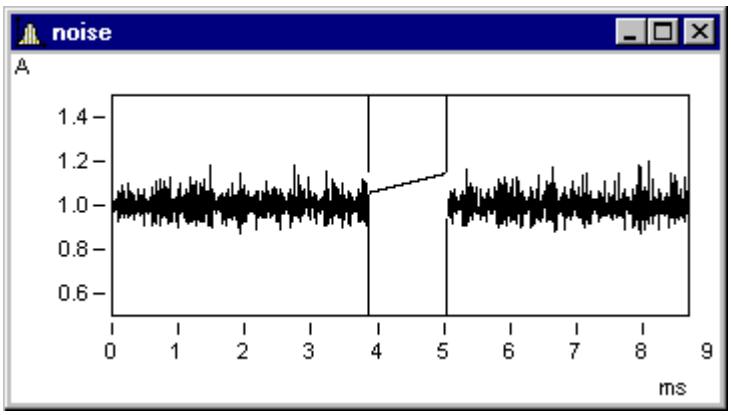
In the example below, the delineated region is to be replace by a straight line (between the boundary values):



In the dialog *Reprocessing signals...*, the following settings are made.

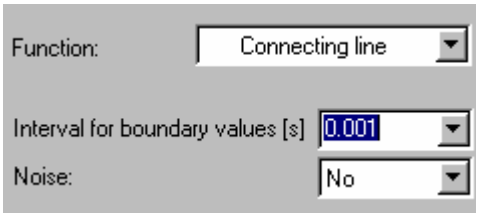


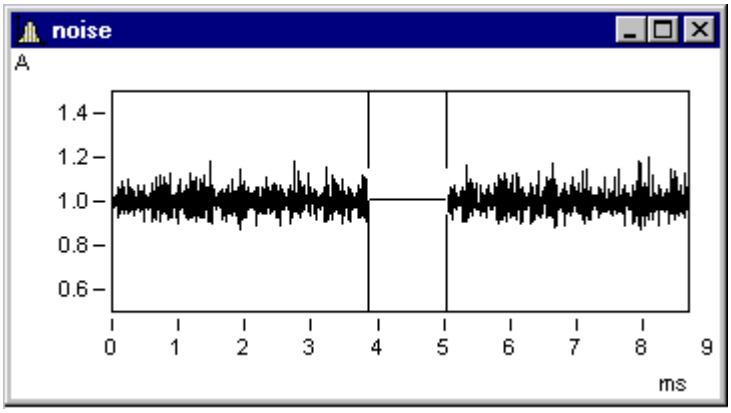
The following connecting line results:



This isn't what one would expect. By chance, the values at the edges (under the cursors) are random peaks.

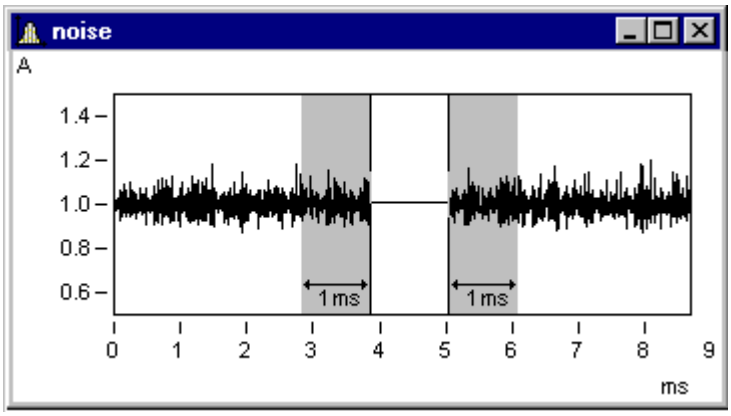
Setting a boundary interval of 1ms (0.001s !) produces proper results:





The boundary interval always is located outside of the delineated region:

Here, the boundary intervals are shaded in grey.



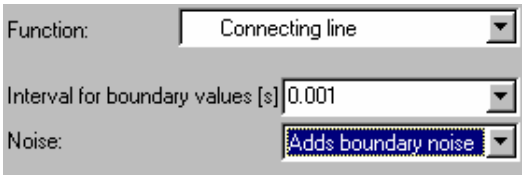
Remarks

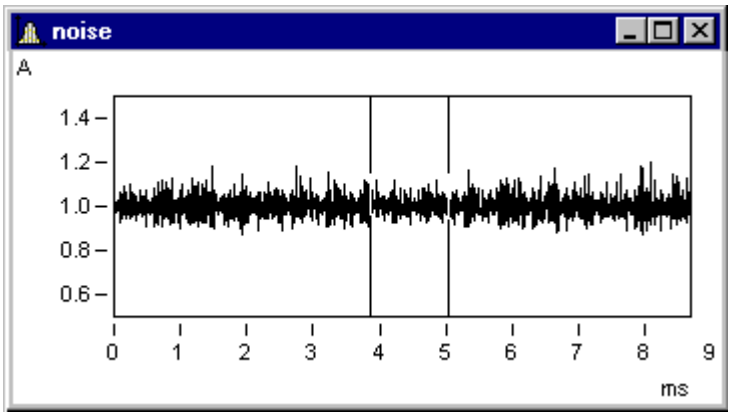
If 0.0 is taken as the boundary interval width, the value at the exact edge is used. Thus, the minimum width of the boundary interval is 1 reading.

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.

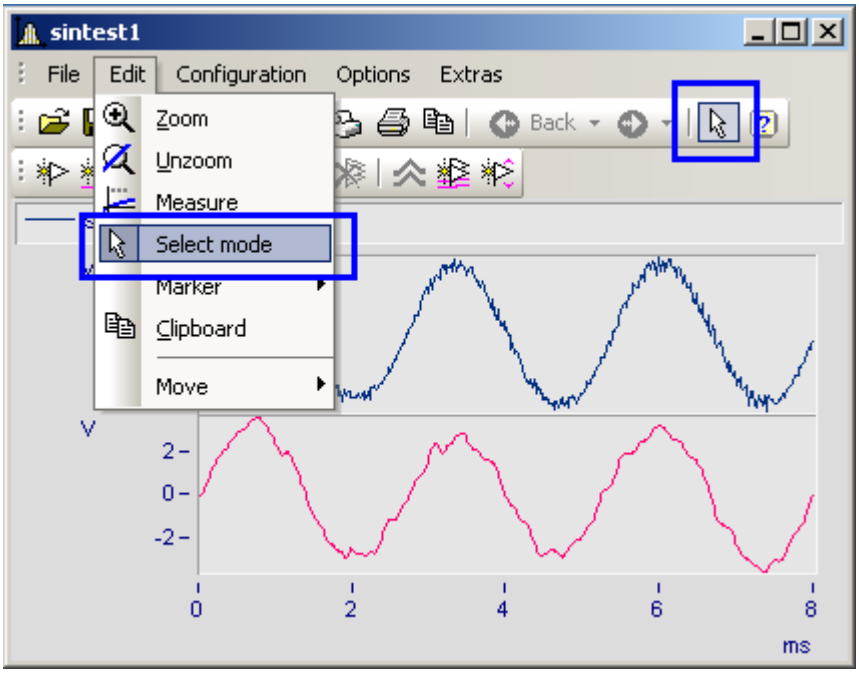
With the following setting, noise from the boundary intervals is added to the delineated region. This produces a realistic signal appearance:





8.6.3.4 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.



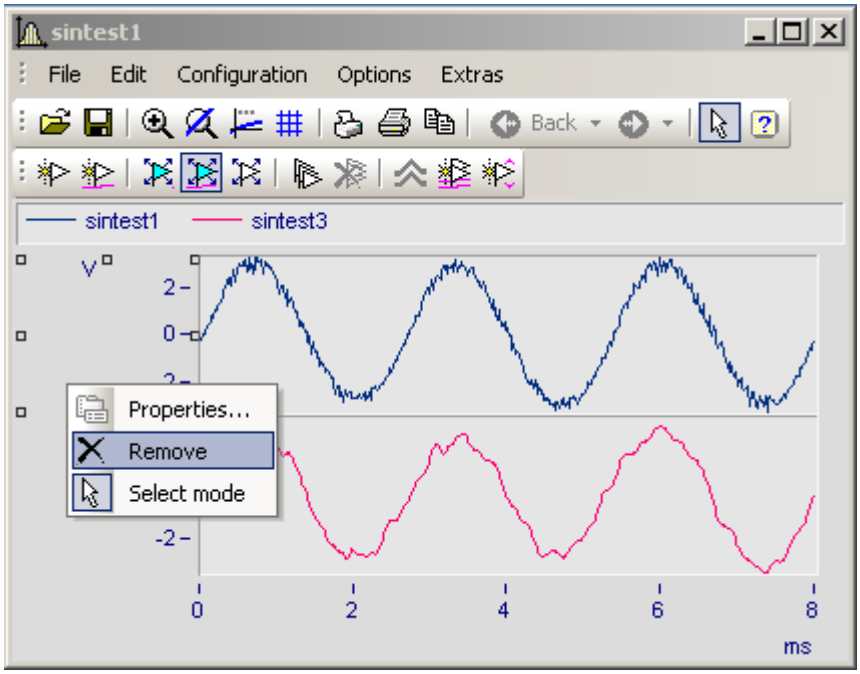
If you are in *Select mode*, the current object is highlighted. This is indicated by the presence of small empty squares at the object's edges. Any lines not selected appear dulled in color.

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.

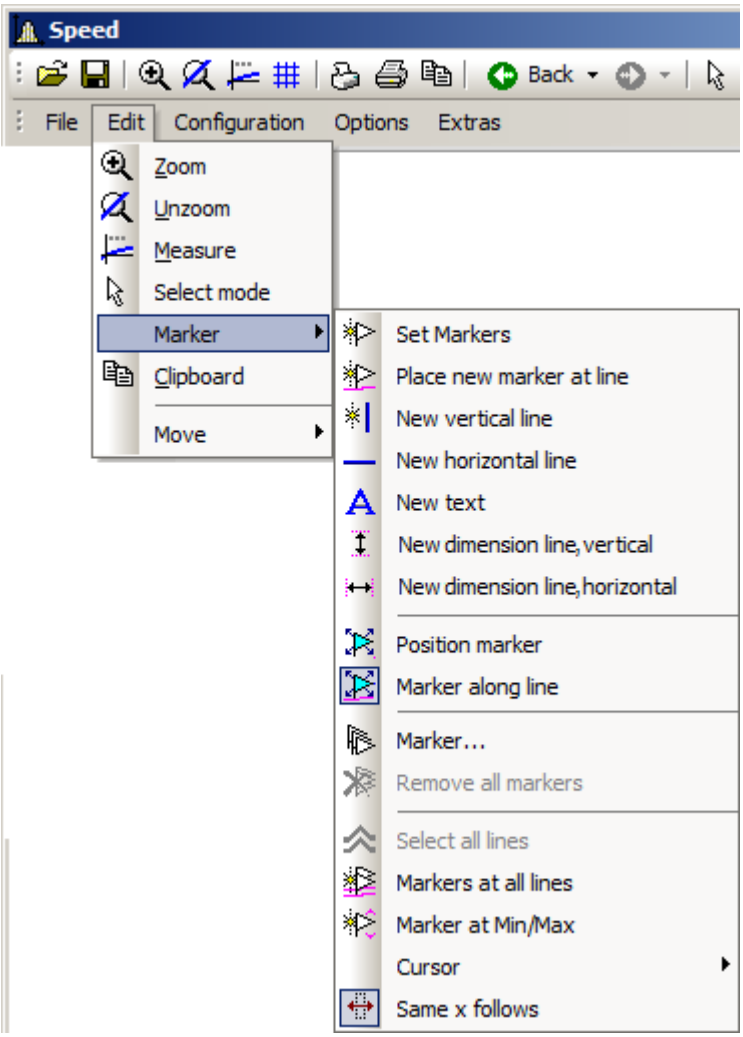
Among other things, the context menu contains the function *Delete*. This is only available if objects which can be deleted are available, such as a second y-axis. If such an object is deleted, any associated objects are also deleted. In the example, the lines in the deleted y-axis would also be deleted.

Example

You select the upper y-axis and deleted it. In the process, the sintest1 lines with which it is associated are also deleted.



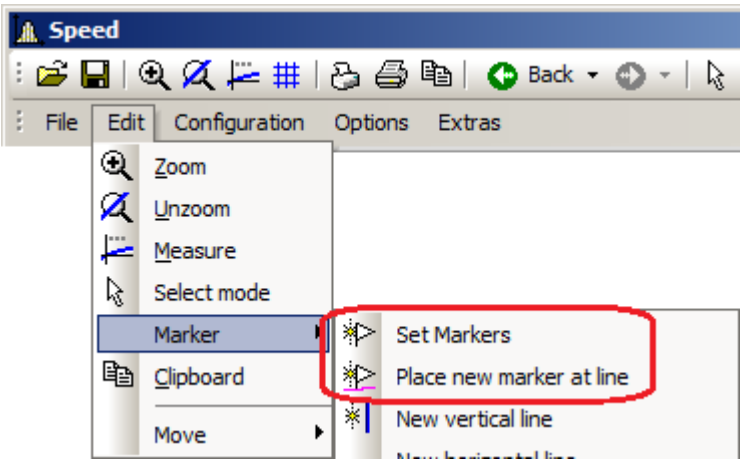
8.6.3.5 Marker



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.

8.6.3.5.1 Set Markers

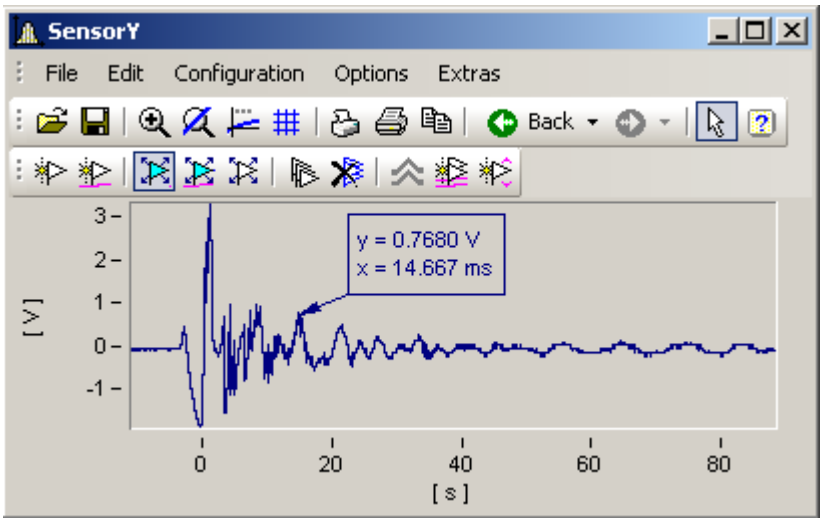


There are two ways of setting a marker, which are accessed either from the Marker toolbar or via the menu item *Edit/ Marker*.

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 892.



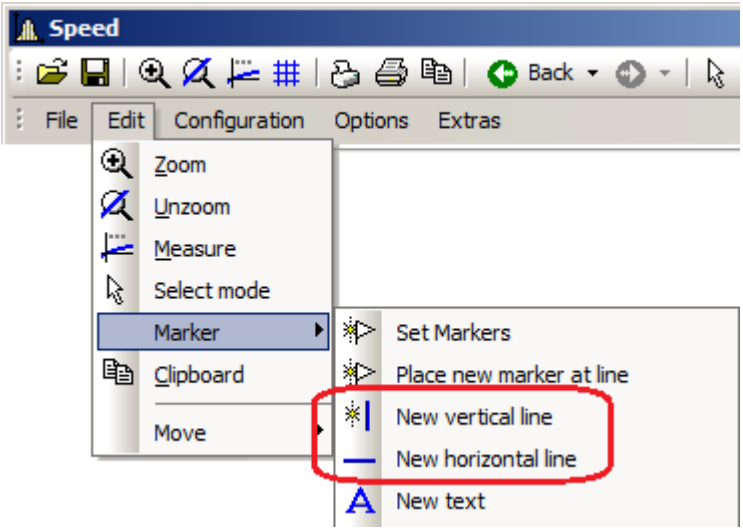
Markers can be placed anywhere in the curve window. Zoom more intensely if a marker should be placed near a line. Accuracy depends on the resolution. To place a marker exactly at a certain point in the curve, select the function *Place new marker at line*.

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.

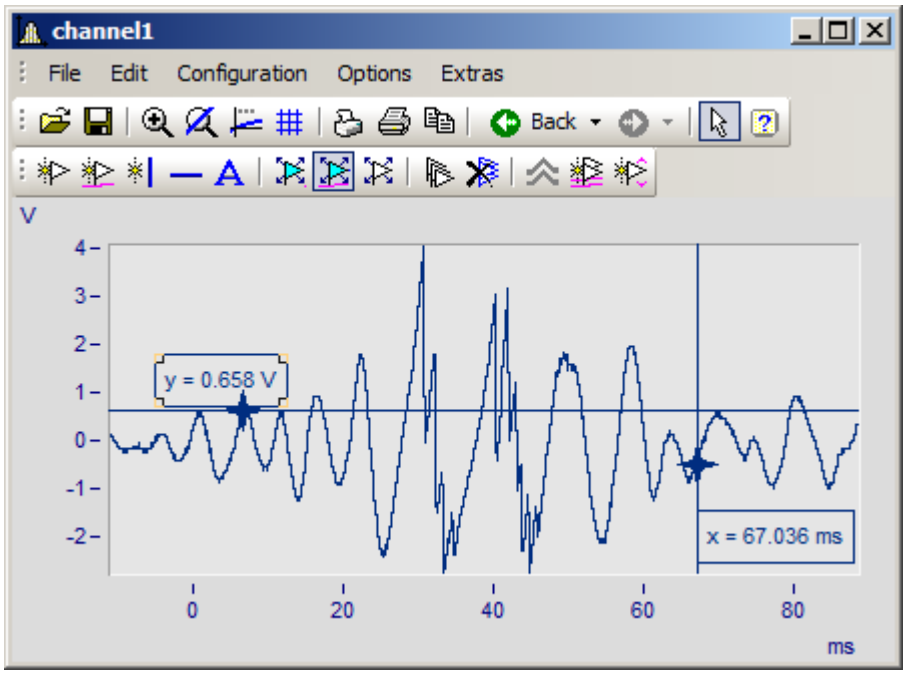
Both modes can be exited by a click of the right mouse button.

8.6.3.5.2 New line

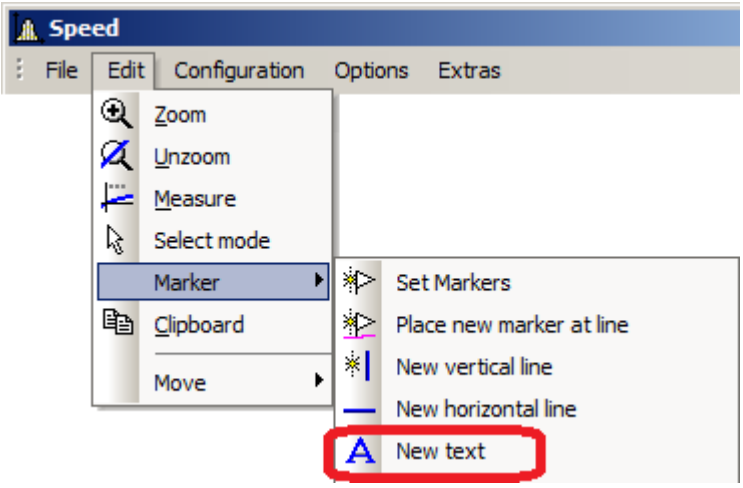


Vertical line new and Horizontal line new

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.



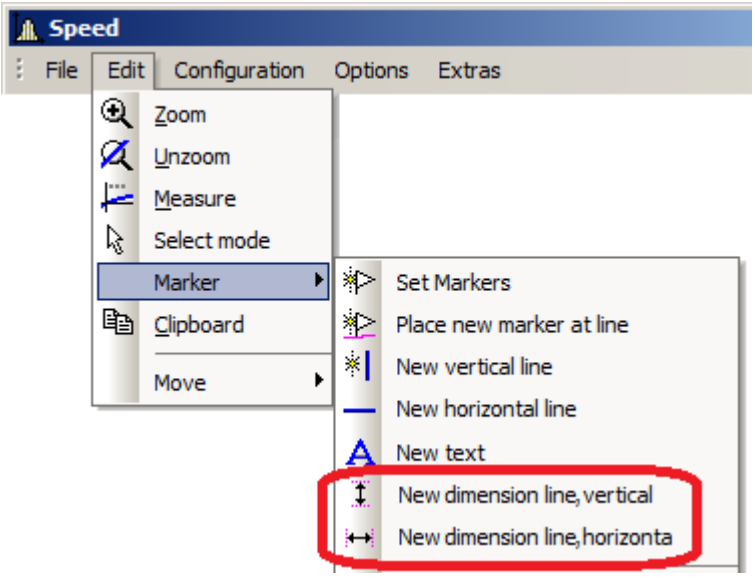
8.6.3.5.3 New text



New text

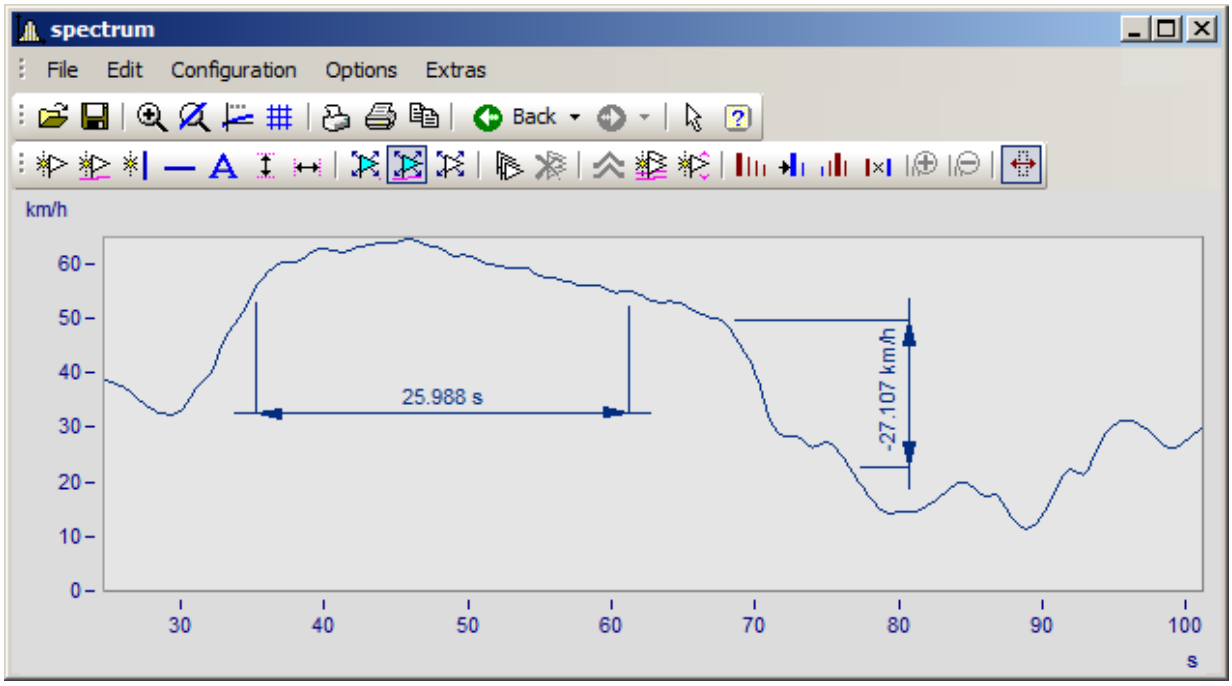
A text box without an arrow can be placed freely. Otherwise, the same settings are available as for markers.

8.6.3.5.4 New Dimension Lines

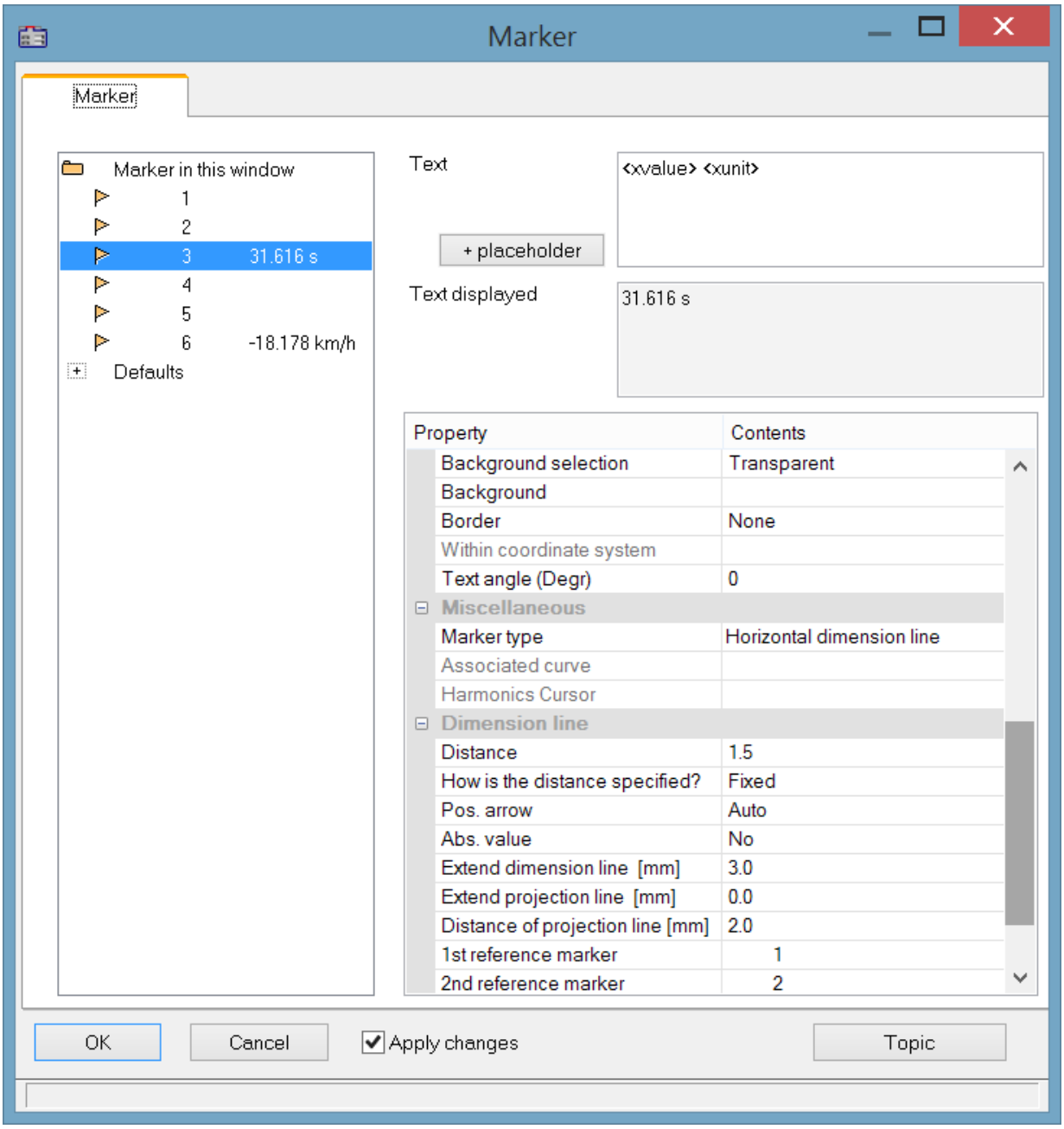


New Dimension Line: vertical and New Dimension Line: horizontal

Select this entry to create a vertical or horizontal dimension line between two points in the curve window where you click the mouse.



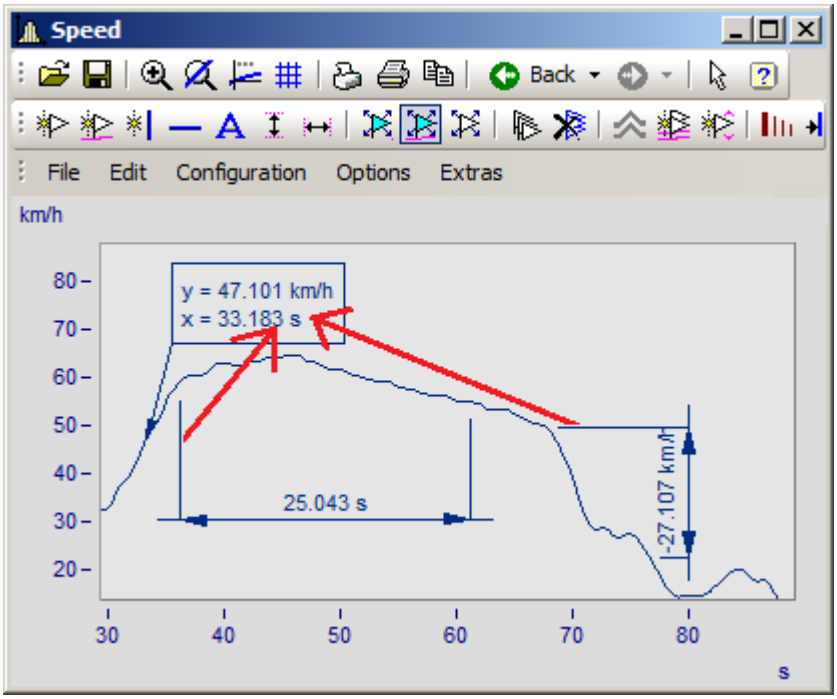
You can change the width of the dimension lines using Drag & Drop. The placement of the legend as well as of the line is positioned by Drag & Drop. Making settings for the markers is done as shown below:



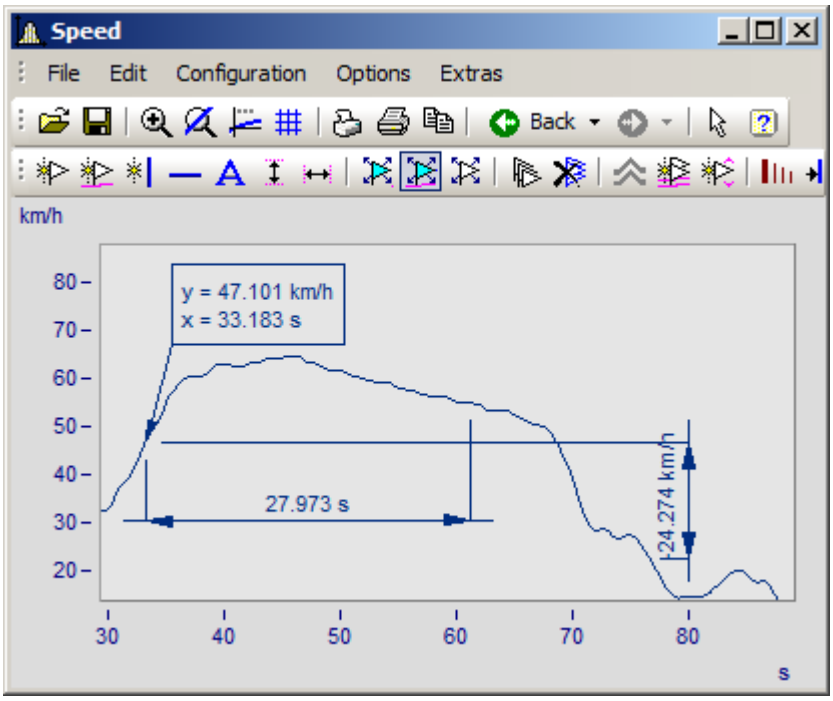
Position

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.

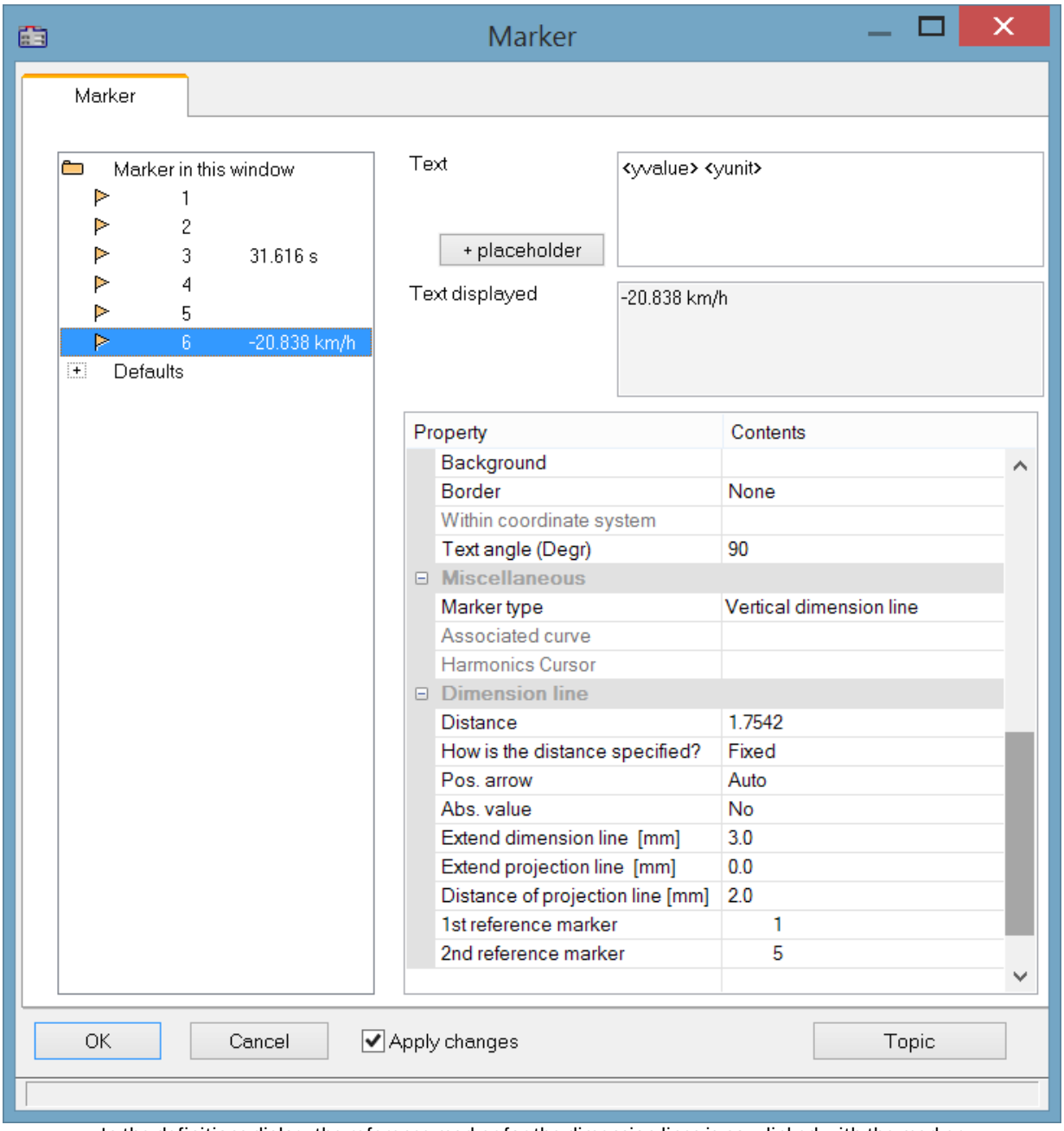
The example below shows how two dimension lines are linked to a marker. When the marker is subsequently moved, the dimension lines are adapted automatically.



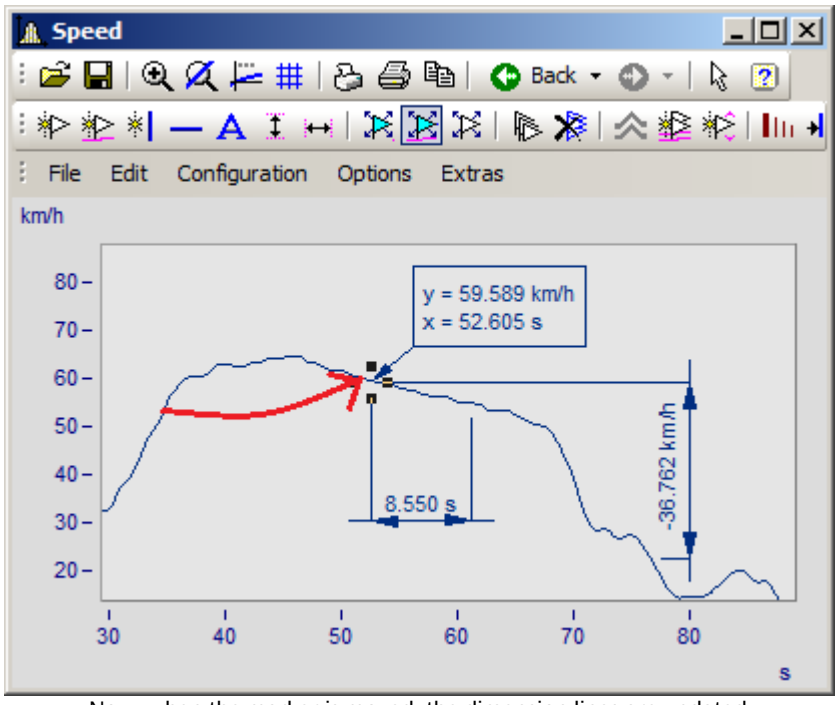
The marker and dimension lines are not yet linked. The extension lines are dragged to the marker.



Now the marker and the dimension lines are linked with the marker.

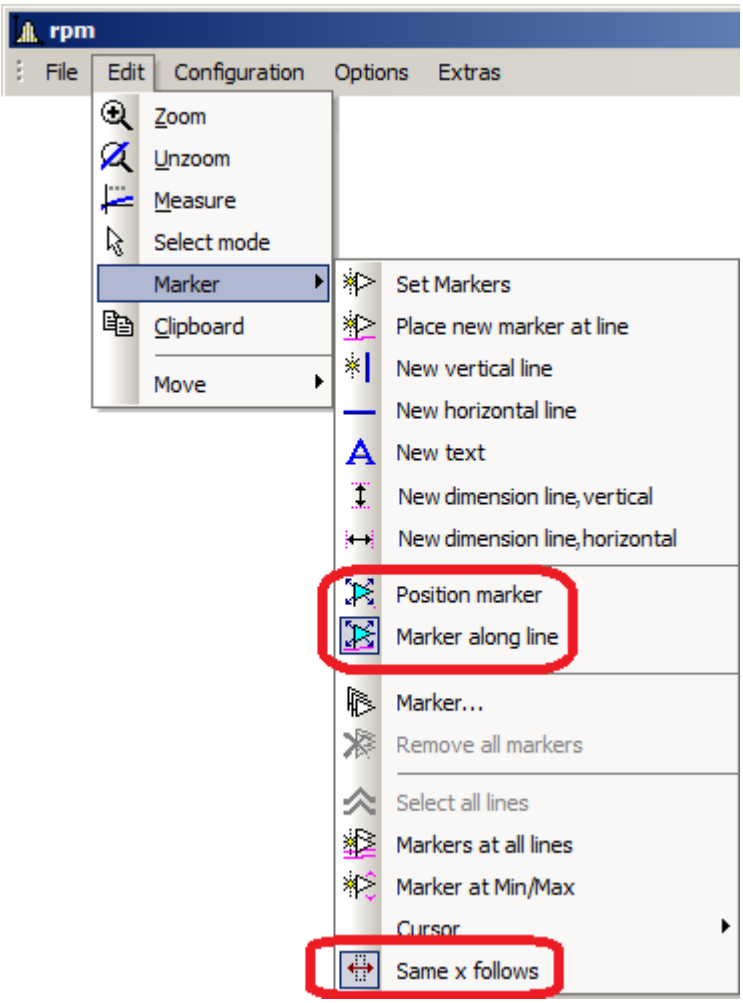


In the definitions dialog, the reference marker for the dimension lines is now linked with the marker.

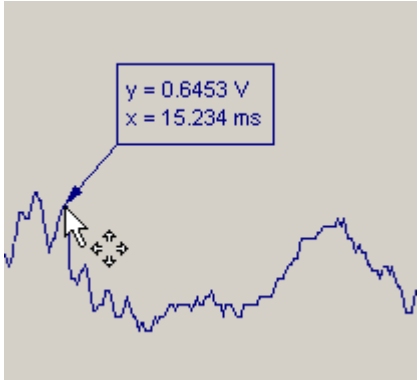


Now, when the marker is moved, the dimension lines are updated.

8.6.3.5.5 Move Markers



Existing markers can be moved subsequently. Both the position and the text box belonging to the marker can be re-positioned. To do this, use Drag&Drop to move the arrowhead point.



There are three different ways to change the marker position :

Position Marker ( )

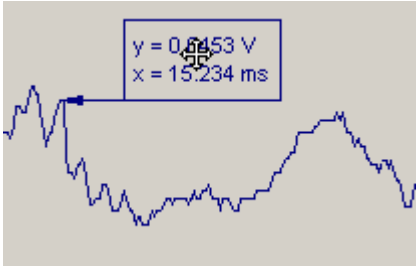
This function lets you move a marker freely, while the marker text is automatically updated to reflect the respective new position.

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 marker ( )

Free movability for the marker, without updating, to reflect the respective new position. 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.



Same x follows ( )

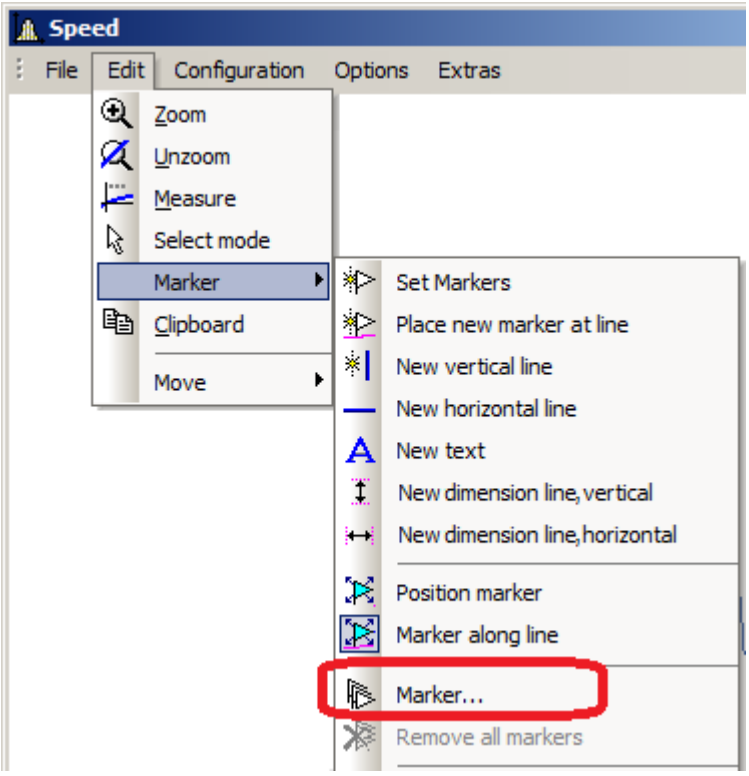
Markers assigned to the same x-position can be moved jointly by means of this function.

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

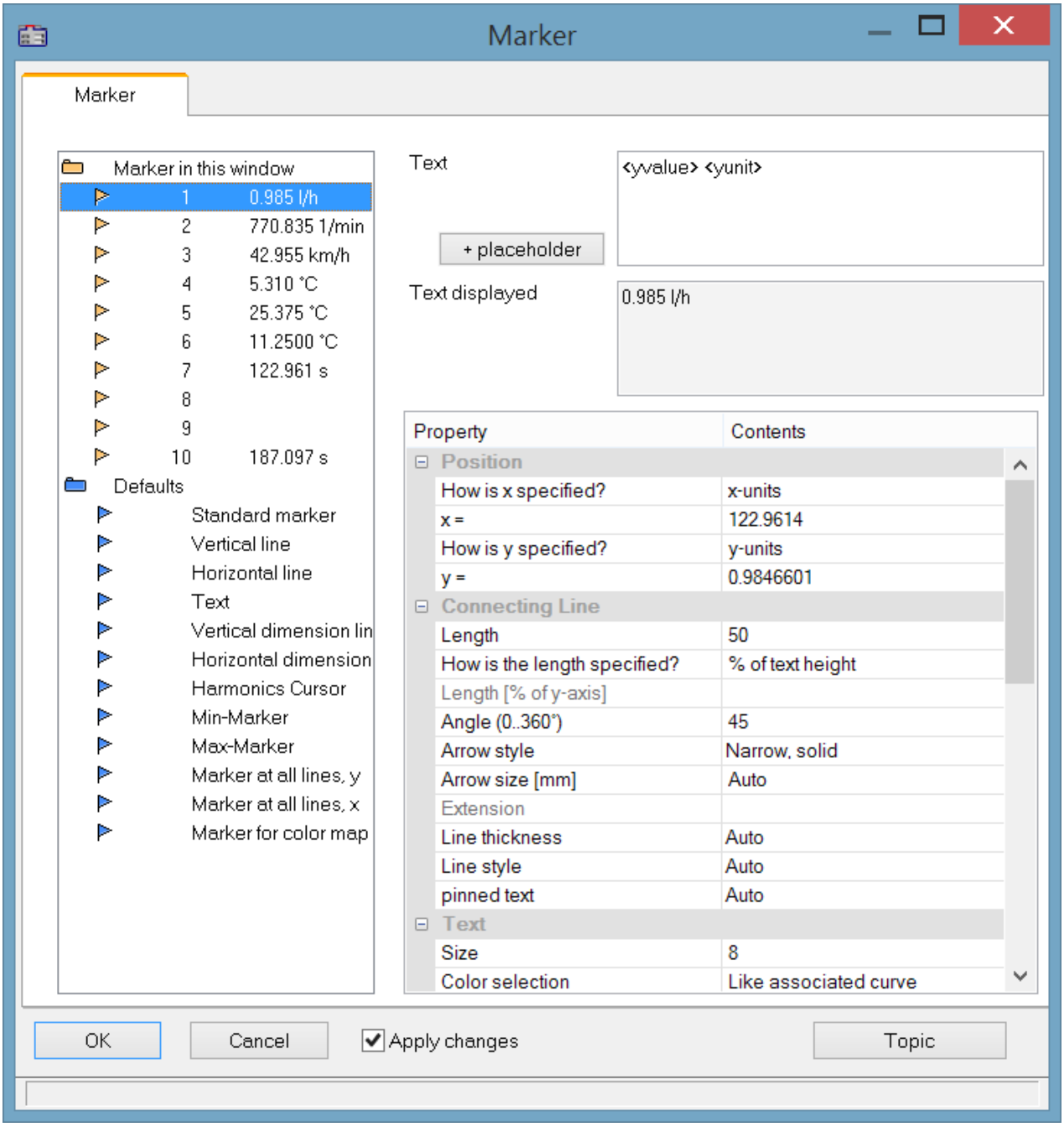
8.6.3.5.6 Marker Definition



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.

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
- wide
- narrow
- wide, solid
- narrow, solid
- large
- large, solid
- circle
- Point
- Diagonal dash
- Star
- Default

The size of the arrow is oriented to the size of the symbol, see also "Clipboard settings 1044".

Line thickness, Line style: Settings like for Lines 1018. At printout, the line appears in the thickness of the cursor-line, see Clipboard Settings 1044.

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:

The following description also appears as a tip-help, if you let the mouse pointer over a placeholder unmoved.

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 1069 can also be displayed.

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 1053").

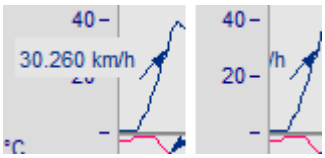
Color: The color can be selected as absolute (e.g. red) or relative. A relative color definition refers to a color already defined for the curve window. For example, if the color of the first curve in the window is selected, it is possible to select another color for the printout and the screen.

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.

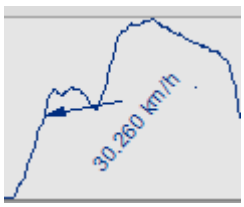
Border: The text may have a border around it. Various options are available for the border joining the connecting line with the marker:

- *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.

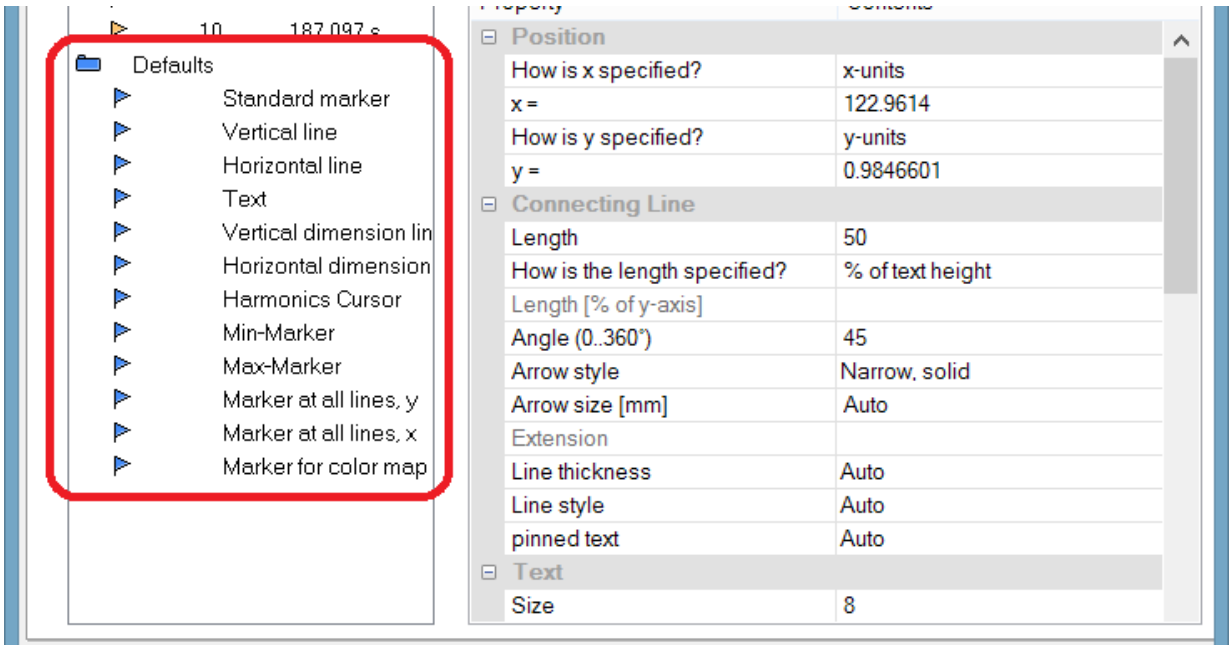


Delete

All lines selected in the marker list are deleted.

Edit defaults

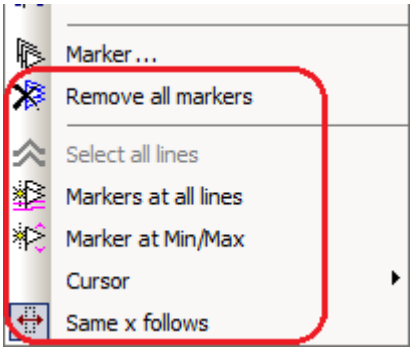
The *Defaults* for the *Marker types* are listed at left links and can be edited at any time using the Properties table.



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.

8.6.3.5.7 Extra marker functions



There are four additional marker functions available:

Delete all markers ( )

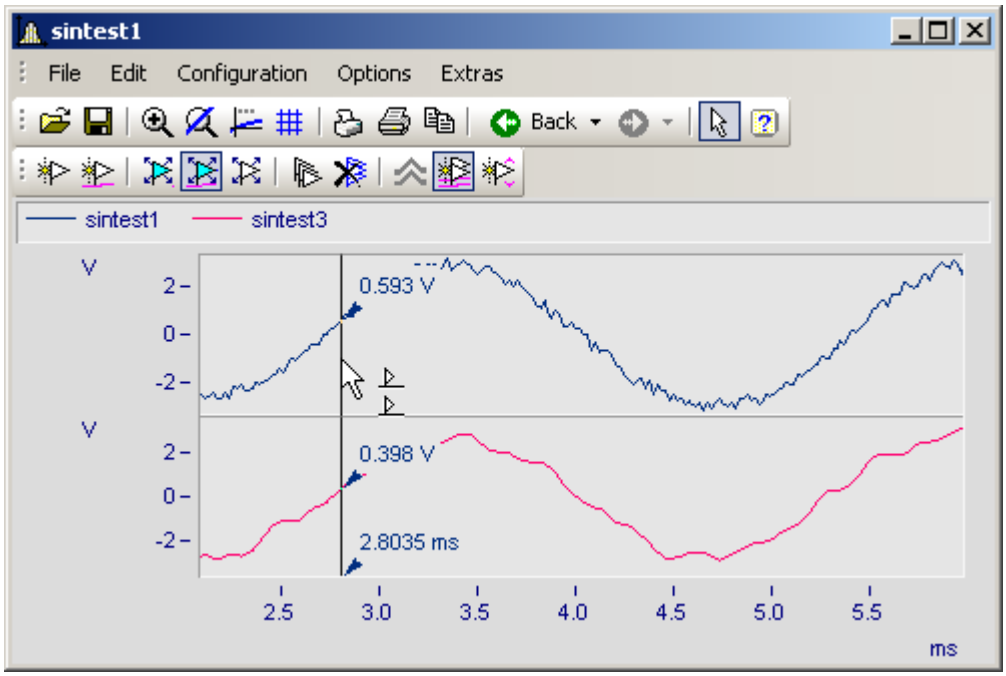
With this function, all markers set in the curve window can be deleted at once.

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".

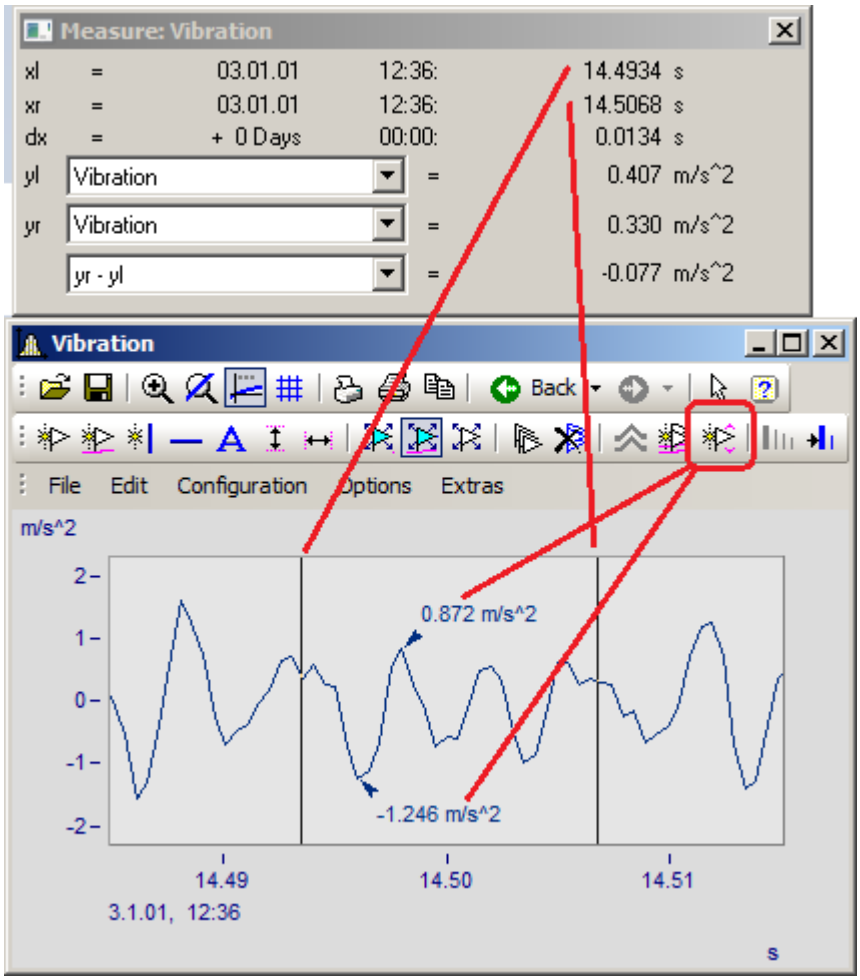


Markers at Min/Max ( )

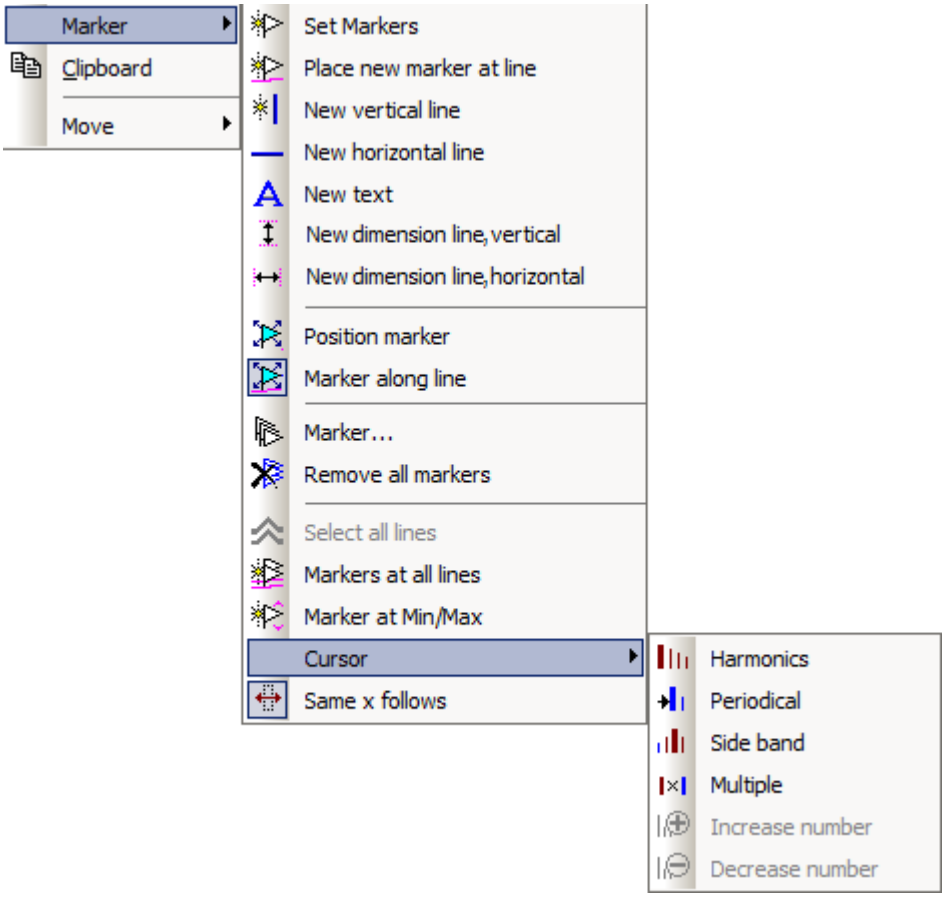
Markers are set for the maximum and the minimum of the line selected. If multiple curves are in a curve window, the function only applies to the line selected. As described in the section "Move Markers 890", the marker's position and its label can be moved as desired.

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.

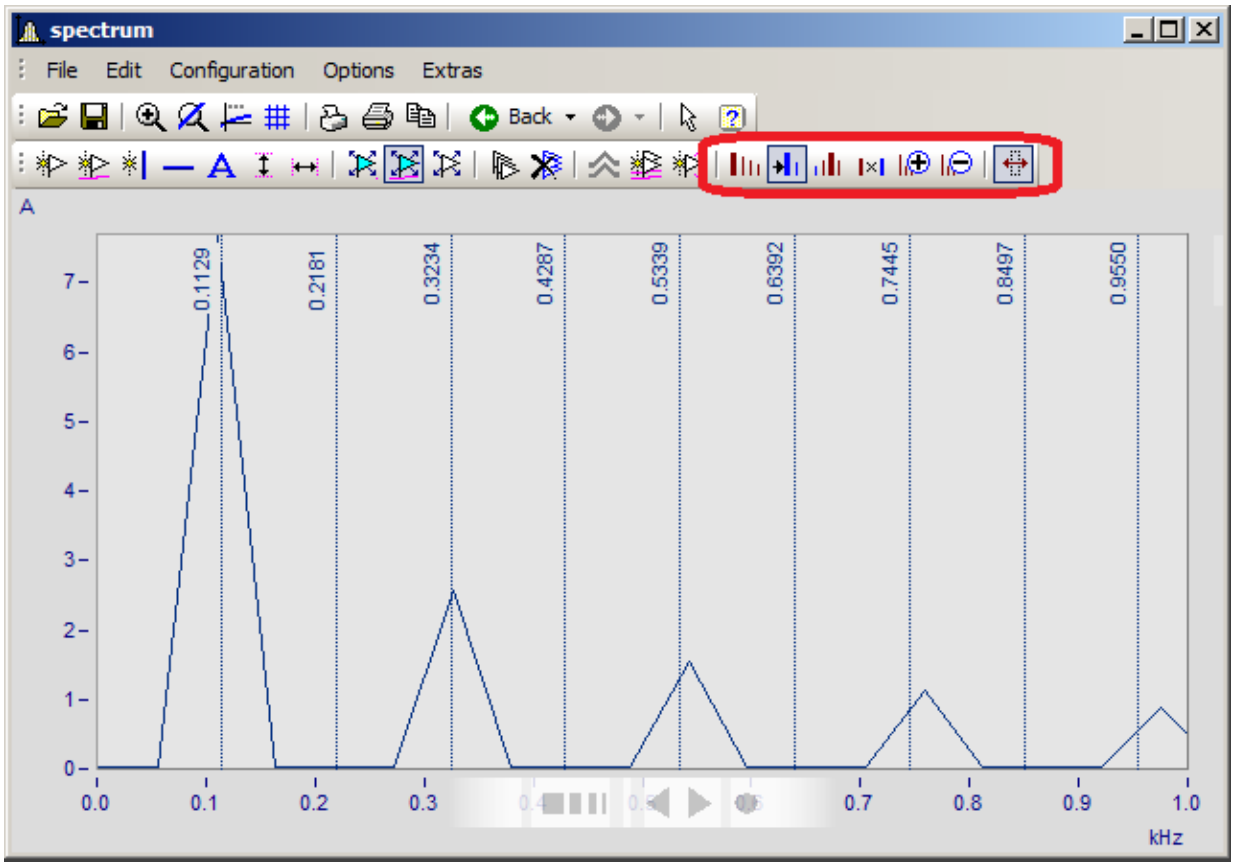


8.6.3.5.8 Harmonics cursor



Function

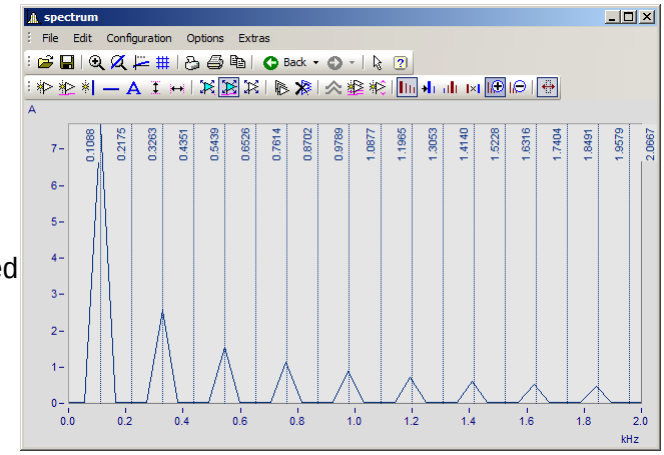
Harmonic cursors indicate periodic multiples of a fundamental frequency. Set the marker at the position of the fundamental oscillation. The position can be readjusted subsequently, if desired.



The following types are available:

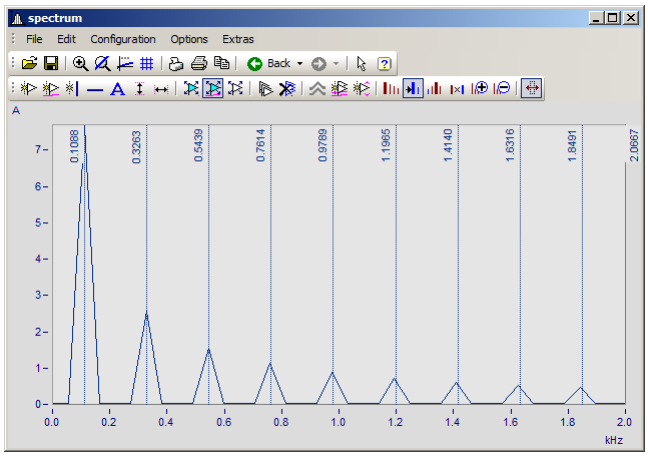
Harmonics ( ):

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.



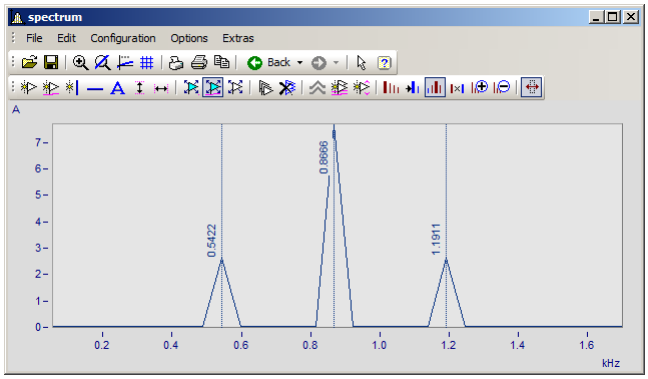
Periodic( ):

Periodic processes with arbitrary start time. The fundamental frequency can be freely placed. The distance between the harmonics is independent of the fundamental oscillation.



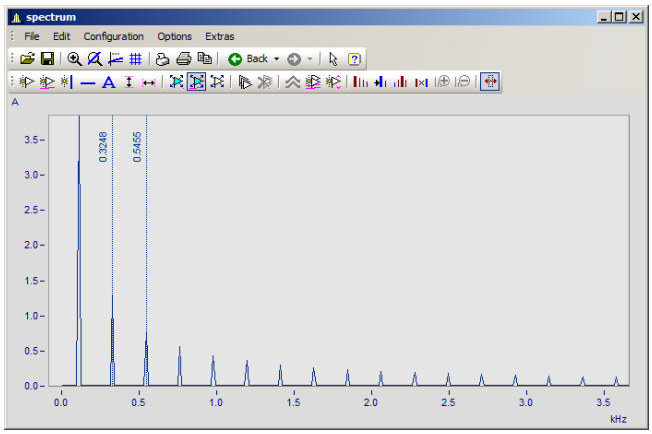
Side band ( ):

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.



8.6.3.6 Clipboard

Function

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.

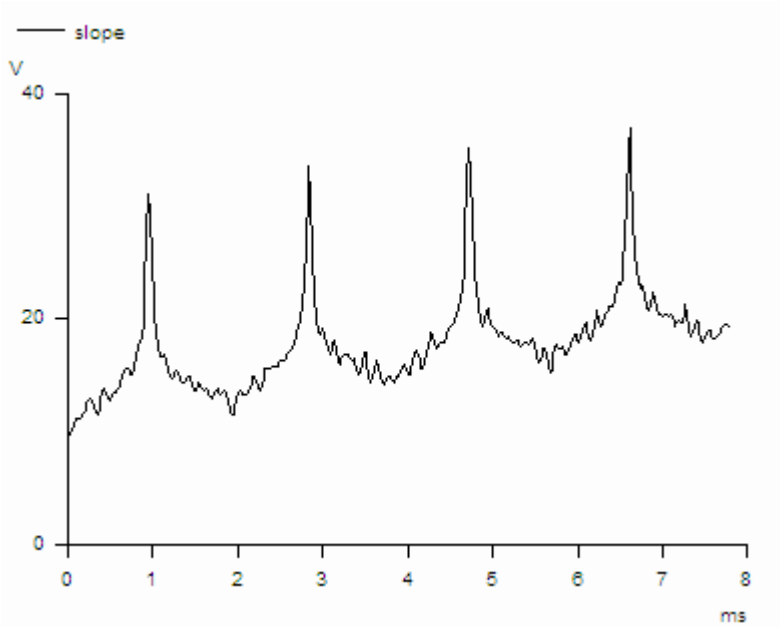
When one application sends graphics to the clipboard, another application can read and retrieve these graphics from the clipboard. In general, data are not damaged when they are read from the clipboard, which means that applications can retrieve the same graph as often as desired. Existing graphics are 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:



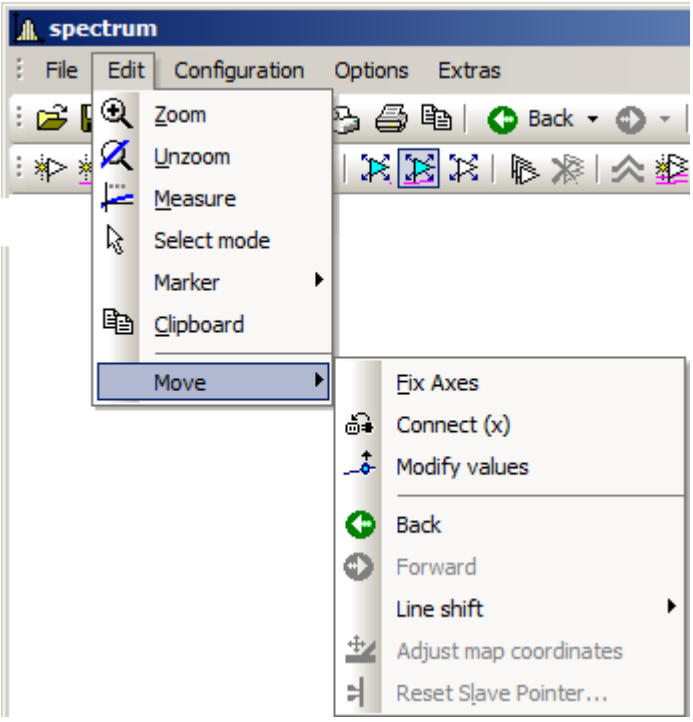
Remarks

- 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 861"). 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 859' 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

Section 'Clipboard Settings 1044'

8.6.3.7 Move



The menu item *Move* contains various curve window functions for working in the curve window:

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 905.

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

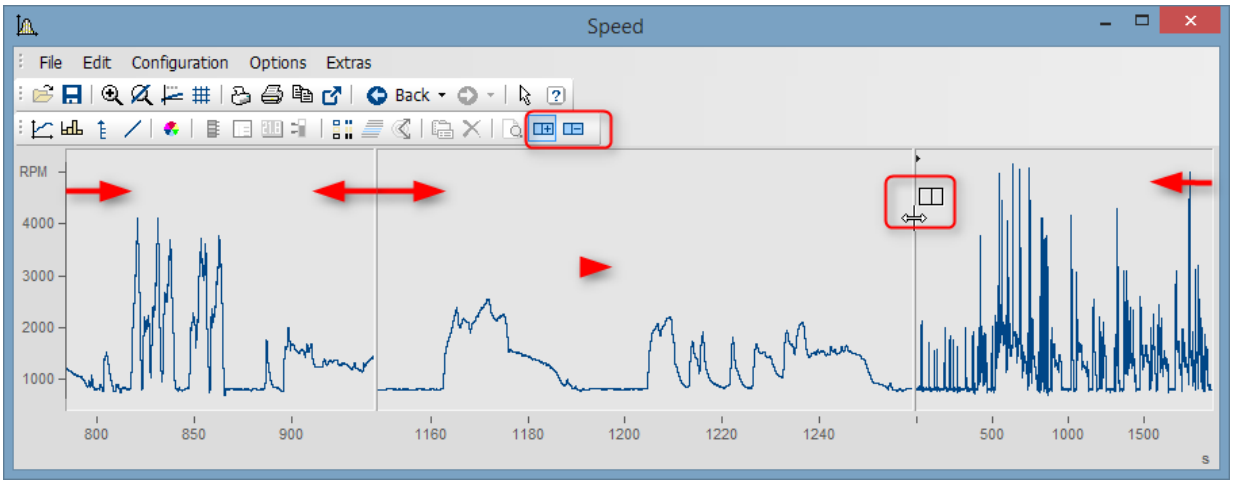
For moving selected lines in horizontal and vertical direction. More on this topic is presented in the section Line-Shift 907.

Reset Slave Pointer...

For resetting the slave pointer. For more on this topic, see the section Bar meter 935.

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

8.6.3.7.1 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.

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* 1053 and 1053 *This window follows* 1053.

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.

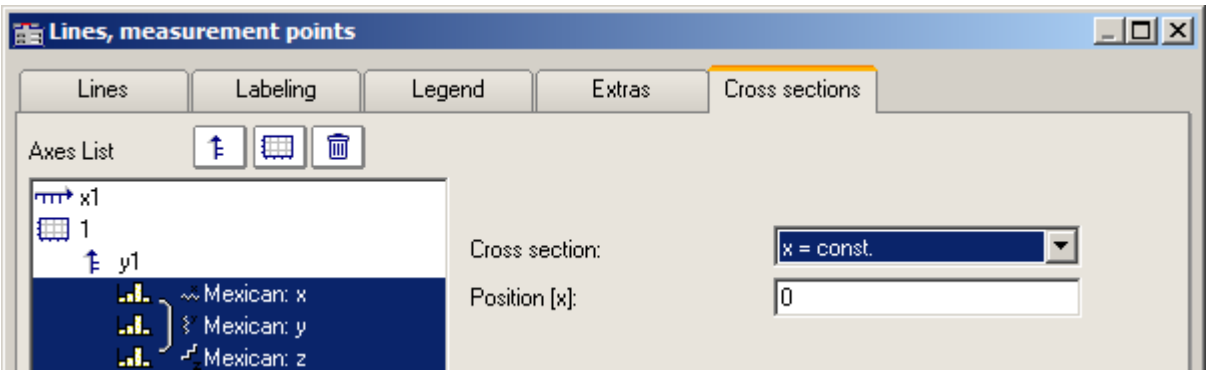
All connected windows should have the same x-axis mode (relative x-axis or display with absolute time).

To undo an x-linkage, simply click on the button again.

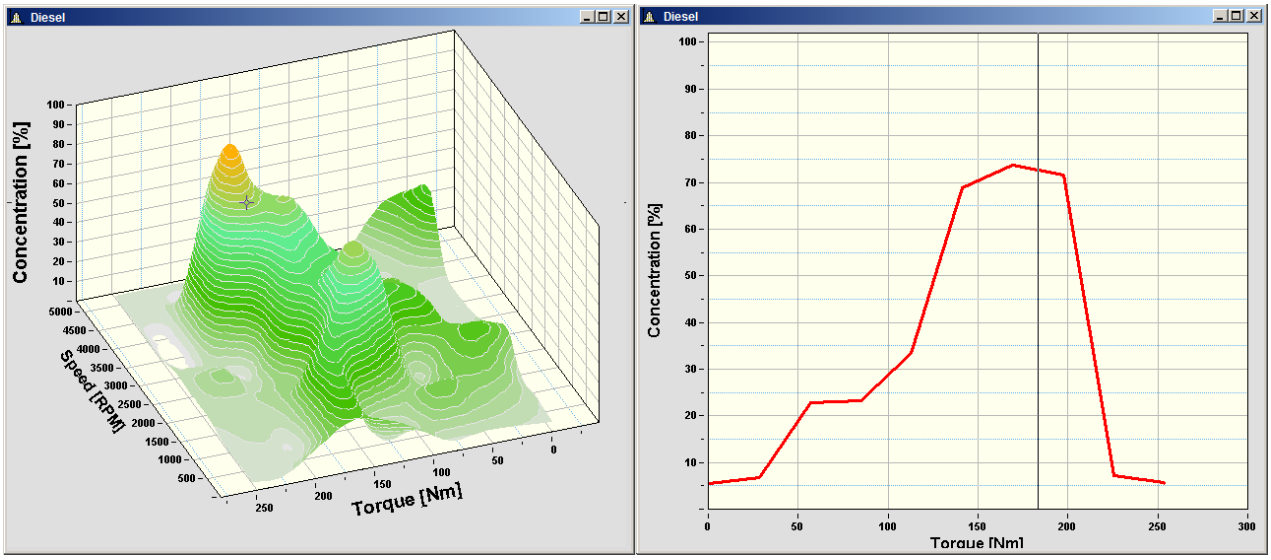
8.6.3.7.1.1 Connect with 3D

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*.



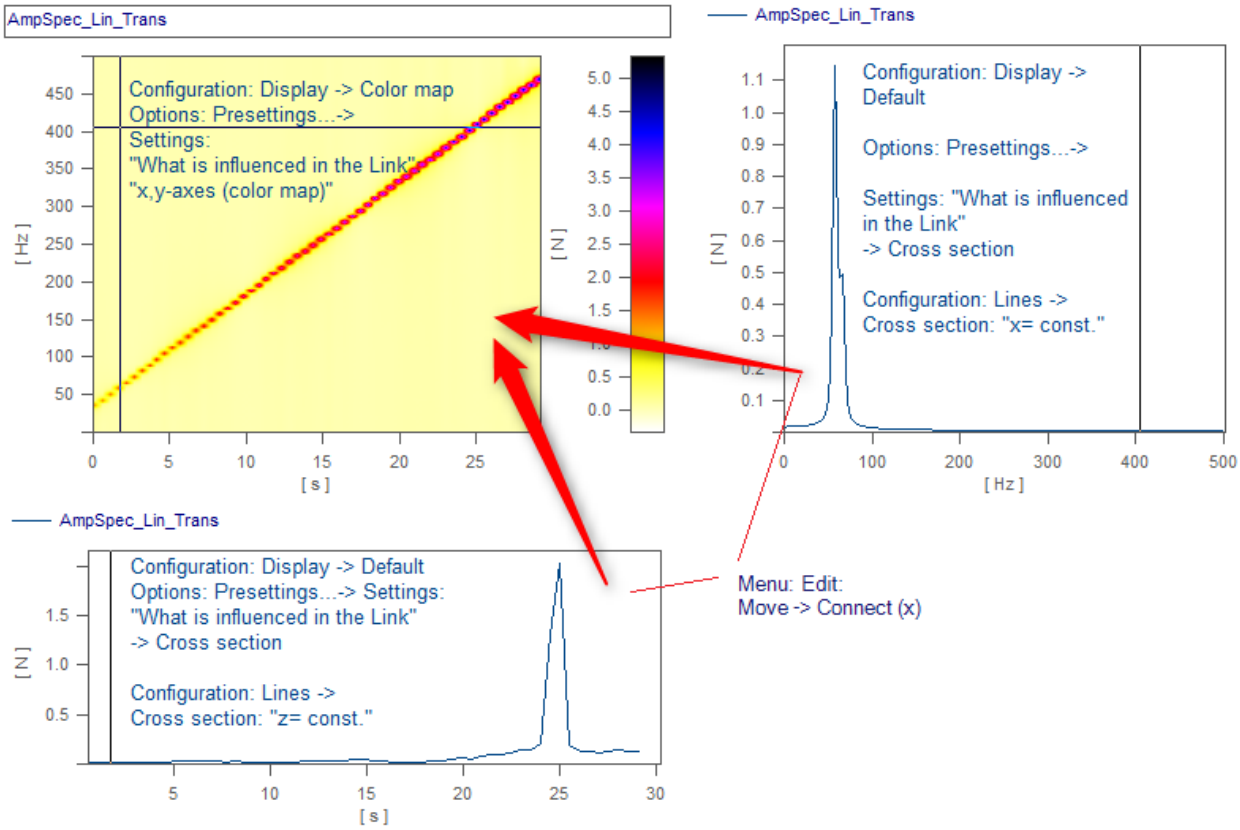
In one of the two curve windows, activate the *Connect*-function, so that the cursor changes shape. Then move the changed cursor into the other curve window's coordinate system and click on the left mouse button. In the curve window with the 3D-display, a crosshairs appears at the upper left edge. Move 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 with two cross sections

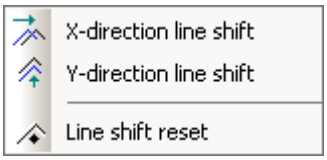
Color maps can also be linked in the y-direction. For this purpose, in the Options for the curve window 1053, under *Settings*, it is necessary to set the *Link*-property "*What is influenced in the Link*" accordingly:

- 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.

8.6.3.7.2 Line Shift



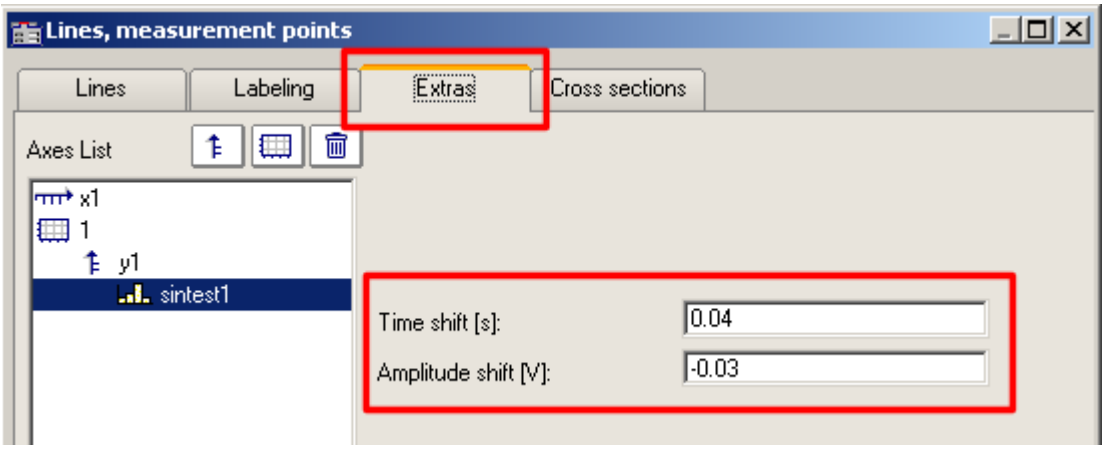
Using this function, any lines selected in the Select-mode, or all lines at once can be moved in the either the x- or y-direction.

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.

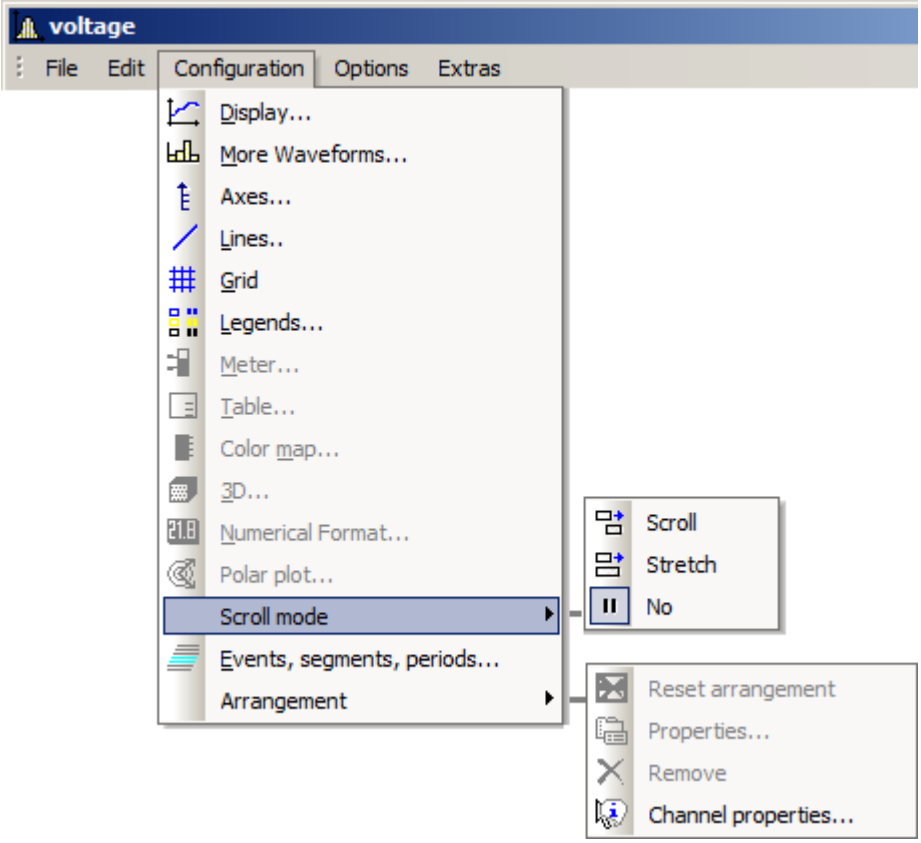
Line-Shift of a selected line in x-direction

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

8.6.4 "Configuration" menu



	Menu item	Description
	Display 910	The structure of the curve window and other attributes, such as the "Freeze" mode and time/date display can be selected here.
	More Channels 995	Selects additional waveforms to be displayed in the current curve window.
	Axes 1005	Sets x- and y-axis parameters.
	Lines 1018	Sets line (measurement cure) parameters.
	Grid 1026	A grid can be added to the coordinate system.
	Legends 976	Sets the display of legends in the curve window.
	Meter 935	Sets properties of bar meter display.
	Table 931	Sets properties of tables display.
	Color map 946	Sets properties of color map display.
	3D 959	Sets properties of 3D display.
	Numerical format 927	Sets properties of the display mode "Last value as number".
	Polar plot 960	Sets properties of polar plot.
	Scroll mode 1035	In the curve window, a particular section of the data set can either "scroll past" or continuously "grow" from the beginning.
	Events, segments, periods 1035	Selects the display of individual events (in imc Devices: trigger events), individual segments (e.g. spectra or rows in a matrix) or even periods (period comparison).
	Arrangement 1042	Restores the curve window arrangement, access to properties and deletion of selected objects.

8.6.4.1 Display

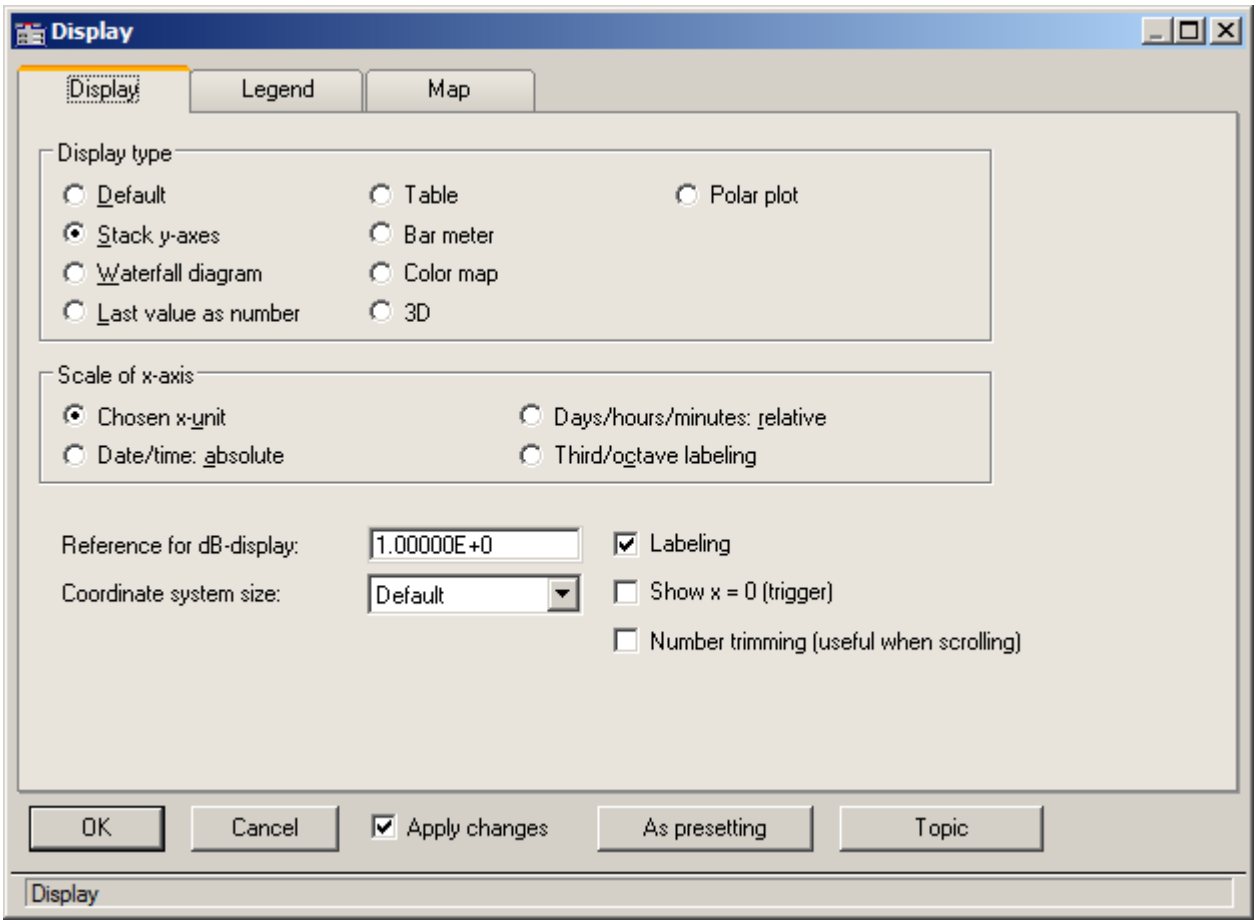
8.6.4.1.1 Display

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

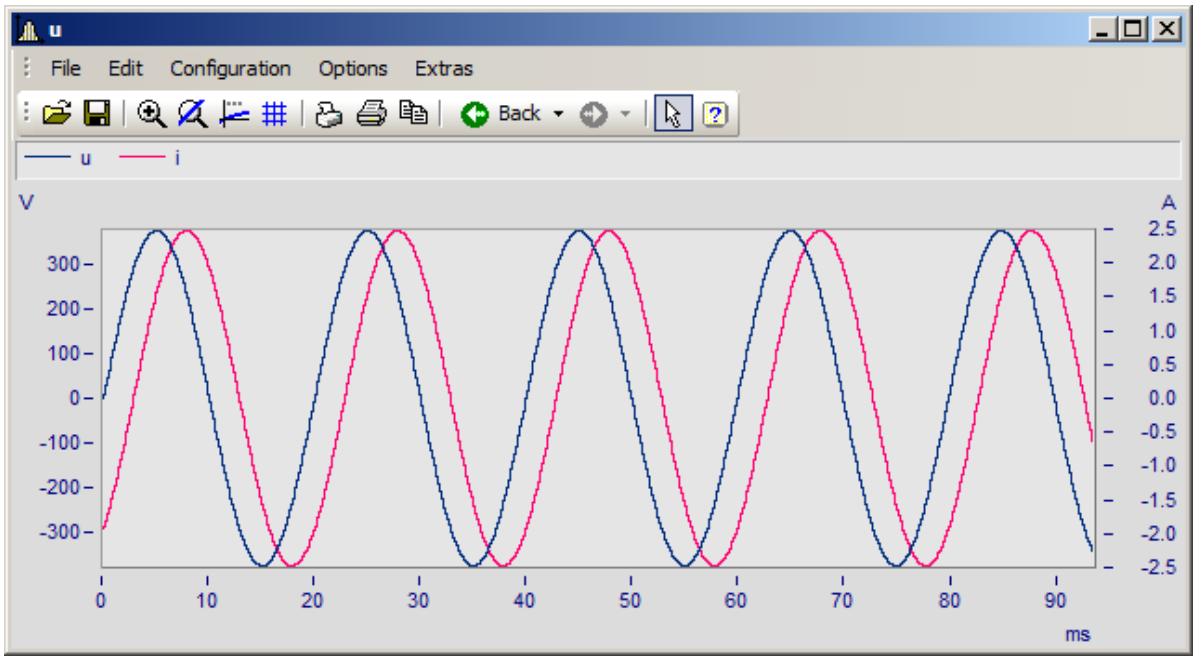
Select menu option *Configuration/Display...* in the curve window. A dialog appears in which the curve window display can be defined.



Display type

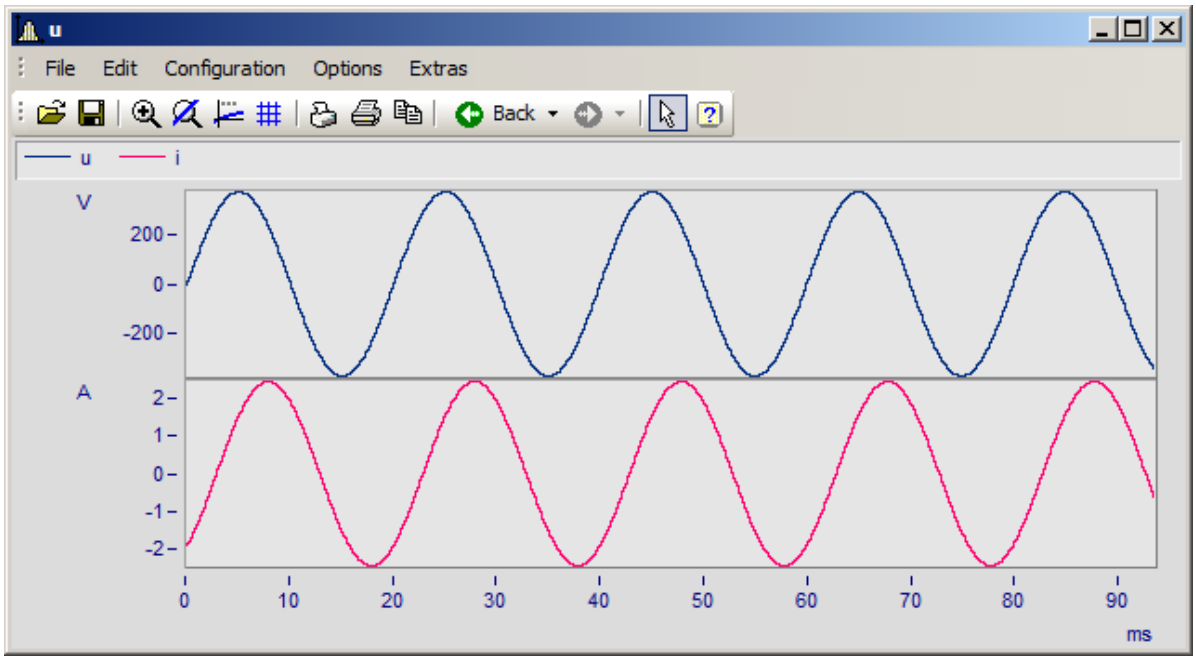
Default

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.



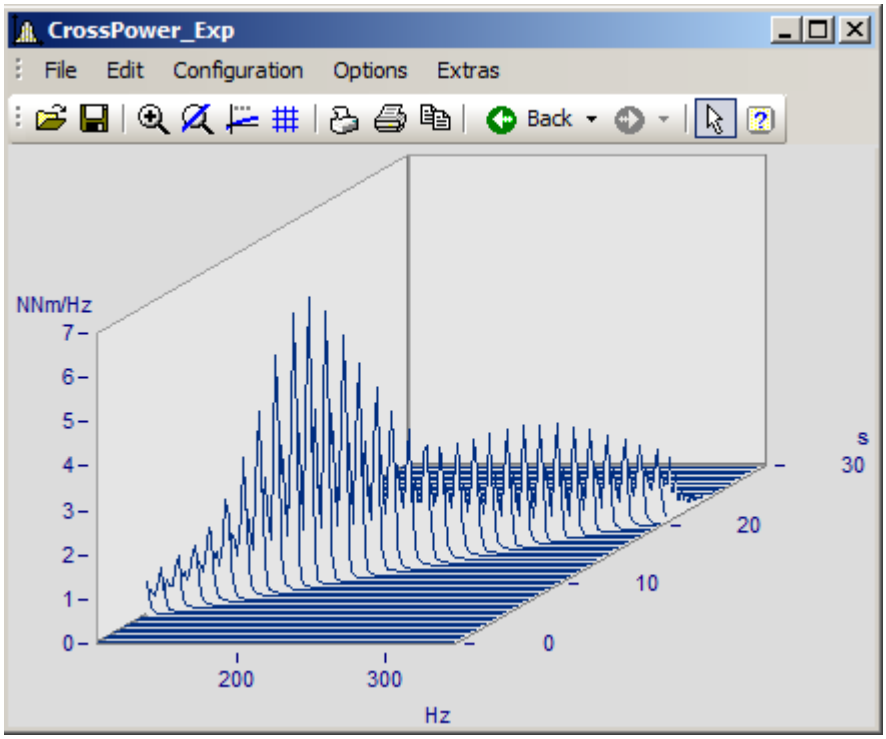
Y-axes stacked

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.

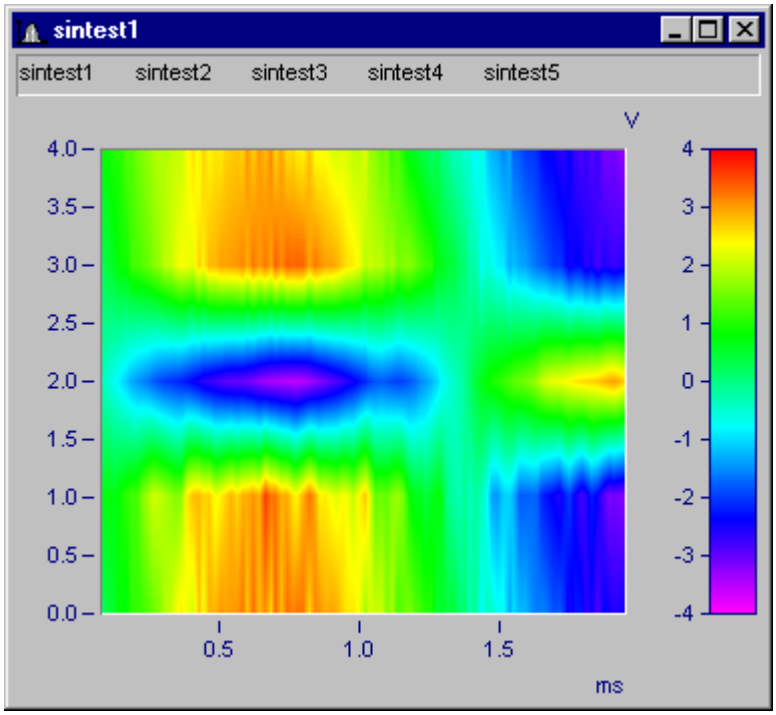


Reference

See "Waterfall Diagrams 921"

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.



Reference

See "Color Map Display 946".

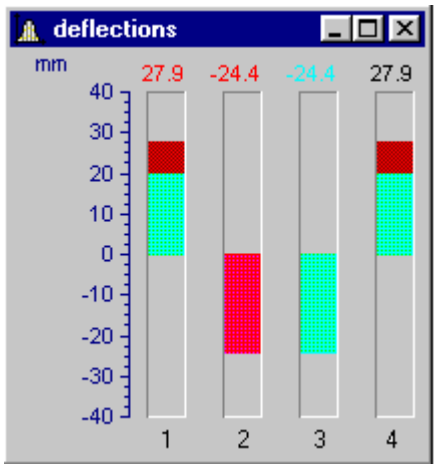
Last value as number

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.

Reference

See "Last Value As Number 927".

Bar meter

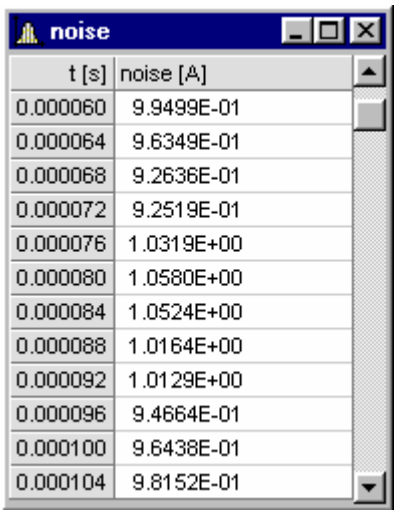


Particularly for online display of measured data, in this mode the last value is displayed in a bar meter.

Reference

Section 'Bar gauge' 935

Table



noise

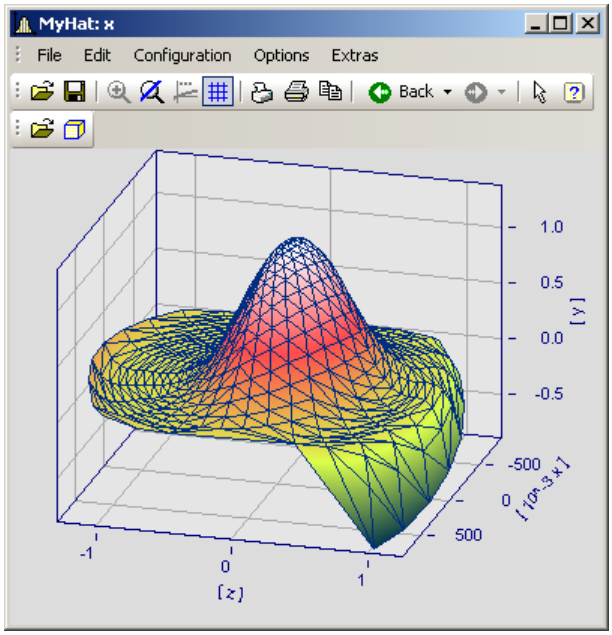
t [s]	noise [A]
0.000060	9.9499E-01
0.000064	9.6349E-01
0.000068	9.2636E-01
0.000072	9.2519E-01
0.000076	1.0319E+00
0.000080	1.0580E+00
0.000084	1.0524E+00
0.000088	1.0164E+00
0.000092	1.0129E+00
0.000096	9.4664E-01
0.000100	9.6438E-01
0.000104	9.8152E-01

The readings are displayed in tabular form, in chronological order.

Reference

Section "Tabular display 931"

3D

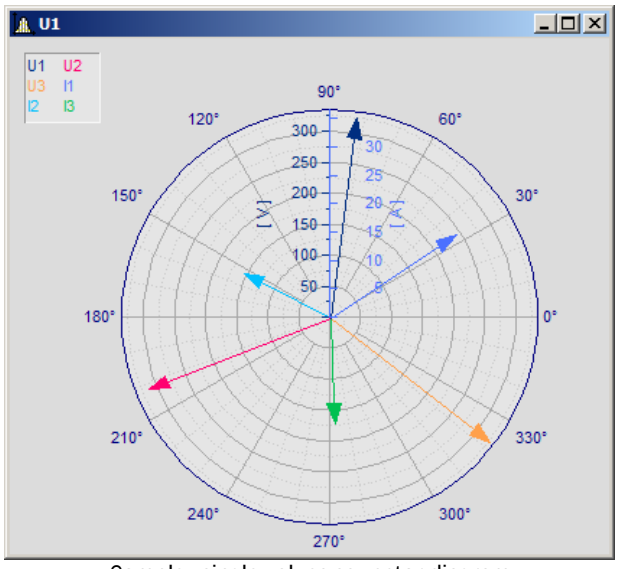


3-D display of waveforms having x,y,and z coordinates with various color map surface patterns, and rotating perspective in all dimensions.

Reference

Section on 3D display 959

Polar plot



Complex single values as vector diagram

Complex data sets can be displayed as a polar diagram.

Reference

Section on "Polar plot" 960

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 1005".

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.

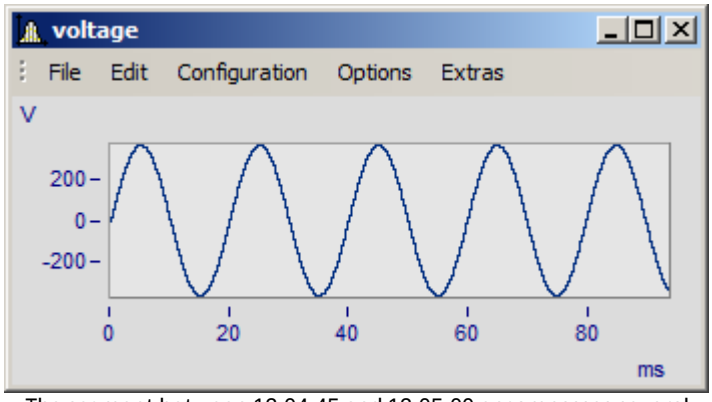
The generation time of a waveform along with the x-coordinate in "s" determines the resulting axis labels.

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

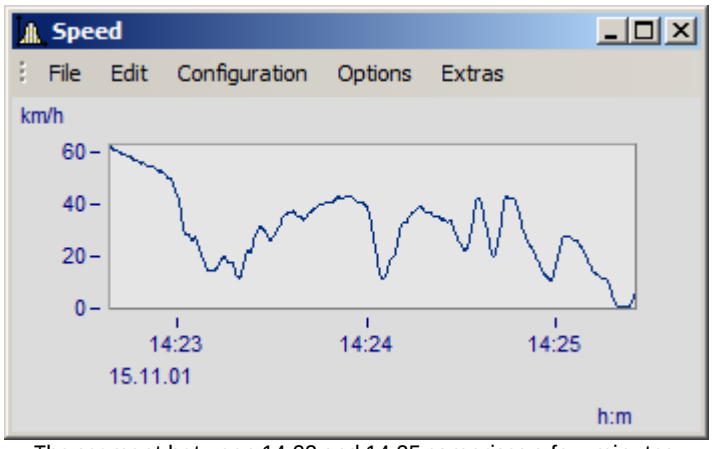
1.1.92, 13:00:00 to 2.1.92, 13:00:00

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.

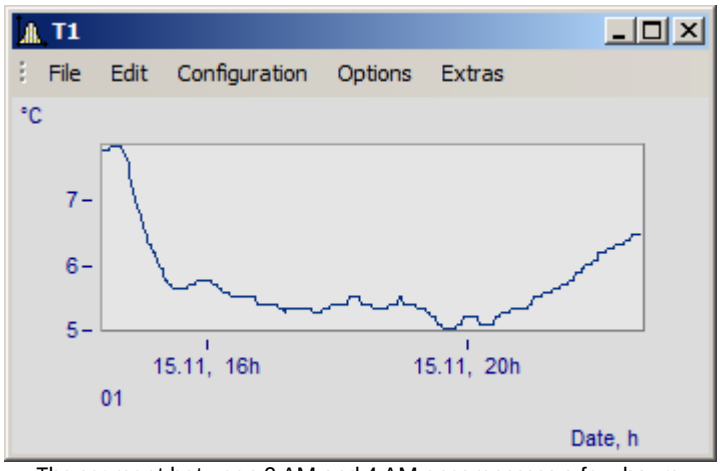
The labeling of the x-axis depends on the length of the displayed time interval, for example:



The segment between 13:04:45 and 13:05:00 encompasses several seconds.



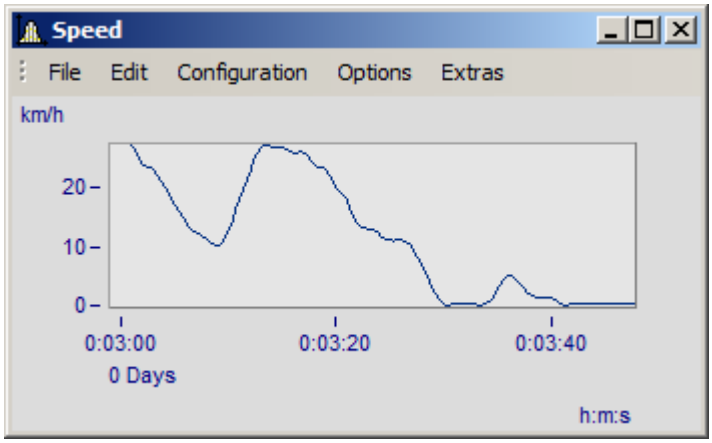
The segment between 14:23 and 14:25 comprises a few minutes.



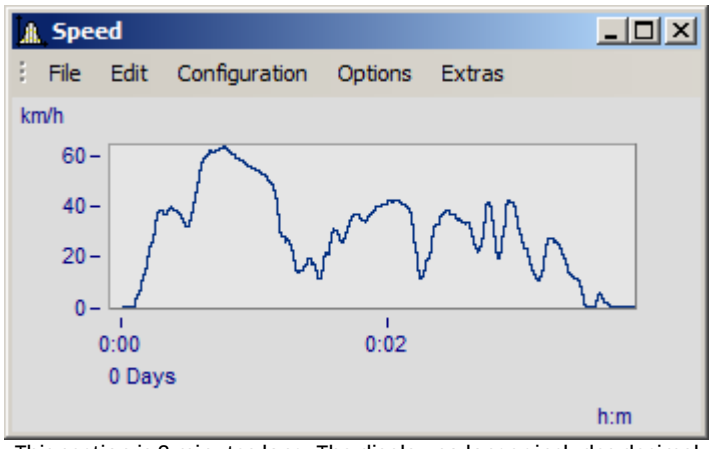
The segment between 2 AM and 4 AM encompasses a few hours.

Days/hours/minutes: Relative

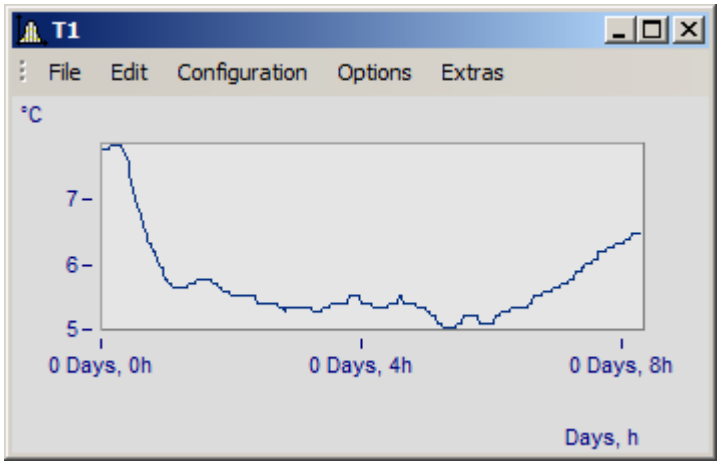
If you wish to view the whole duration of a long-term measurement, then display of relative time in days, hours, minutes and seconds is recommended. The selected unit depends on the interval. Possible variations of this labeling are as follows:



A very narrow section is shown, significantly less than one minute



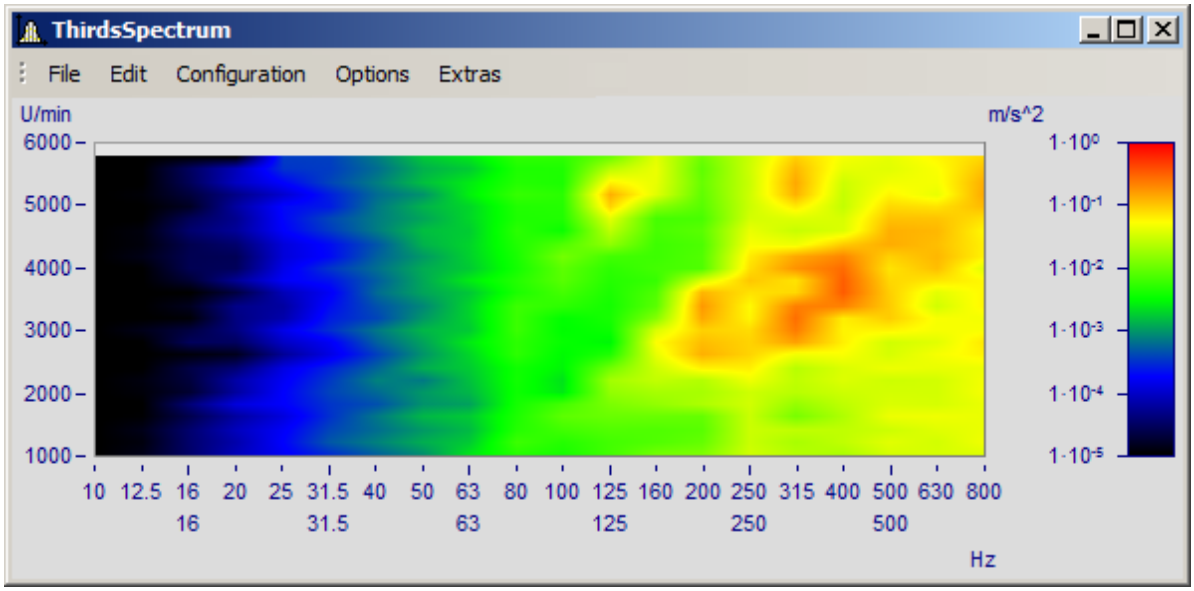
This section is 3 minutes long. The display no longer includes decimal places after the seconds.



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.



Reference

See "Third-Octave Display 965".

Further settings

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 1026 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

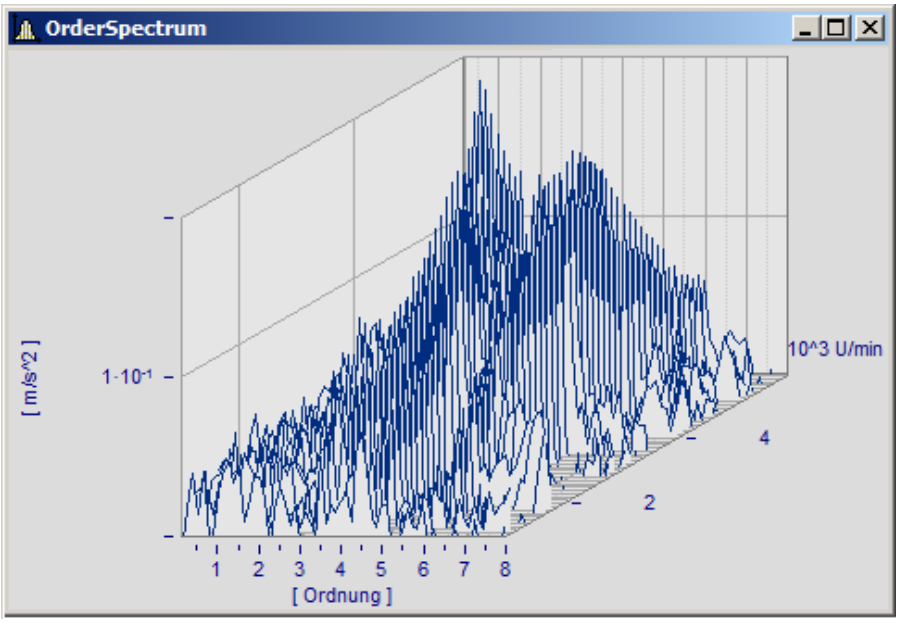
For details on operation, see the chapter Confirmation bar 1043

Remarks

- 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.

8.6.4.1.1.1 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.

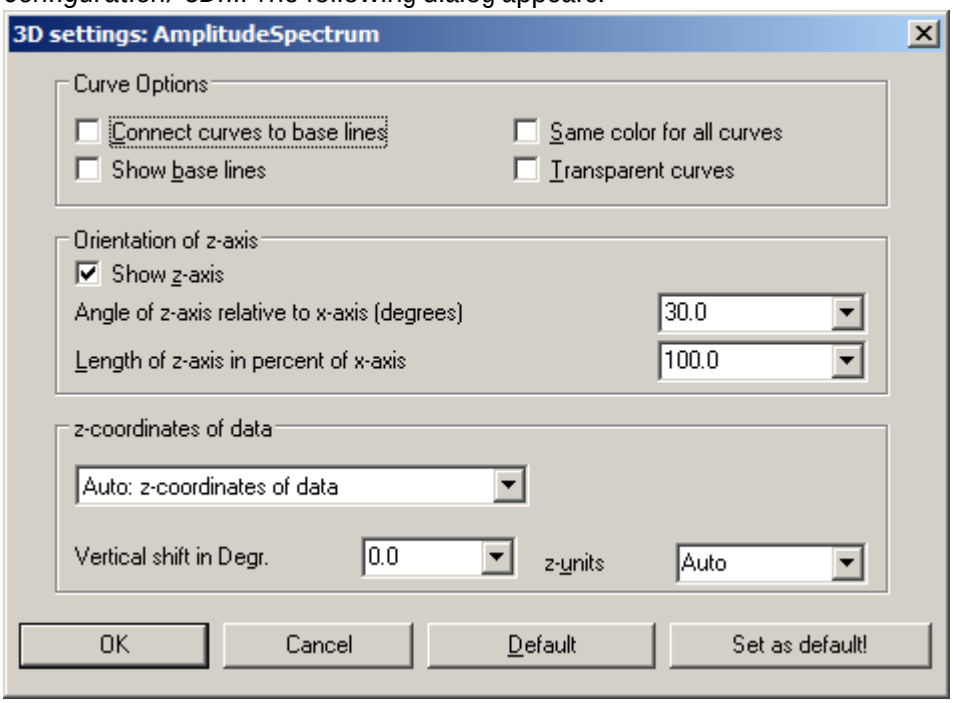
A waterfall display is drawn from back to front. When a new curve is drawn, the area under the curve is filled in with the color of the screen background, covering all lower parts of curves behind it.

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:

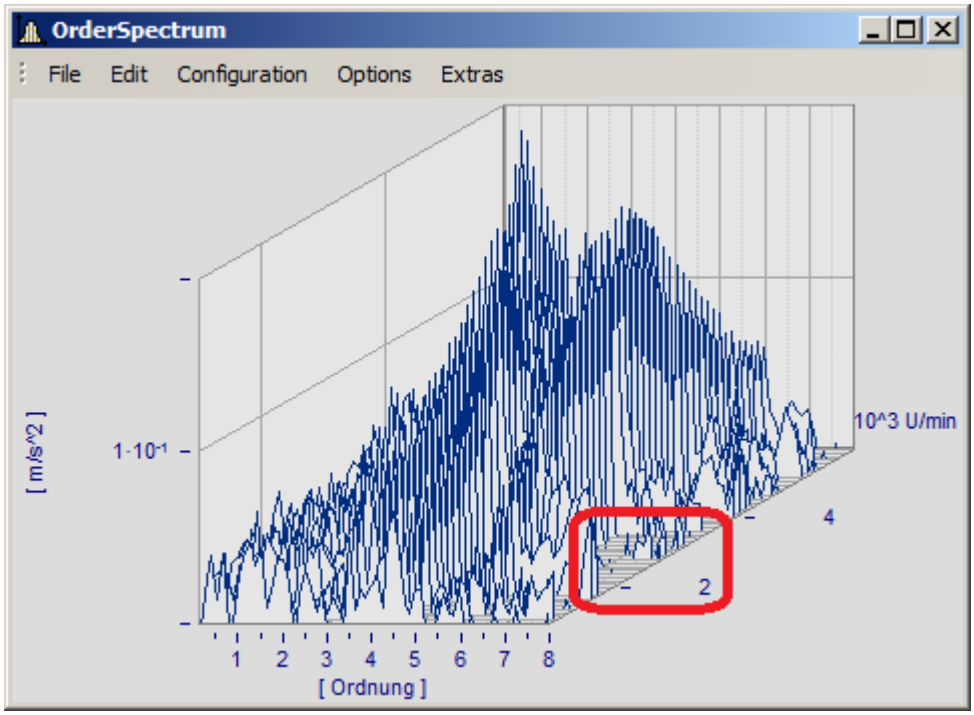


Curve options

Enter here the settings for the appearance of the curves in the window.

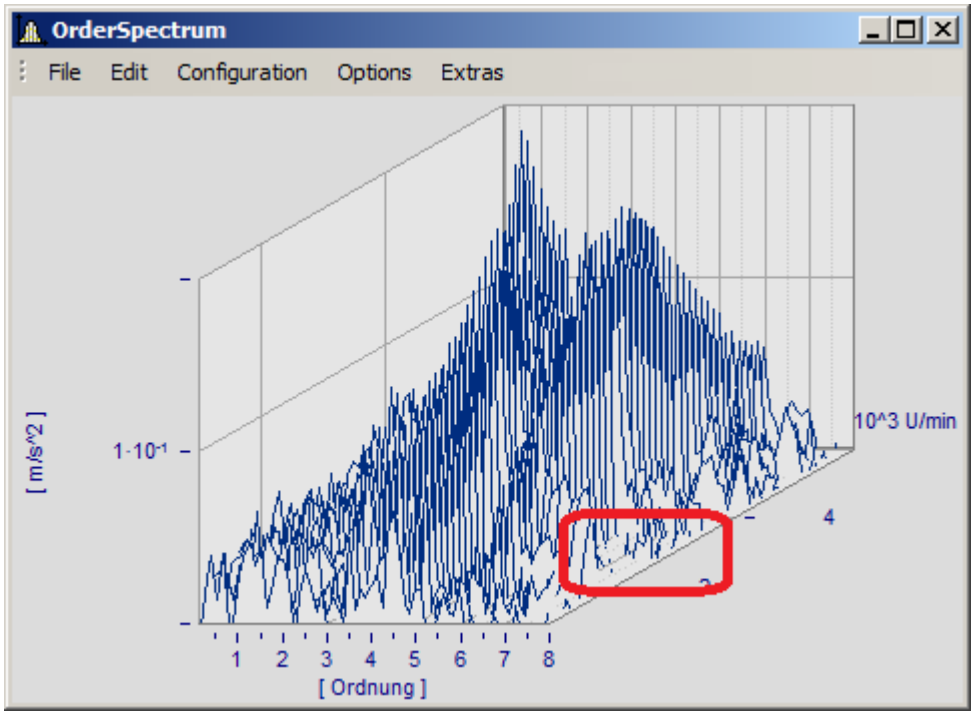
Connect curves to base lines

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:



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.

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:

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

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.

Caution! 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

Section 'Scaling the x-axis' 1005

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 1043

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.

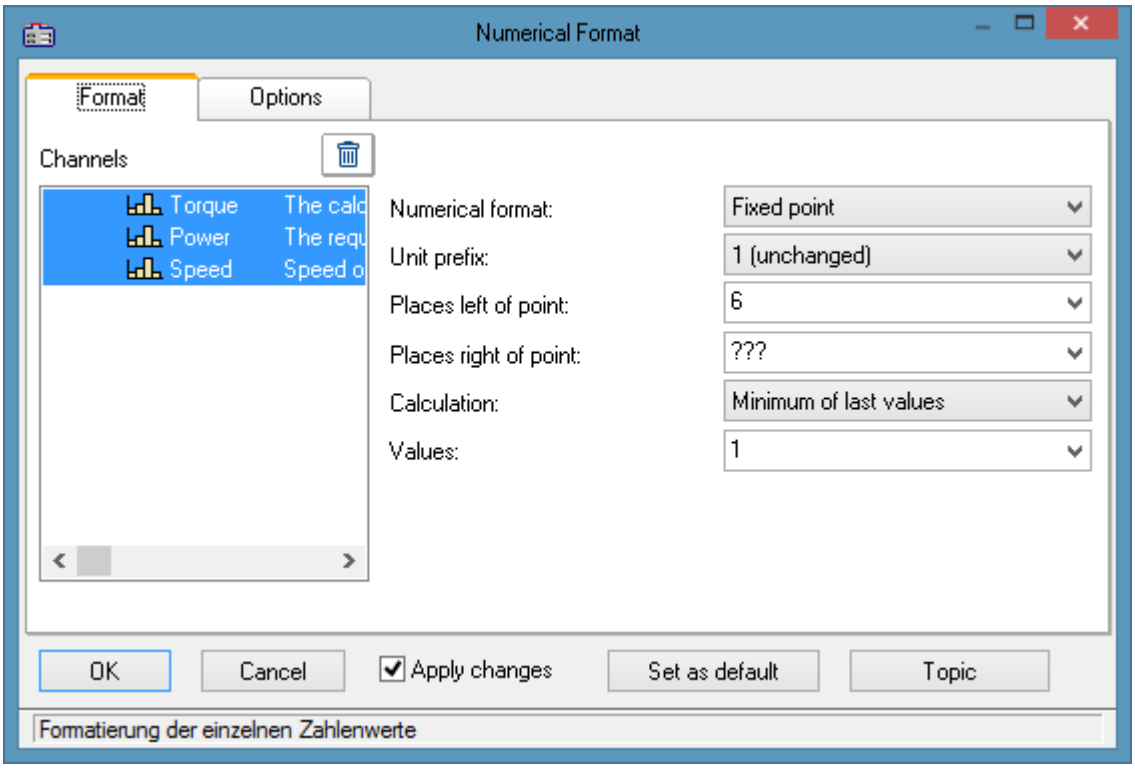
8.6.4.1.1.2 Last Value As Number

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.

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*.

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 910.

Time relative-, Date/time fixed: The formatting of the time and date is specified by means of placeholders, see the chapter Axes-Scale 1006.

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.

Calculation: The numerical value displayed can be calculated from the most recent values:

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.

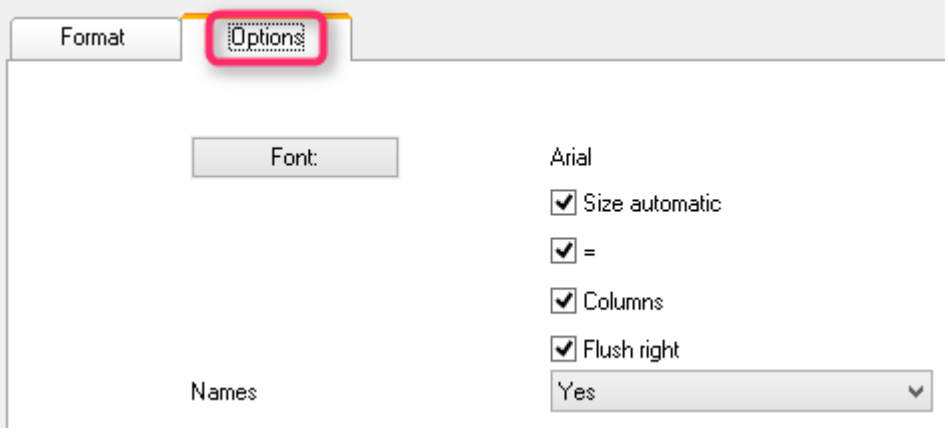
Minimum of the last values: The minimum is calculated from the last *n* values in the waveform.

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.

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 1043.

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*

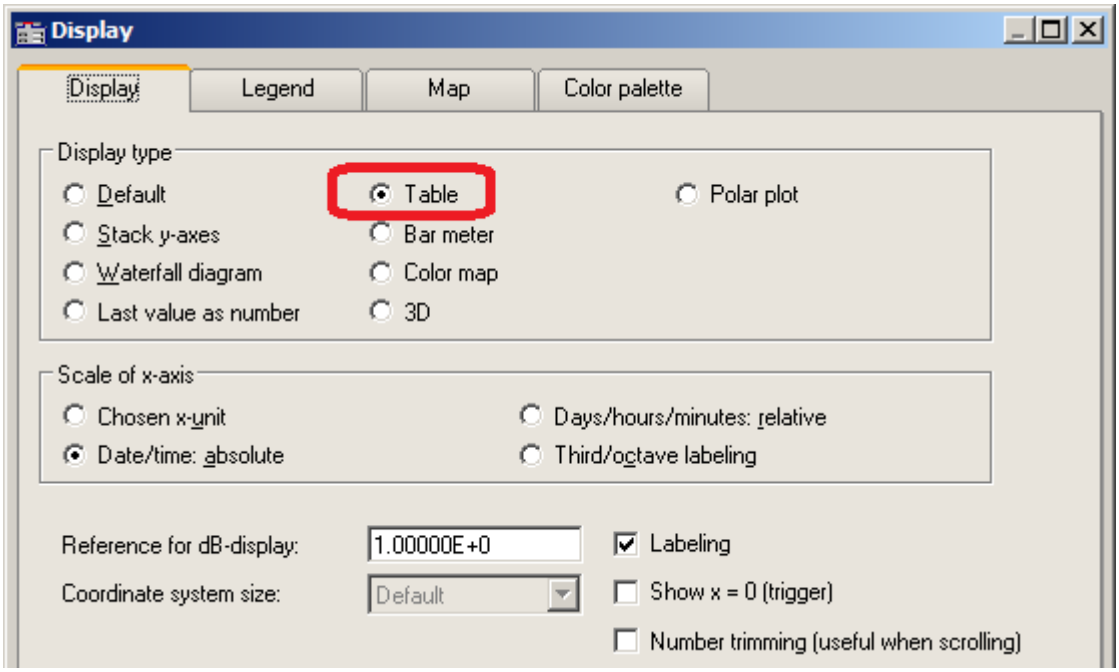
The display mode *Last value as number* is automatically activated if real waveforms with a length of 1 are displayed.

8.6.4.1.1.3 Tabular representation

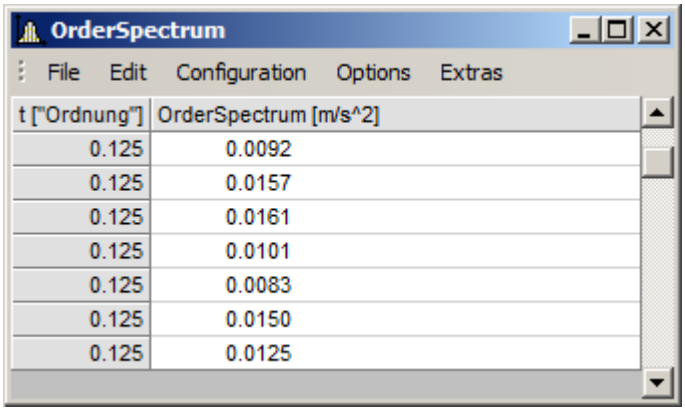
Measurement readings can also be displayed in a table, where the readings are listed in chronological order. This can appear as a protocol readout or as a log book.

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.

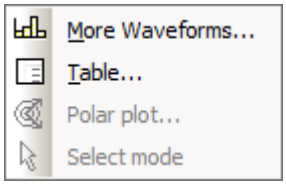


The curve window then appears as shown below:

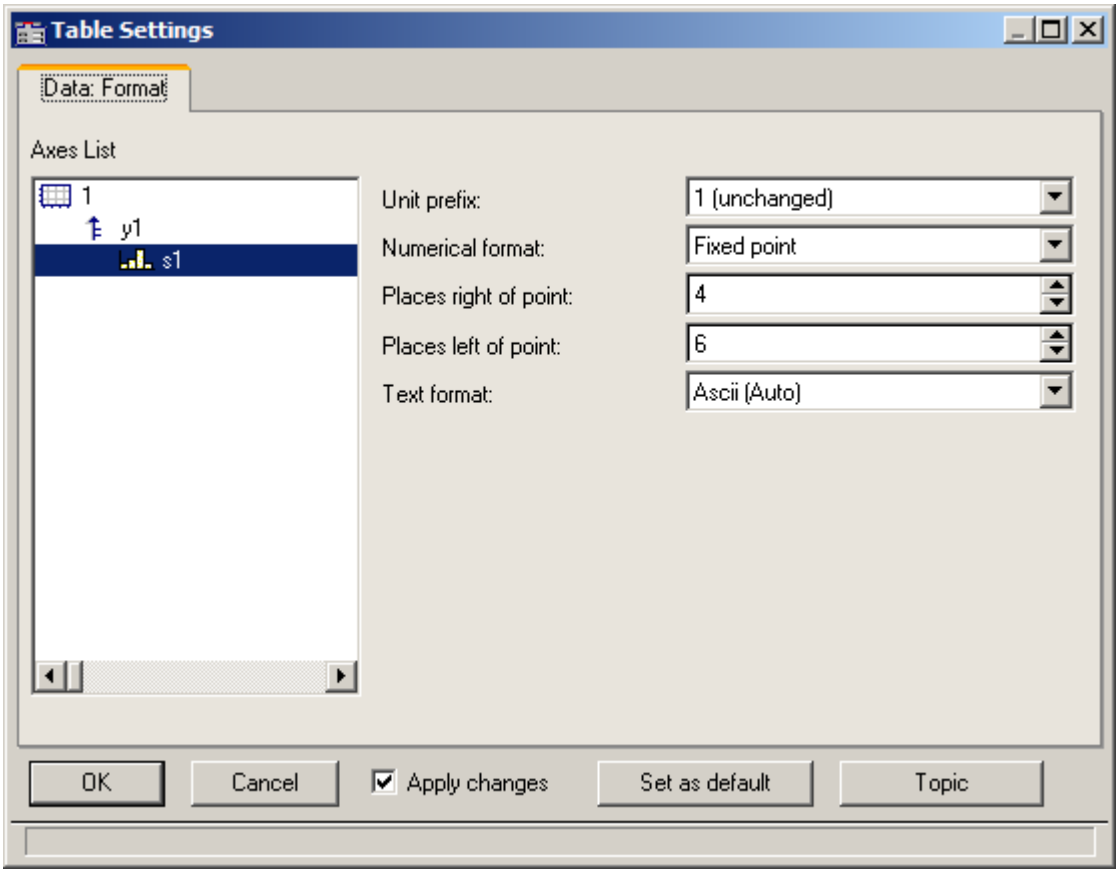


It's possible to move through the list using the scrollbar at right.

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.

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.

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

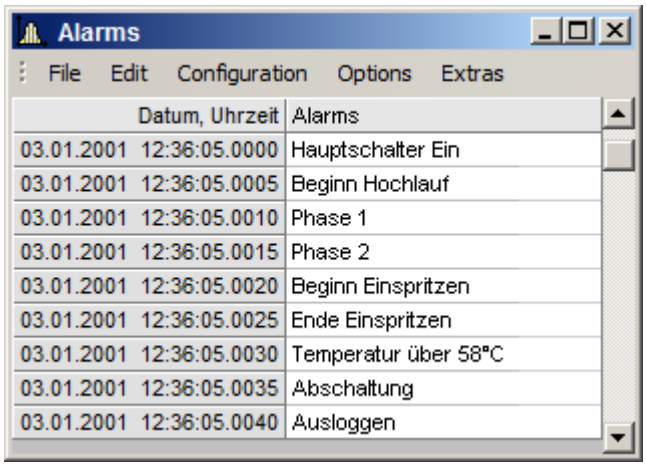
This is the number of decimal places to show. 0..15 places are possible.

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.

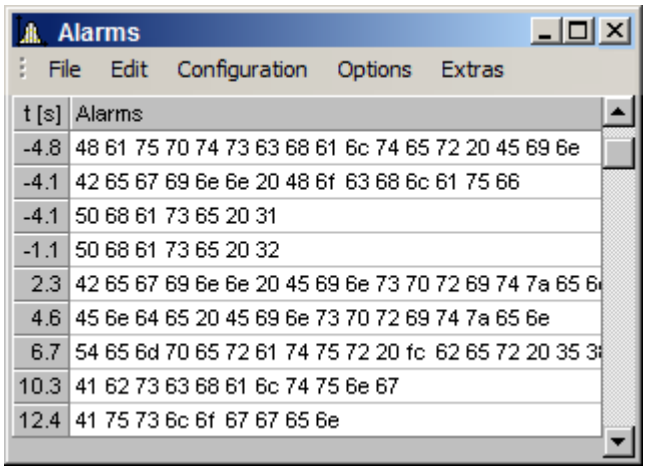
Text format

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 imc 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:

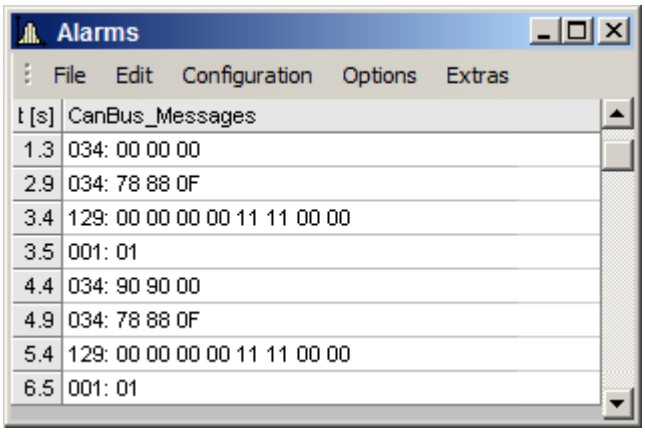


The following options are available:

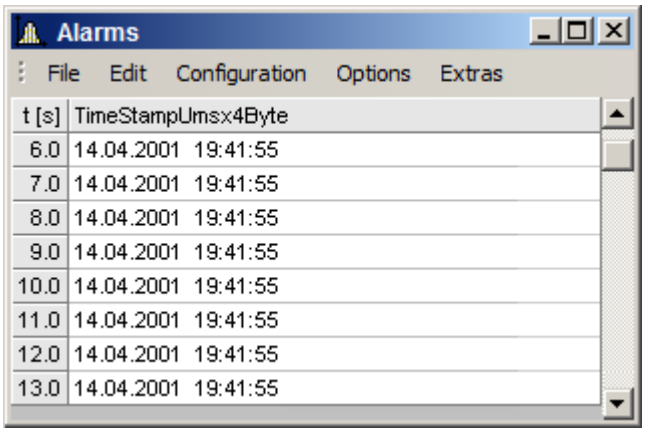
- ASCII (Auto): Display of ASCII characters in legible form (see above.).
- Hex: Display of hexadecimal values 00H .. FFH.



- imc DEVICES CAN-message: for all devices with a CAN-connection: display with identifier and contents of the CAN-messages. The identifier (appearing before the colon) and the bytes belonging to the message are displayed as hexadecimal values.

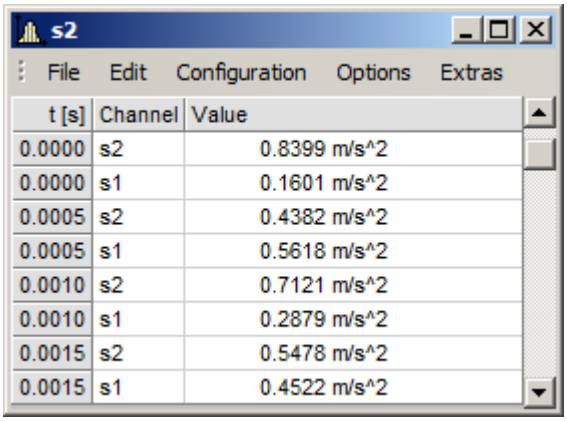


- imc DEVICES 4 byte abs time: For all devices which can read in 4 Byte time stamp data (seconds since 1.1.1980), there is a specialized display style.



Multi-channel displays

Display of multiple channels at once also uses correctly synchronized ordering. For every measurement reading, the source channel is indicated:



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.

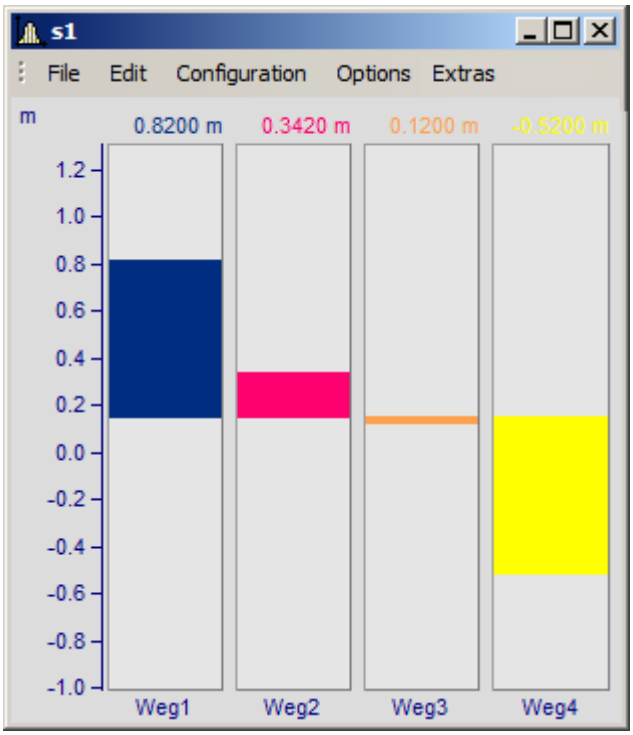
OK, Cancel, Set as default

For details on operation, see the chapter Confirmation bar 1043

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 characteristic curve which is an 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 1043.

8.6.4.1.1.4 Bar meter



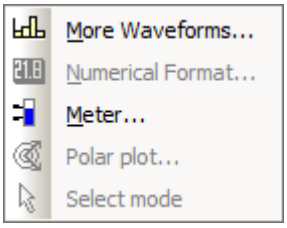
In addition to the numerical data display *Last value as number*, there is also a data display style having the form of a vertical meter bar. The height of the bar is determined from the last value in the waveform. This representation style is particularly useful for online viewing, as it offers a convenient 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 (context menu - "More Channels in Curve Window 995").

To configure the bar meter itself, right-click on the curve window to open the context menu:



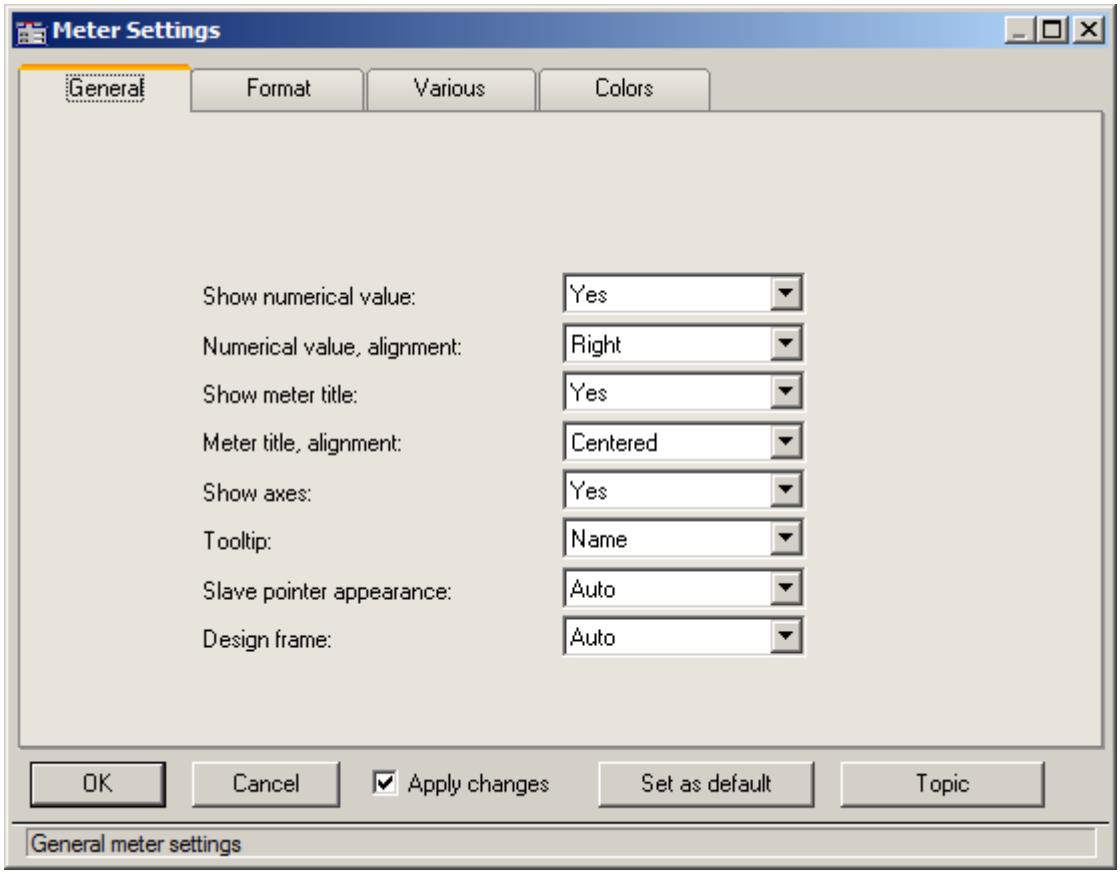
Select *Meter...* to open the *Meter Settings* dialog. This dialog consists of several sub-dialogs, each of which contains controls for different settings.

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.



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*.

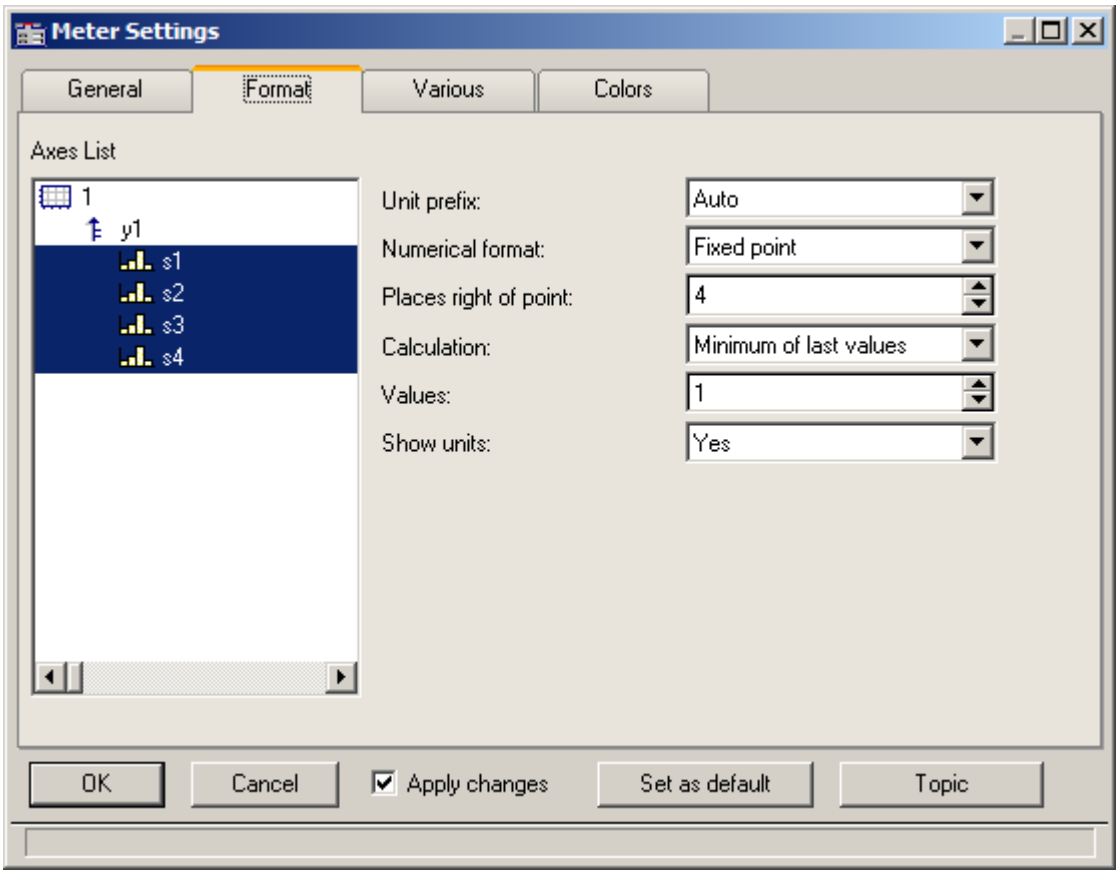
Design frame

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* 996. If multiple, selected channels have mutually different settings for a property, the setting indicated for that property will be "???".



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

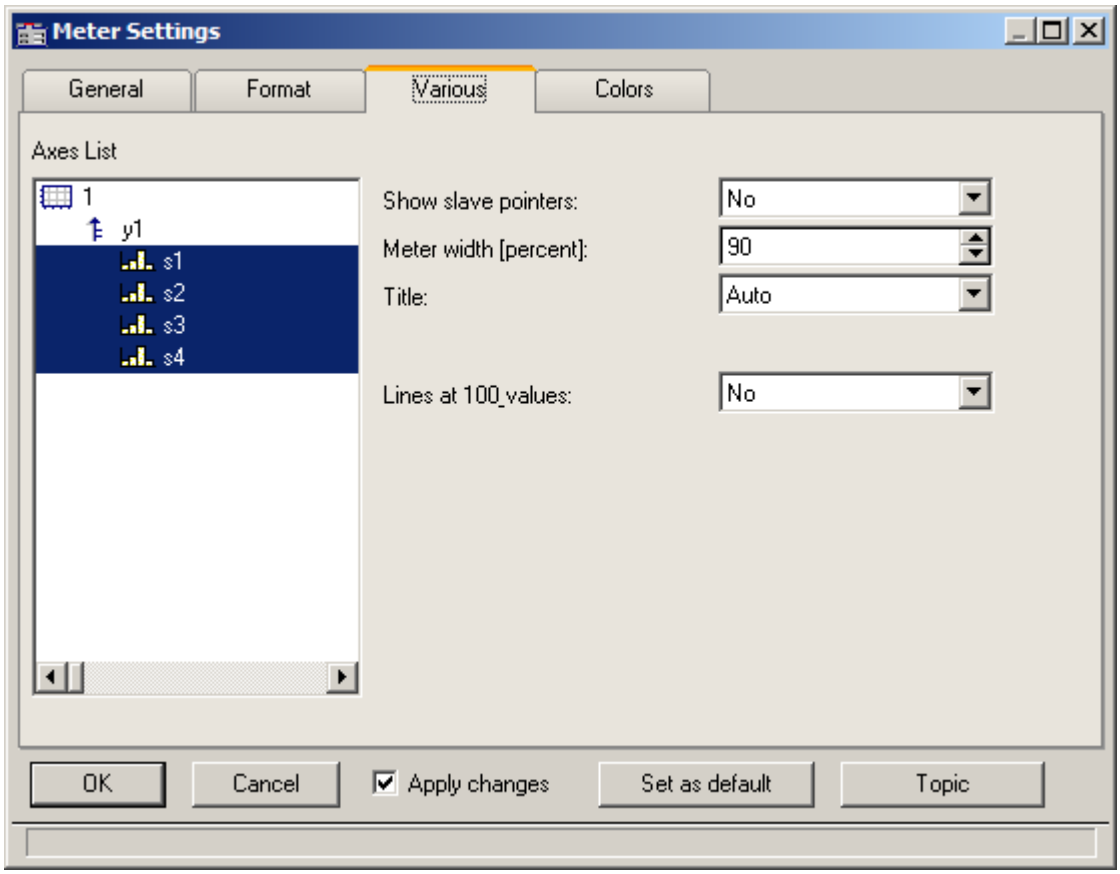
Enter the number N of values, from which the mean, minimum, or maximum is to be calculated. Only integers >= 1 are permitted. A large value for N needs much time for computation, but 100 values is an amount the system can comfortably and quickly handle.

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

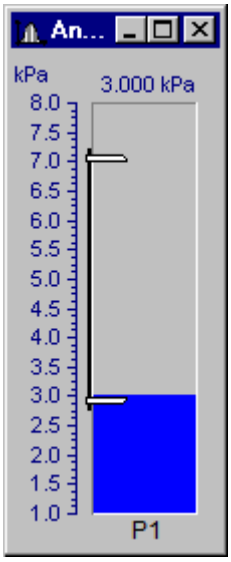
This sub-dialog contains an assortment of setting controls, such as the width of the meter bars, or settings concerning the slave pointer.



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.

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.

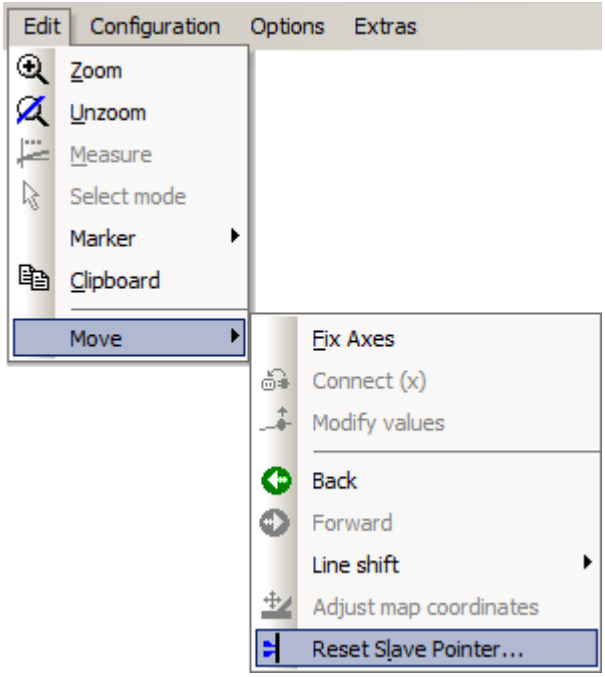
Starting a new measurement on a channel with slave pointers does not reset them. They retain their former value until new data forces them to a new position.

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...*.

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.

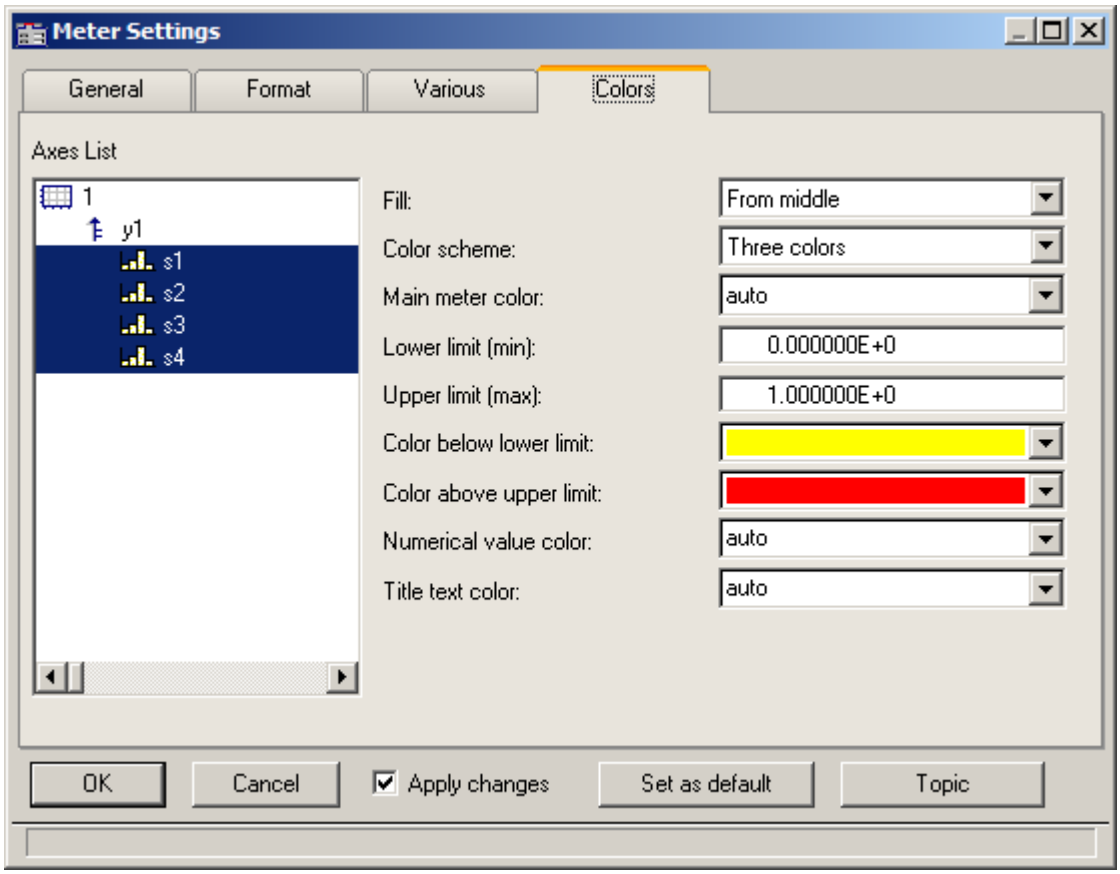
Title

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.

Settings: Colors

This sub-dialog lets you decide on colors for the bars.

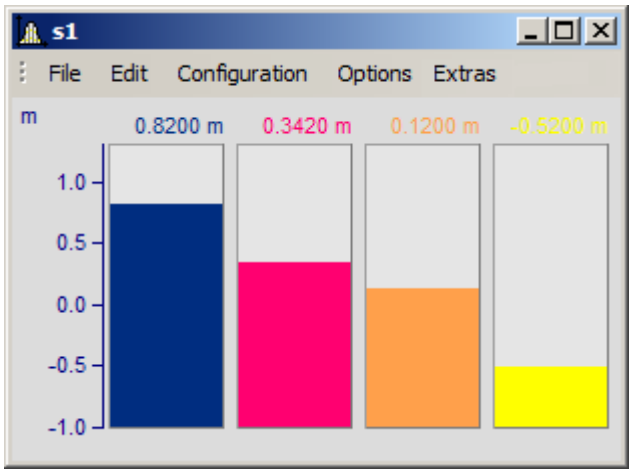


Fill

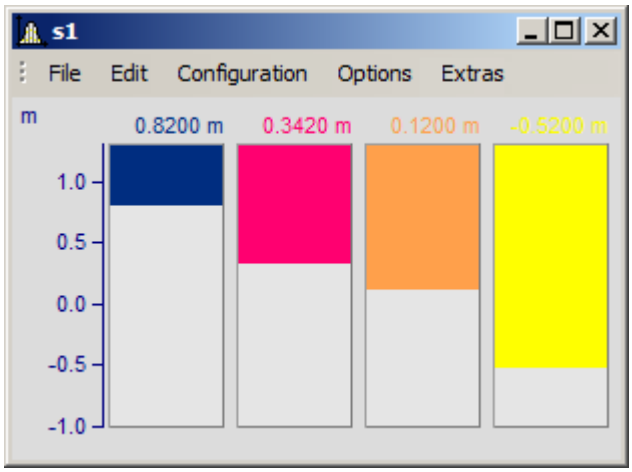
This setting determines from where the meter bars grow with increasing signal measurement values, as represented visually by a column filling with color.

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).

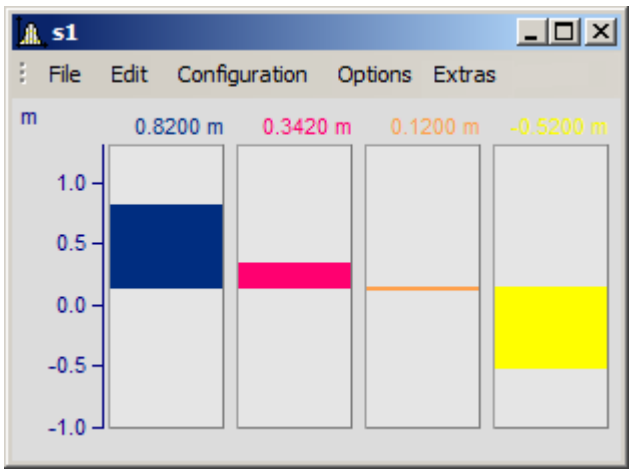
From below



From above



From y = zero



Alternating up and down from the 0 line of the axis

Color scheme

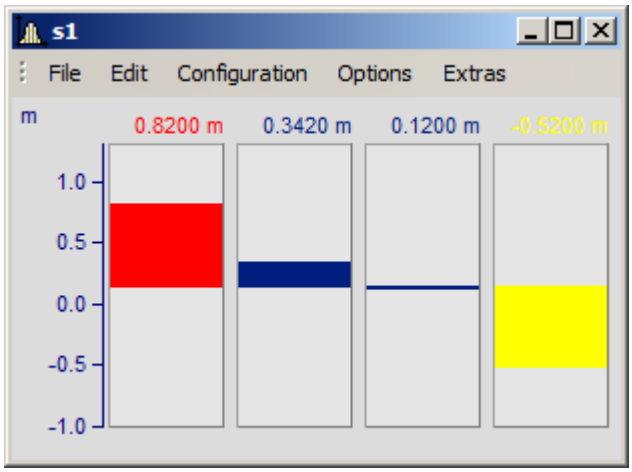
The bar can have only one color, three colors in succession or up to three colors at once.

One color

A one-colored bar has the same color at all times.

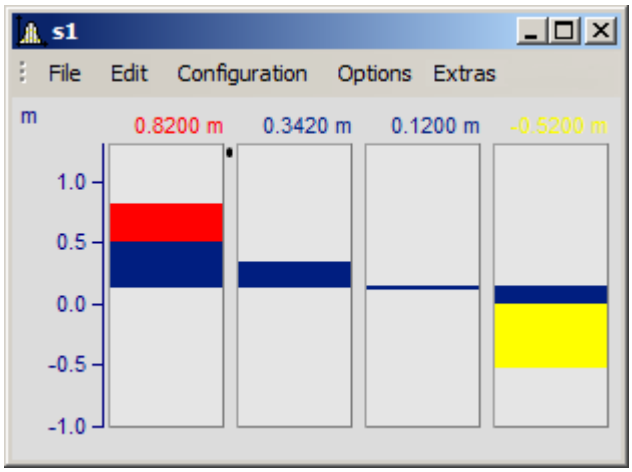
Three colors

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.



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



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.

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 1050 of the curve window.

If a waveform doesn't contain any values, the bar meter remains empty as well.

Limits

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.

Colors

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.

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.

8.6.4.1.1.5 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* 1023.

Prerequisites

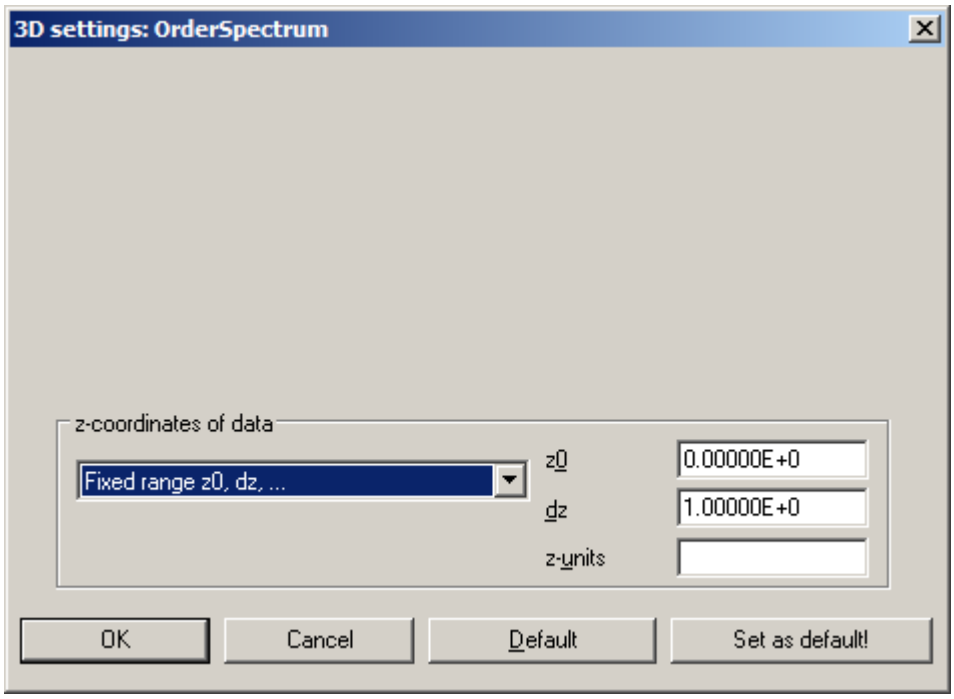
- Color video card with more than 256 colors, in other words 16 bit or 24 bit colors.
- Color printer of DIN A4 format.
- A fast PC (>= 586, 100MHz) with plenty of RAM (>= 32MB)
- Since the graphics display is based on large bitmaps (pixel graphics), large displays in particular require much storage 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

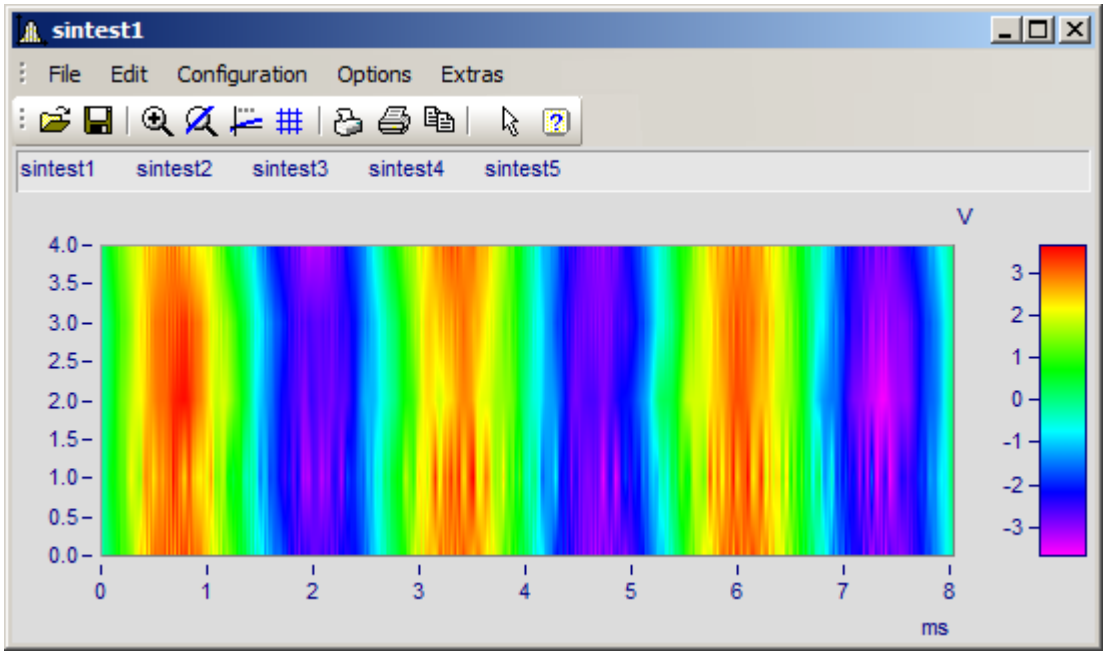
In the dialog: *Configuration / 3D* 959 of the curve window, select an appropriate z-coordinate (see below):



Reference

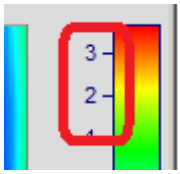
More on the topic of Waterfall diagrams z-coordinate 921

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* 949.

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

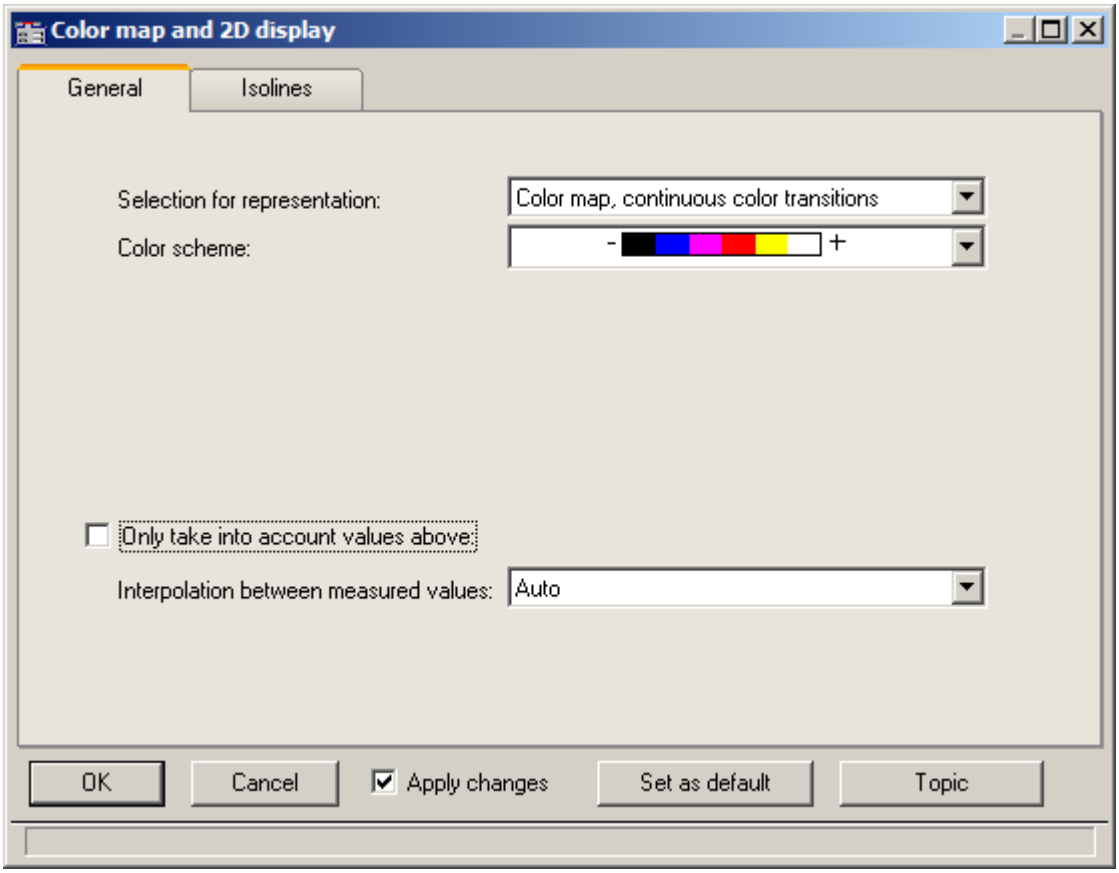
The component waveforms in the map are represented as multicolored horizontal lines, whose x-range is plotted from the x-minimum on the left edge of the graph to the x-maximum on the right edge. The height on the screen of a waveform's line or strip is determined by that waveform's specified z-coordinate; 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.

The options for color map display are set in the dialog called by means of the menu item *Configuration/Color Map*:



Selection for representation

auto

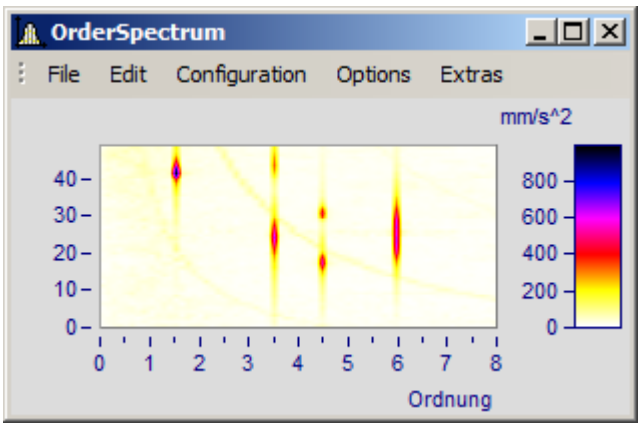
Default colors from magenta to red, or from white to black for B/W display. No other options.

Color map, continuous color transitions

Interpolated color transitions.

For each amplitude to represent, an appropriate color is calculated.

The color scheme can be selected.



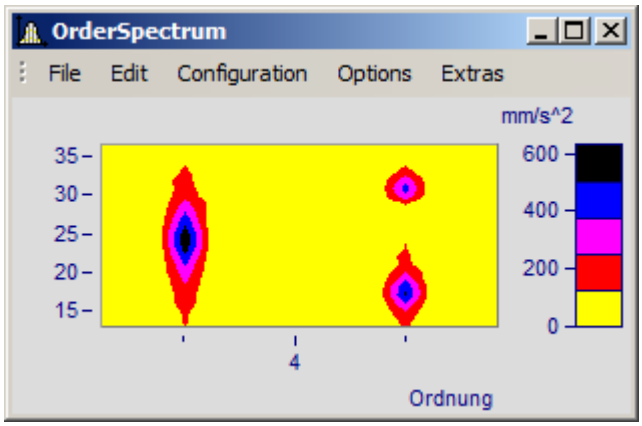
Color map, graduated colors

Specified fixed number of colors without interpolation.

The color scheme can be selected.

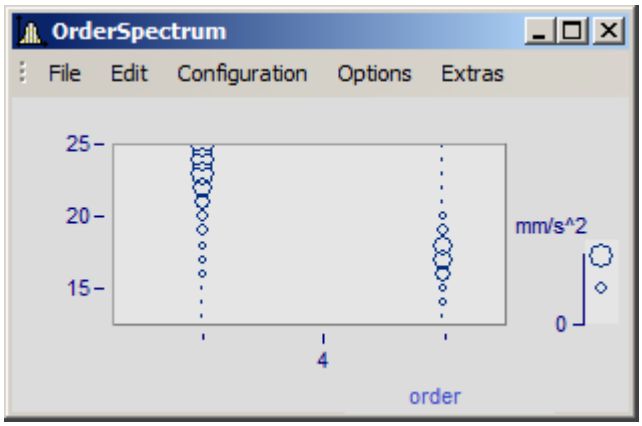
Number of different colors

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.

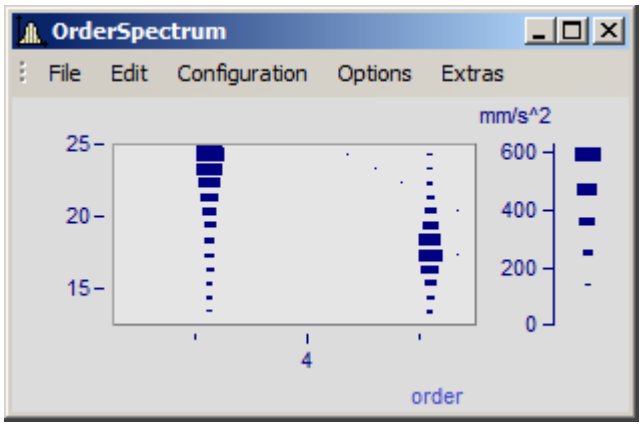
Under *Symbol selections*, ellipses, rectangles and diamonds, either drawn as outlines or filled, are available.

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.
- Symbols, filling corresponds to amplitude

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.

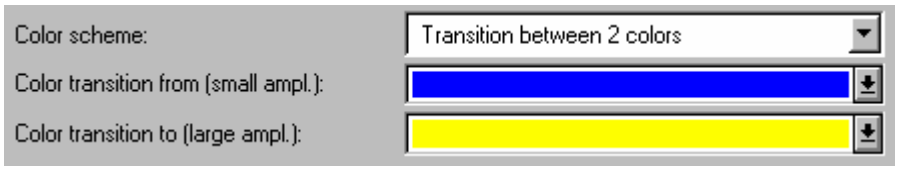
The color scheme can be set for color map display with graduated or continuous color transition.

All color schemes (except *auto*) are independent of the capabilities of the output device.

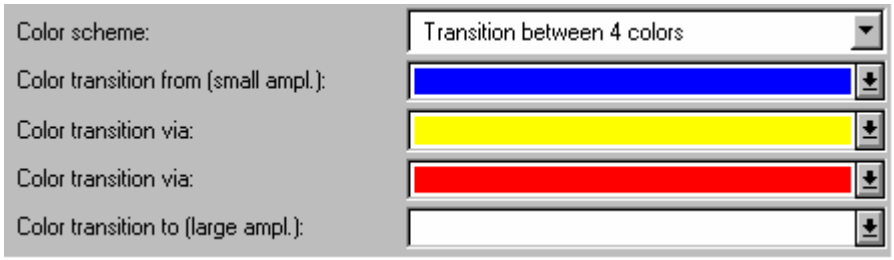
auto: For color output devices, the colors Magenta to Red are used; for b&w printers, the shades from White to Black.

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.

For 2 Colors, the dialog looks like this:



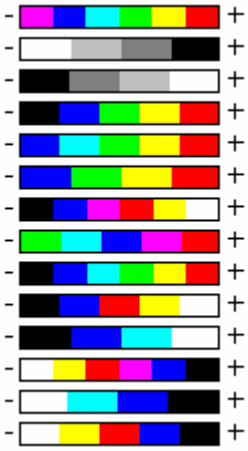
And correspondingly for 4 colors:



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.

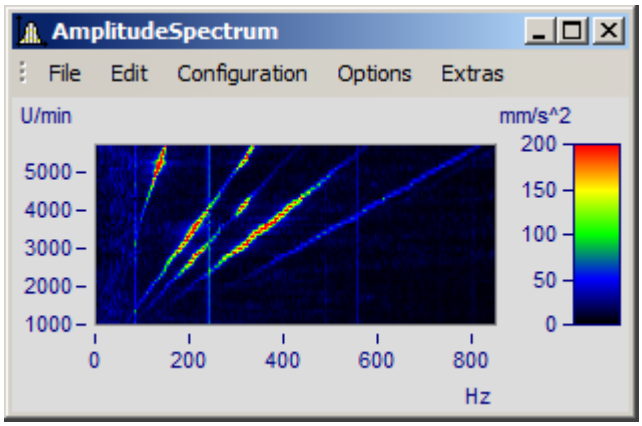
Fixed color schemes: These color schemes are available for direct selection. At left (where "-" appears), the color for the smallest amplitude is shown, at right ("+") the color for the highest.



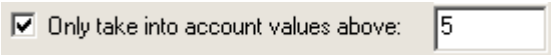
Only take into account values above

The options depend on what display type is selected.

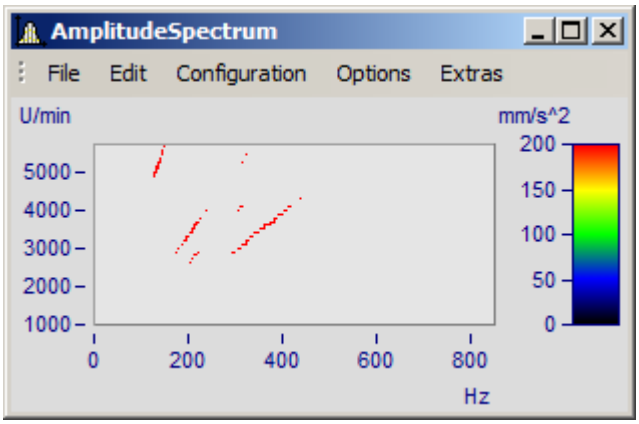
If this option box is not check-marked, all readings are taken into account (default).



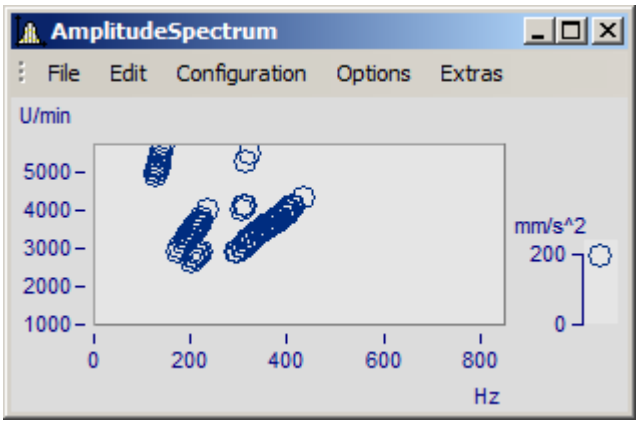
If the checkbox is marked, then a lower limit for the amplitudes to be displayed can be specified.



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)*

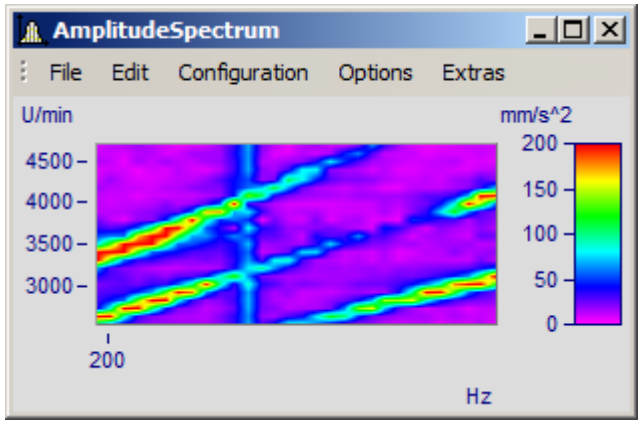


Interpolation between measured values

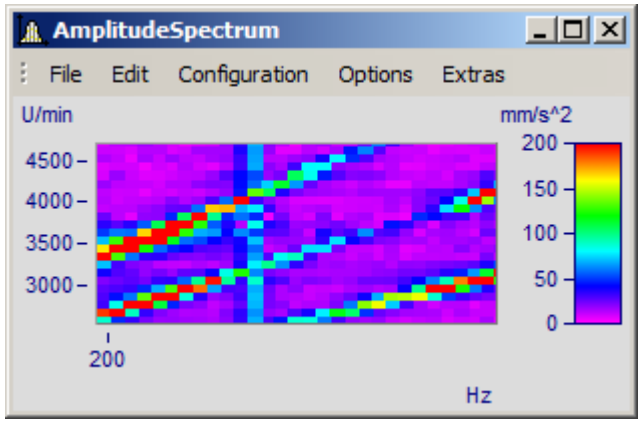
The options depend on what display type is selected.

This refers to interpolation in xy-plane between readings. Not to be confused with interpolation in the z-direction (interpolation of colors).

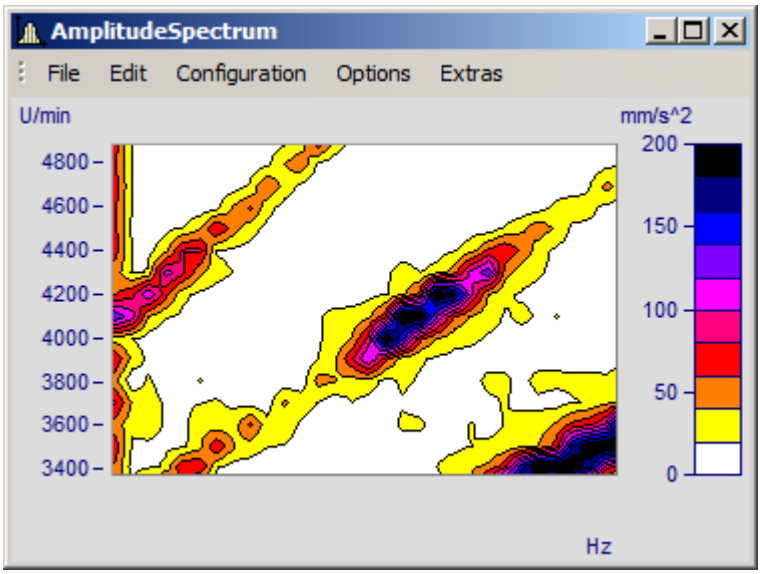
auto, linear: In the x- and y-directions (horizontal and vertical on-screen, respectively) interpolated values between the specific readings are determined. This corresponds to linear interpolated lines between plotted points.



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.

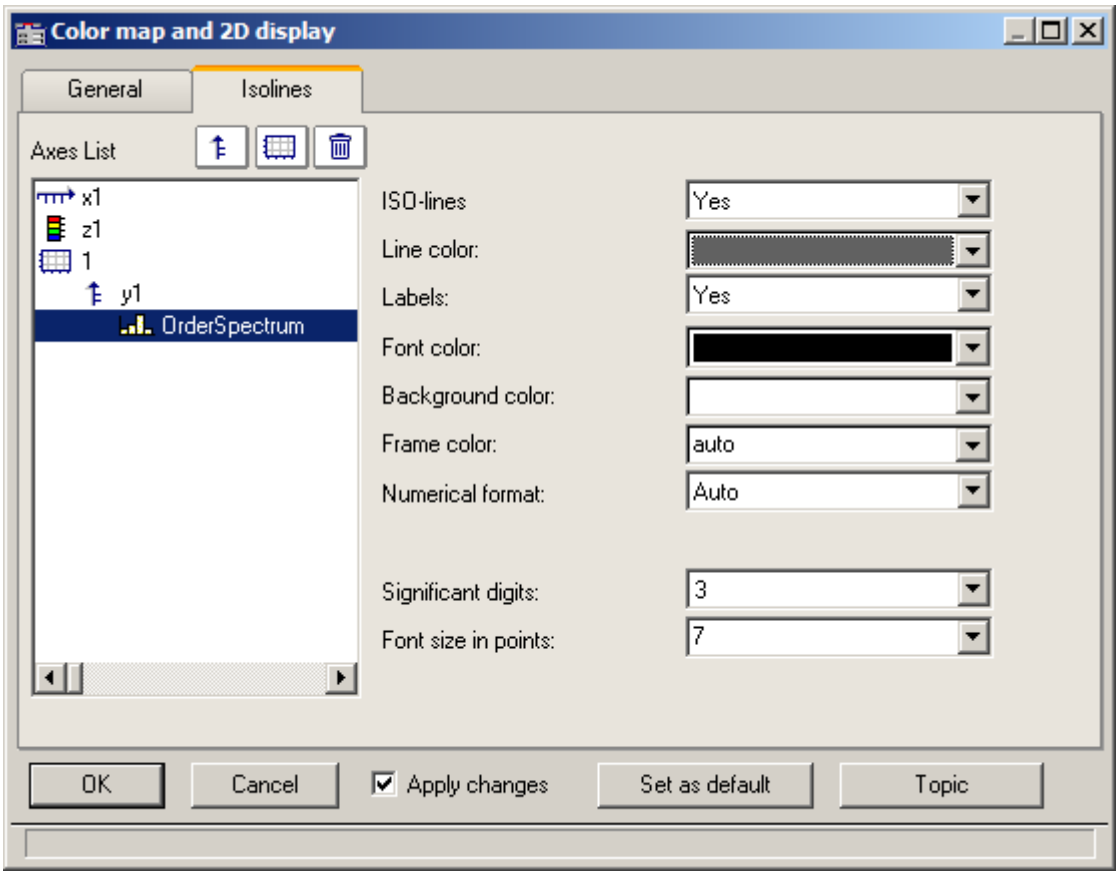


ISO-lines are contours, in other words, lines joining coordinates of equal height. They are used here in the same way as in maps. Since the lines appear in conjunction with graduated colors, they mark the boundaries between the colors.



ISO-lines can only be set for *color map display with graduated colors* 950.

The *Isolines* index card of the Color map settings dialog:



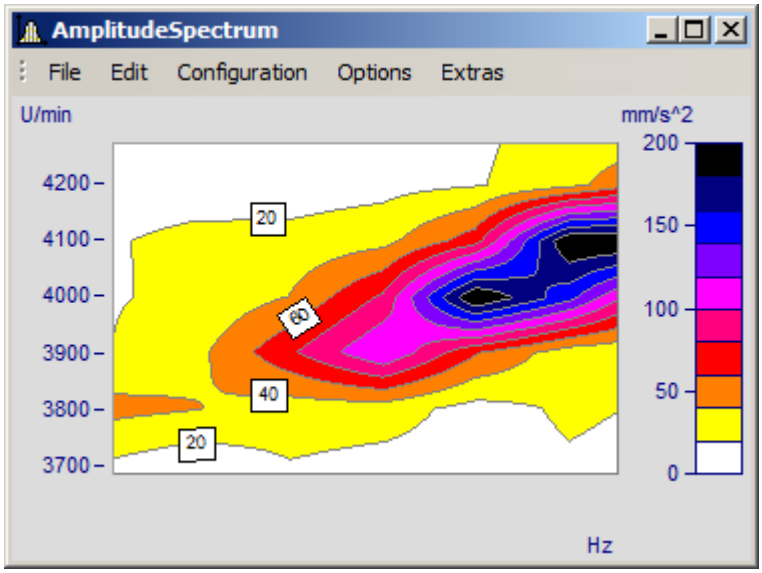
The following options can be selected:

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.

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.

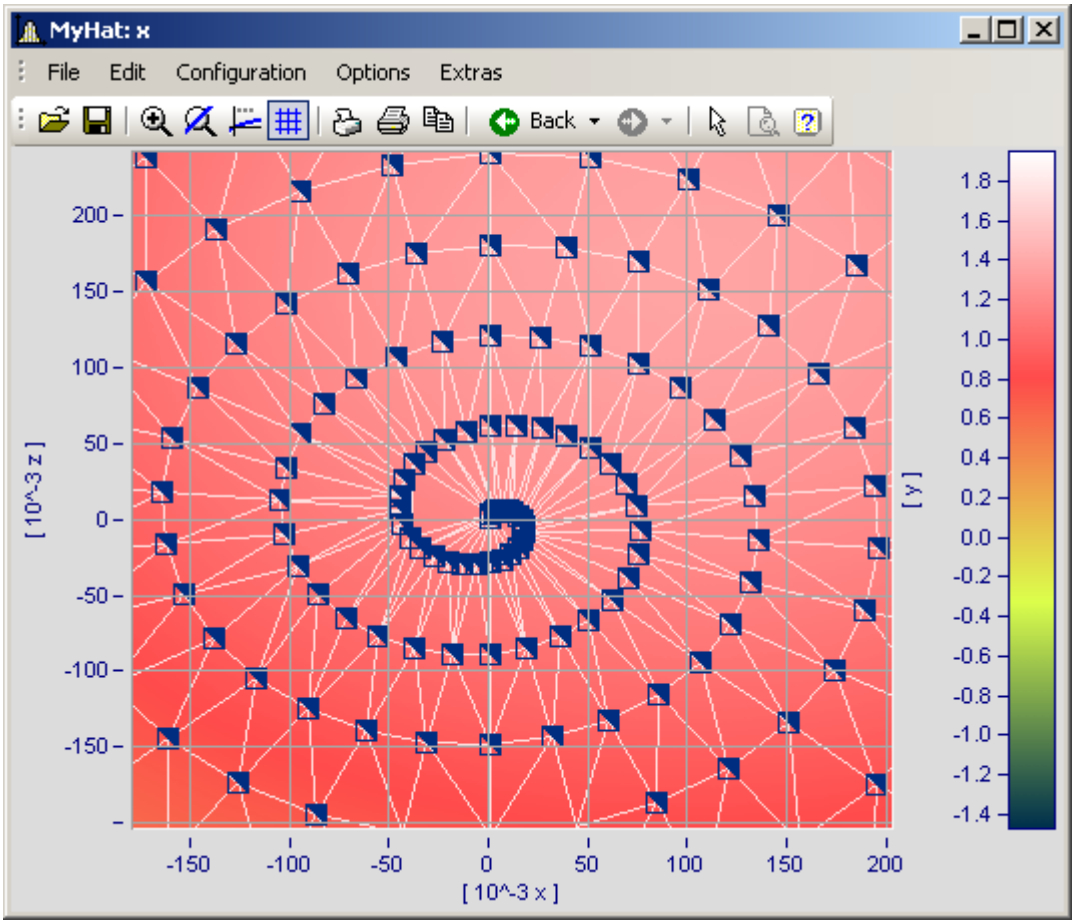
Valid digits: For *numerical format auto* the number of decimal places cannot be set, instead, the number of significant digits. Thus, the numbers 3.4, 3.4E-4 and 0.034 each have 2 significant digits.

Font size in points: The font size in points (pt): 4pt to 10pt. Typical values are 6 and 8 points.

Reference

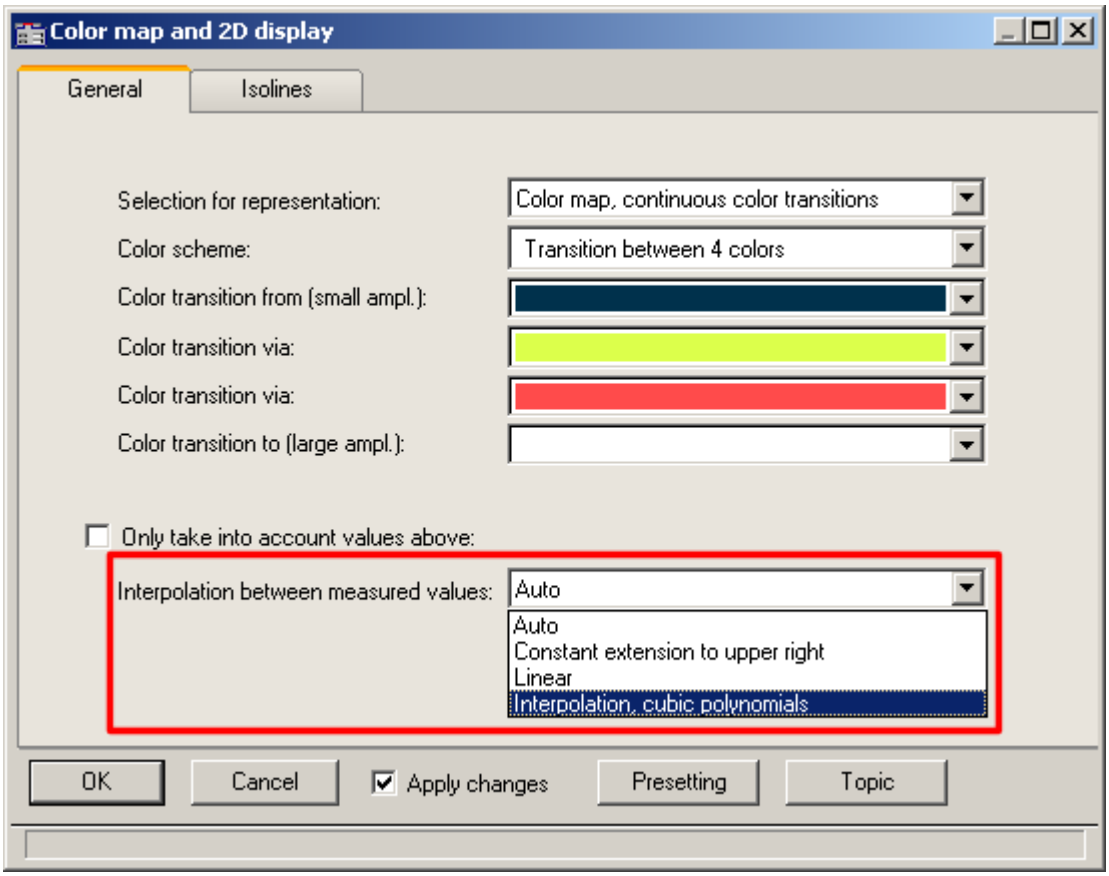
See chapter Waterfall Diagrams 921

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

8.6.4.1.1.6 3D Display

Another display style for datasets with associations of x, y, and z variables 1002, for segmented waveforms is *3D* display.

In this display style, the perspective can be freely rotated, and the Axis Navigation Bar 1073 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.

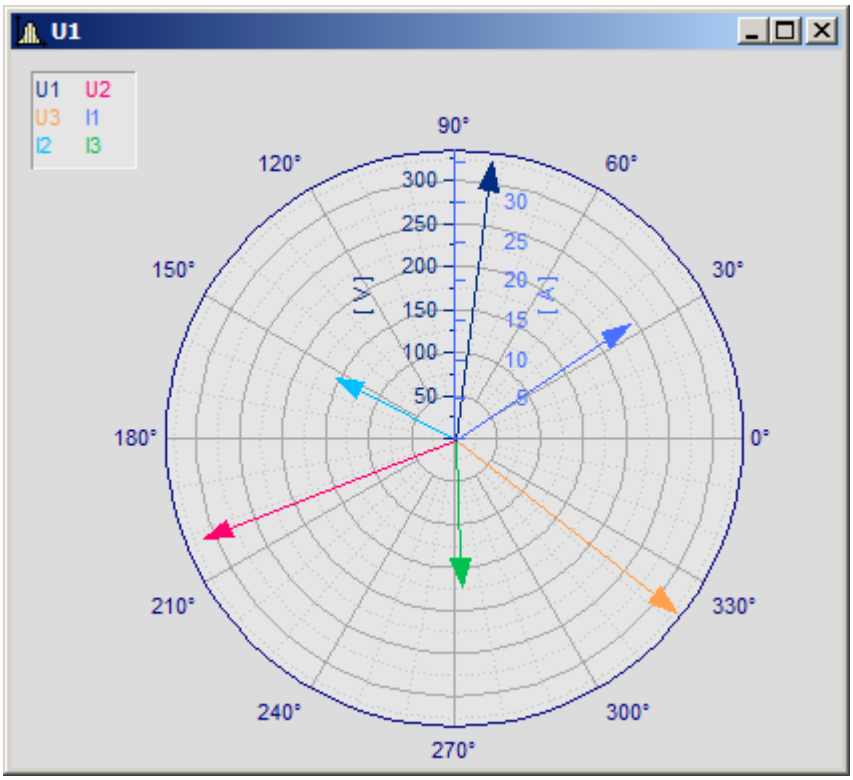
Other curve window functions such as *zooming*, *setting markers* or *line-shifting* are not available here.

For more on using 3D-display, see the section on *Configuration / 3D* 1027.

8.6.4.1.1.7 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



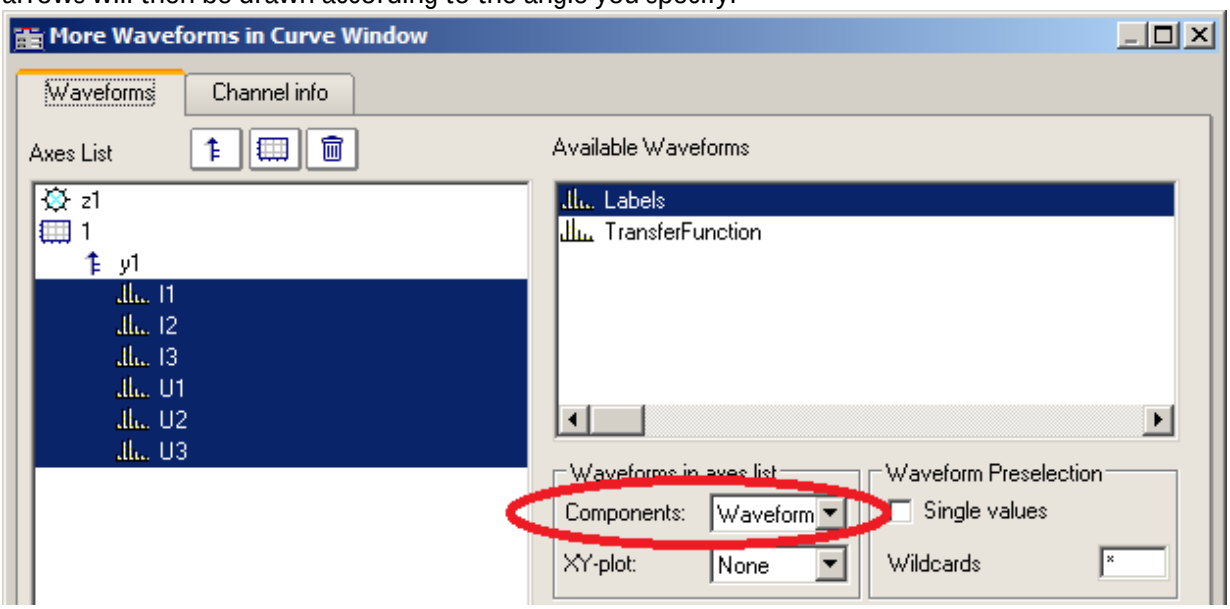
Complex single values as vector diagram

In imc FAMOS you can create a complex single value using the function Cmpl().

The pointer graph above is created as follows:

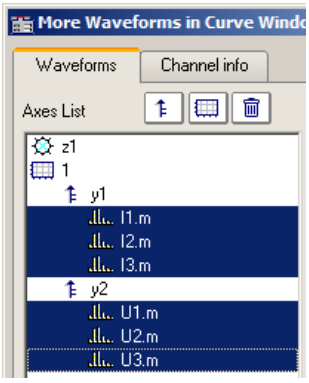
1. Create the desired single values:
   I1= Compl (27'A', 33'Degr')
   I2= Compl (17'A', 153'Degr')
   I3= Compl (19'A', 273'Degr')
   U1= Compl (327'V', 82'Degr')
   U2= Compl (315'V', 202'Degr')
   U3= Compl (331'V', 322'Degr')
2. Display the single values in a curve window and select the display style *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.

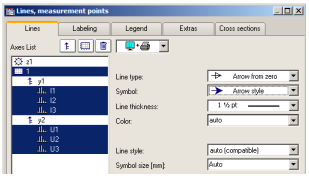


Observing each component of a complex waveform

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...*

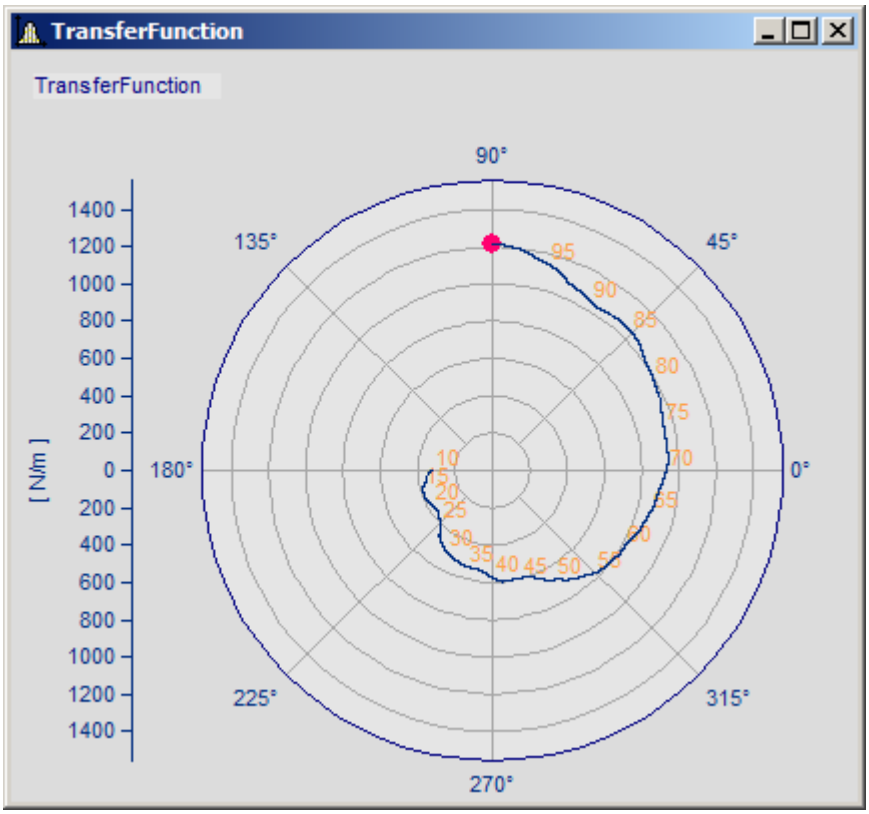


5. Using the Lines dialog, you can now adapt the display of the arrows. Right mouse button: *Lines*



Dialog lines: Display with arrows

Example: Transfer function



Example transfer function

There are further possibilities for displaying a complex data set:

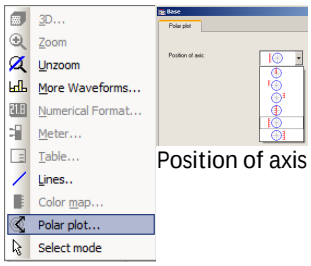
1. Use imc FAMOS to create a complex data set:

```
; 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.

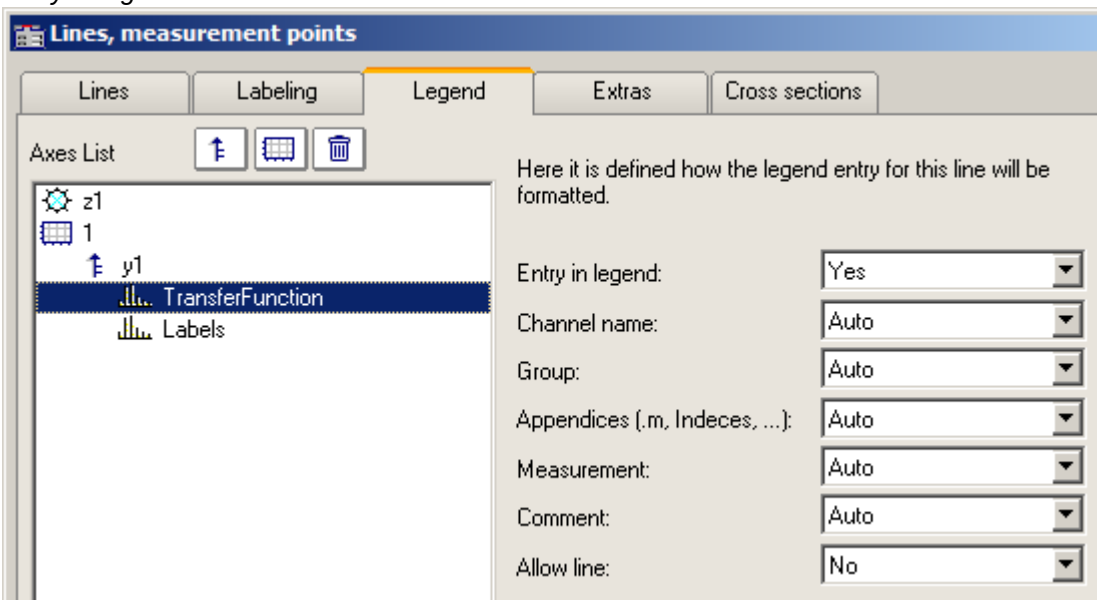
   Select the right mouse button: *More Channels...* and assign *Labels* to the same axis.

   Click the button *Topic* and select *Lines*.

   Go to the page *Labeling* and activate it.

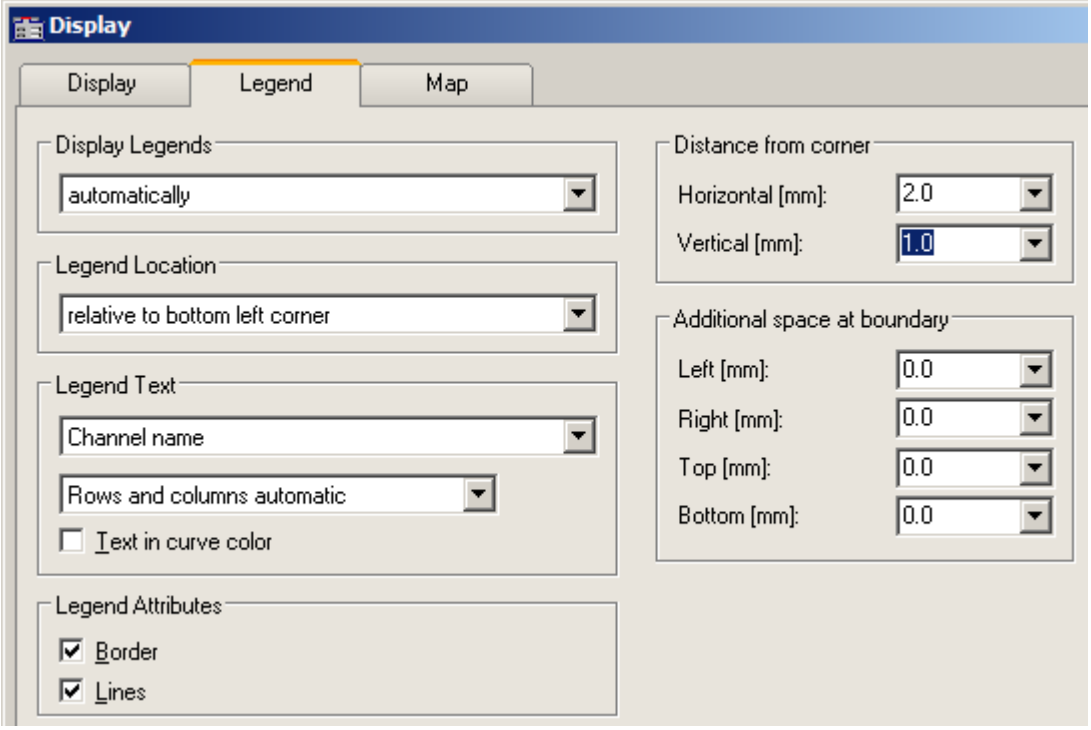
   By default, *Labeling* displays the magnitude. In our example we wish to display the frequency. For this reason, select for *Value selection* the entry *Parameter*.

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.

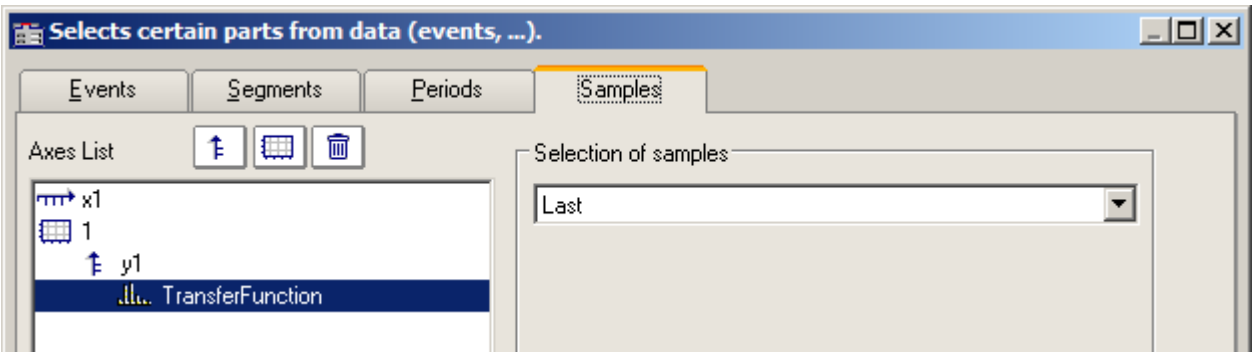


Position of legend

Notes on online display

For measurements with *imc DEVICES* or *imc STUDIO*, 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...* 1002.

If one wishes to display the complex data as a pointer diagram 960 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*.



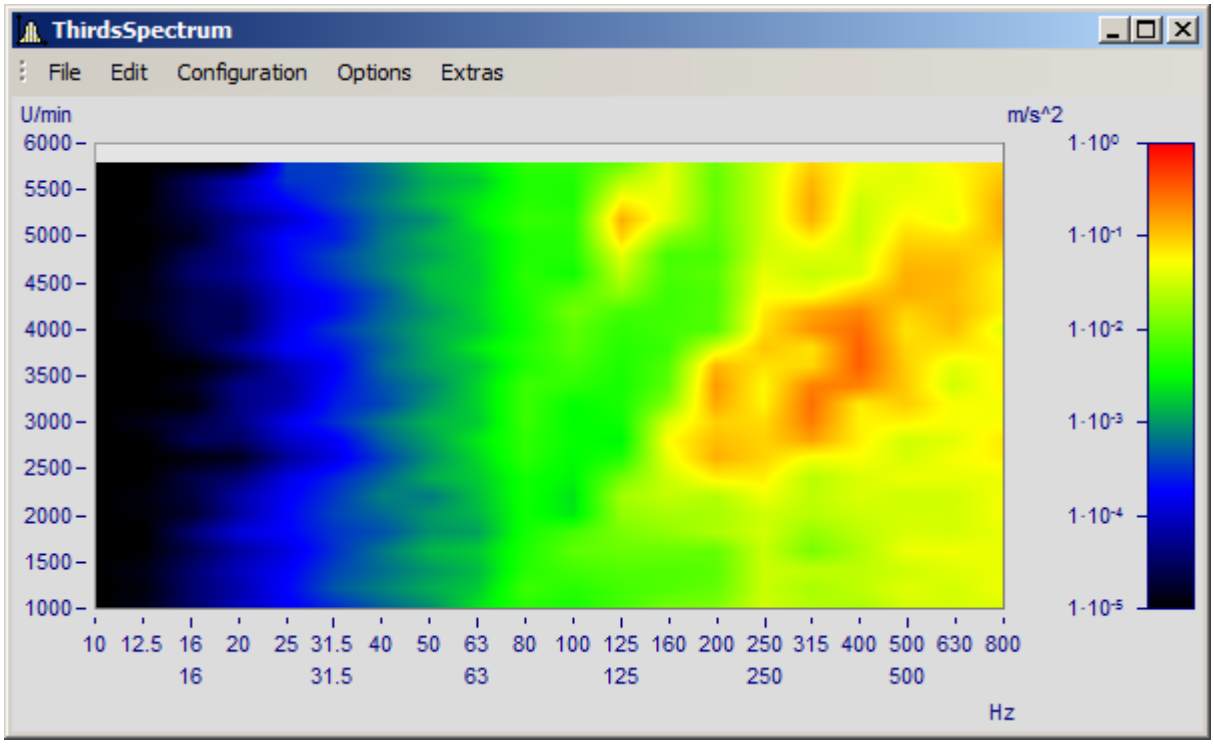
Displaying only the last sample

8.6.4.1.1.8 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.

An example for 1/12-octaves:

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
..	..	..

An example for 1/24-octaves:

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
..	..	..

© 2017 imc Meßsysteme 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.

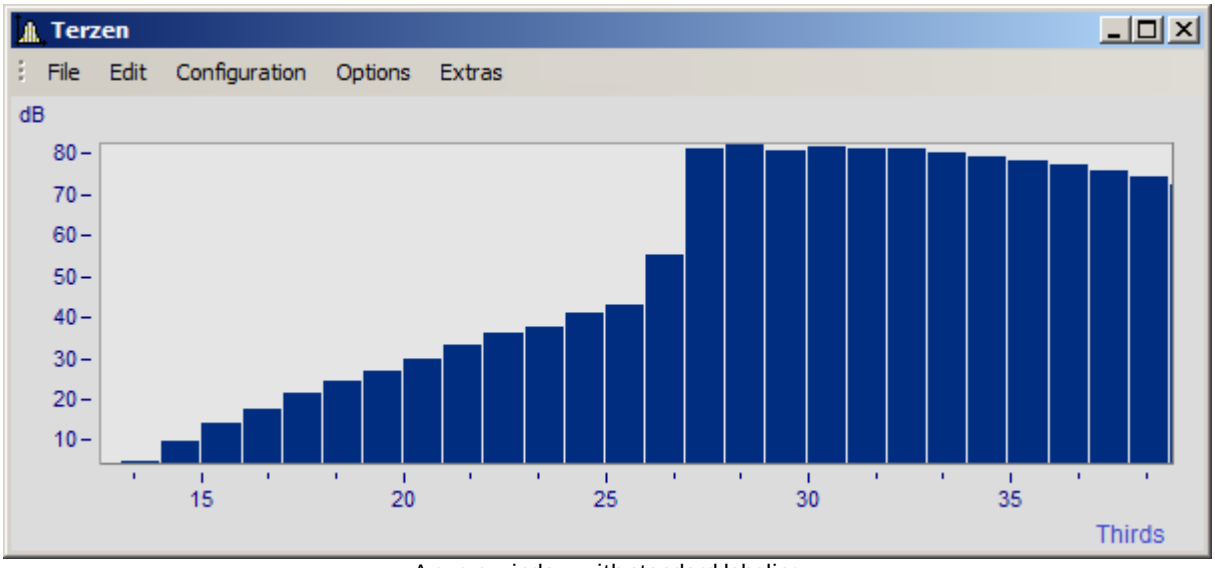
x-scaling	Center frequency (Hz)
	..
-3	0,5
-2	0,63
-1	0,8
0	1
1	1,25
2	1,6
3	2
4	2,5
5	3,15
6	4
7	5
8	6,3
9	8
10	10
11	12,5
..	..
20	100
30	1000
40	10000
41	12500
43	20000
..	..

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.

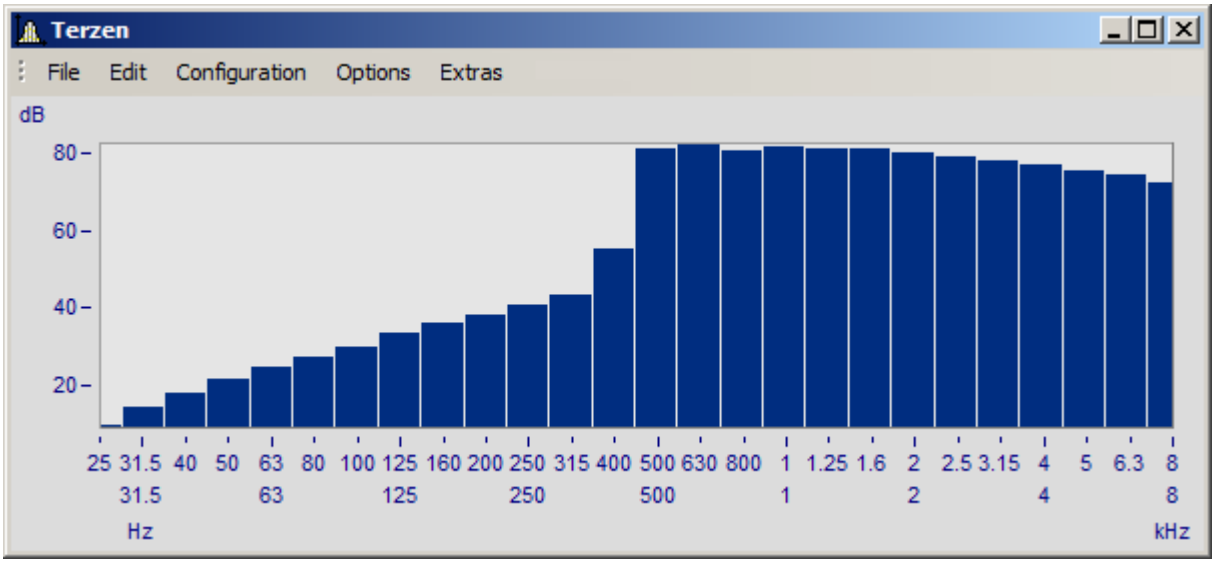
This means that the various bandwidths are expressed as a multiple of a 1/3-octave. Therefore, the delta-x of the x-axis is set to the following values for frequency scaling:

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

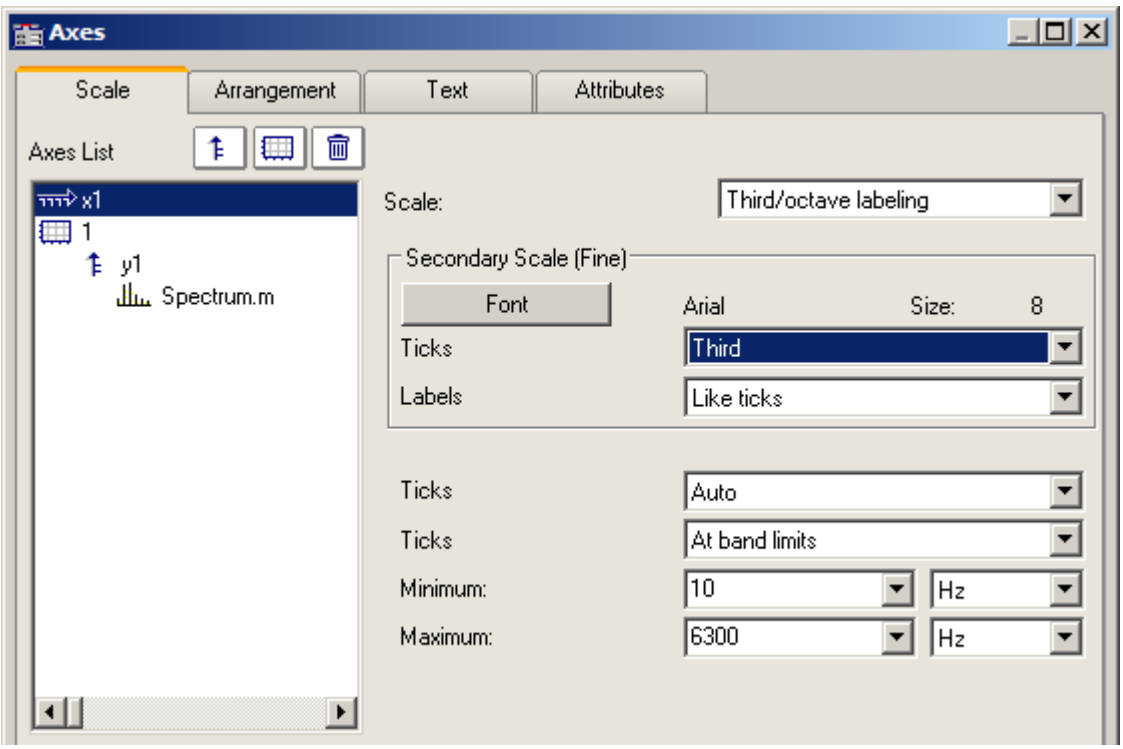


In 1/3-octave / octave labeling

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.

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.

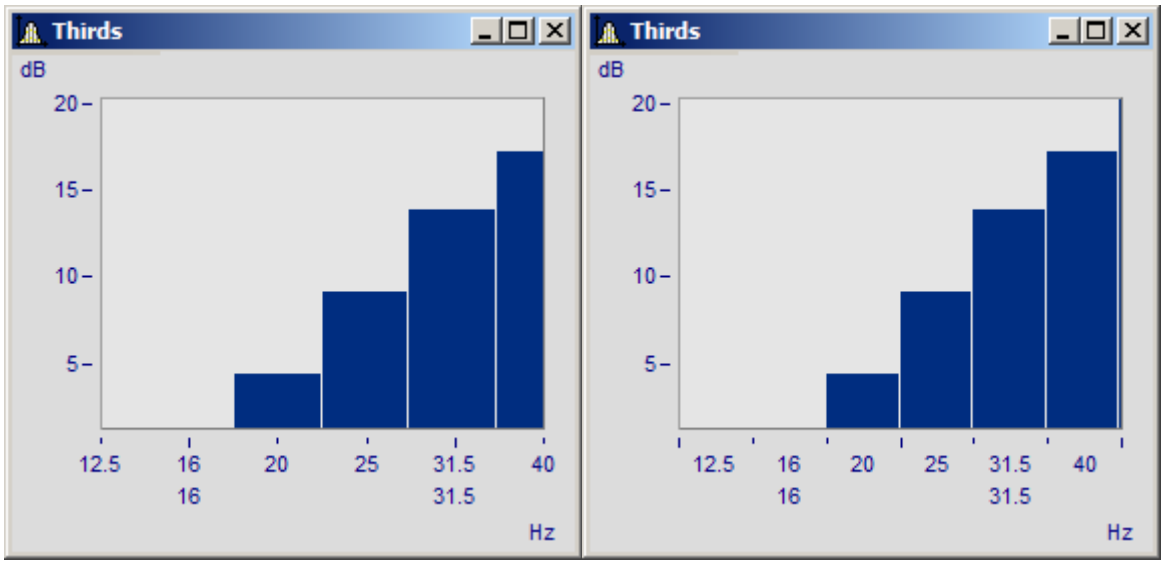
Ticks

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

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.

8.6.4.1.1.9 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.

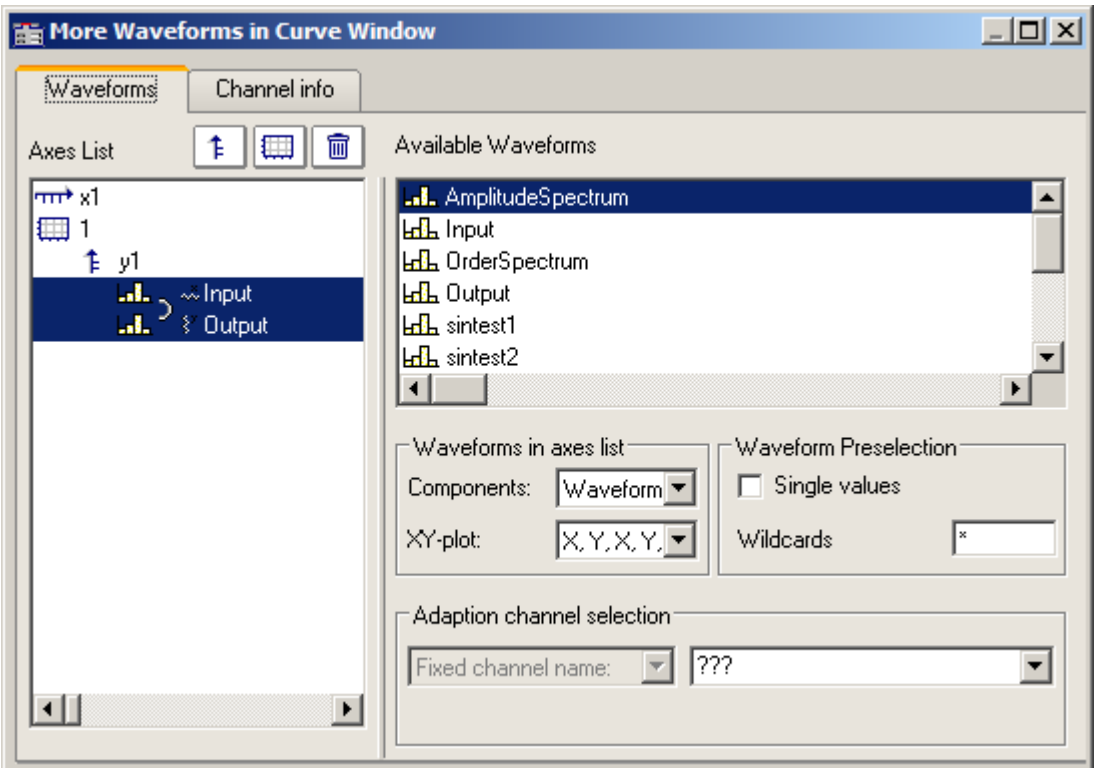
The curve window is also able to display XY-plots, meaning plotting one waveform's values in correlation to another waveform's values.

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* 995 (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.*

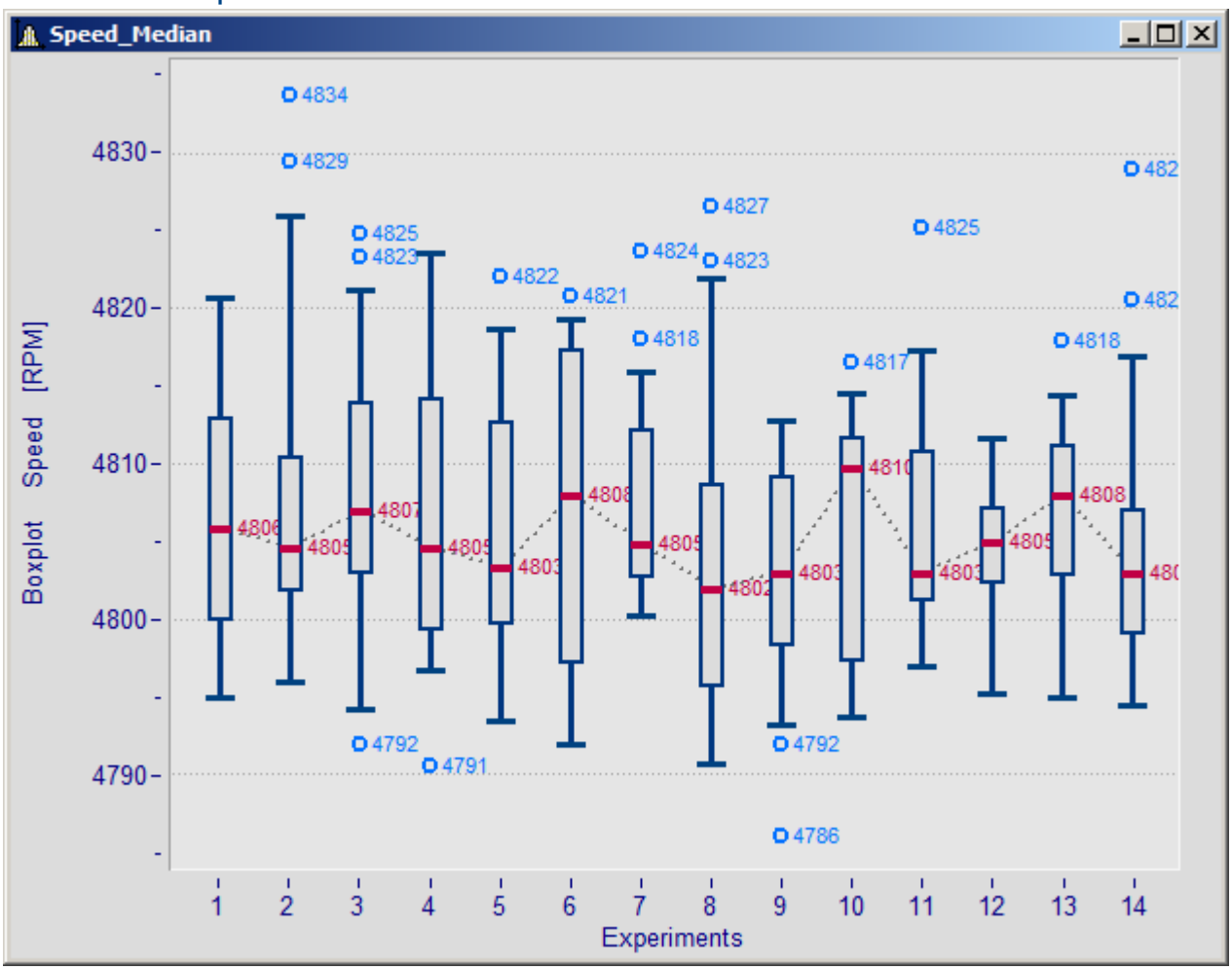


Remarks

- In all XY-plots, the measurement cursors follow the parameters during operation in "Measure" mode. See chapter section '*Measure* 867'.
- 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.

8.6.4.1.1.10 Boxplot

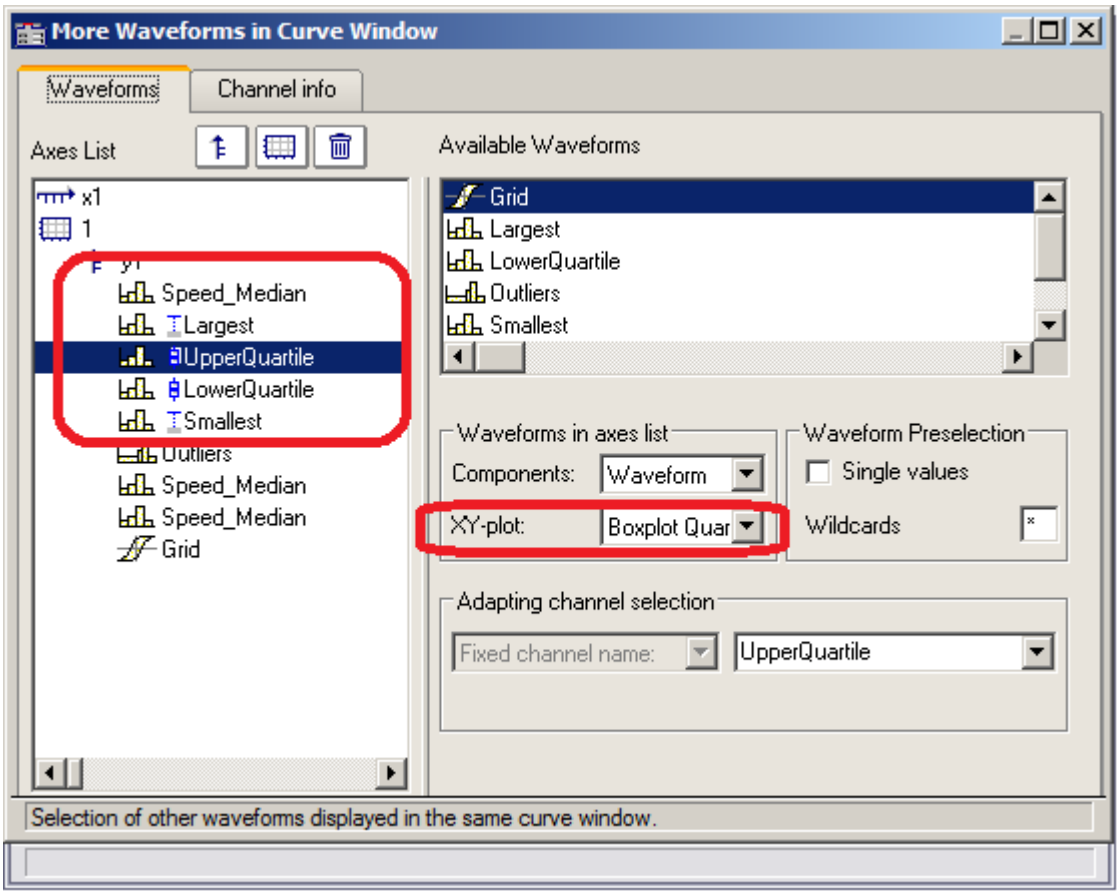


The Boxplot (also called Box Whisker Diagram) depicts a statistical distribution by displaying the minimum value, the maximum value, bottom quartile, top quartile and the median.

In this display, the minimum and maximum values are represented by whiskers, the quartile boundaries as a box, the median as a line.

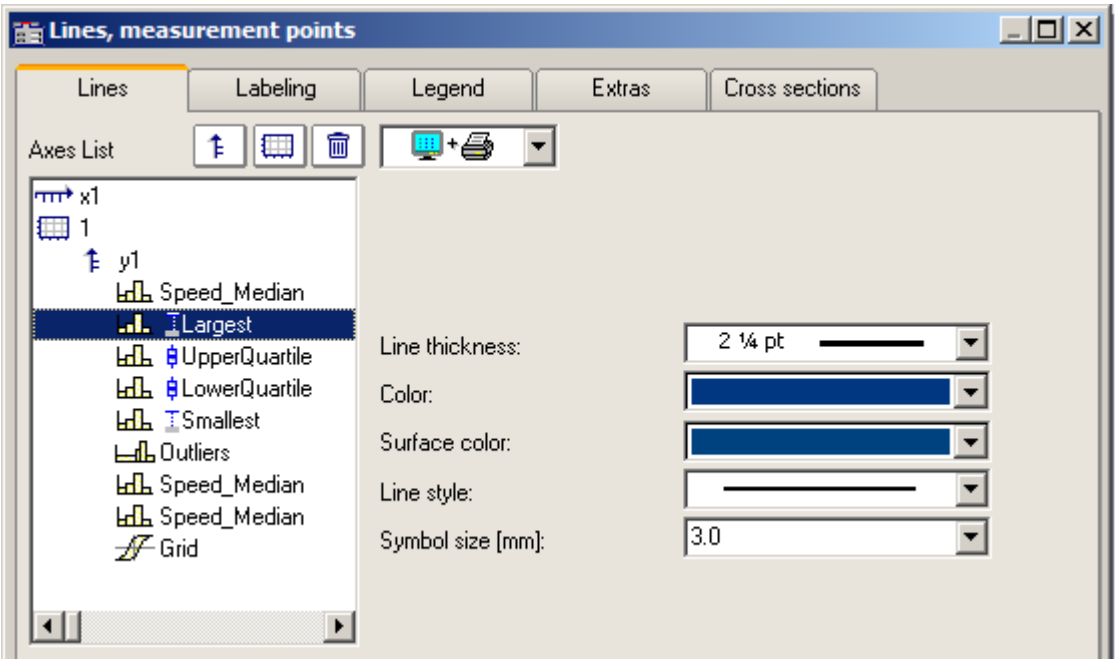
For a curve window display, each of these statistics is saved in a separate channel, e.g. one channel with the median values, another channel of maximum values...

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.

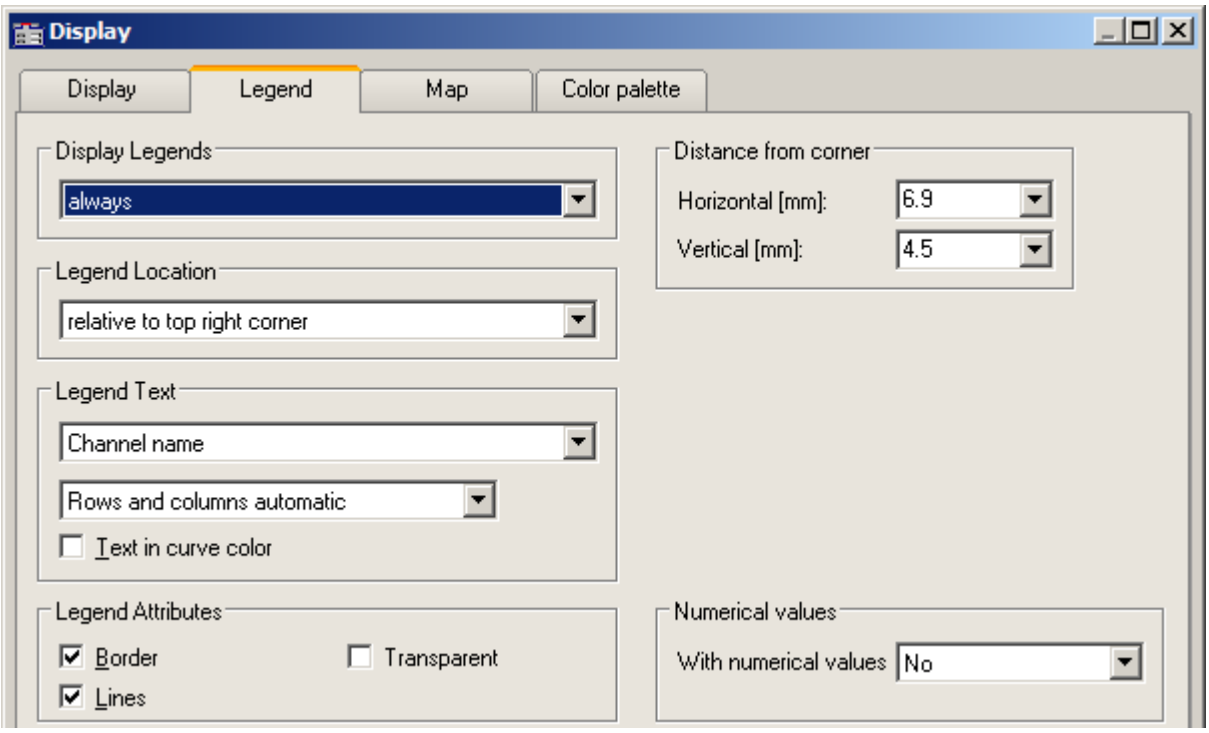
The display is performed point by data set point. All data sets must have the same sampling rate. The median itself may be an XY-channel, but all others may only be normal, equidistant data sets (it may be necessary to display the .y-component alone). In the case of an XY-channel, the sampling time of 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.

8.6.4.1.2 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

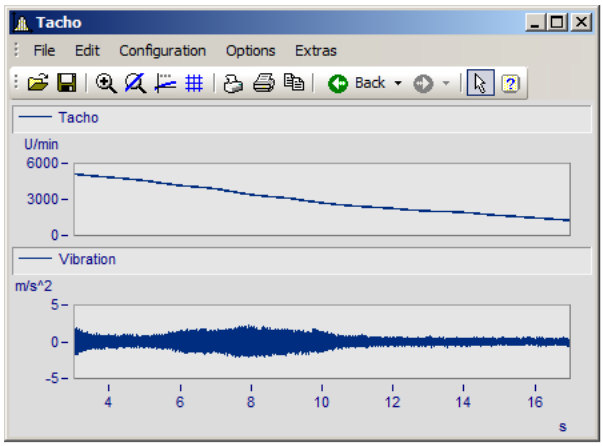
Specify here the conditions under which a legend is to appear. The following options are available:

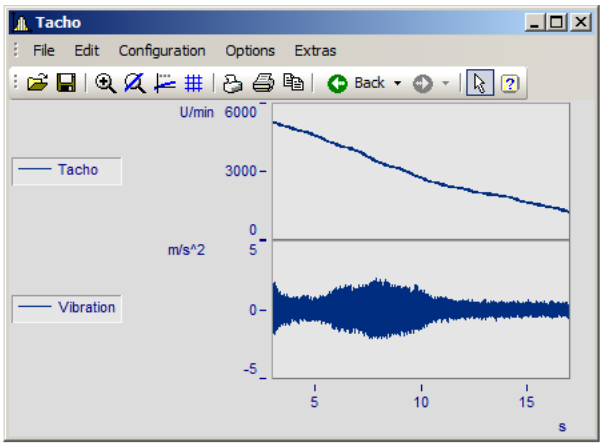
- *Automatically*
- *Always*
- *Never*
- Only if more than one curve is present

Legend Location

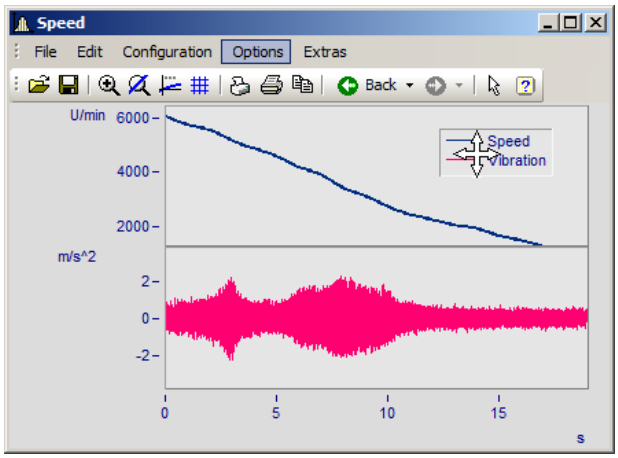
By default, the legend for all variables appears above the curve.

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

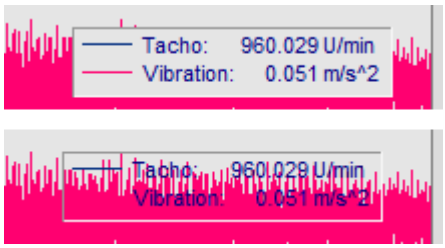
Specify here the text which is to appear in the legend, as well as the format and color of the text. The following options are available:

- 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 1069.
- 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

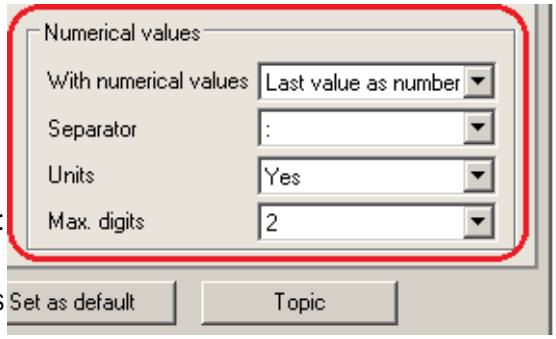


The option *Border* places the legend in a frame, which provides a 3D-effect.

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.*

Separator inserts a colon or equal sign.

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.

OK, Cancel, Set as default

For details on operation, see the chapter Confirmation bar 1043

8.6.4.1.3 Map

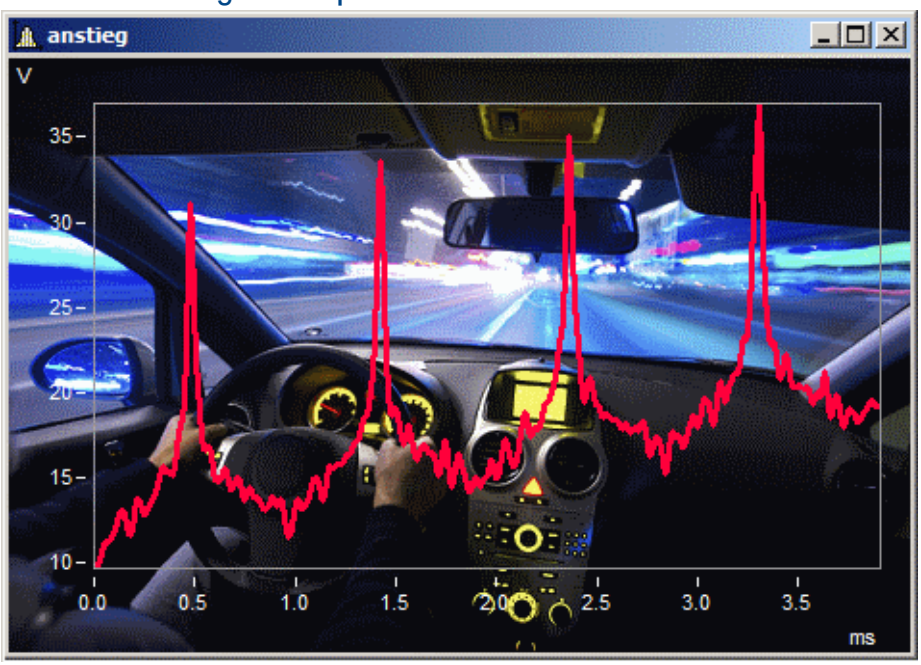
Function

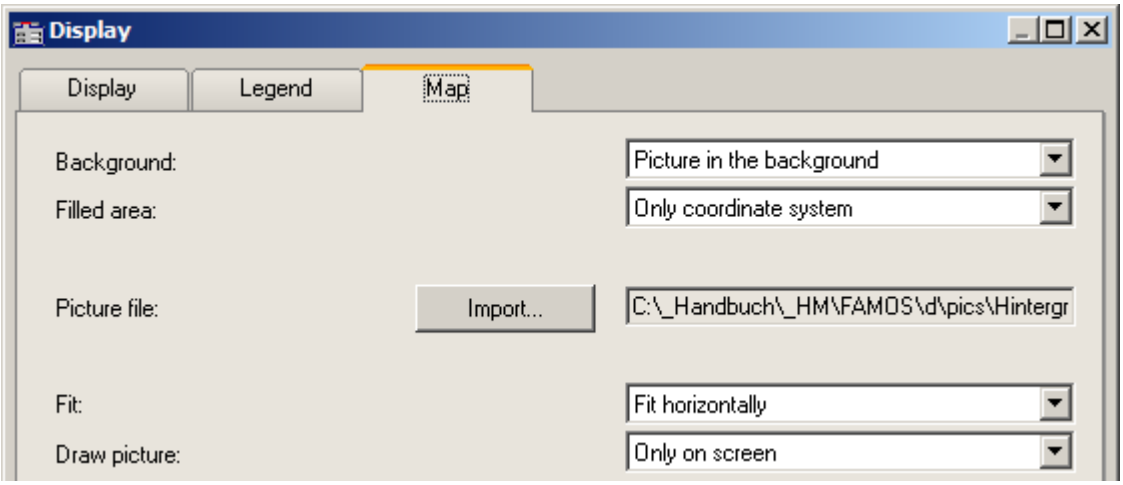
Map lets you place a picture in the background. It is additionally possible to layout the picture in specified coordinates.

Mouse Operation

In the dialog called by the menu item *Configuration / Display*, select the option *Map*.

8.6.4.1.3.1 Background picture





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.

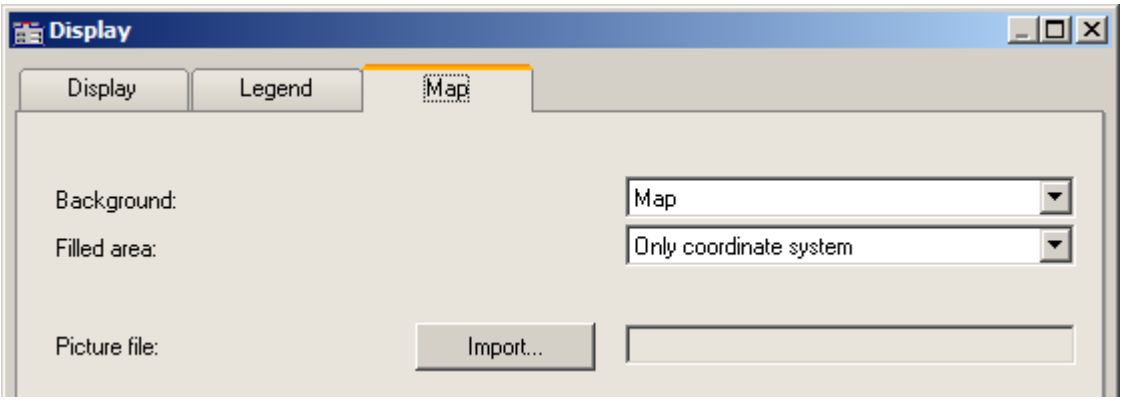
8.6.4.1.3.2 Map

Positioning a map as the background picture

On the following conditions, a picture can be correctly displayed as a map:

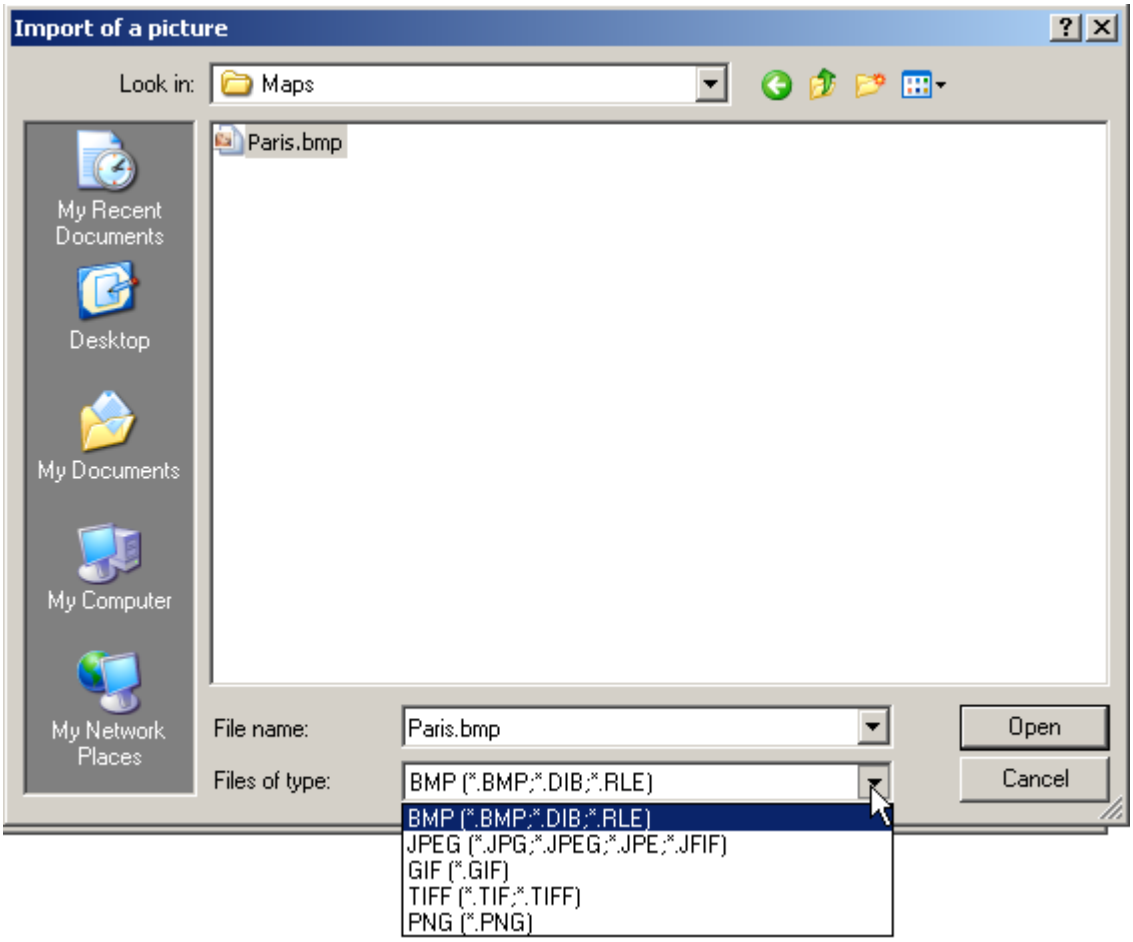
- selected curve window configuration: *Display \ Default*
- linear axes
- only one coordinate system

The following example illustrates how GPS data are linked with a map.



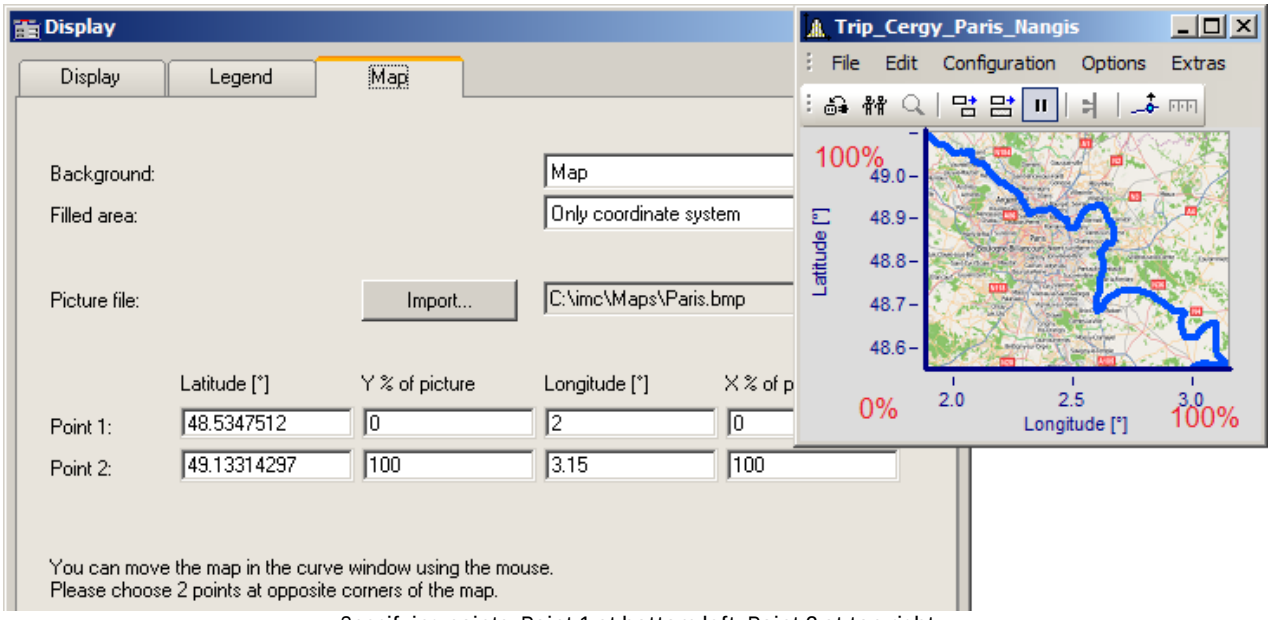
Select the "Map" page and set the background to "Map"

Click on the button *Import* and select a *picture file*. Specify its format.

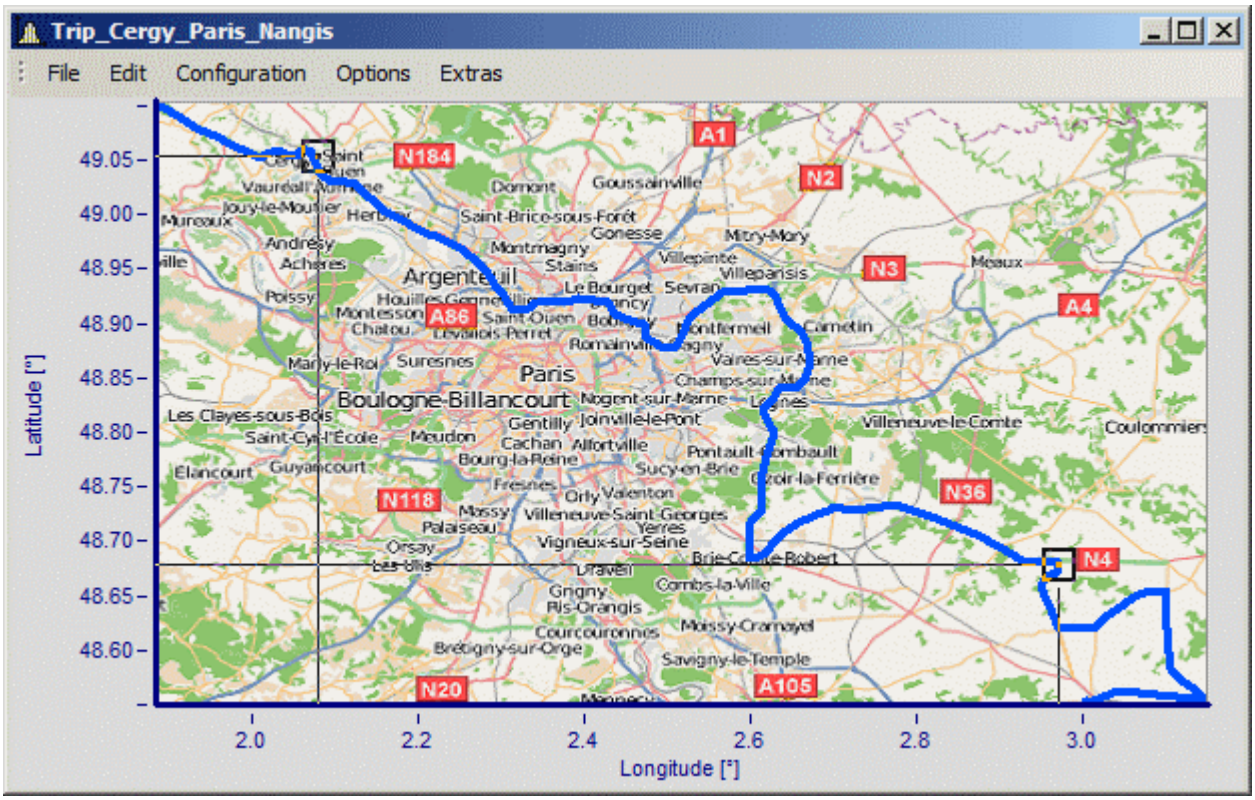


Selecting a picture file. All customary graphics formats are available.

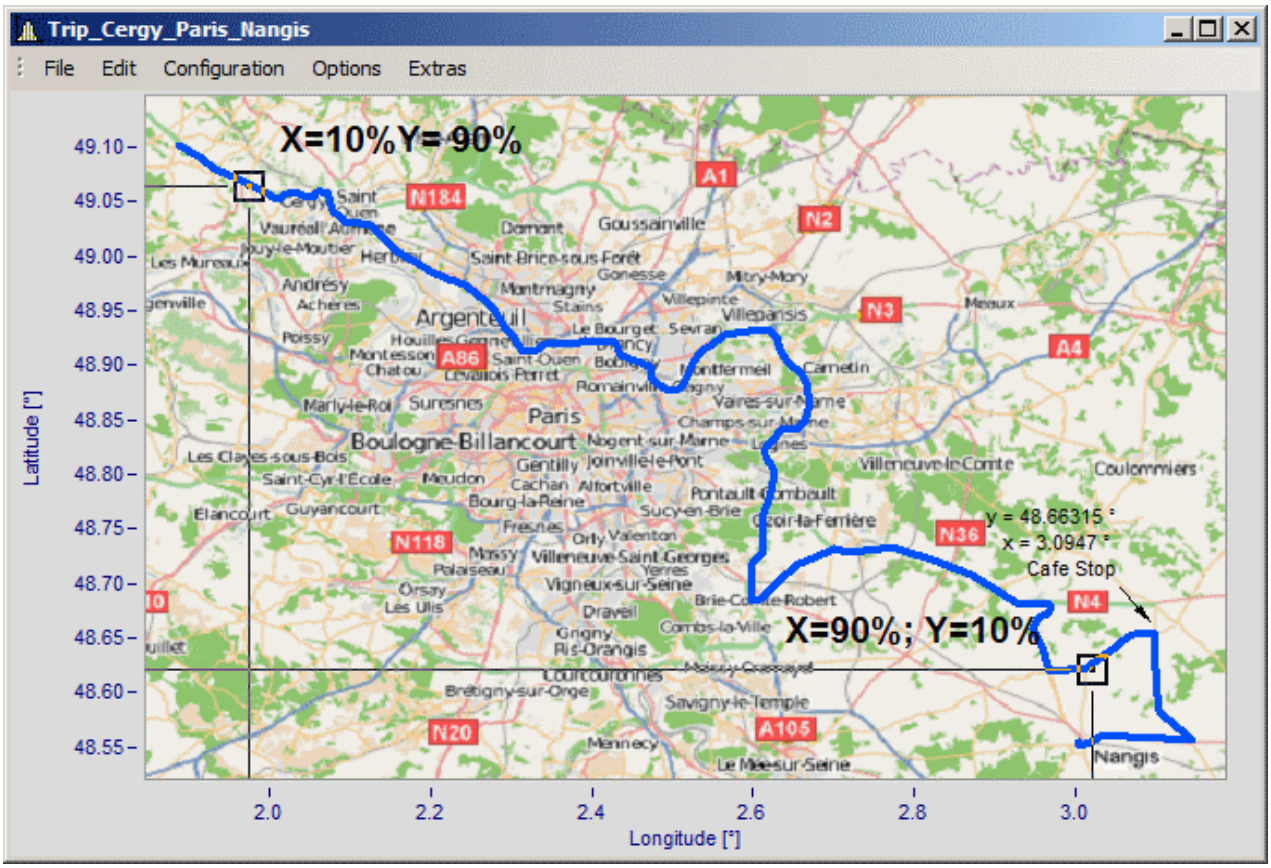
To position a map, you must specify two points. This accomplished by making entries in the input boxes. The curve window is then in Map Mode.



Specifying points: Point 1 at bottom left, Point 2 at top right



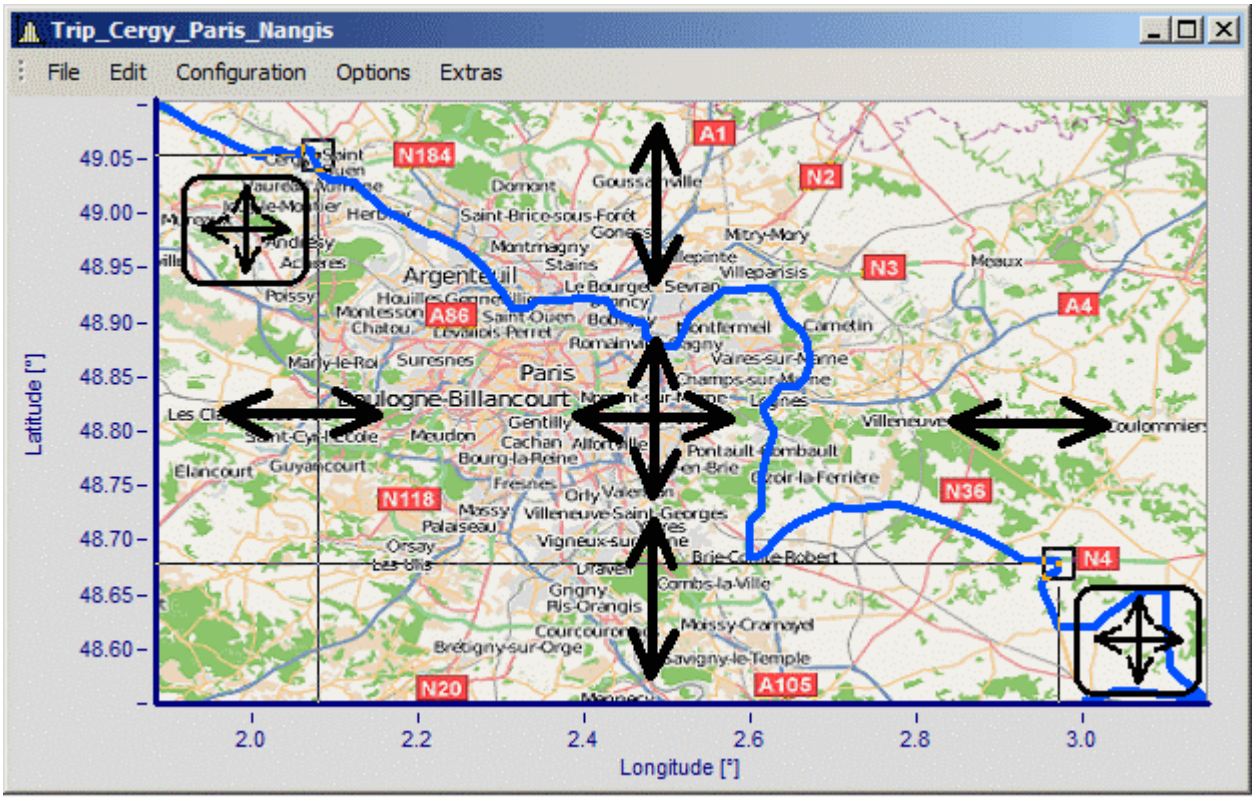
The points are entered into the curve window while the Map dialog is open. Move the points by clicking within the squares and dragging the points using to the mouse to the correct positions.



By moving, the points' X- and Y-positions are updated.

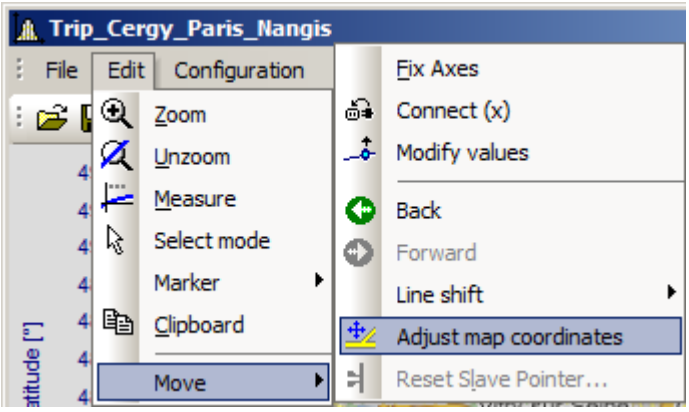
The mouse pointer changes appearance in response to the position in the curve window.

- 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.

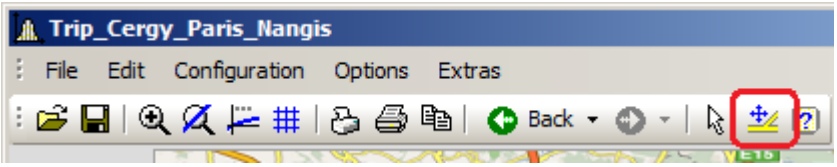


Adjusting the map and points in response to the mouse position

Once the map dialog has been closed, you can use menu commands to adjust the map: *Edit\Move\Map*.

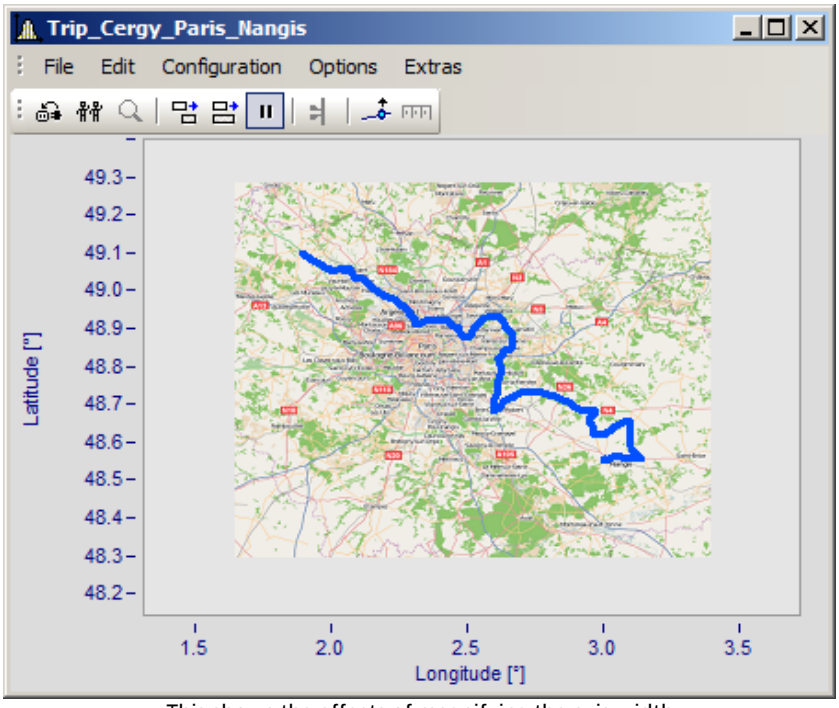


Alternatively, activate the mode by means of the toolbar. To do this, adapt the Toolbar 1061.



You can also drag this function into the toolbar.

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.

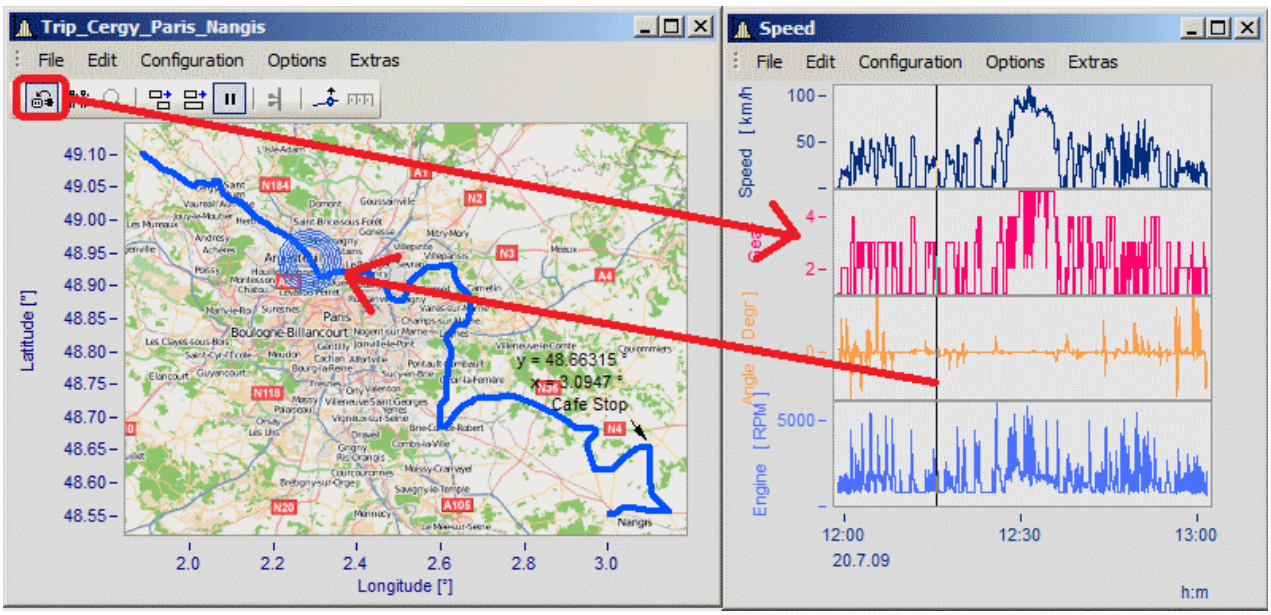
Alternatively, you can use a static picture in the background 979.

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.

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.

8.6.4.1.3.3 Map (from Internet)

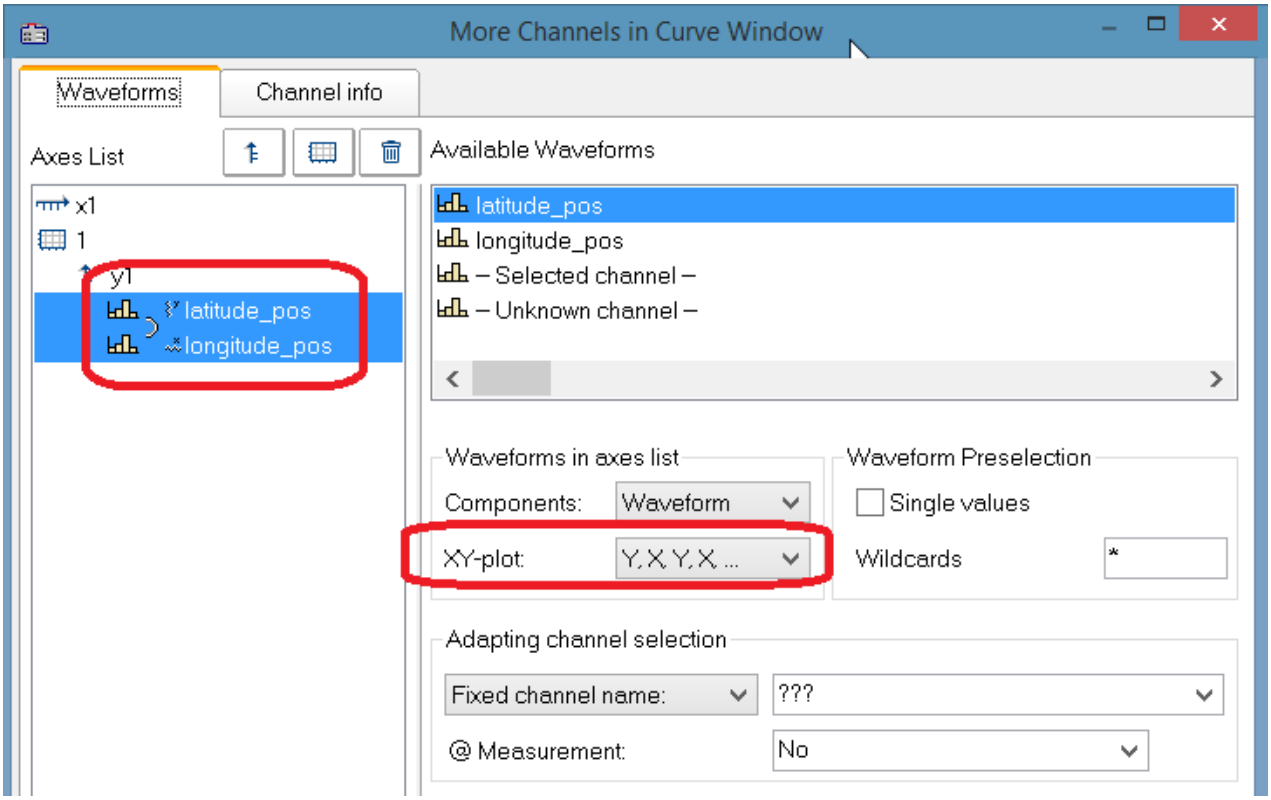
Automatic loading of a map from the Internet

On the following conditions, a picture can be correctly displayed as a map:

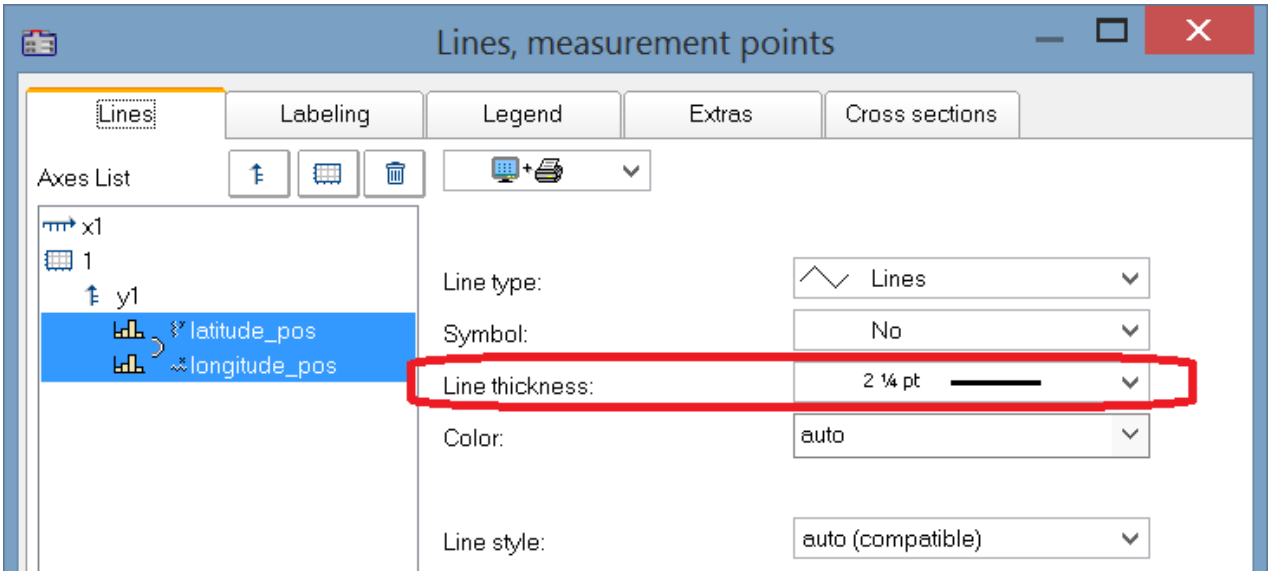
- selected curve window configuration: *Display \ Default*
- linear axes
- only one coordinate system
- a data set with plausible position data is displayed.

The following example shows how GPS data are automatically superimposed on a map from the Internet.

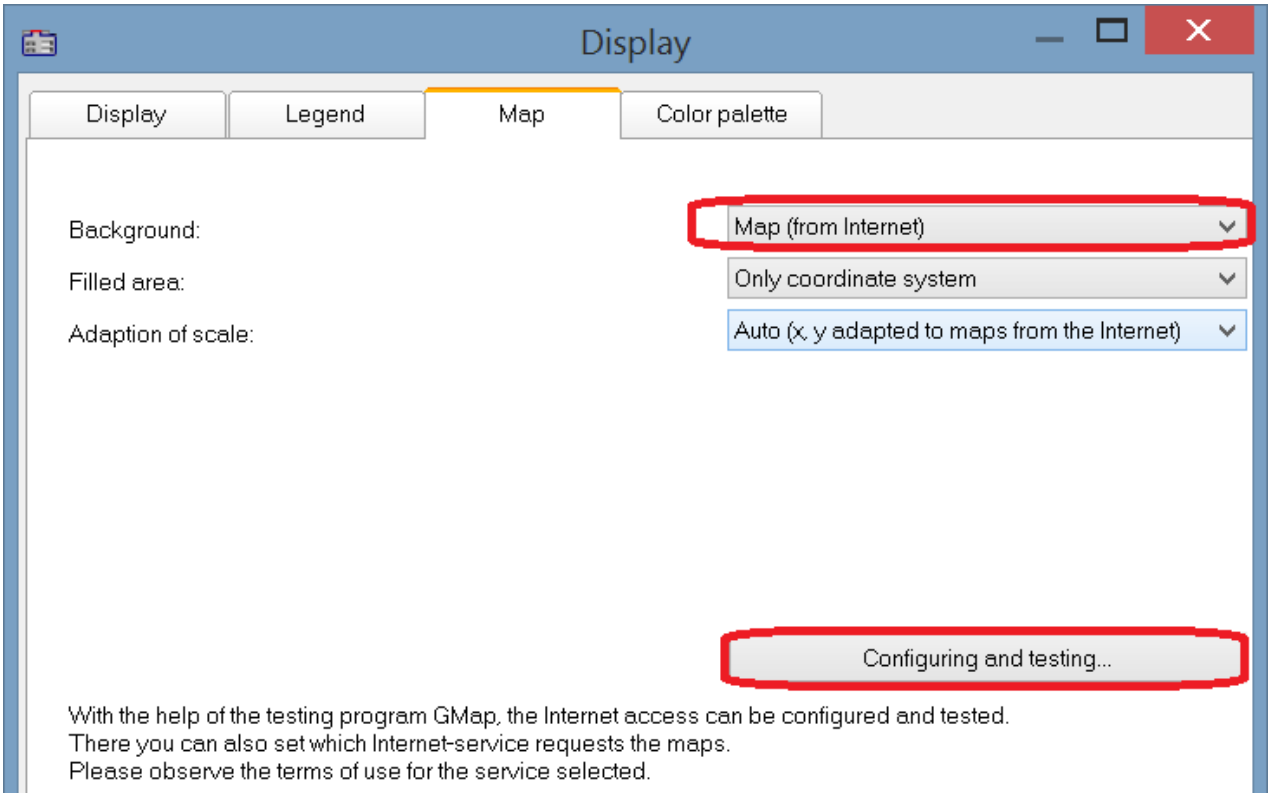
Load the data sets for Longitude and Latitude and display them as an XY-plot. To do this, open a curve window with the two components and superimpose them by means of the dialog "*More Channels in Curve Window*":

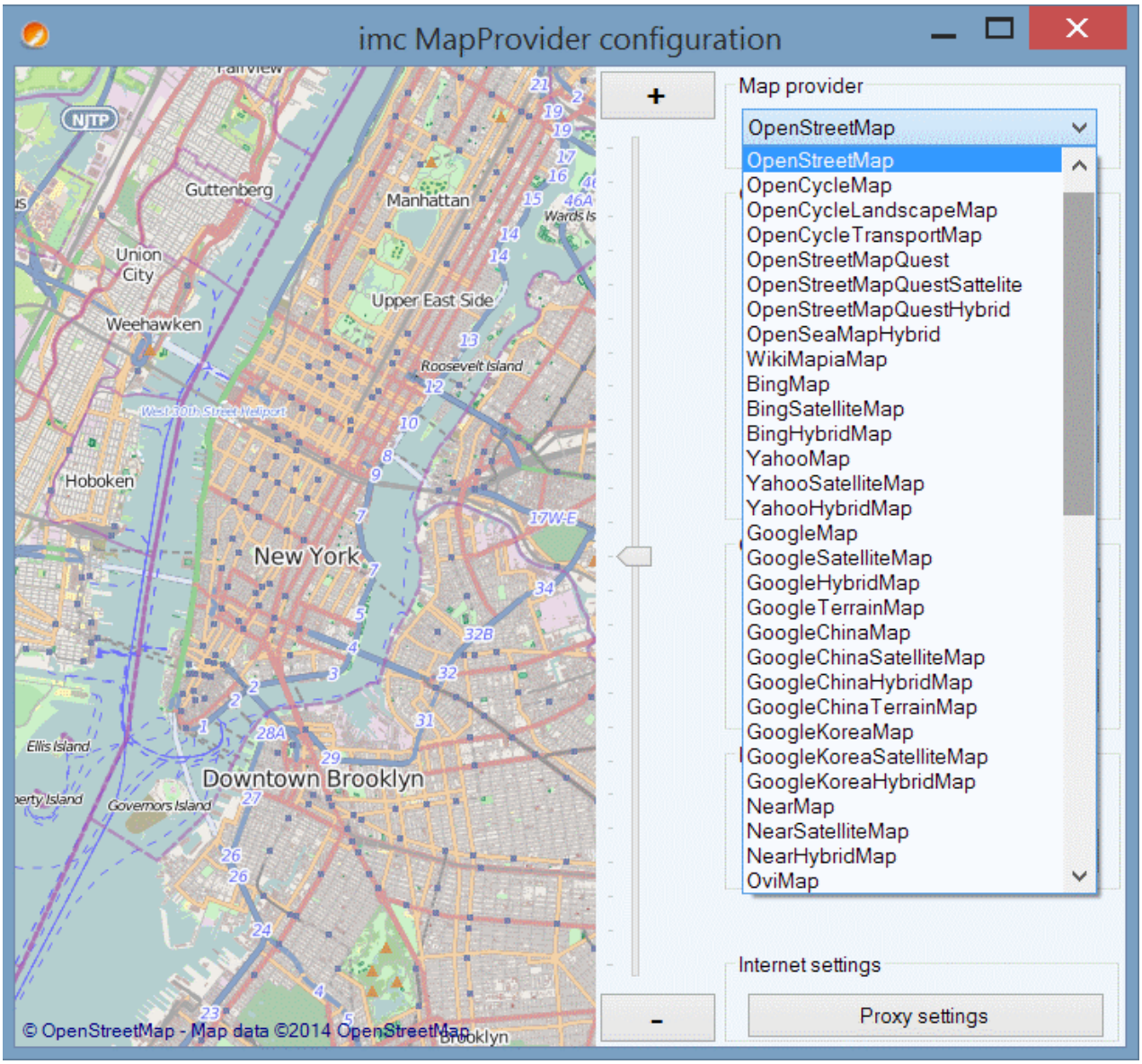


It is advisable to draw the lines more thickly:

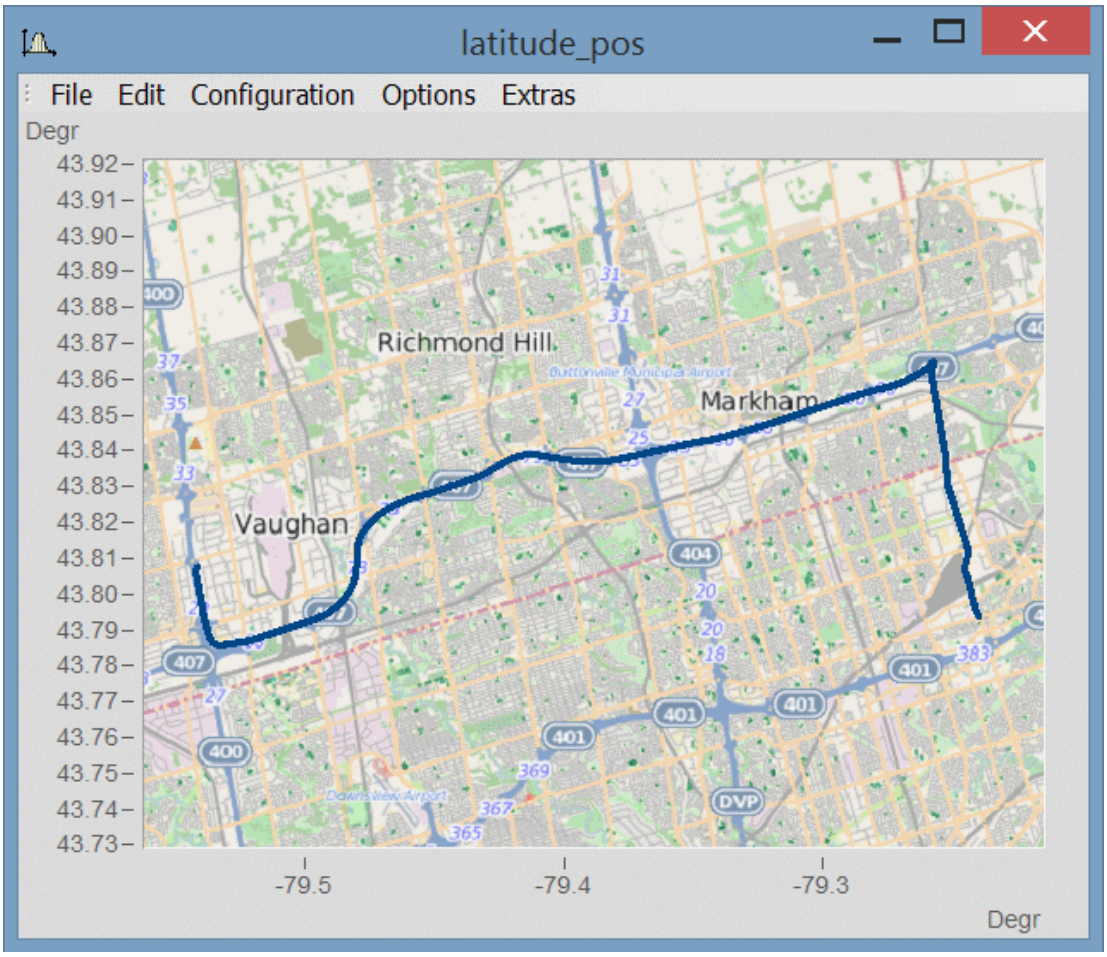


Select "*Map (from Internet)*" on the page "*Map*" in the dialog "*Display*". The setting "*Configuring and testing*" selects the Internet-service:

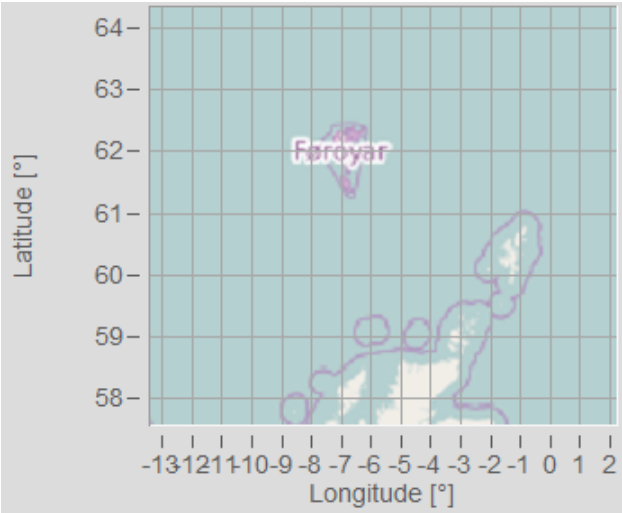




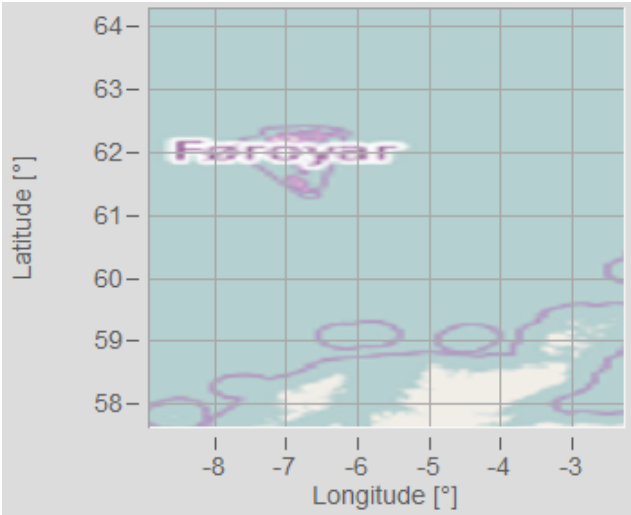
After selecting the map service, the map section loads already if an Internet connection already exists. In the curve window, this may appear as shown below, for example:



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

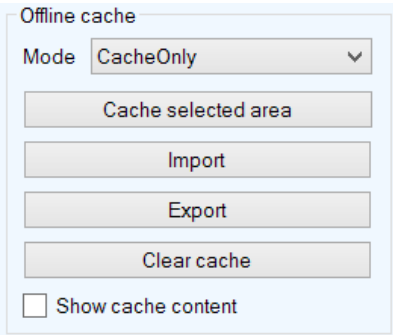


Coordinates equidistantly scaled -> Map distorted

Note

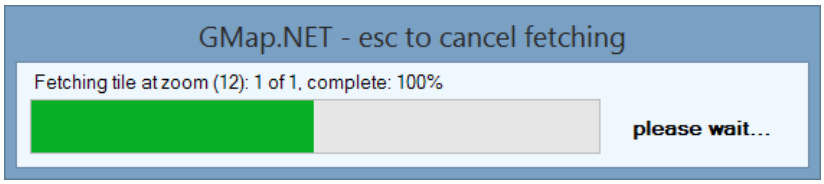
This projection named after Gerhard Mercator (1512-1592) causes imprecision in the curve window, since the coordinates are displayed equidistantly in it. The imprecision increases with the distance from the Equator and the size of the map.

Offline cache

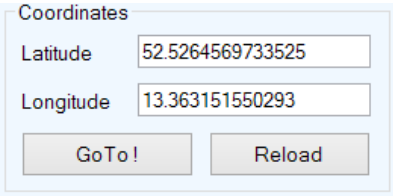


You can export, import, or save a map section locally. To do this, select "*Mode*" *ServerAndCache*" or "*Cache only*".

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.

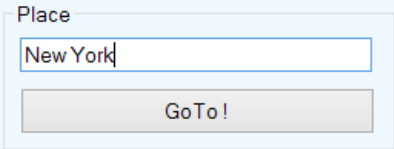


Coordinates



Select the map section by its coordinates and zoom to the desired section.

Place



Select the map section by entering a location.

Internet Settings

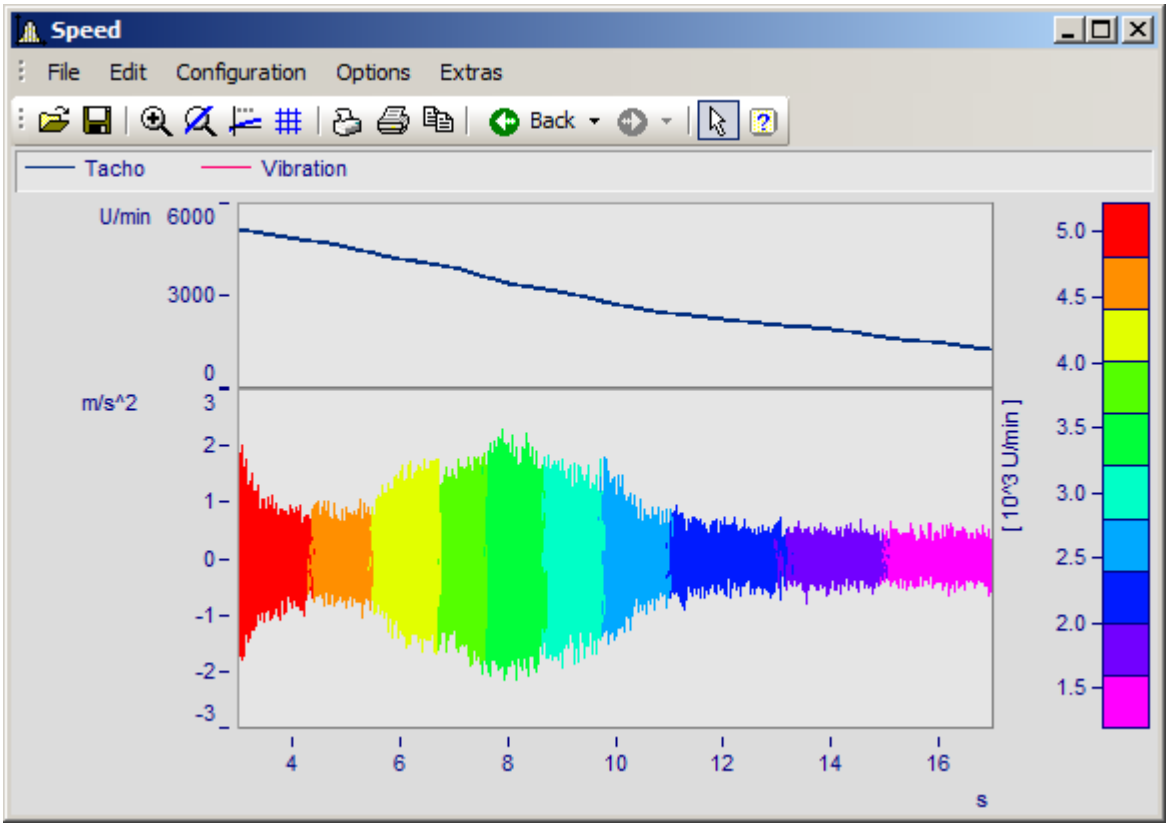
Here you can specify a proxy-server.

8.6.4.1.4 Color palette

A channel's color can be governed by a reference signal's amplitude.

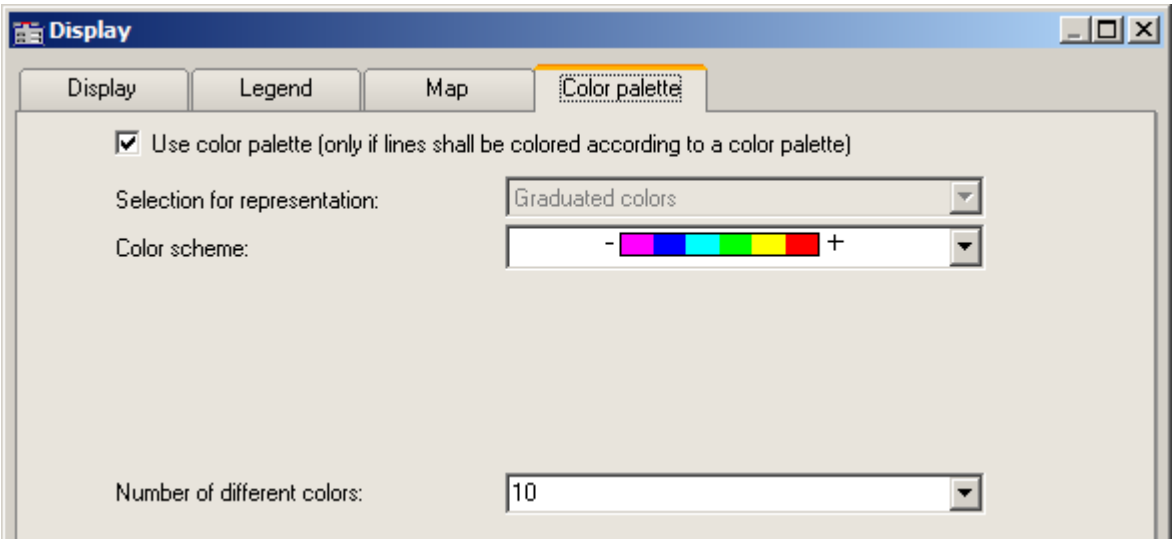
Note

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

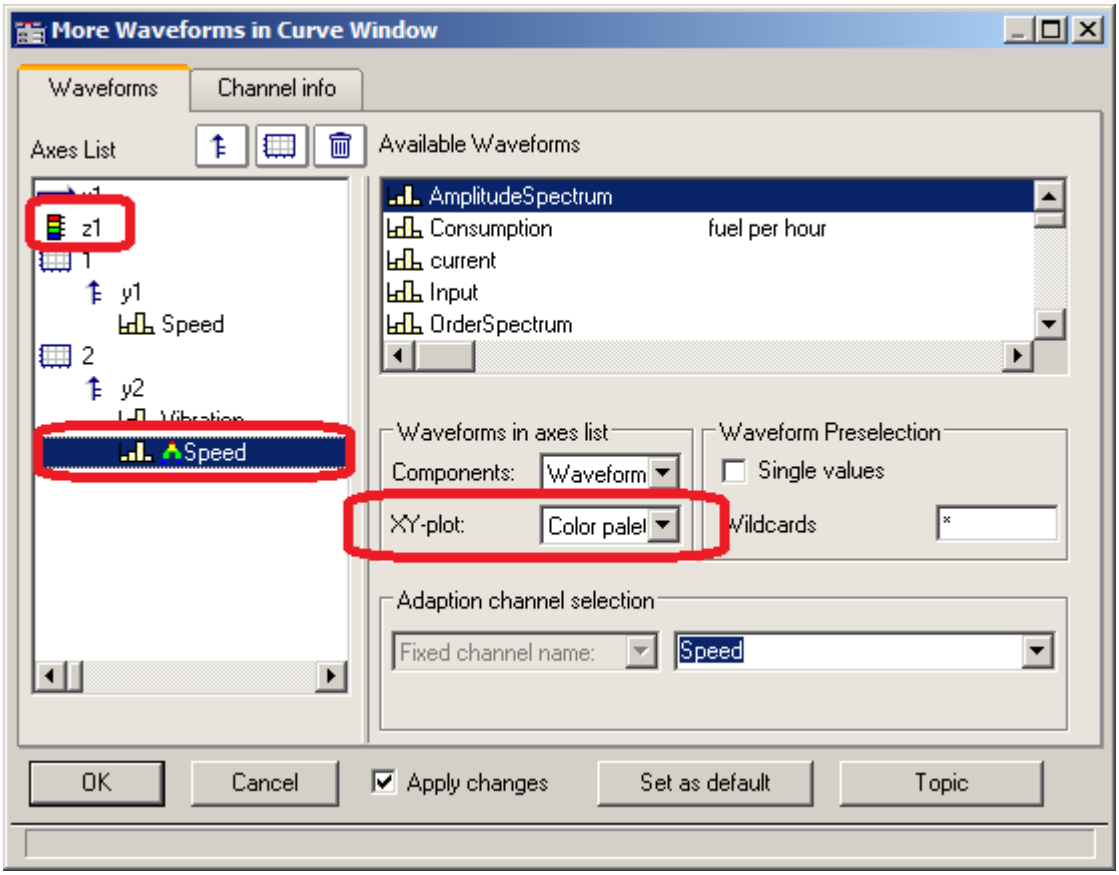
Select the page *Color palette* (not *Display type Color map*) in the dialog called by *Configuration\Display*.



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.

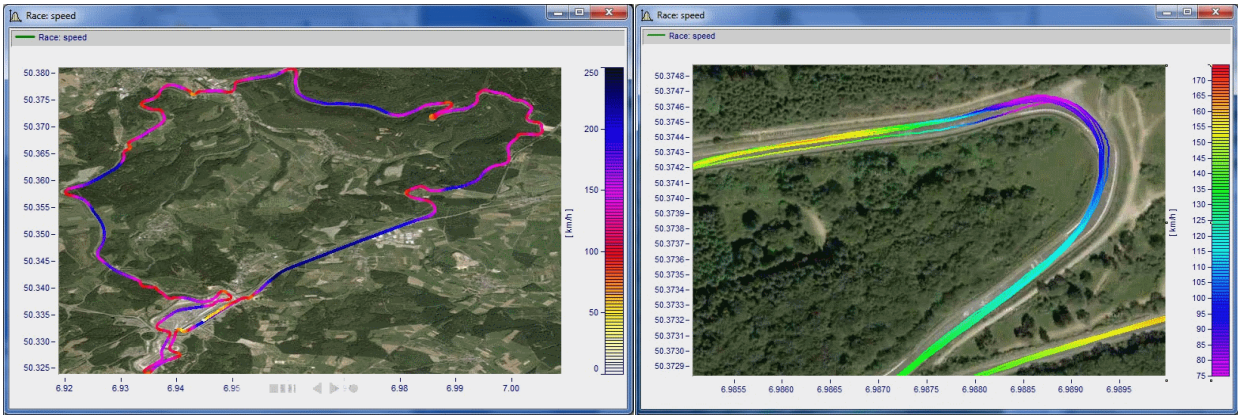
The same settings options for the color scheme and amount of colors are available as for Color map 949.

The assignment is performed in the dialog *More Channels...*, which is called from the curve window's context menu.



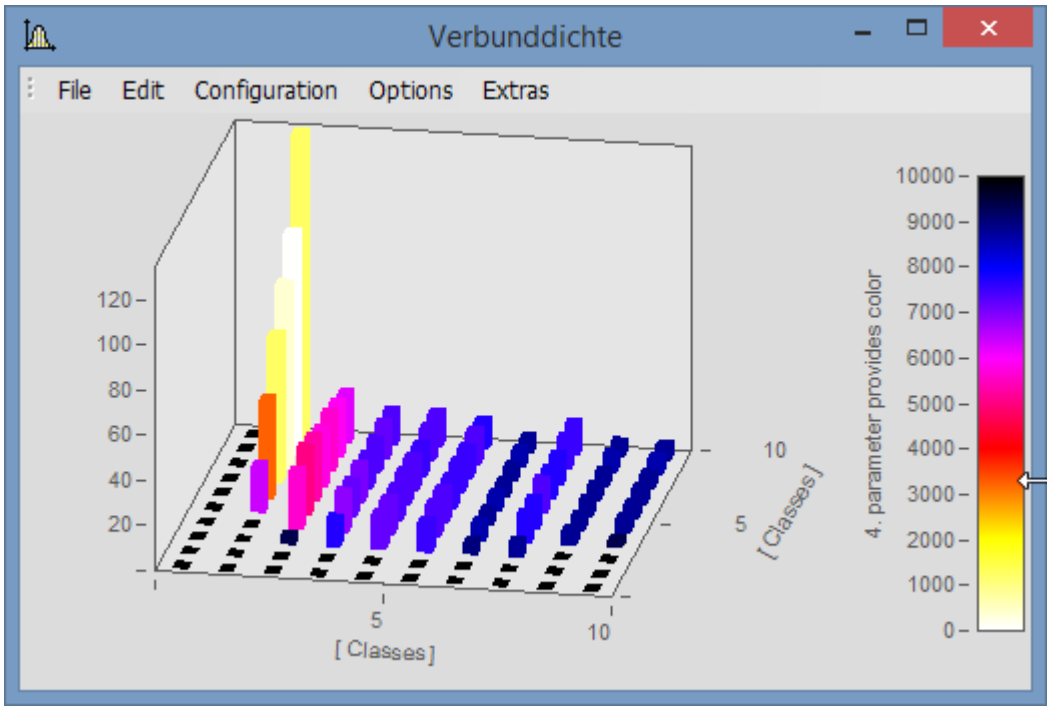
The reference channel is assigned to the same y-axis as the channel to be colored. Select a reference channel and set the *XY-plot* to *Color palette*.

Example:



Color palette as 4th parameter in 3D

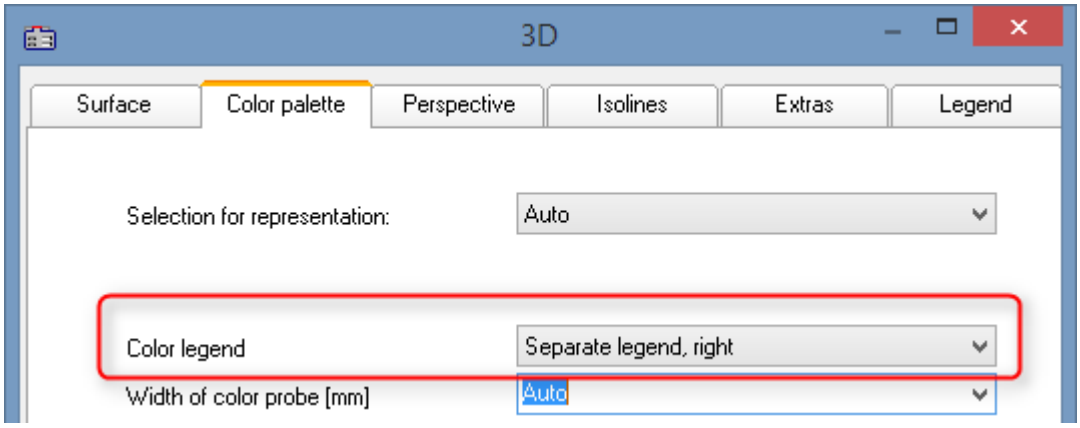
A 3D display can be colored according to a fourth parameter. Thus, 4 dimensions can be displayed. See also here 1030.



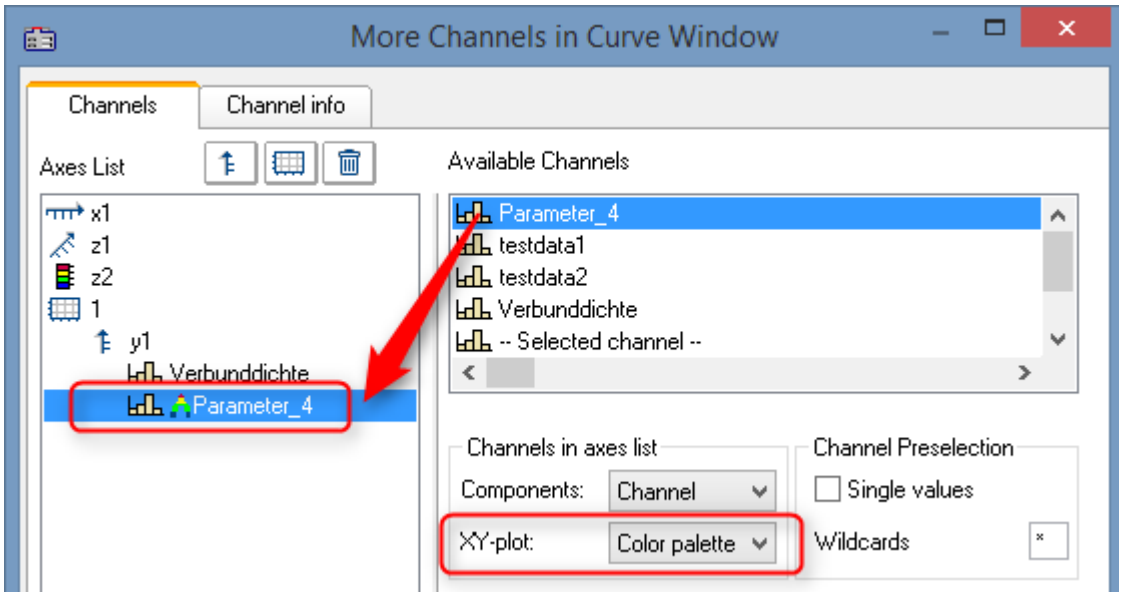
3D display with colors governed by a fourth data set

Procedure:

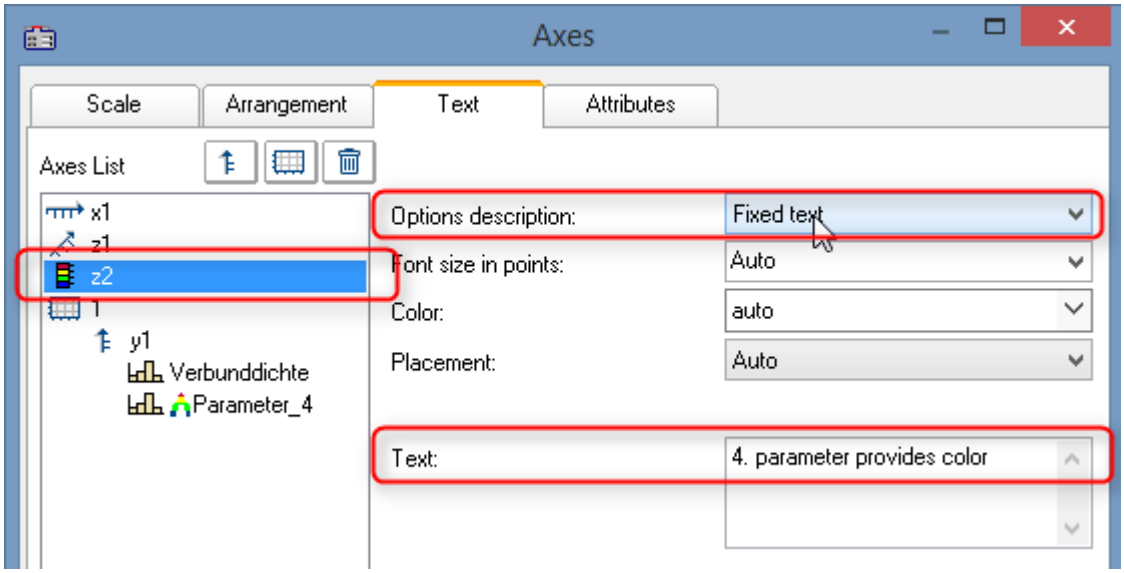
1. Set *3D* display with *Separate legend right* or left in *Color plaette* 1030. With the settings *Auto, No* and *Integrated into y-Axis*, it is not possible to generate a colored overlay.
2. Under *More Channels...*, place the fourth variable directly below the 3D data set.
3. Under *XY-plot*, select *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.

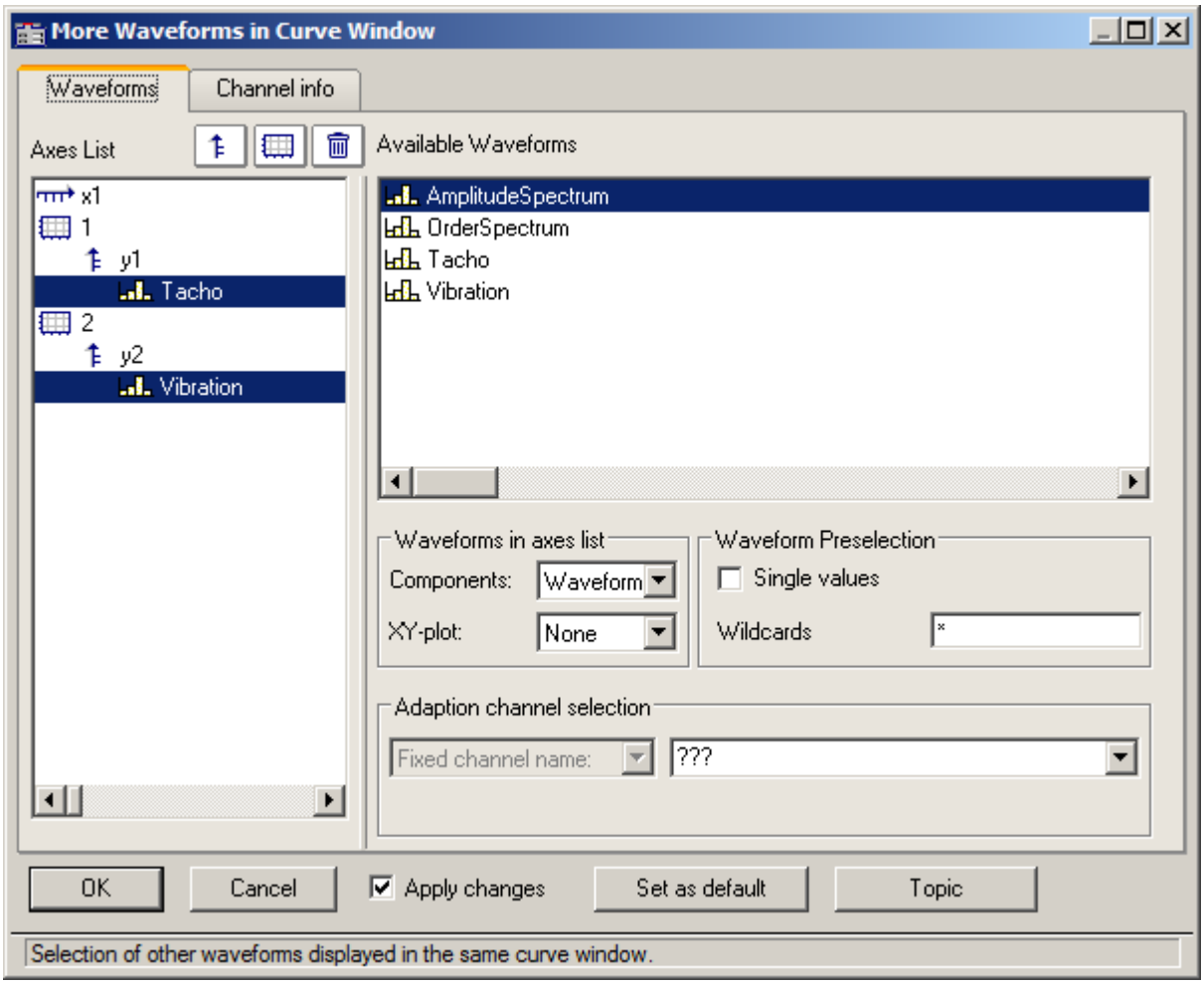
8.6.4.2 More Channels in Curve Window

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 Waveforms respectively can be altered by dragging the border between them using the mouse.

The subsections below respectively describe the individual dialog elements.

8.6.4.2.1 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 Waveforms* using Drag and drop.

Move or Select:

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.

Using Drag & Drop on an entry which is not selected, you can select elements in the axis list .

When you hold down the CTRL-key, individual entries can be selected or de-selected.

Buttons for editing the axes list

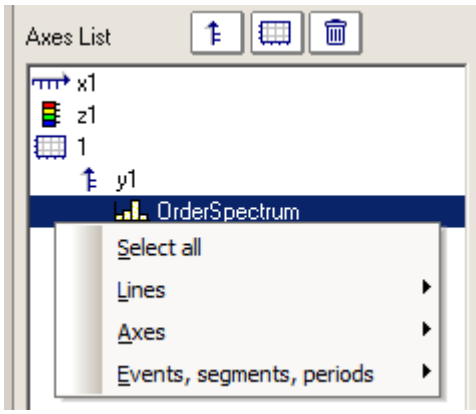
	A new y-axis is created. To do this, click briefly on this button, which changes the mouse pointer to a suitcase symbol. Click on the target position desired. The new y-axis is inserted above the row over which the mouse pointer is located.
	A new coordinate system is created.
	The lines selected in the axes list (coordinate systems, axes, waveforms) are deleted. Use the <Del> button as an alternative.

Symbols used in the axes list

	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 (characteristic curve)
	A digital waveform
	A complex waveform in magnitude/phase- or real/imaginary display

Context menu in Axes List



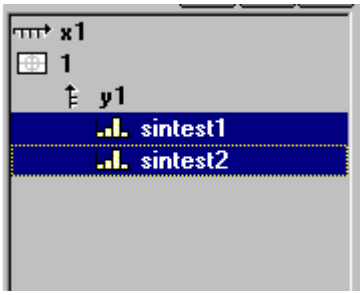
The following menu items are available:

Select all: Selects all lines in the list.

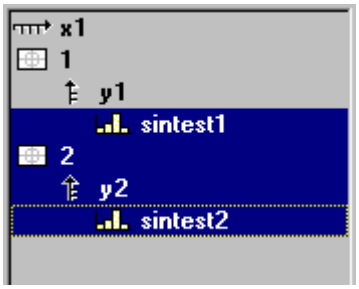
Lines: A *line* is the graphical representation of one or more data sets, in other words a *waveform*. Normally an entry in the axes list will indicate the simple representation of one data set as a line.

The following items affect the selected lines:

Individual coordinate systems: This produces a *stacked* representation of curves.



before



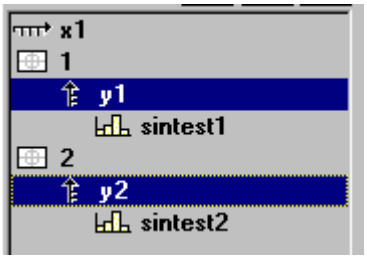
after

All on 1 axis: All selected lines are displayed with only one axis. Any obsolete axes and coordinate systems are deleted.

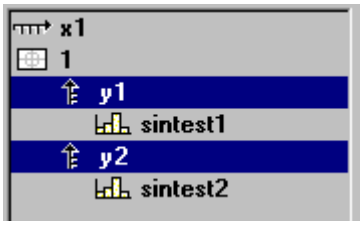
Individual axes: A separate axis is displayed for each line.

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

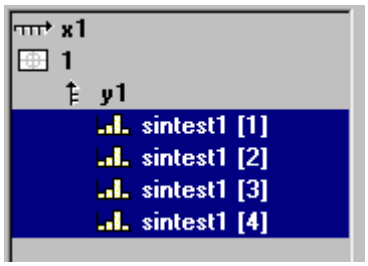


after

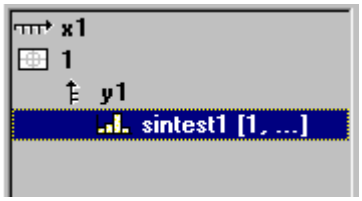
Individual coordinate systems: A separate coordinate system is displayed for every (y-) axis.

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.

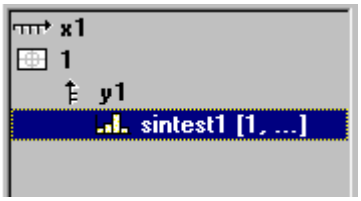


before

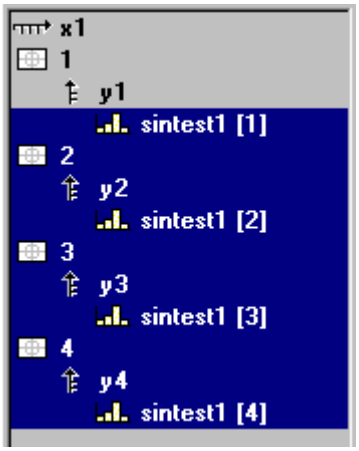


after

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

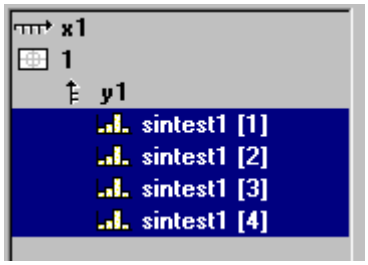


after

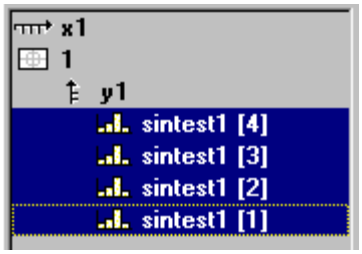
Separate into individual lines: This is the contrary of *Combine*. Multiple lines, which each amount to a structural element, are created.

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

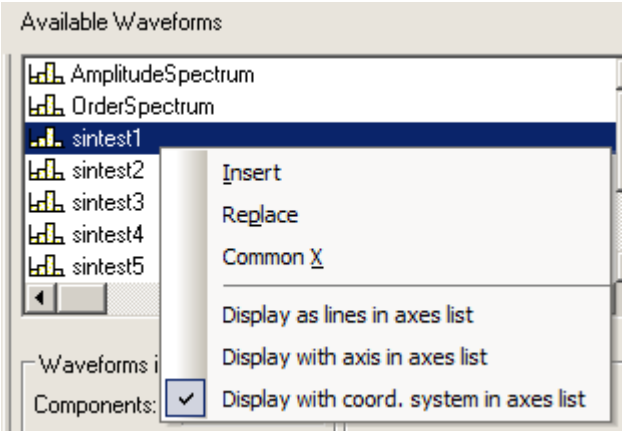
To make any of these format changes permanent, save them by clicking on *As presetting*.

8.6.4.2.2 Available Waveforms

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 Waveforms

Right-clicking mouse in the *Available Waveforms* list to open the context menu:

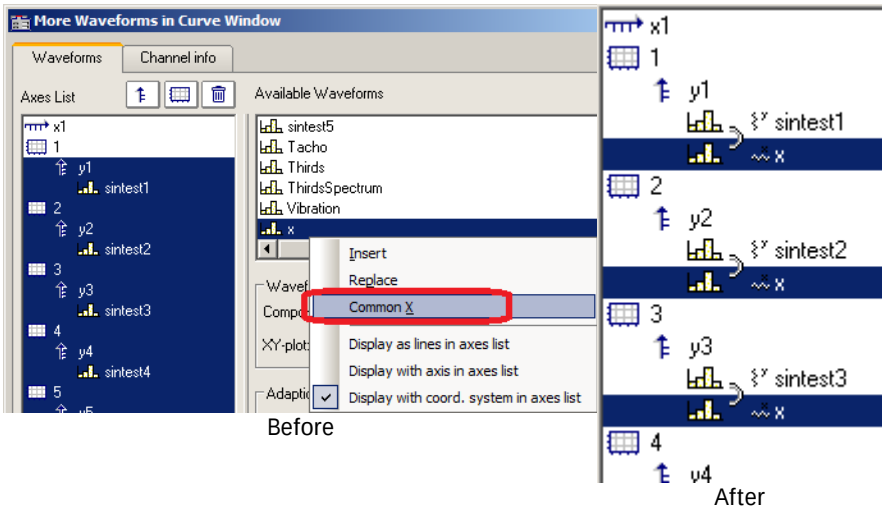


Insert: The waveform selected is added to the axes list. Depending on how the controls below are set, the waveforms are automatically created along with y-axes or coordinate systems:

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 Waveforms*. If more than one waveform is selected under *Available Waveforms*, this menu item is not enabled.

Common X: The waveforms selected in the axes list contain an x-component.



8.6.4.2.3 Waveform Preselection

Waveform Preselection: Single values

Single values are only indicated in the list of available waveforms 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 ";"

8.6.4.2.4 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.

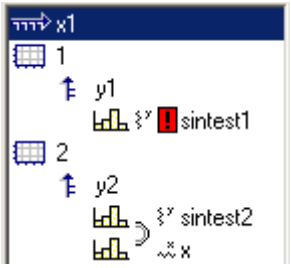
???	The settings for the selected waveforms do not match.
Waveform	Complete display of variables according to type. XY-Waveform: the entire waveform "Y over X" Complex data: polar plot
.x *.y*	The specified components of XY-waveforms are displayed
.r *.i*	Real or imaginary part of a complex waveform 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.

???	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.

???	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

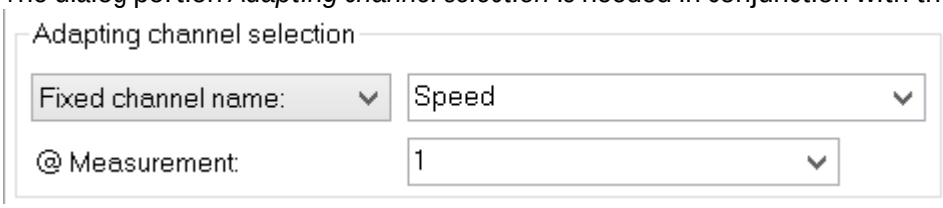
The multi-dimensional relationship always applies to three consecutive waveforms belonging to the same y-axis. All waveforms must be one-dimensional (so either normal waveforms or defined as one component of a 3-component waveform).

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.

8.6.4.2.5 Adapting channel selection

The dialog portion *Adapting channel selection* is needed in conjunction with the Data Browser 818.



Fixed channel name

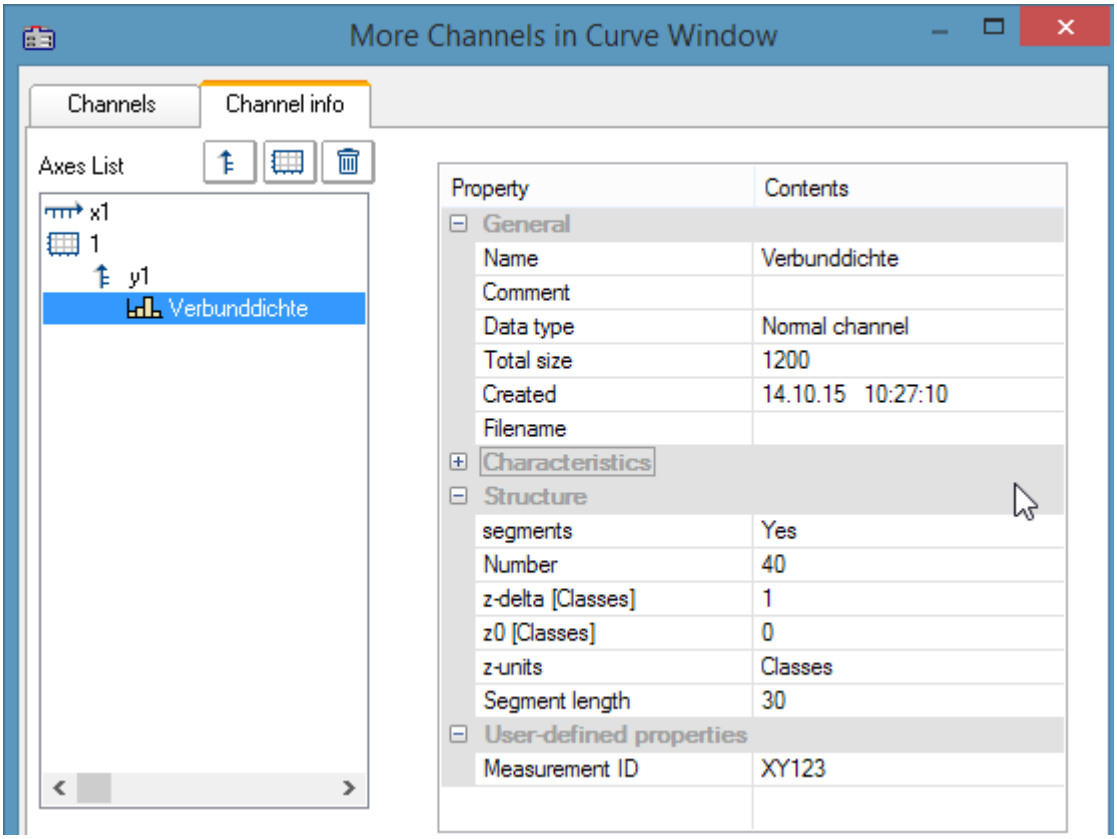
Lets one specify that only a waveform having the channel name entered in the edit box at the right is displayed. Toward this end, "Unknown channel" from the window *Available Waveforms*: is added to the axes list and configured accordingly.

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 Waveforms*: is added to the axes list and configured accordingly

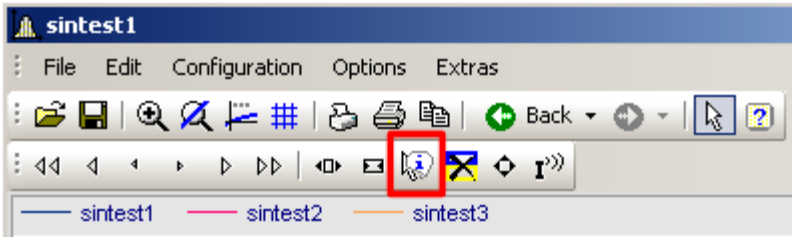
Additionally, the measurement selected in the Data-Browser can be set as a further display criterion in the dropdown-list *@Measurement*.

8.6.4.2.6 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* 1042 or via the corresponding symbol in the *Navigation Toolbar*.



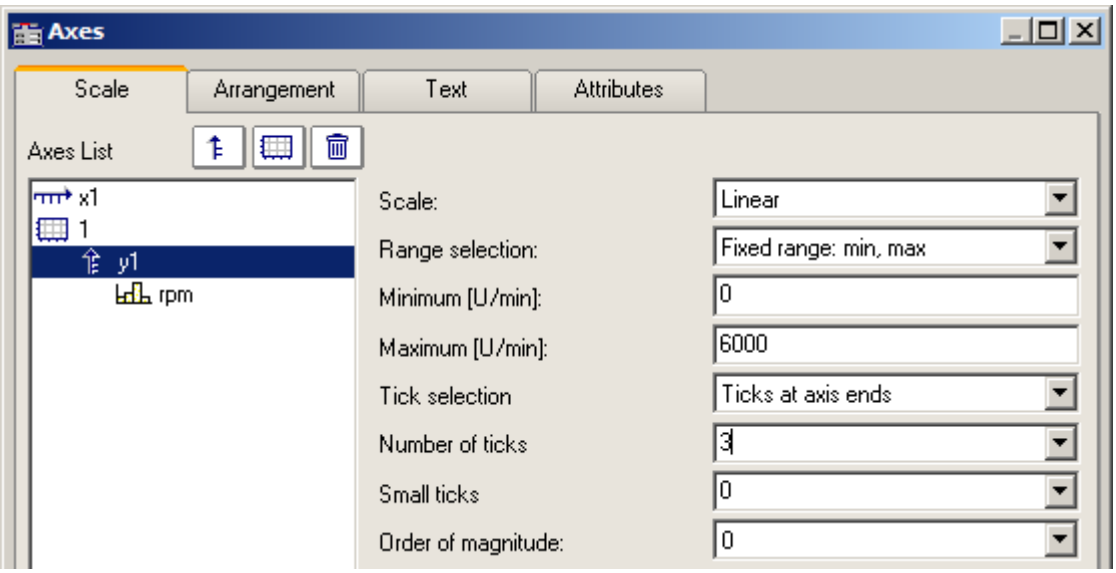
Channel info in the Navigation Toolbar

8.6.4.3 Axes

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.

8.6.4.3.1 Scale

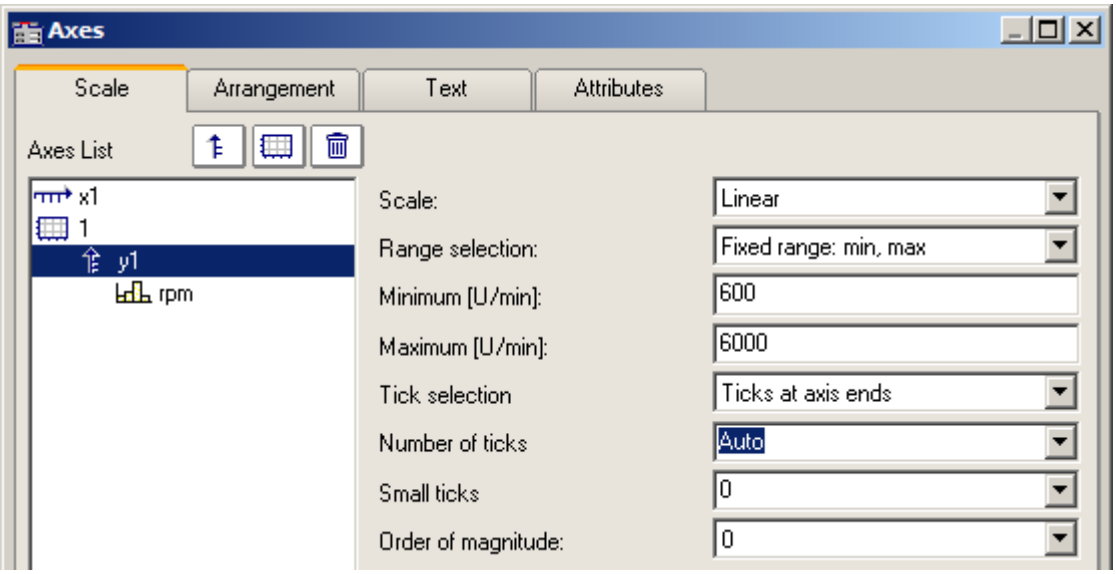
Scaling of the axes. An axis' domain can be defined in a number of ways.

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.

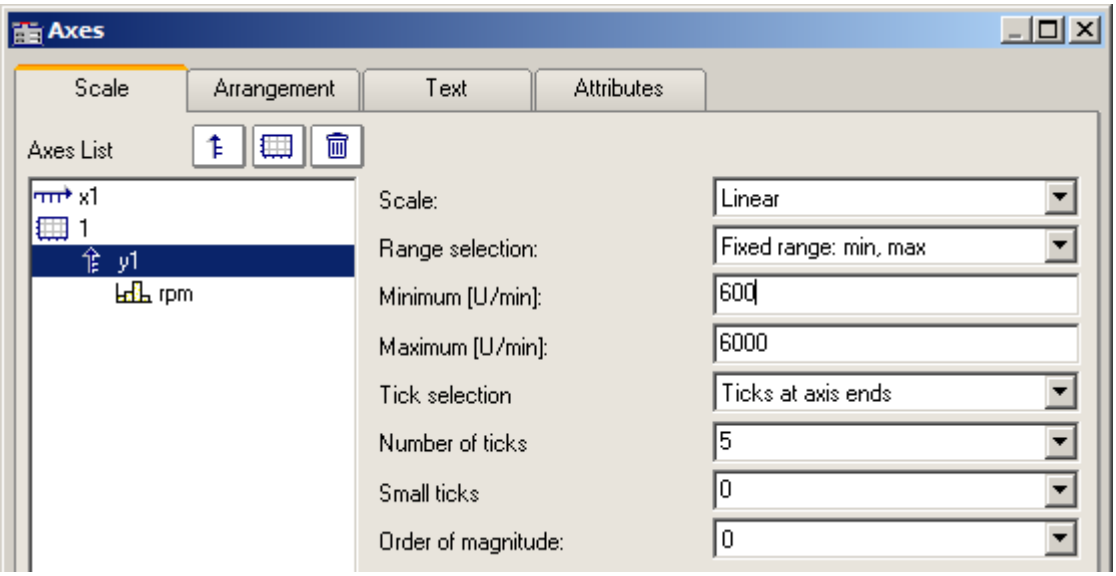
Range selection

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*.

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.

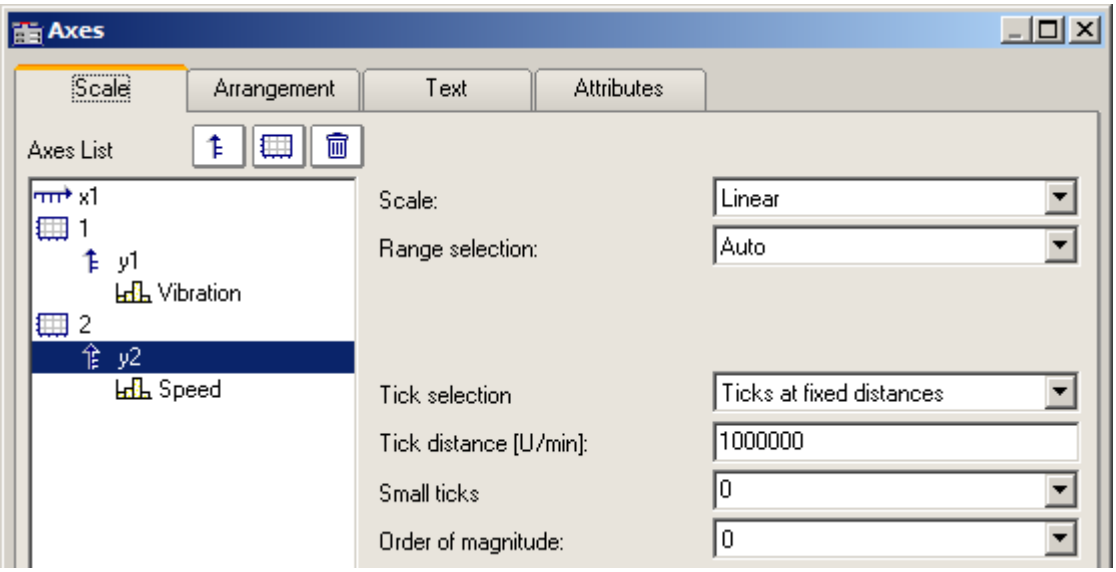
Auto: Determining the range is automatic. The entire curve is displayed in the curve window.

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

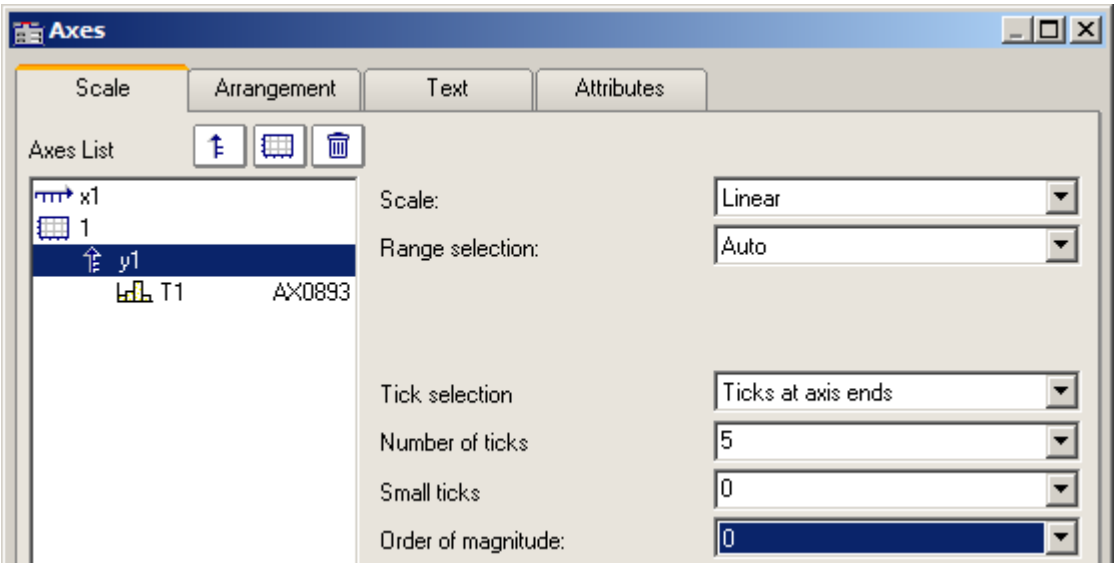
The main ticks, along with which the axes labeling appears in the curve window, can be placed anywhere on the axis. If you select the option *Ticks at fixed distances* in the *Tick selection* control, the additional control *Tick distance* becomes available. Here you have the ability to set the amount 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.



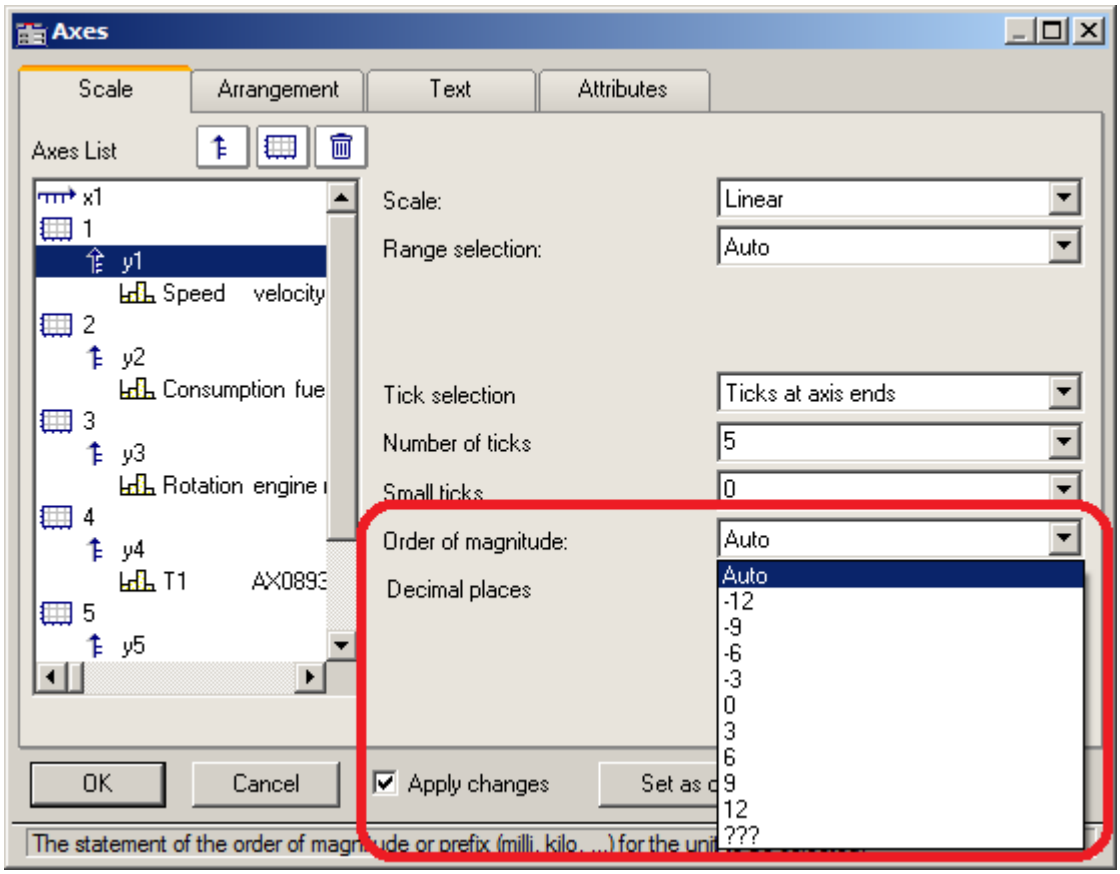
The number of ticks per unit is calculated by subtracting one (1) from the number of markings and dividing the difference by the displayed interval in units. For example, with an interval of 3ms and seven markings specified, two ticks per unit (ms) will be placed on the axis.

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

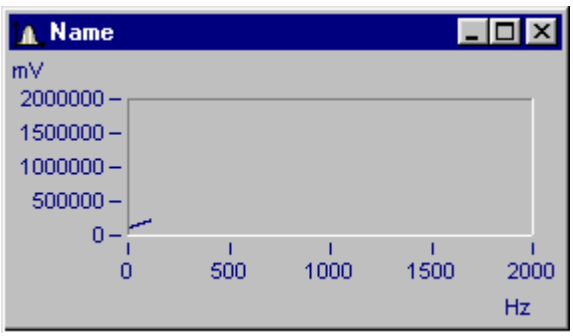
It is possible to arbitrarily determine the power-of-ten of the axis-scaling. If the axis is labeled with a unit, and appropriate prefix will be appended to the unit symbol, if applicable (e.g., mV or MV).



Example

For an x-axis with a fixed scaling range of 0...2000V:

Auto	Display: 0..2kV
Fixed exponent: +6	Display: 0..0.002 MV
Fixed exponent: 0	Display: 0..2000V
Fixed exponent: -3	Display: 0..200000 mV

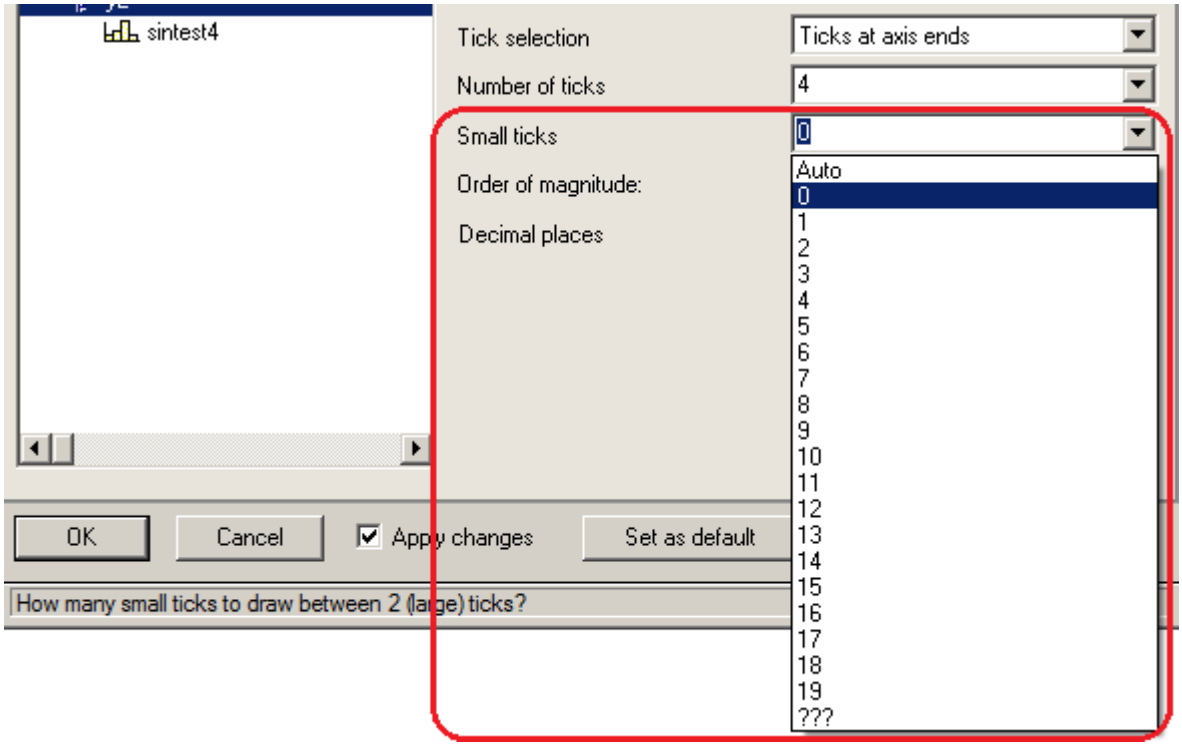


Curve window with inappropriate scaling

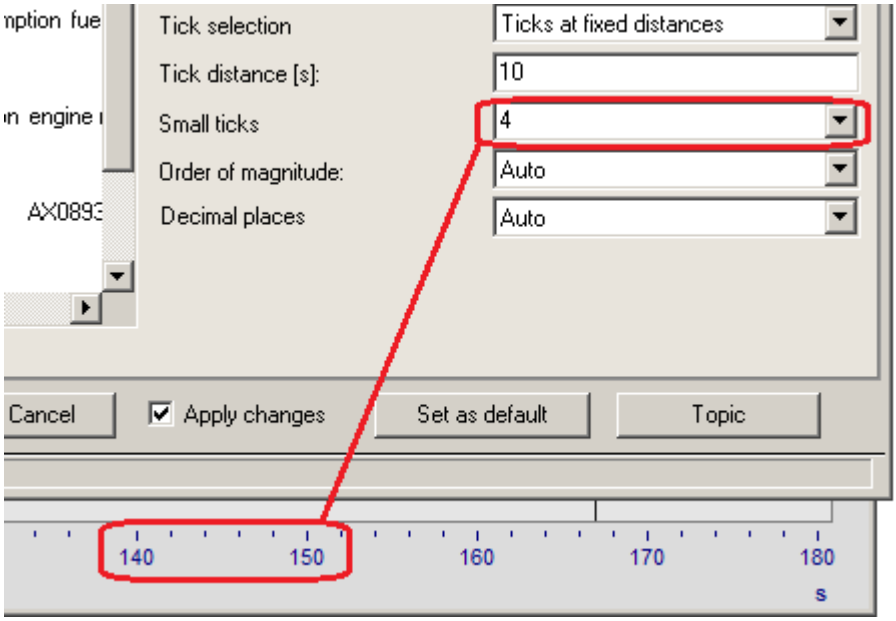
x-axis scaled with exponent 0, y-axis with exponent -3.

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* 1026.

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

Examples:

<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.

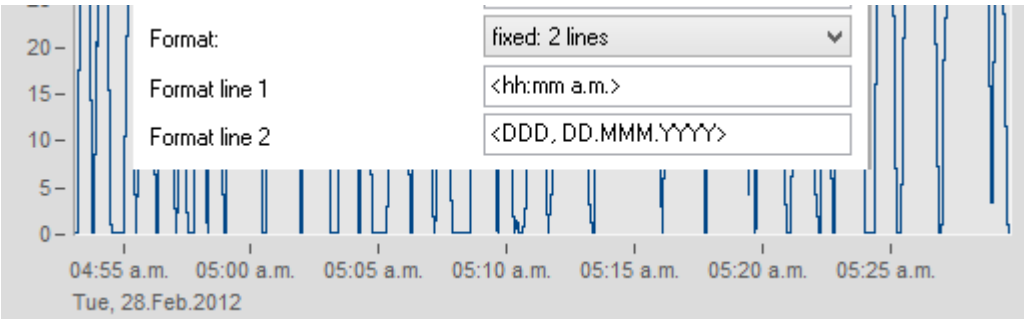
Examples:

<hh:mm:ss.ss>

<D> days

<o:mm:ss>

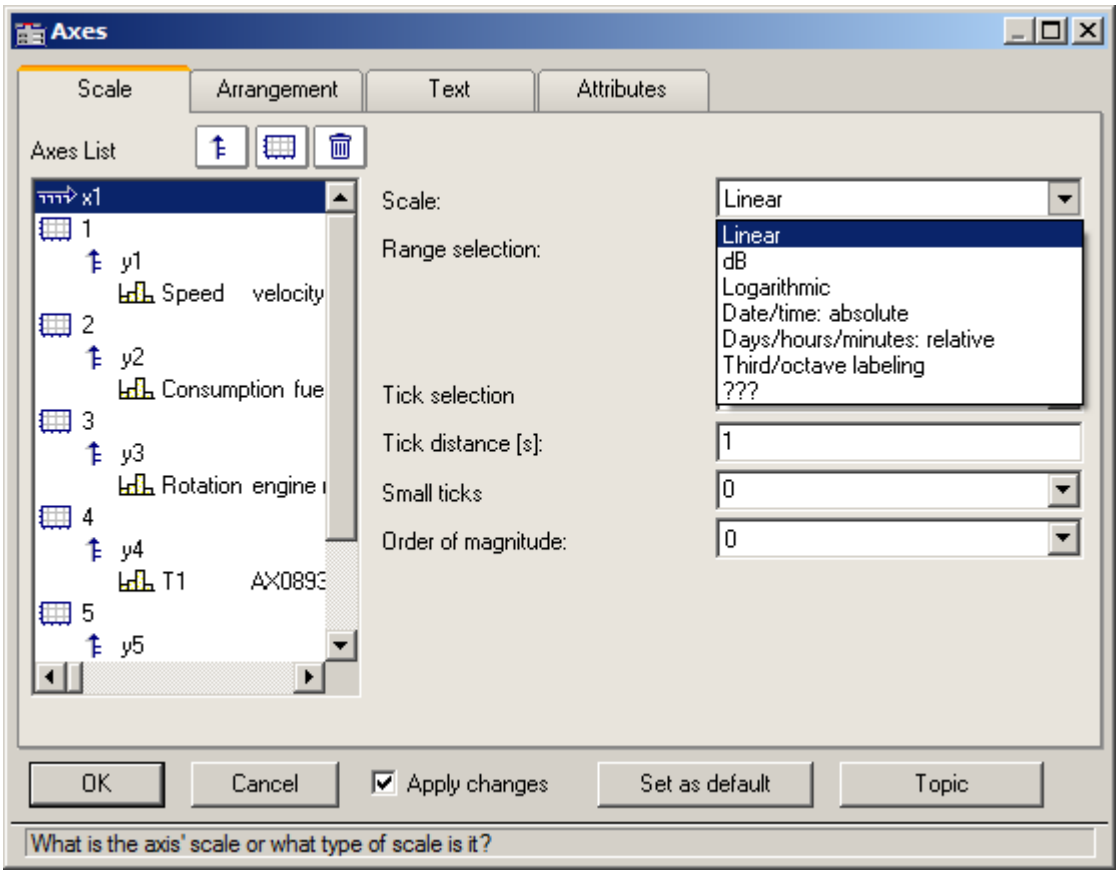
<o> Std, <mm:ss>



Example: line 1: <hh:mm a.m.> line 2: <DDD, DD.MMM.YYYY>

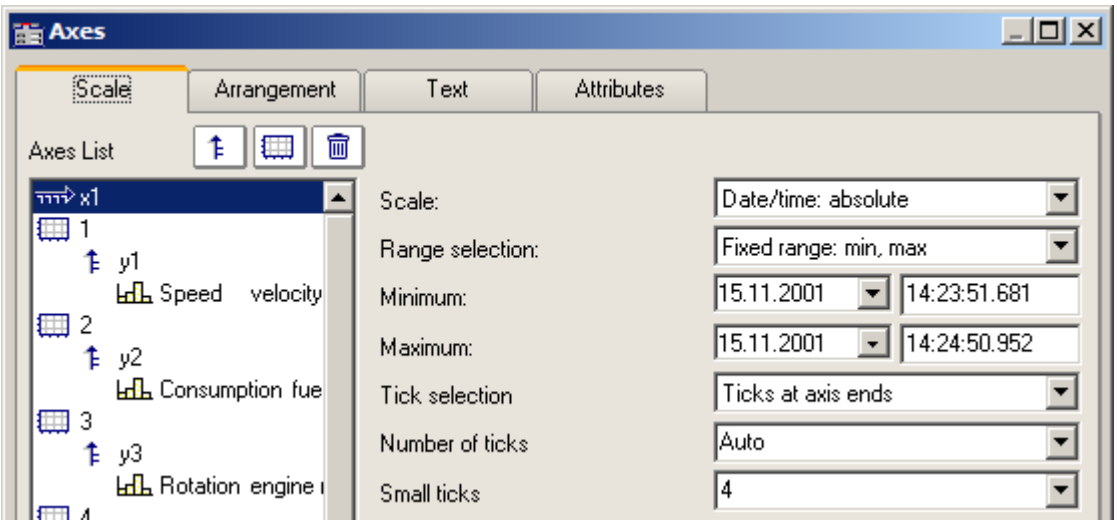
8.6.4.3.1.1 Particularities of the x-axes

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.

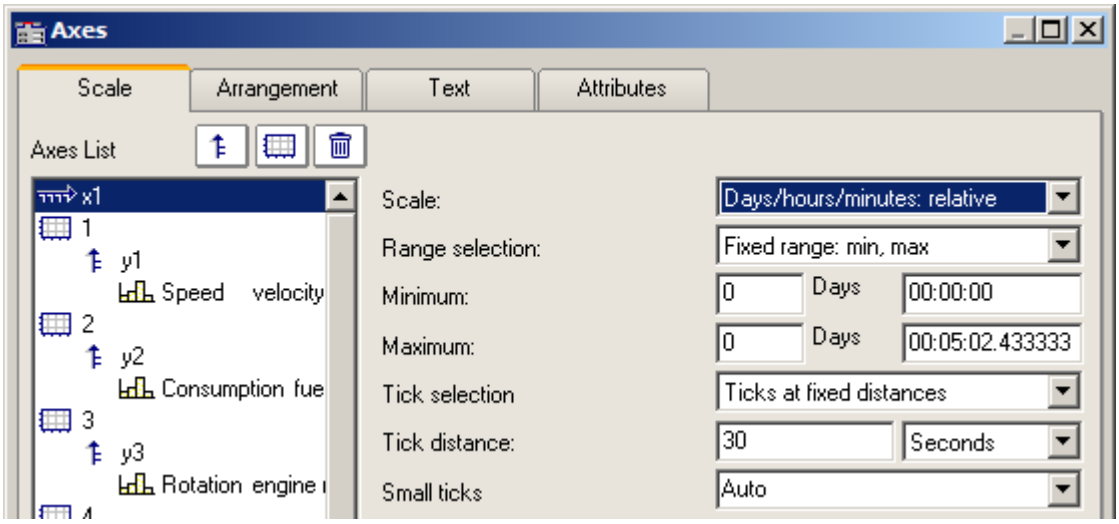


The distance between ticks can be set to be determined automatically or to be fixed. With a fixed setting, the unit can be anything from seconds to days.

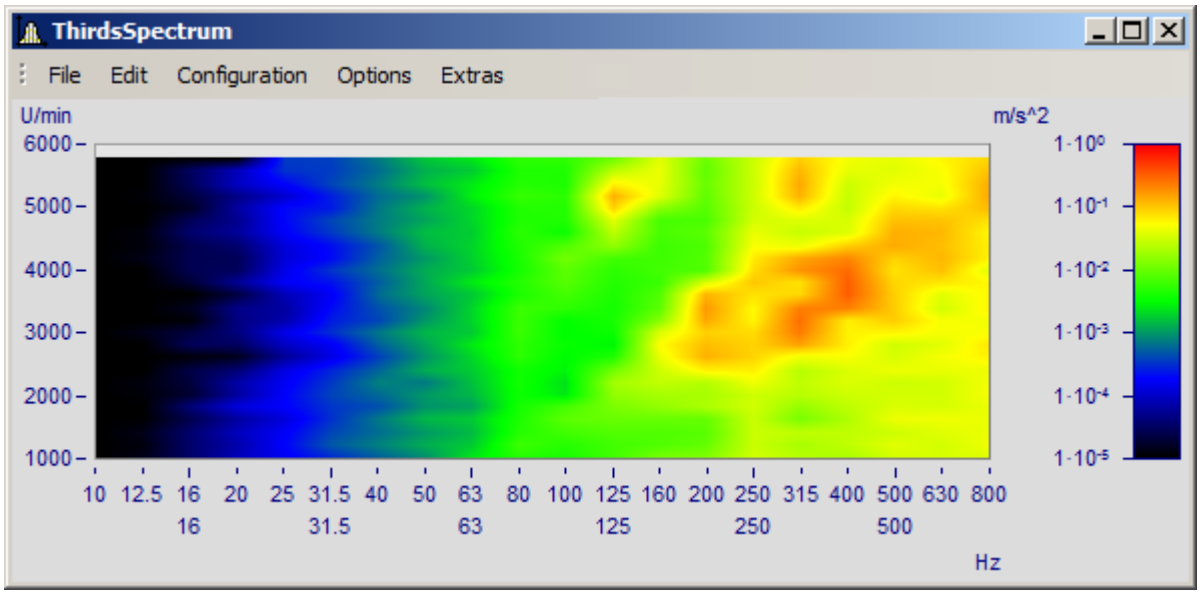
The channels are displayed in reference to absolute time. The absolute time of the measured points is usually derived from the sum of the specified absolute trigger time and the relative time for a point measured after the measurement's start.

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.



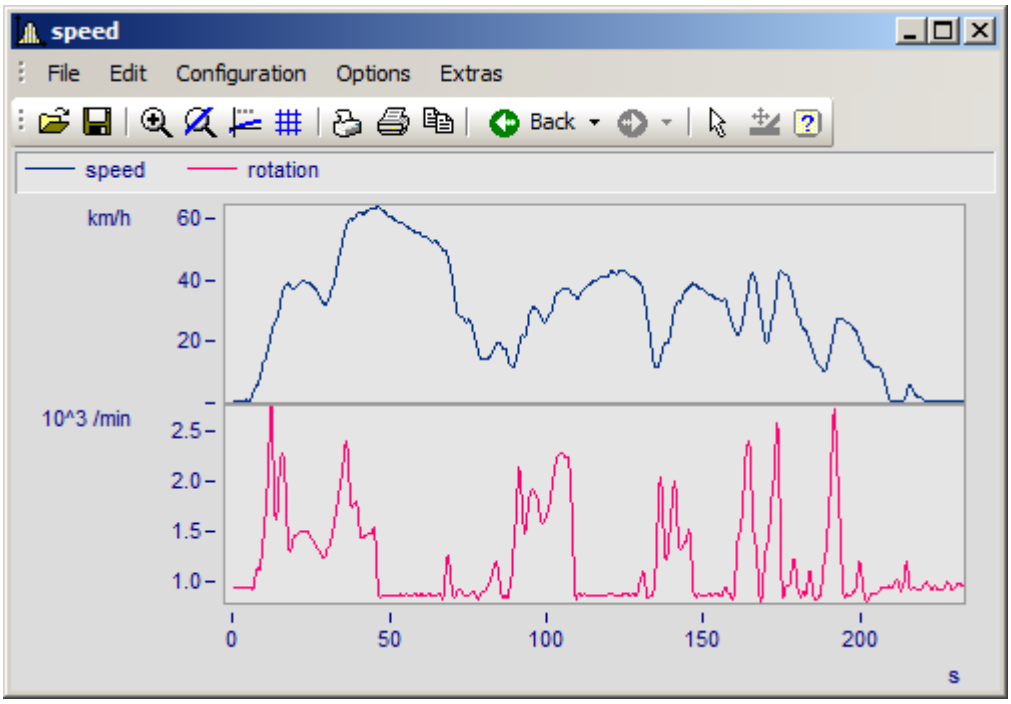
Third/ octave labeling: If a waveform contains a 1/3-octave or octave spectrum and the waveform's x-axis is scaled in 1/3-octaves, then the x-axis can be drawn with numerical values of the 1/3-octaves and octaves, in accordance with DIN.



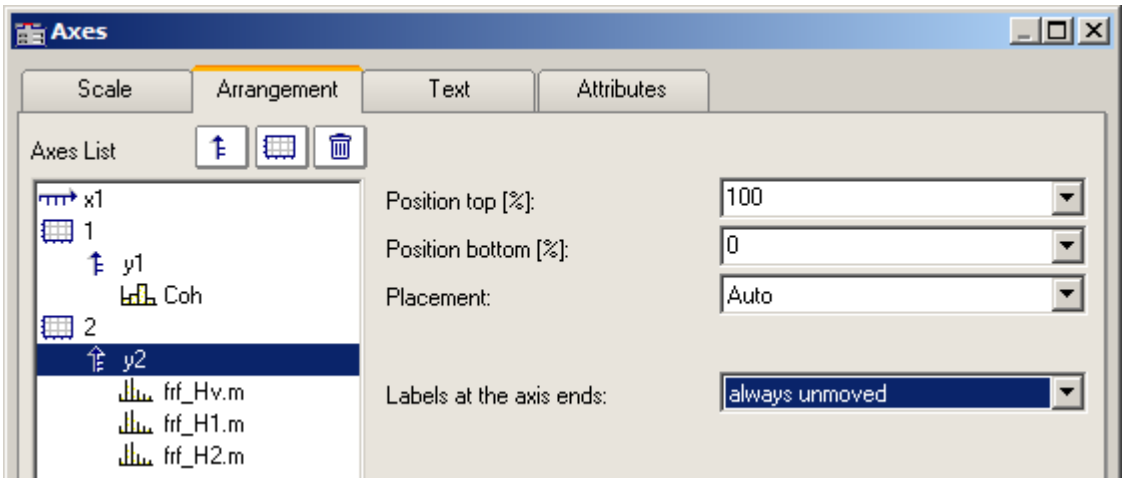
For details, see the dedicated section on "Third / octave labeling 965".

8.6.4.3.2 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.

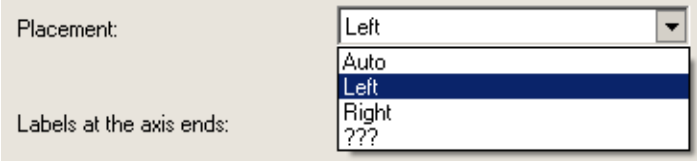


To do this, select the page *Arrangement* in the dialog *Axes*.



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.



The control *placement* sets whether to position the axis at the left or right of the coordinate system.

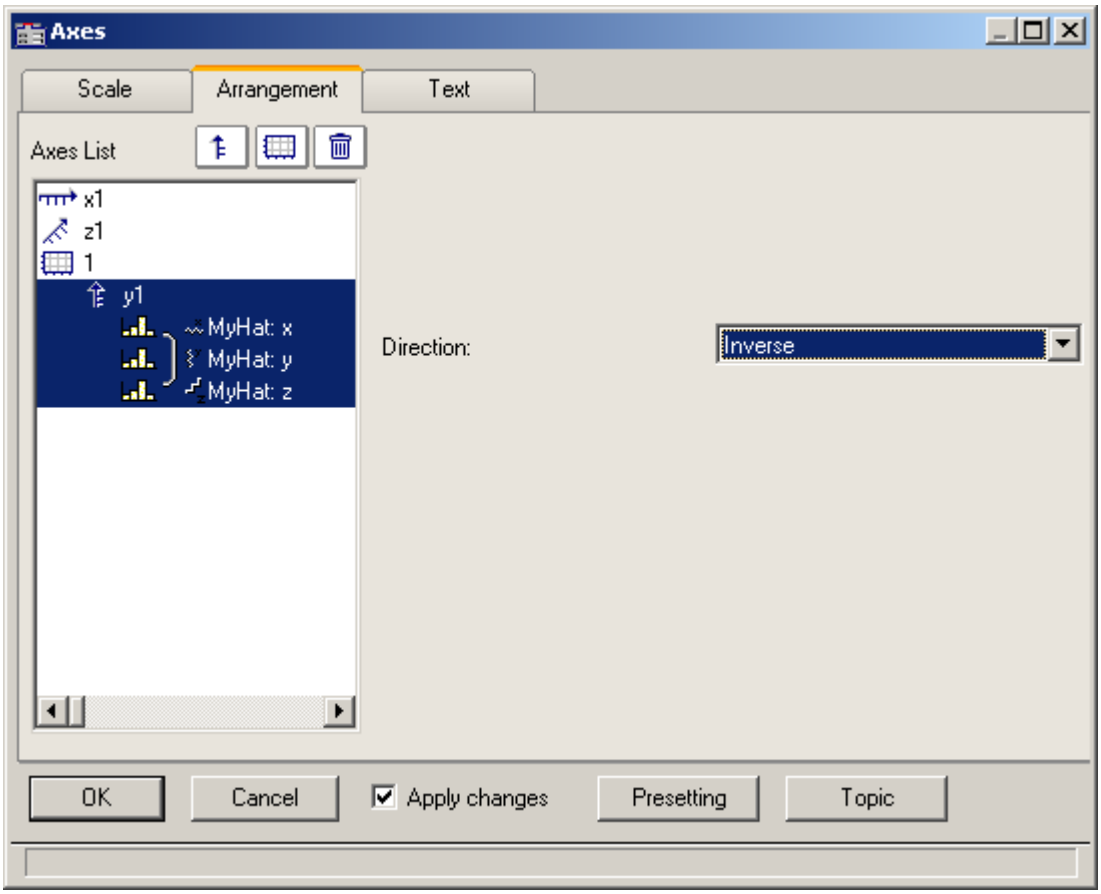
Labels at the axis ends:

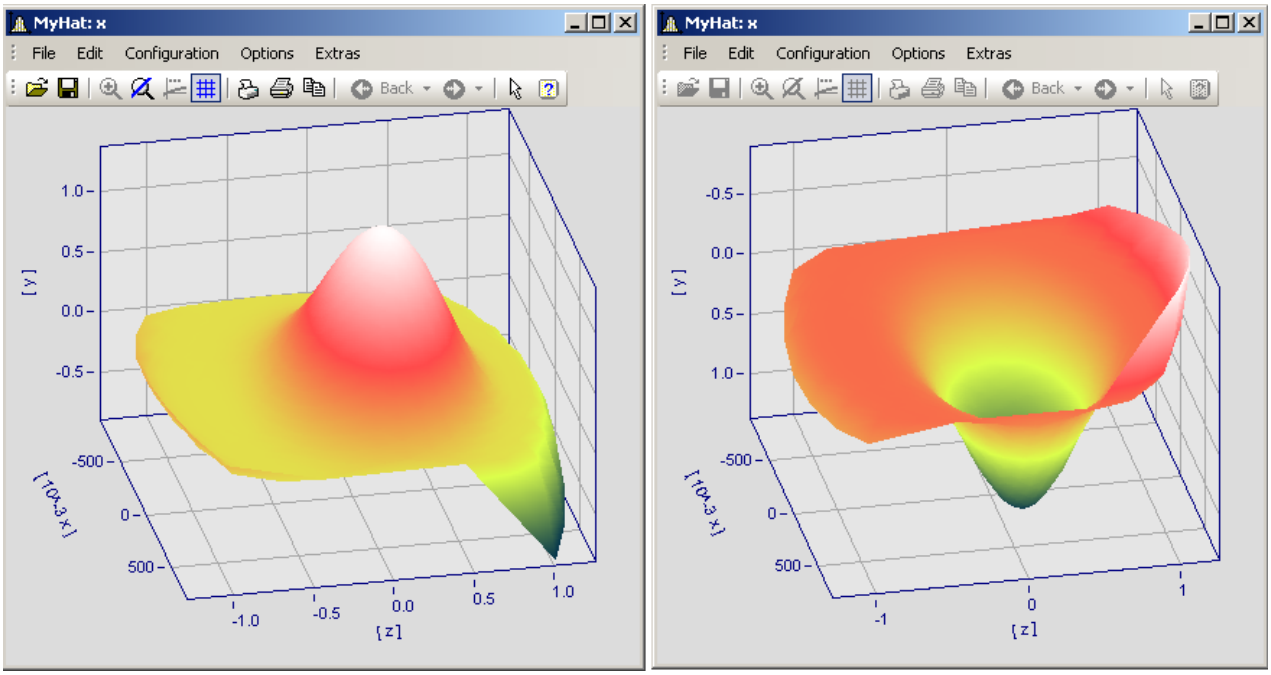
Alongside automatic orientation of the labeling, this control also offers *always unmoved* and *move if necessary*.

Inverting the axis direction

With 3D display, you can invert the direction of the Y-axis.

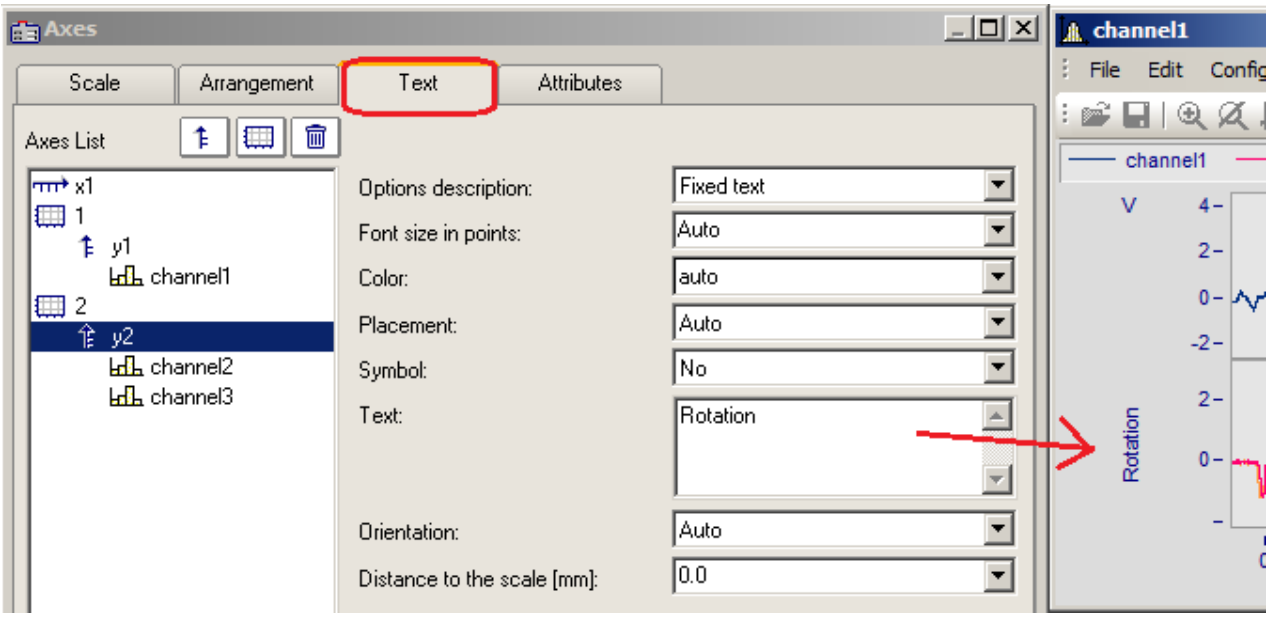
Use the menu: *Configuration\Axes* and select the page *Arrangement*. In the dropdown list you will find the entry *Inverse*.





Example: Invert the axis direction
At left the normal orientation, at right the inverted Y-axis

8.6.4.3.3 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:

Options description

No text: The default setting, in which the unit associated with the waveform is displayed horizontally at the upper end of the axis.

Fixed text: Here you can enter any desired text, such as "Length [m]", as a permanent feature. Seed also the notes on Greek letters 1069.

Unit: The unit saved with the waveform is displayed.

[Unit]: The unit saved with the waveform is displayed in angular brackets.

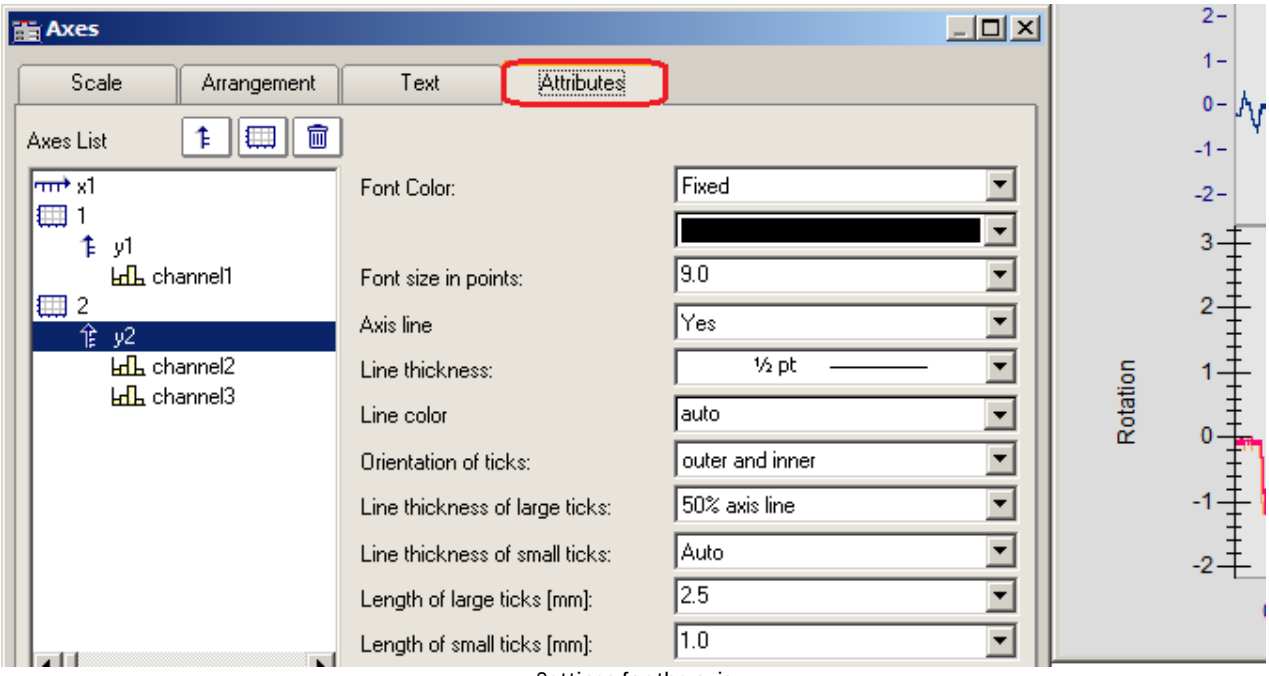
Name, Unit: The channel name is displayed in the first line of the coordinate system and the associated unit saved with the waveform are displayed.

Definable with placeholders: 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

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.

8.6.4.3.4 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.

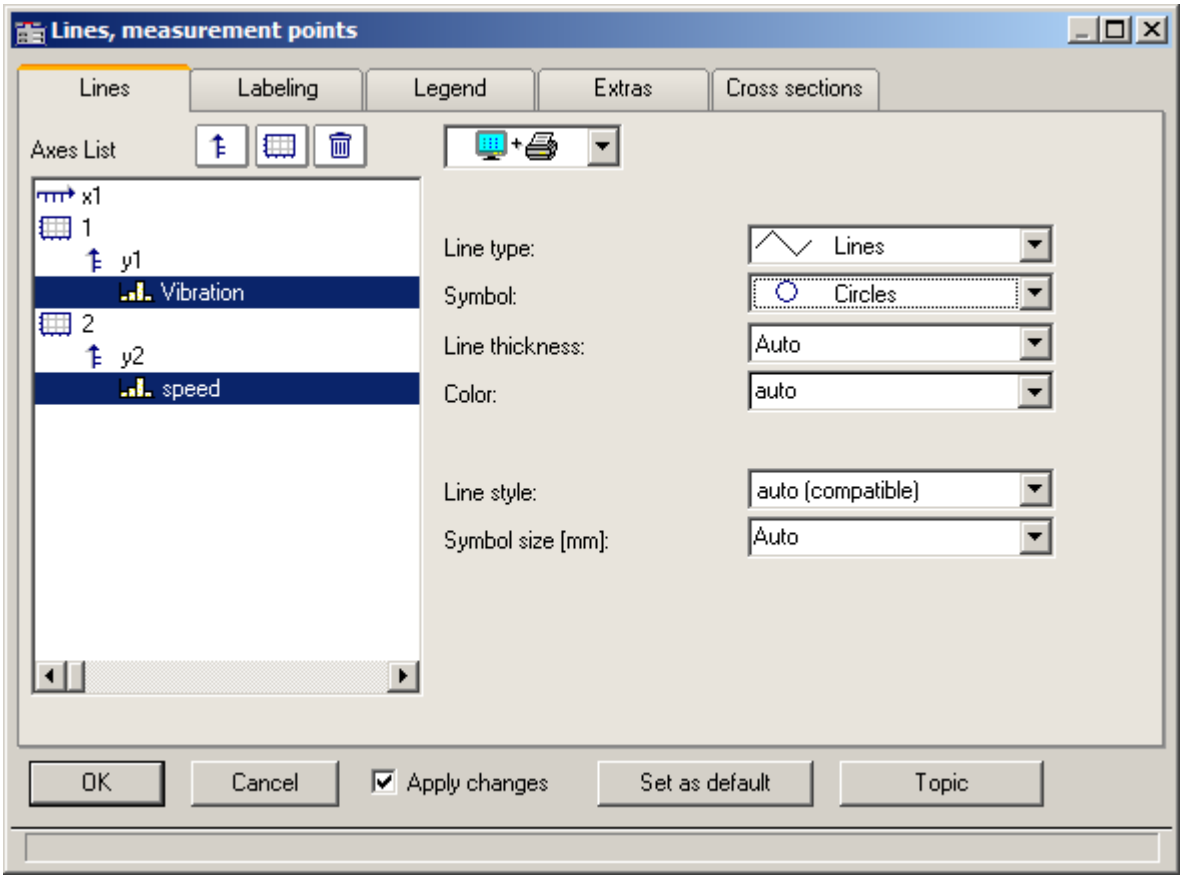
8.6.4.4 Lines

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:



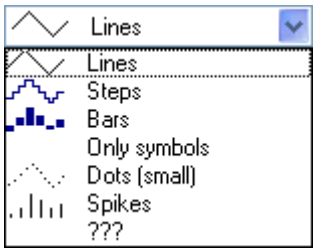
8.6.4.4.1 Lines

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



The *Lines* style displays the curve as a polygon, i.e. as straight lines connecting the points of the waveform.

When the option *Dots* is selected, only the points of the waveform are displayed as dots of one pixel each.

The *Bars* display shows every point of the waveform as a bar rising from the horizontal axis.

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).

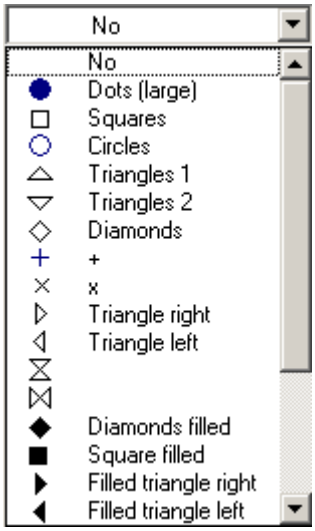
With the setting *Only symbols*, there is no continuous line. At each measured point, a symbol (e.g. a square) is drawn. The particular symbol desired must then be selected under the *Symbol* control.

Spikes are vertical lines extending from the x-axis to the height of the measured value.

The line type is always the same for the printer and the screen.

Symbol

Every measured point is indicated with a symbol.

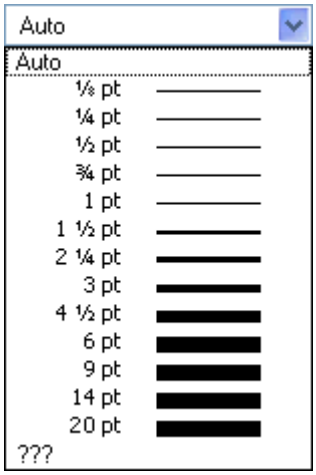


The symbol type desired is selected form this pop-down list. Symbols can be displayed either together with a line or all alone.

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* 910, in connection with the property *Number of symbols*.

The symbol always appears the same on screen or printed out.

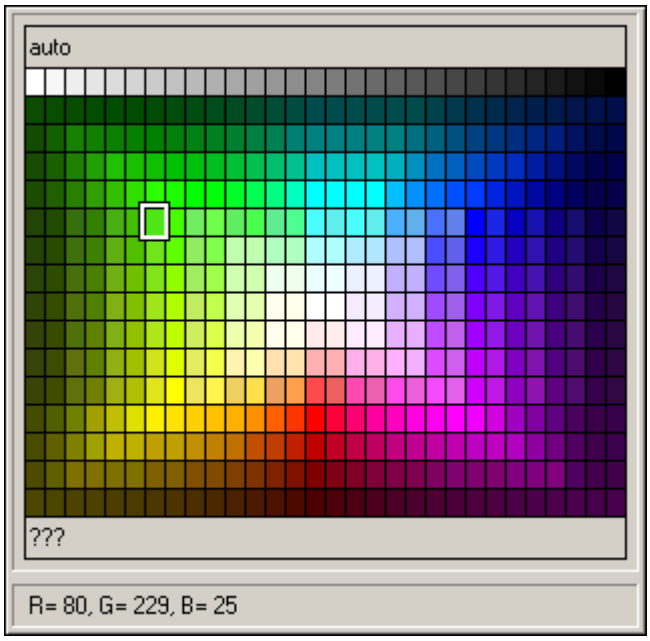
Line thickness



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

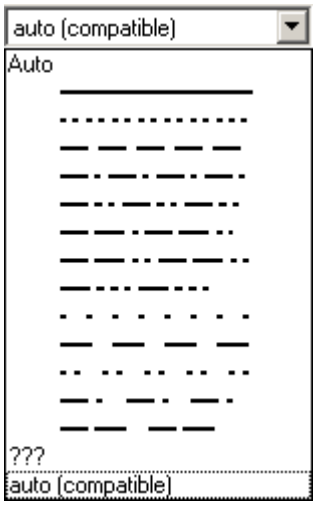


Here, the color of the line or symbol can be specified. If the color is set to *auto*, it is selected automatically by the system. In that case, a line's color depends on its order of appearance among the lines, and on the corresponding global settings for color assignments in the curve window; see the discussion of the menu item "Options / Colors 1050".

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

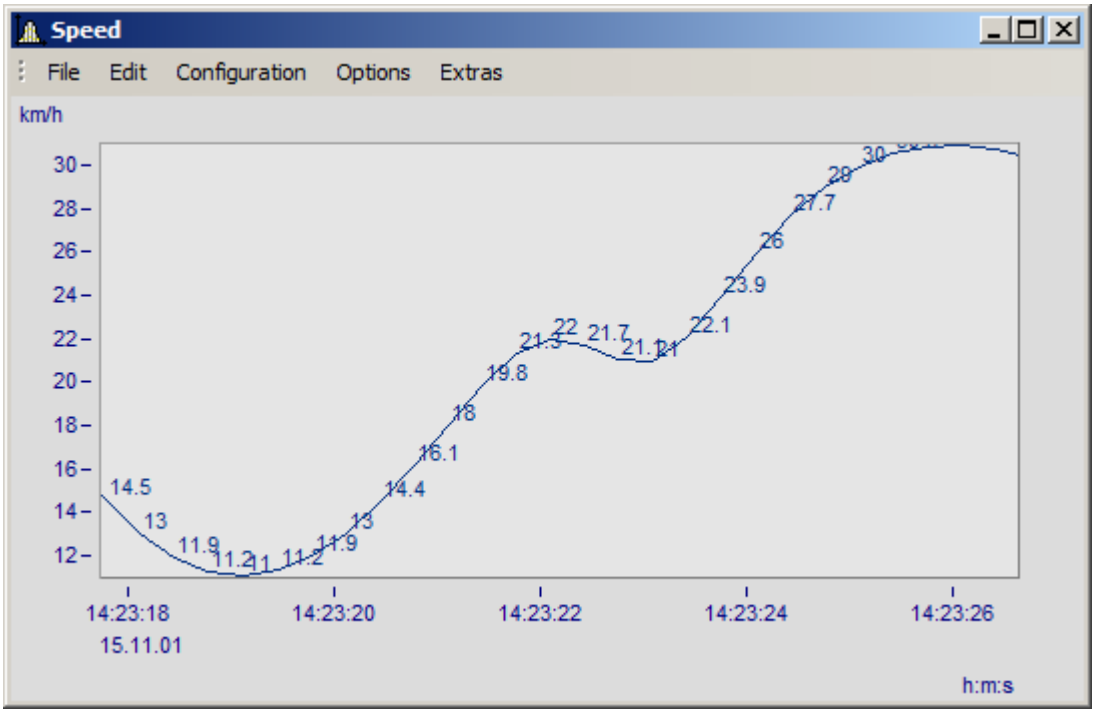


This determines the line's structure. If the structure is set to *auto*, then different line structures are assigned in succession, i.e.the first line is solid, the second dotted, the third dashed etc. For *auto (compatible)*, the global setting *Curves in structure* is observed. This mode is compatible with imc FAMOS 5.0, where the line structure was set globally in the *Colors*-dialog.

Symbol size [mm]

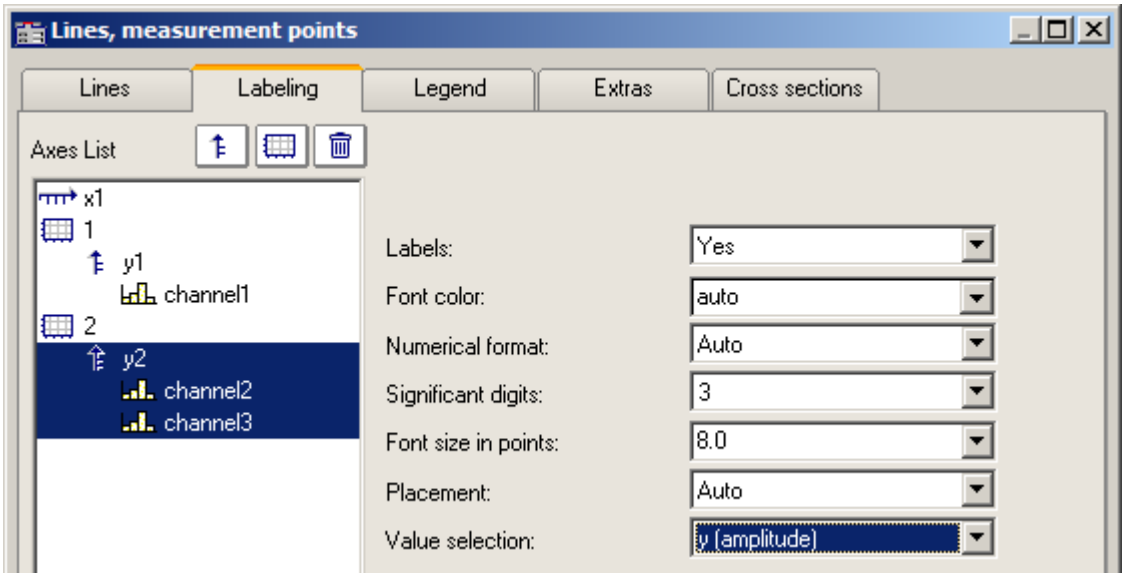
If symbols were selected for display of the measured points, the diameter in mm can be defined here. For *auto*, the global setting *Symbol diameter* is observed.

8.6.4.4.2 Labeling



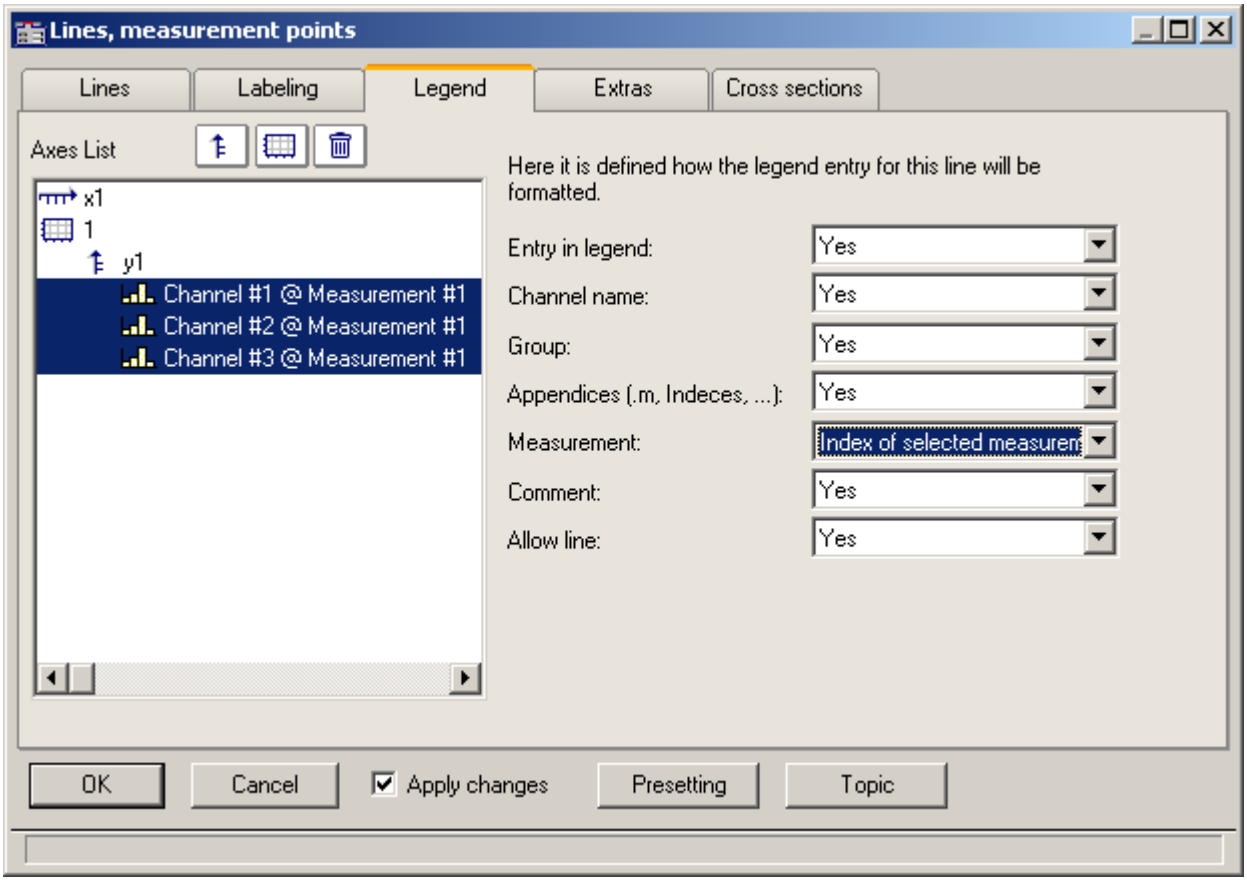
Here you can set whether or not the measurement points are labeled with text stating their numerical values.

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

8.6.4.4.3 Legends



Options for displaying the legend

By selecting this tab, you can select various options for displaying the legend pertaining to the line selected.

Hinweis 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* 976 to *always*.

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.

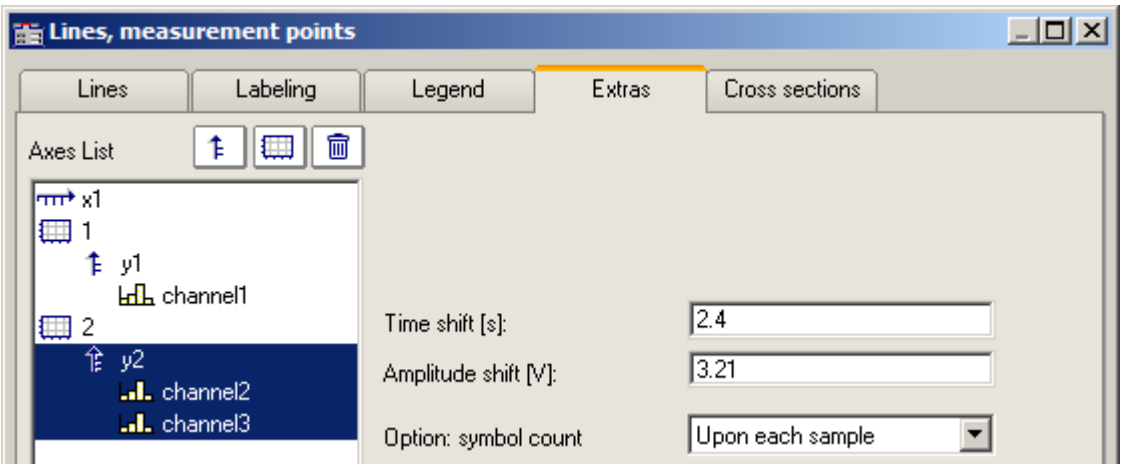
Comment: This is the channel comment as described in the imc FAMOS manual, in the chapter Properties/ Charact. Dialog (Waveforms). For channels recorded using imc STUDIO, the comment can be entered as a channel property.

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* 976.

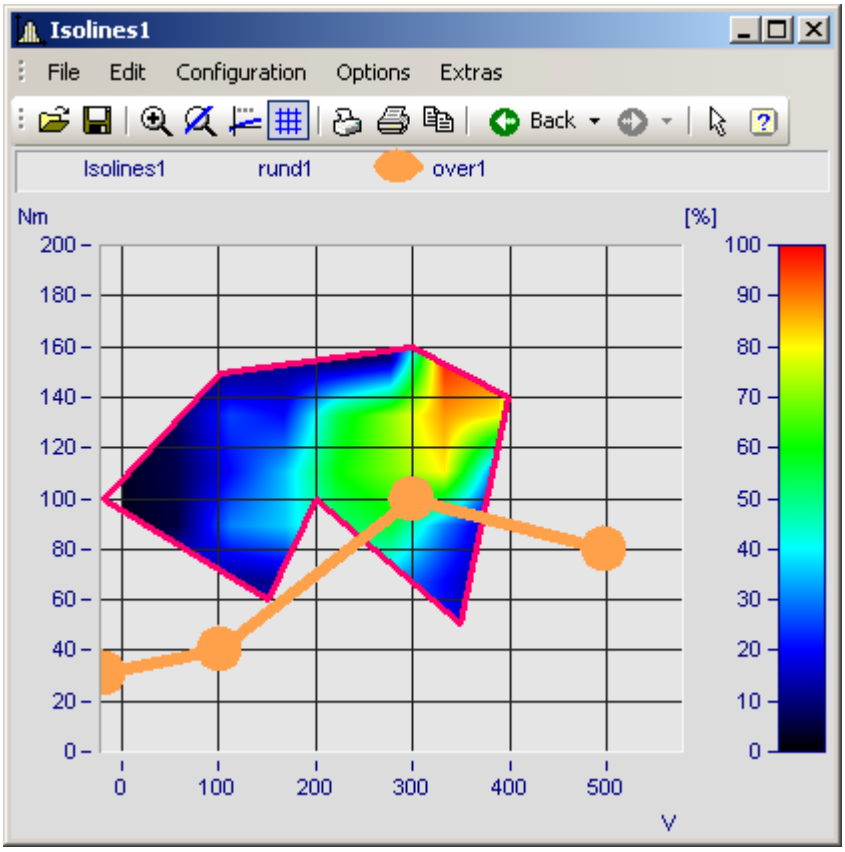
8.6.4.4.4 Extras

In display mode *Default*, you can manually specify the parameters of the Line-Shift function here. For more info, see the section Line-Shift 907.



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.



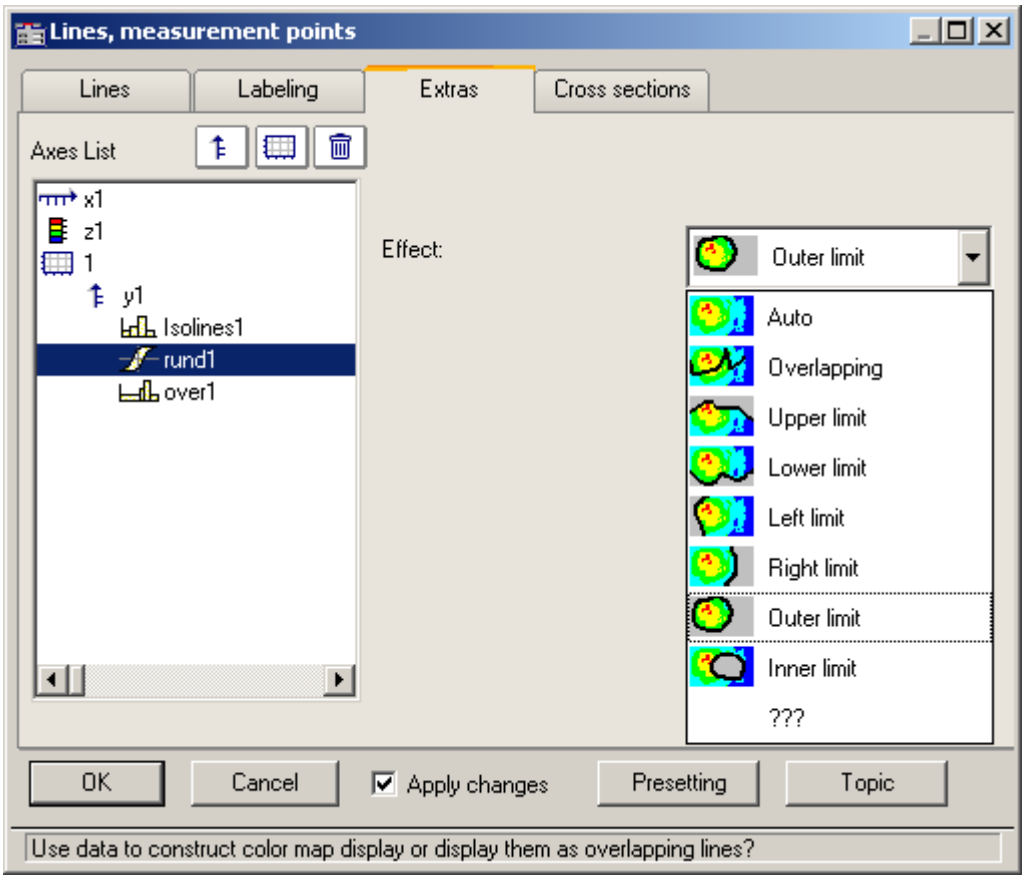
In this picture you see a function for constructing the color map, and a function for superimposing a line over the color map.

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.

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*.

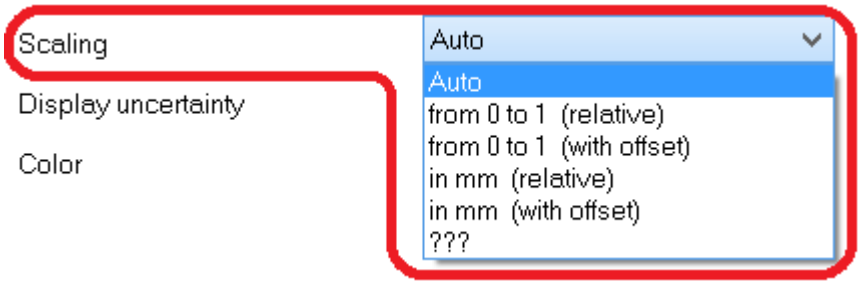


Scaling

Display uncertainty

"Filled area"

Scaling

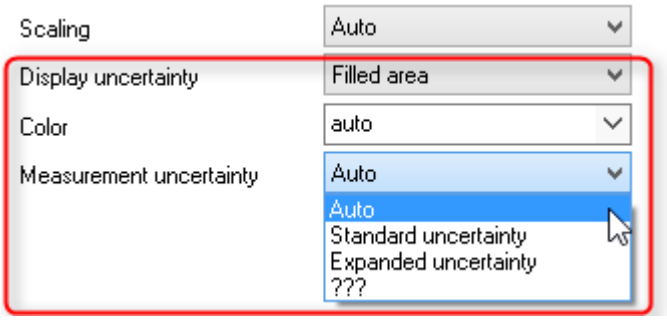


Independently of the coordinate system, the data can be scaled in millimeters. By this means, it is possible to use a data set to define a graphical object, which retains a constant size regardless of the current zoom level.

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.

8.6.4.4.5 Cross sections

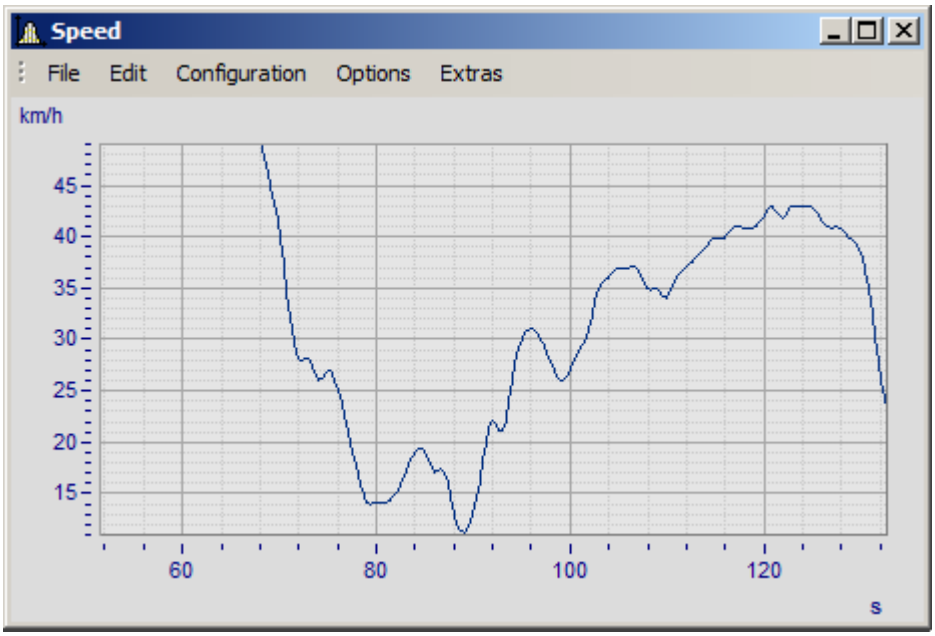
Here you can set the default appearance of cross-sections for x-, y-, and z-data sets or segmented waveforms or segmented waveforms with the respective position. For more info see the section Connect with 3D 906.

8.6.4.5 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* 910). Settings for the secondary grid are made in the Axes dialog by means of the control *Small ticks* 1010.



Various line widths and types can be specified for the clipboard or print-out, see menu *Opt./ Clipboard Settings* 1044.

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

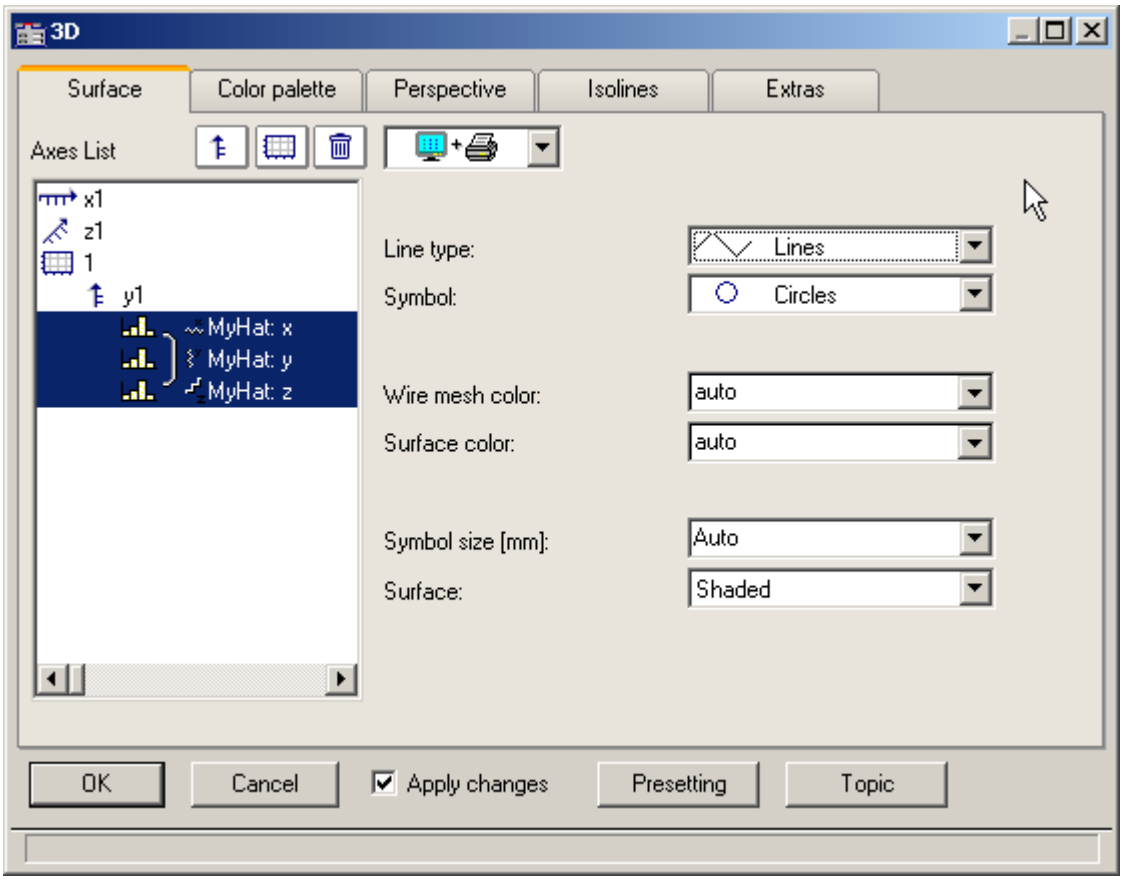
- The color of the grid lines can be determined in the *Colors...* dialog.
- A grid is not drawn if labeling is not visible.
- The menu option *Grid* is marked with a checkmark when grid display is selected.

8.6.4.6 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.

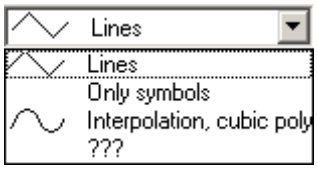
8.6.4.6.1 Surface

In the dialog page *Surface* there are additional settings for 3D-display besides the settings described under Lines 1018.



Line type

In 3D-display, the only available line types are straight lines, no lines, or interpolated lines connecting measured points.



Symbol and Symbol size

See the section on lines 1018.

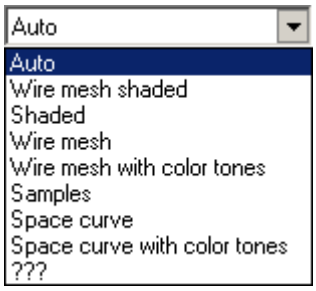
Wire mesh color

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 1030).

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 1030).

Surface



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.

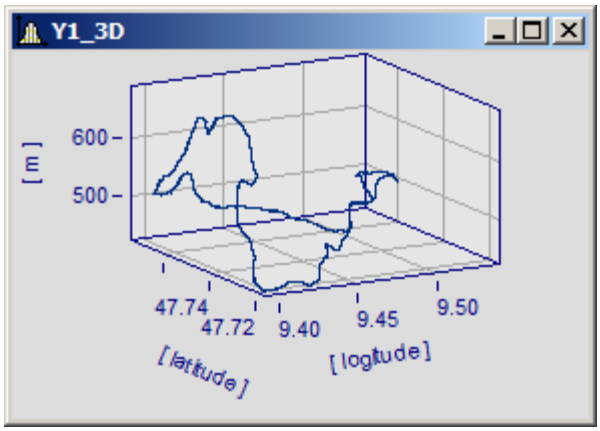
Wire mesh: The individual measured points are connected to lines in a grid.

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.

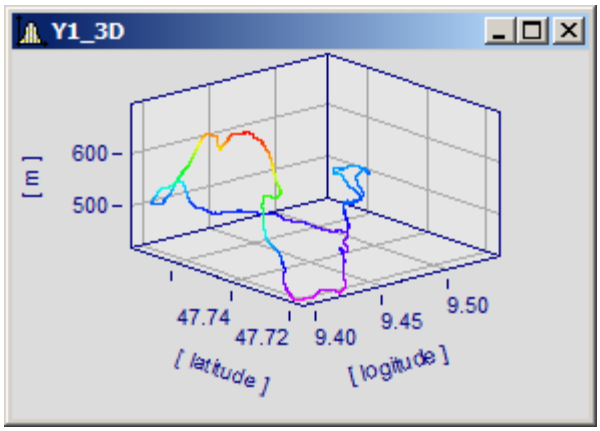
Samples: The individual measured points are displayed as points or, if selected, as symbols.

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.



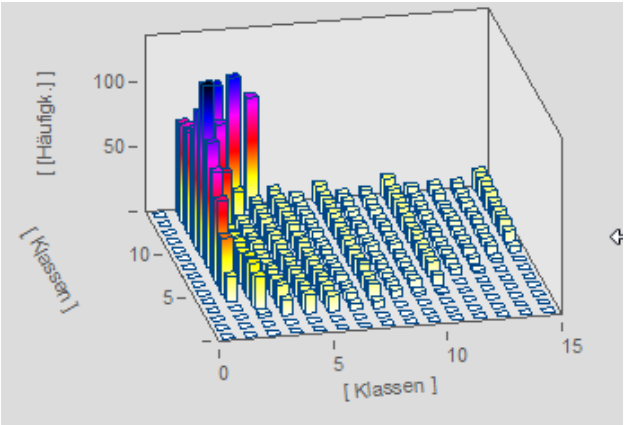
Monocolor space curve



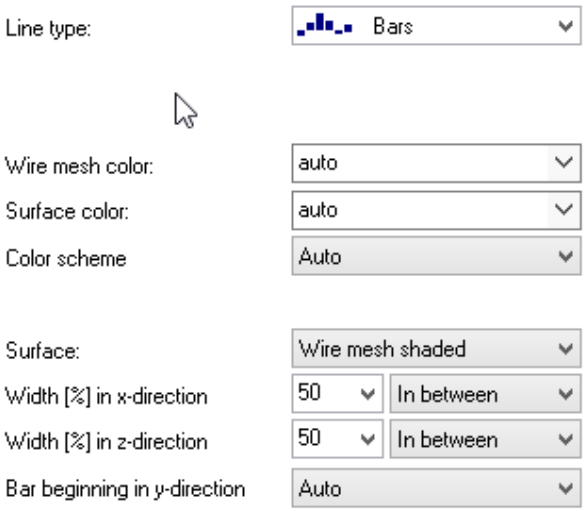
Space curve with color palette

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.

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

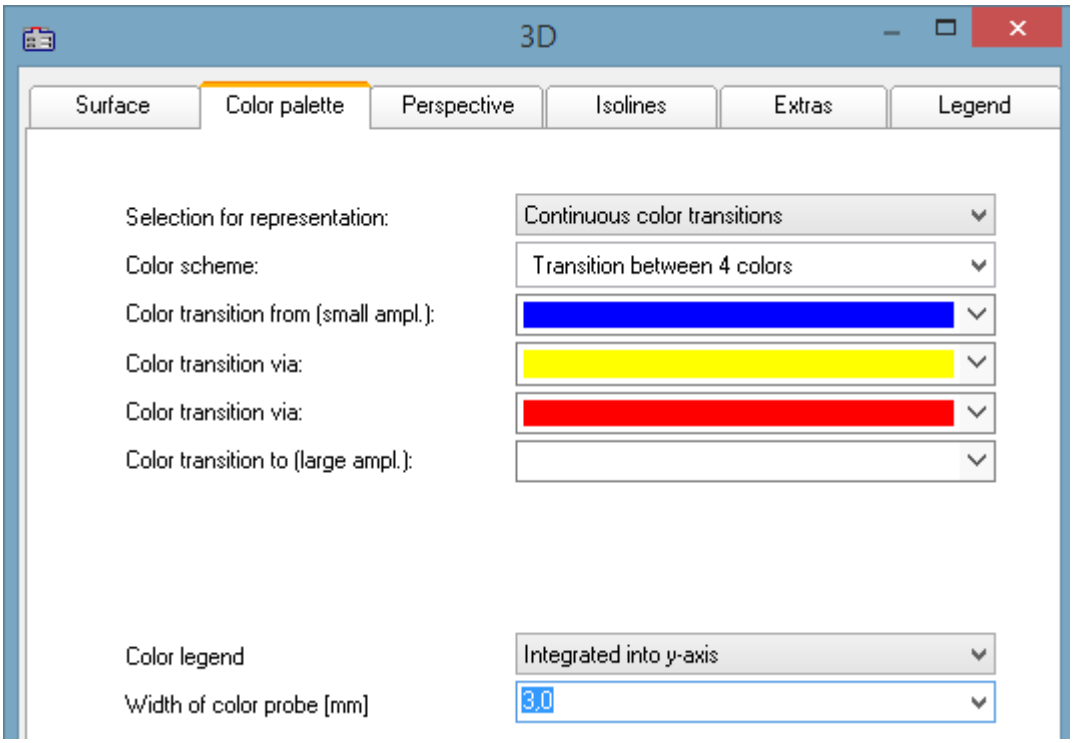
The width and depth of the bars can be set in terms of a percentage. The position may be arranged as *In between, Centered* or *Aligned*.

Bar beginning in y-direction

This sets the height of the bottom surface.

8.6.4.6.2 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" 949



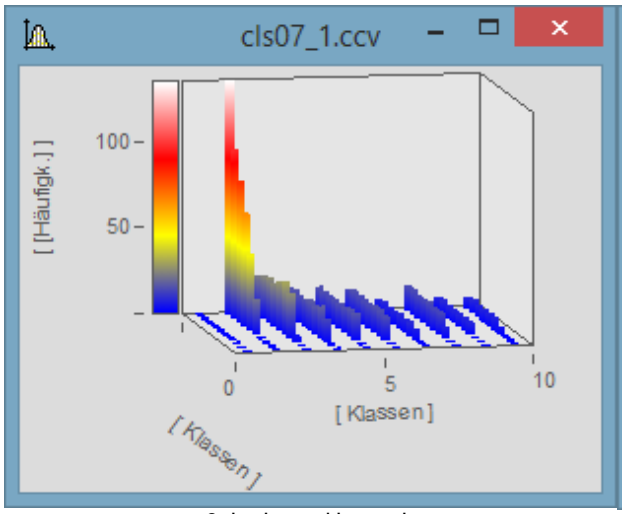
Color legend

The scaling of the colors can be indicated with a *color legend*. The following options are available:

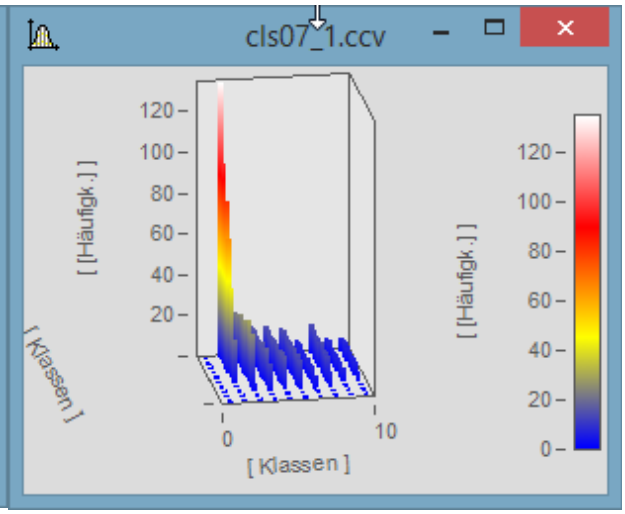
Auto, No: No color legend

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

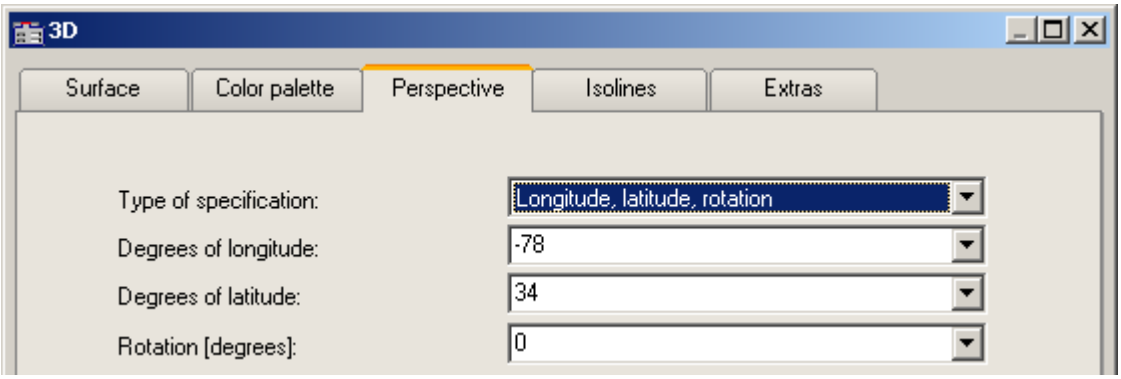


Color legend separate right

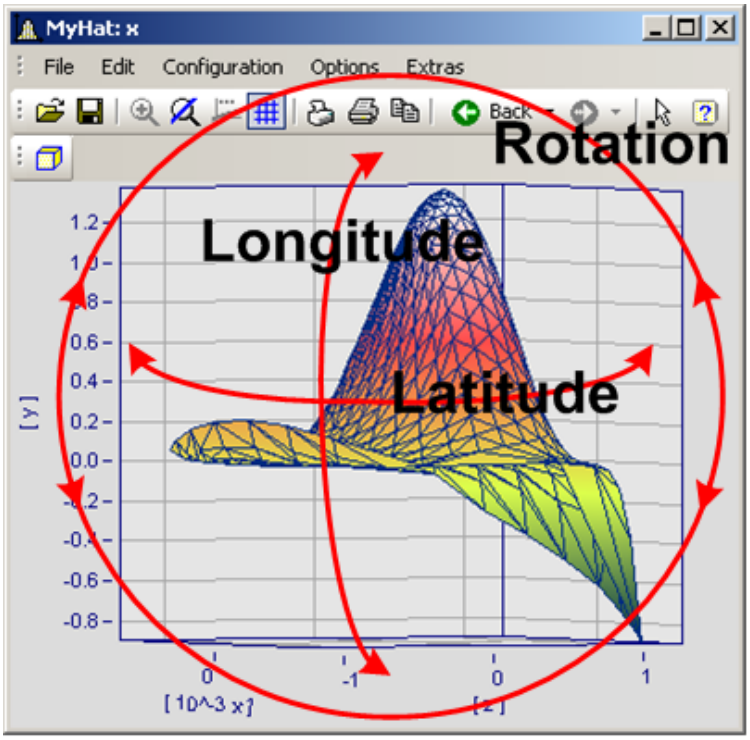
Reference

One way to color the 3D display according to a fourth variable is presented here 992.

8.6.4.6.3 Perspective



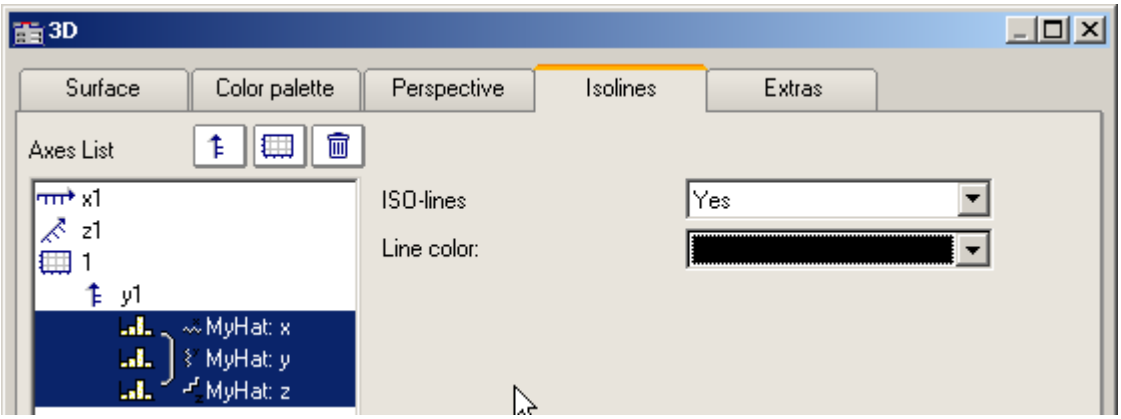
In the dialog page *Perspective*, you can set the viewing perspective as an angle between -180° and +360°. There are three ways to make the specification: *Angle of z-axis*, *Longitude* and *latitude* and *Longitude*, *latitude* and *rotation*.



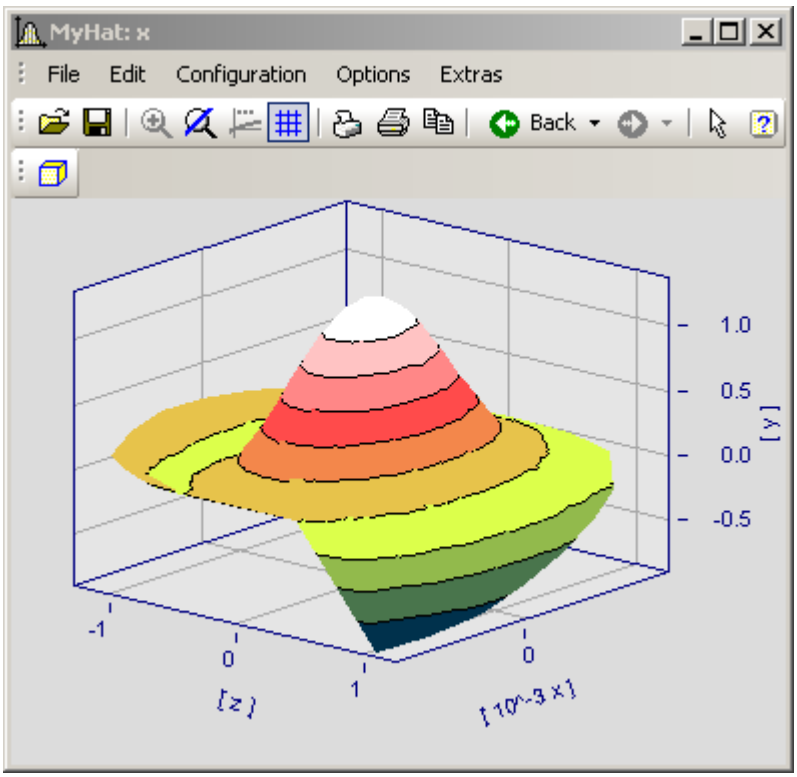
The perspective *Longitude* and *latitude* offers the best mouse control. This view behaves like a survey of the earth from outer space. This is the recommended perspective. If you wish you can additionally specify the rotation.

The three perspective parameters are related as shown below.

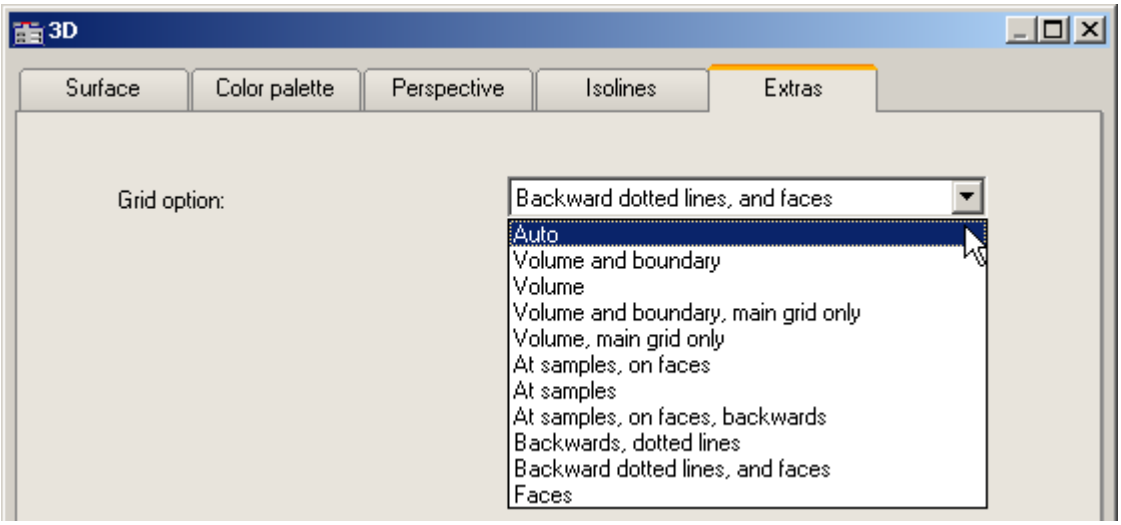
8.6.4.6.4 Isolines



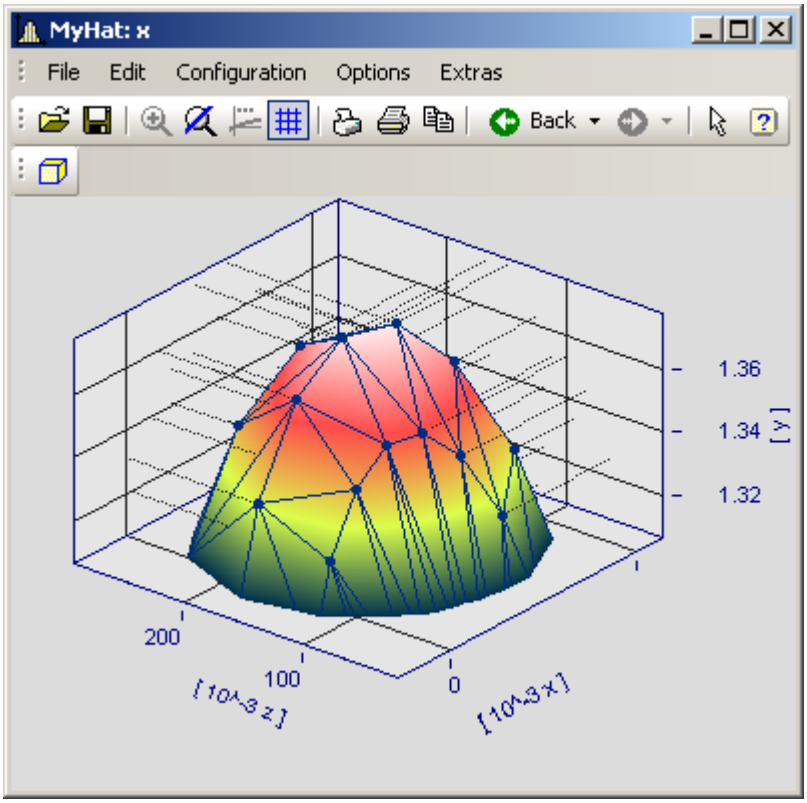
In the dialog page *Isolines* you can select whether to have isolines superimposed on the toned surface, like in color map display. This function can only be activated if graduated colors are set on the *color palette* 1030 page.



8.6.4.6.5 Extras

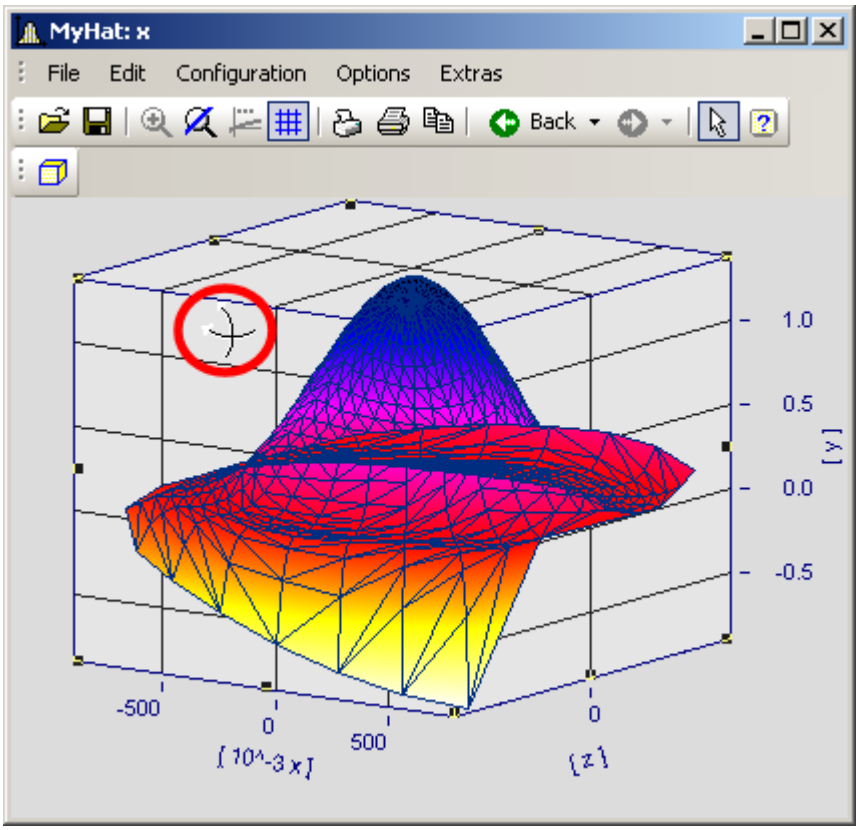


The dialog page *Extras* offers access to various grid options. This grid is then displayed in the space, in addition to the standard coordinate grid, for better legibility of a measurement point's axis values.



8.6.4.6.6 Rotate

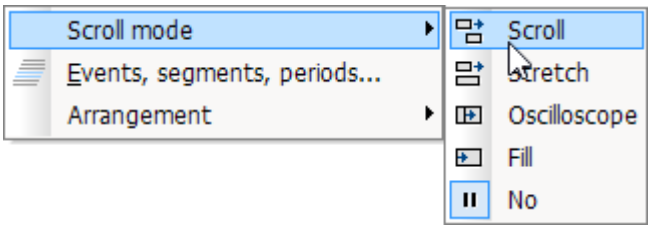
For spatial manipulation of a 3D-display, bring the mouse cursor in the coordinate system, which makes its shape change. If you then click on the mouse and start moving it, the cursor shape changes, as shown in the picture, and the 3D-display can be freely rotated by moving the mouse around.



Cursor for free rotation of a 3D-display

8.6.4.7 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 868 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.

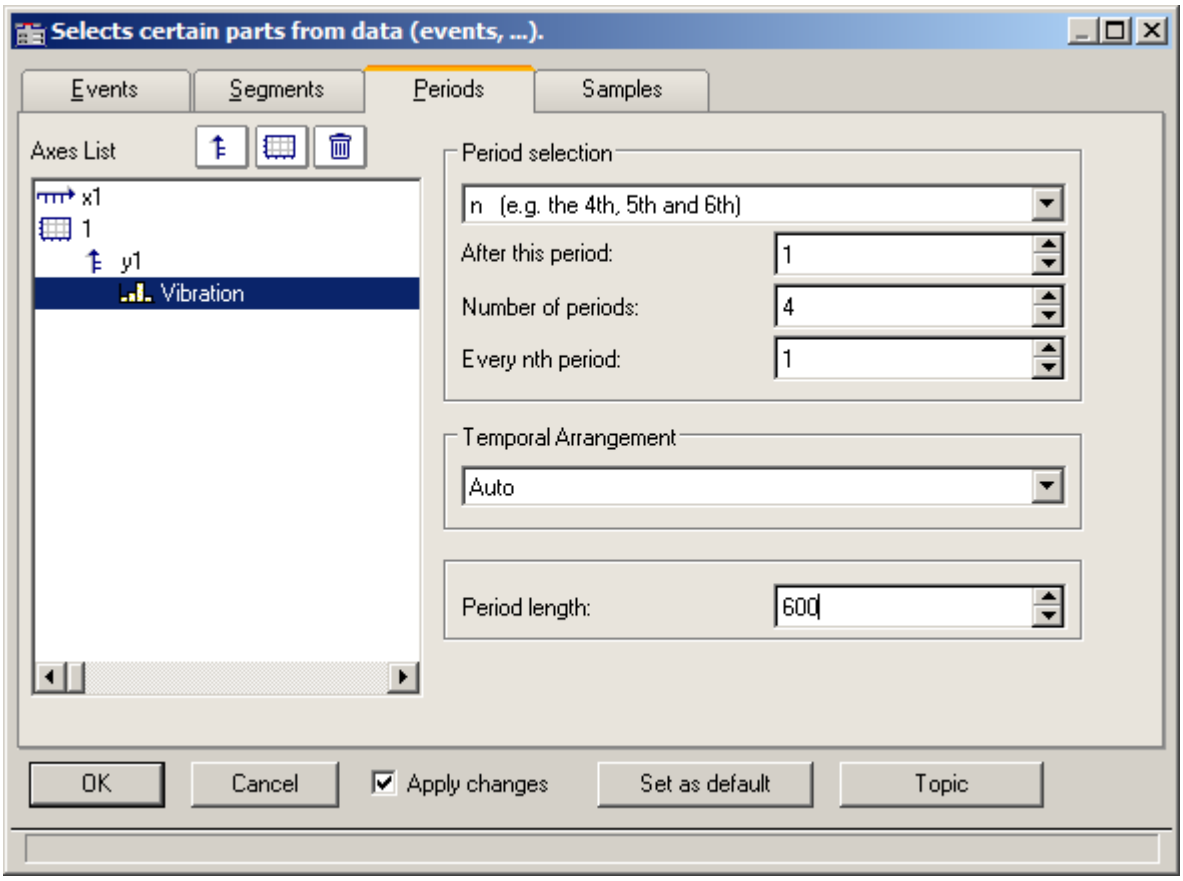
8.6.4.8 Events, Segments, Periods

Function

With normal waveforms, it is possible to perform a comparison of signal periods 1040 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 explained using an example of *Period Comparison* for illustration. Subsequently, the differences between this procedure and the selection of *Segments* and *Events* for comparison are explained.



The dialog is called by selecting the menu item *Configuration / Events, Segments, Periods*.

For general information about period comparison, refer to section Period Comparison 1040.

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:*

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.

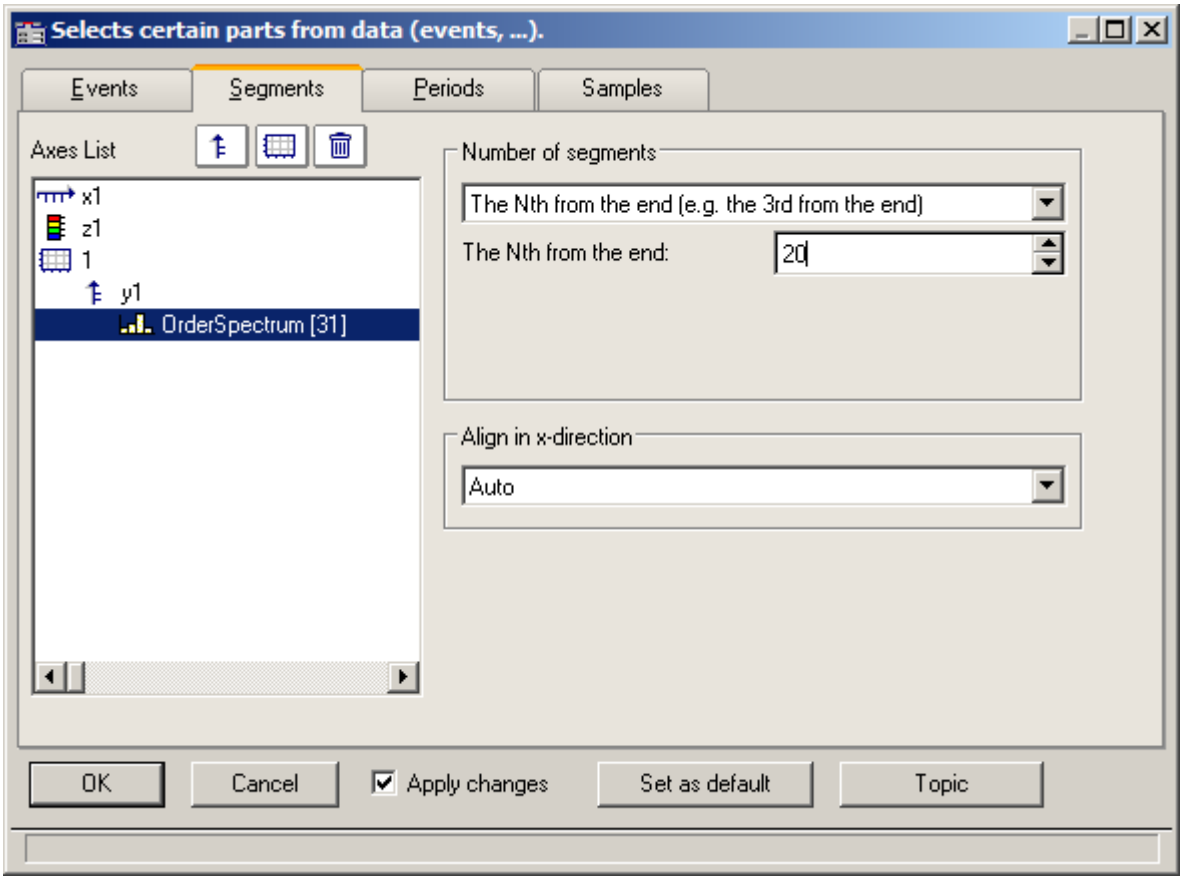
x0 of the last period: The period comparison representation's initial x-coordinate is defined as the x-coordinate of the first measurement value belonging to the last period in the original data set.

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.

Segments

For the display of segmented data. Segmented waveforms are matrices, as generated, for example, by imc DEVICES / 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.

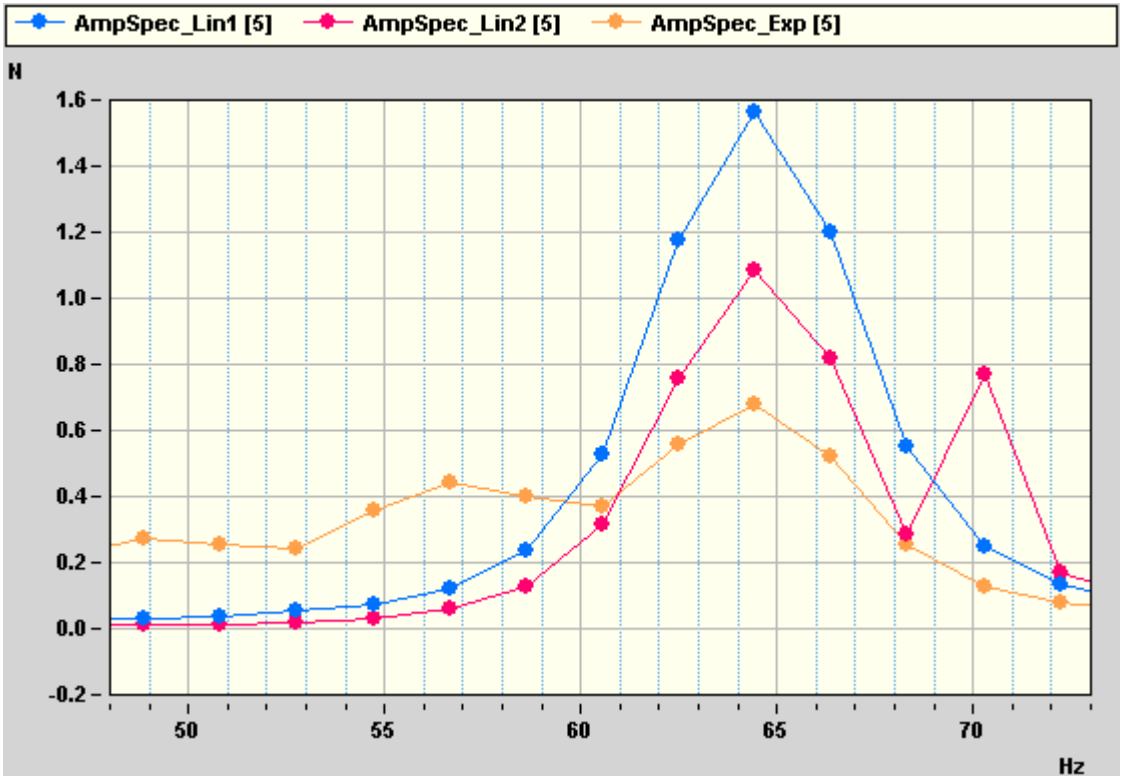
Then select *Align in x-direction:*

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:

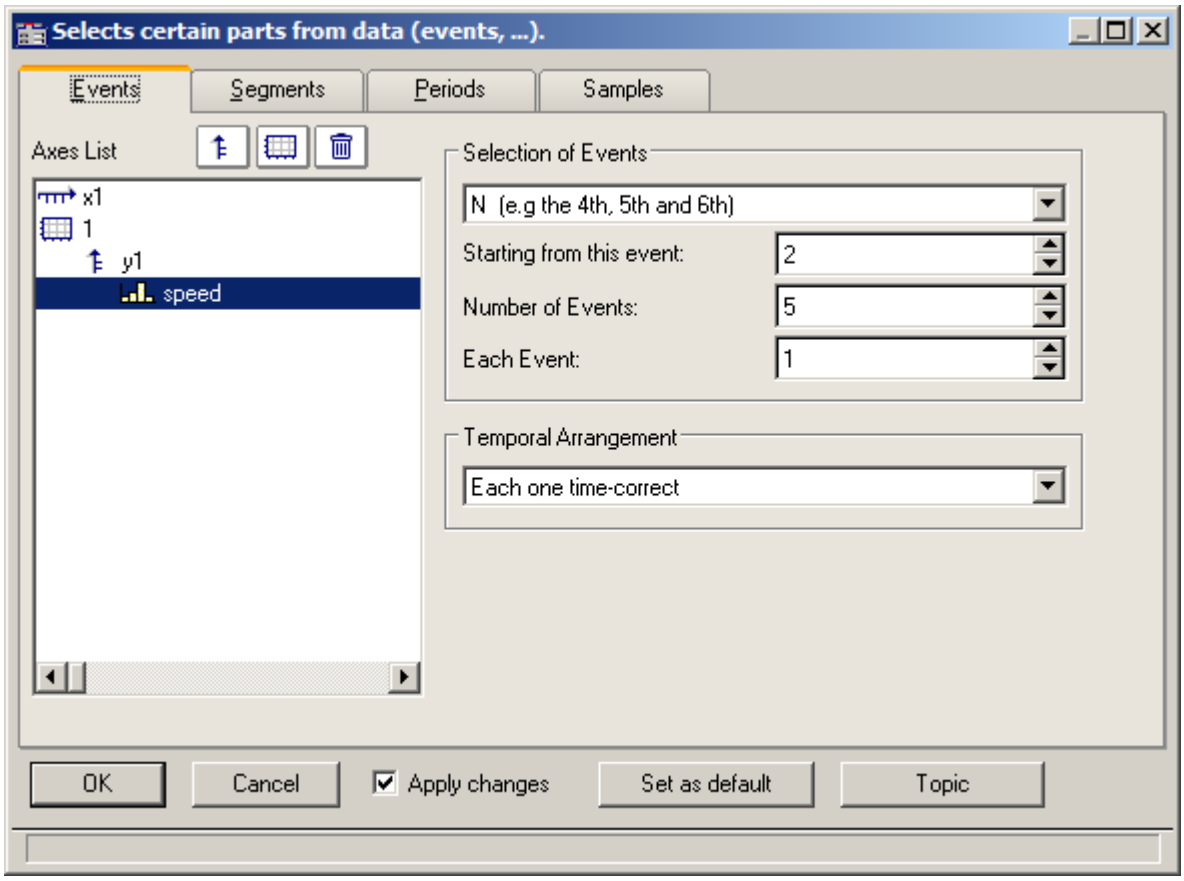


Legend with segment indices

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:*

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.

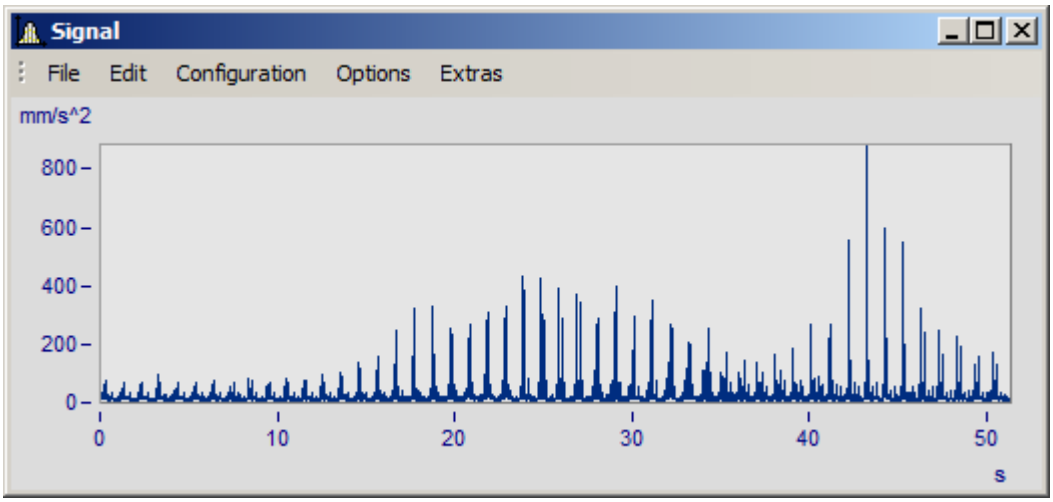
Trigger time difference to last displayed in x0: As above, but based on the last specifically selected event.

8.6.4.8.1 Period Comparison

Periodic signals occur in a variety of applications, such as rotating machinery, where the measured signal almost repeats itself at each revolution.

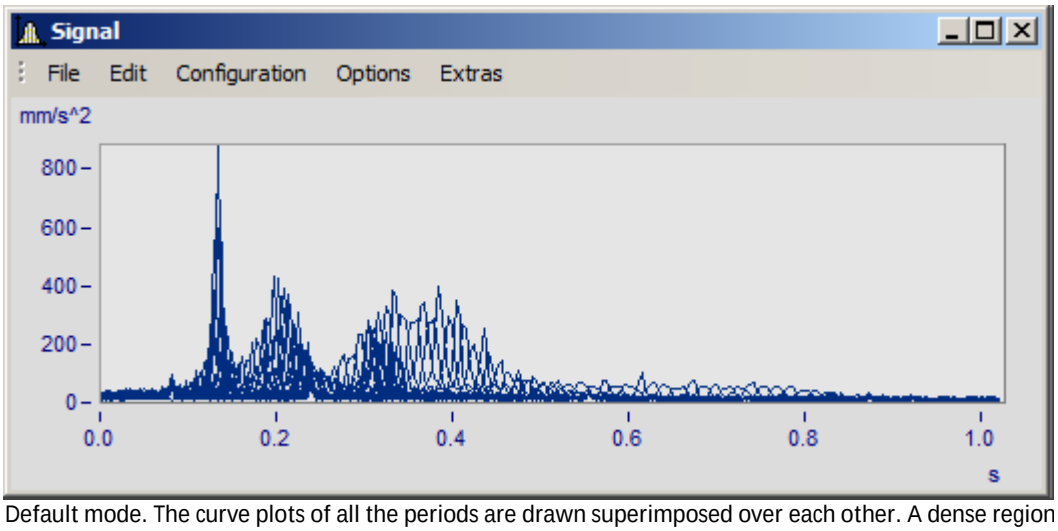
In the curve window's default display mode, entire waveforms appear plotted over time. This means that all periods are drawn in a series. In such a display, it is not always possible to compare individual periods in terms of their maximum values, for instance, or even other properties.

Example:

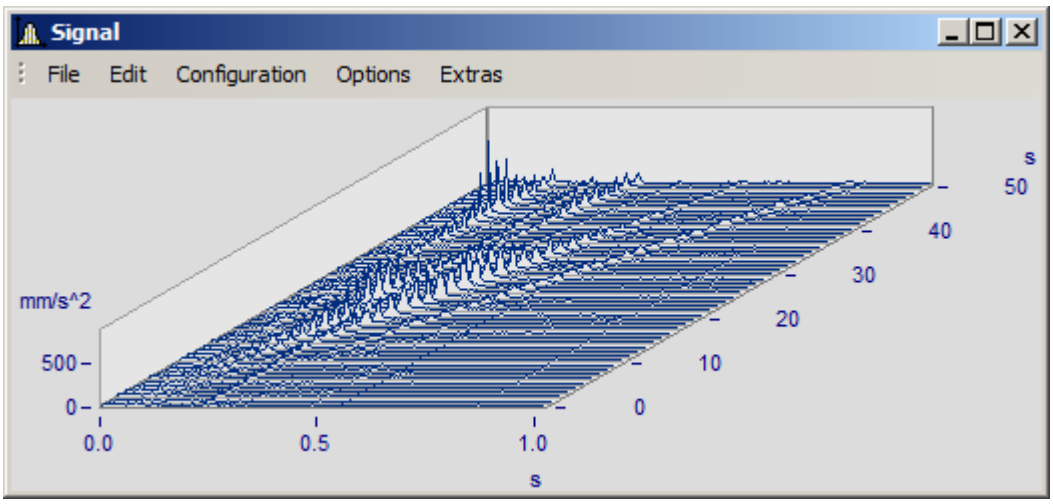


A waveform with 50 periods of 1s duration each is shown

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.

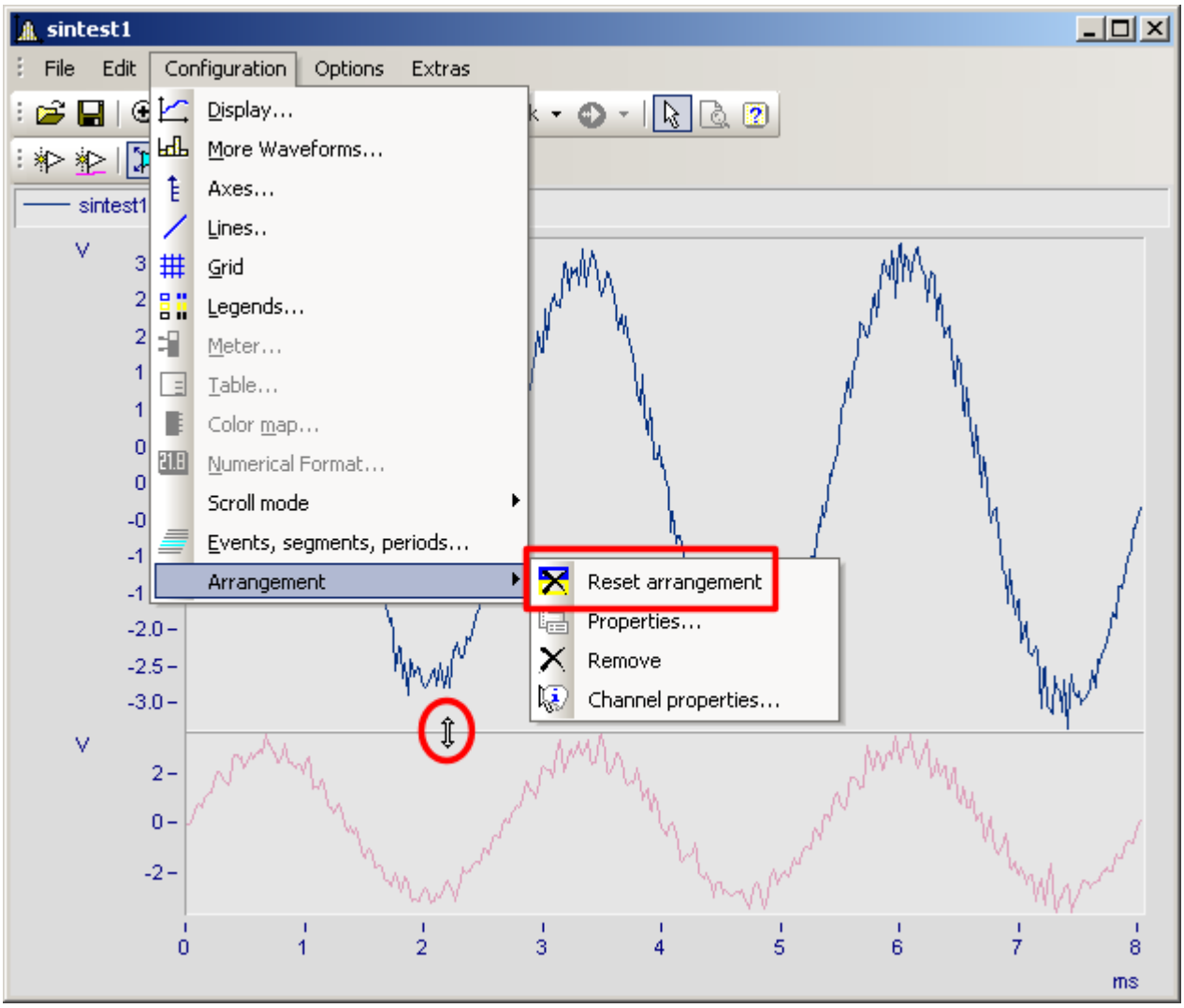


Waterfall diagram: the mountain-range-shaped surfaces clearly illustrate trends emerging across the accumulating signal periods.

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.

8.6.4.9 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 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.

8.6.4.10 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

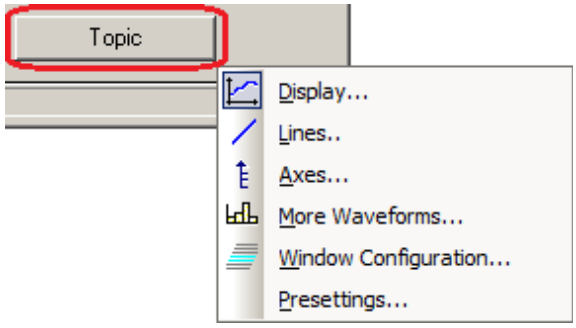
The dialog is closed and any changes made are discarded. This also applies to changes made and displayed in the curve window using the option *Apply changes*.

As presettings or Set as default

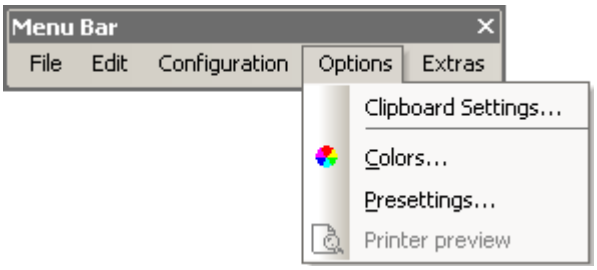
This saves the currently effective settings as default settings for future curve windows. The settings are saved to the Windows Registry. If you wish to transfer the default settings to a different computer, use the program XConfig, which is located in imc FAMOS' BIN folder.

Topic

From here, you can go directly to the properties dialogs for other objects.



8.6.5 "Options" menu



	Menu item	Description
	Clipboard settings 1044	Settings for graphics created for output or export with the clipboard are specified here.
	Colors 1050	Changes the colors in the curve window.
	Presettings 1053	Presettings such as the standard configuration directory and drives for storage of temporary files are made here.
	Printer preview 1059	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.

8.6.5.1 Clipboard Settings

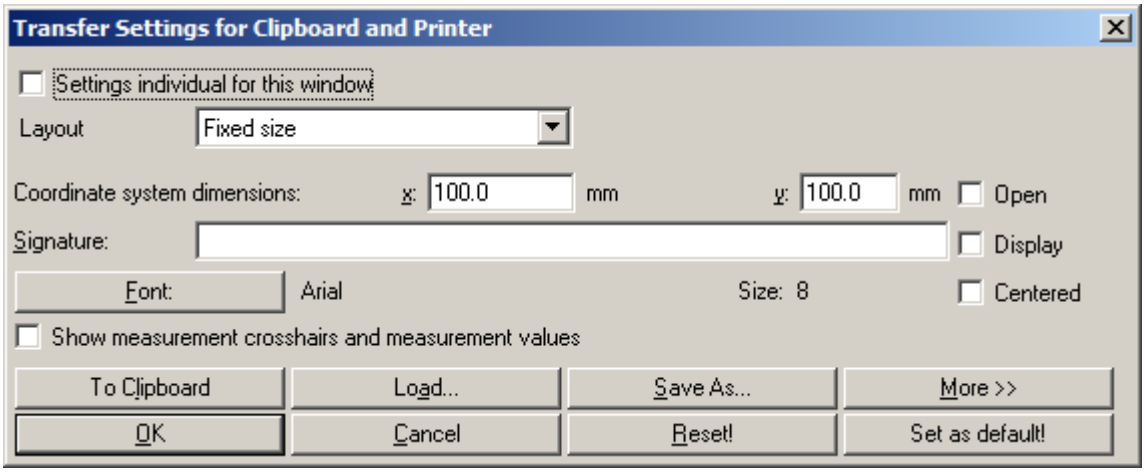
Function

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:



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.

Layout:

If you wish to save the curve window to the Clipboard always having the same proportions as currently displayed on the screen, select *Proportions as on screen* (the default setting!). If instead you select the layout *Fixed size*, the curve window is always saved in the dimensions specified using this dialog .

Coordinate system:

The height and width of the coordinate system are entered in mm the text boxes "x:" and "y:" in the line *Coordinate system 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.

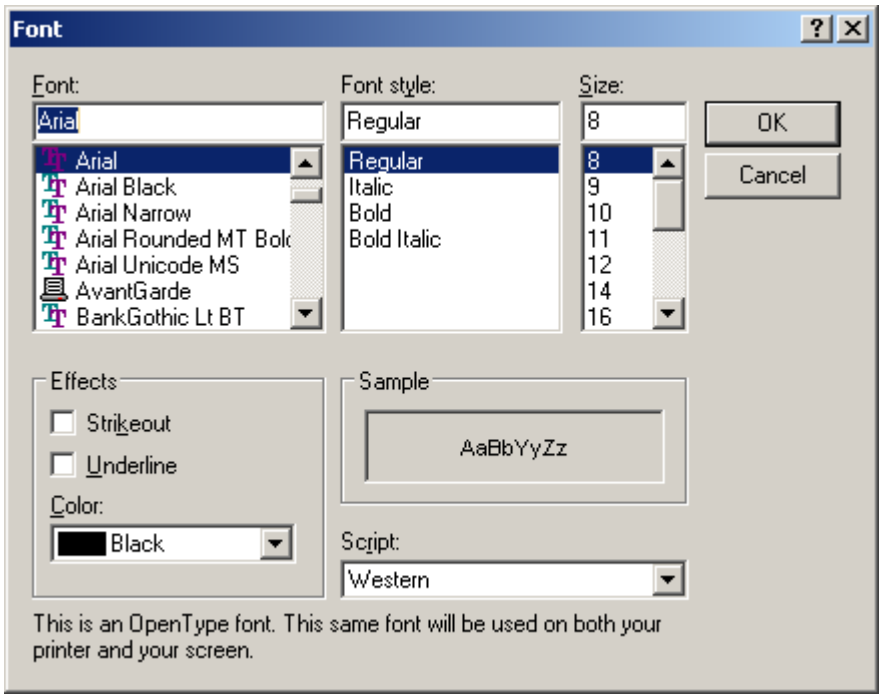
Centered:

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 following 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...

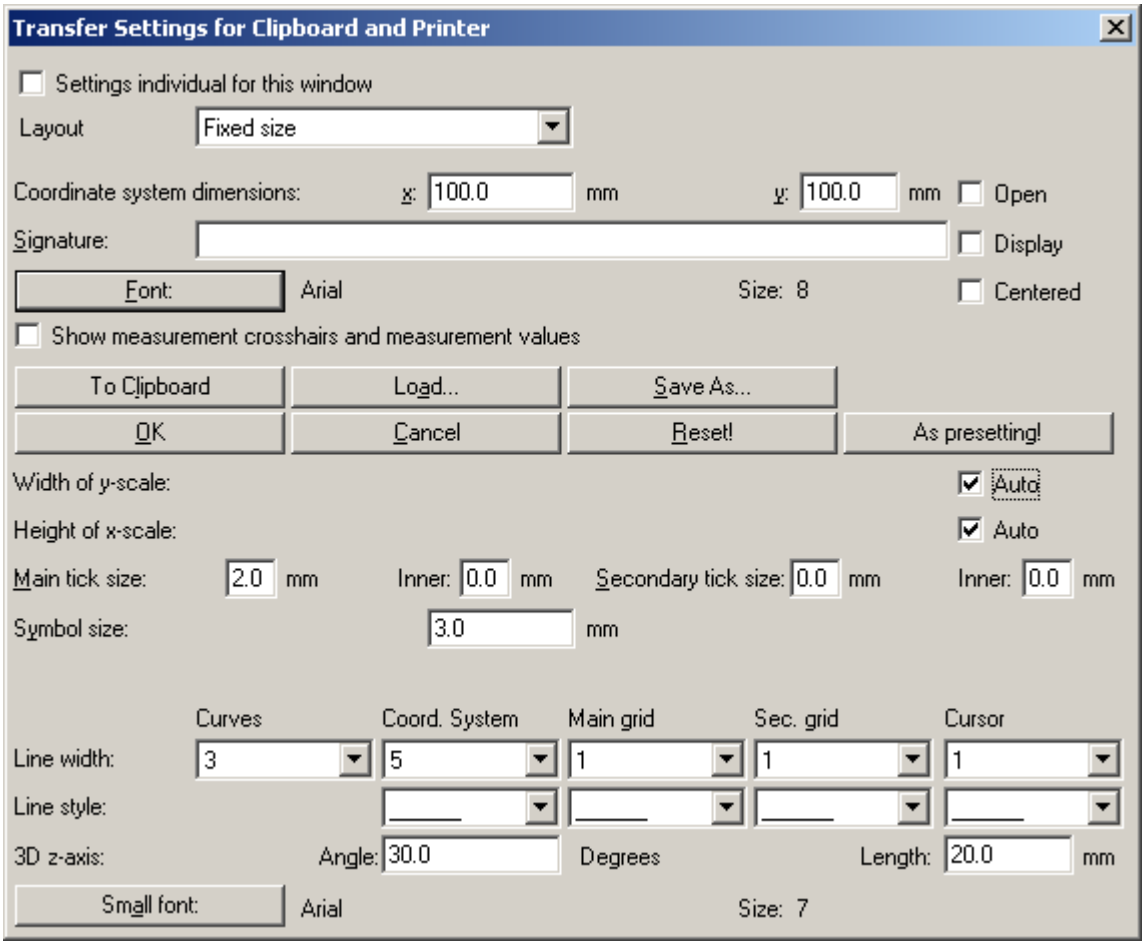
A dialog appears for saving the contents of the *Clipboard settings* in a file.

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.

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:



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* 1005 and *Configuration/ Display* 910. The size of the symbols is specified here. Enter the diameter of the symbols in mm, with up to one place after the decimal.

Line width:

The width can be specified for the following lines:

- curves
- coordinate system
- main grid
- secondary grid
- measurement cursor

Any specified line thickness 1018 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

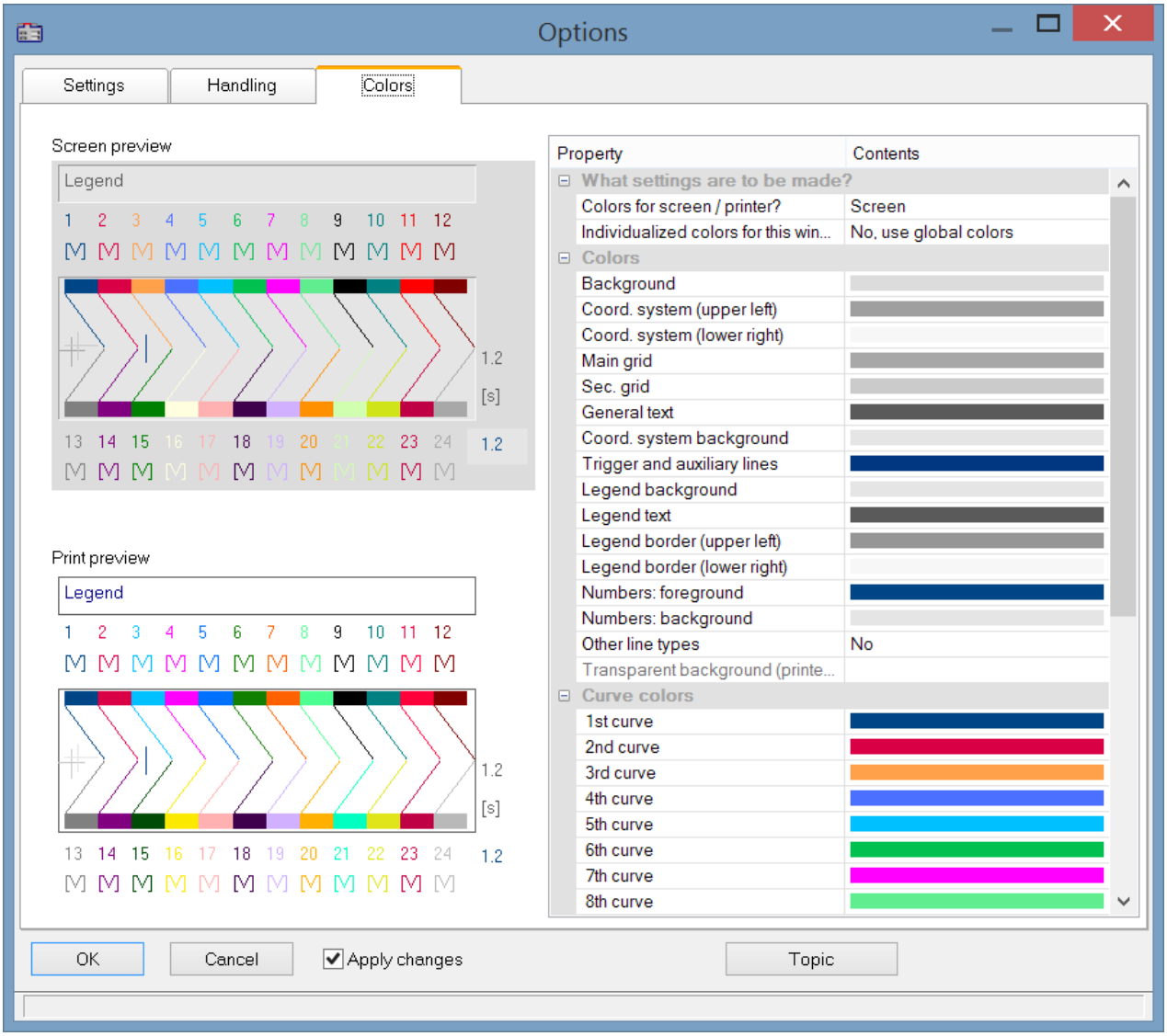
- The settings in this dialog apply to all curve windows.
- The fonts available for selection depend on the printer set for the curve window.
- When graphics are transferred to the clipboard, it is important that the printer set for the curve window is also the printer with which the graphics will later 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.

8.6.5.2 Colors

Function

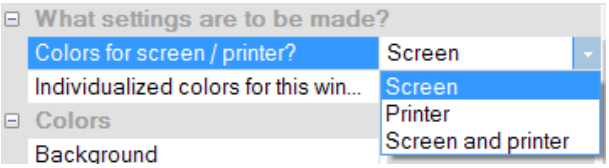
All colors used in the graphic interface of the curve window can be defined as desired in a special dialog. Since imc FAMOS 7.0 single windows can use an individual color scheme.

The following dialog is used to define the colors:



The colors are defined by a palette

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...*

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.

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.

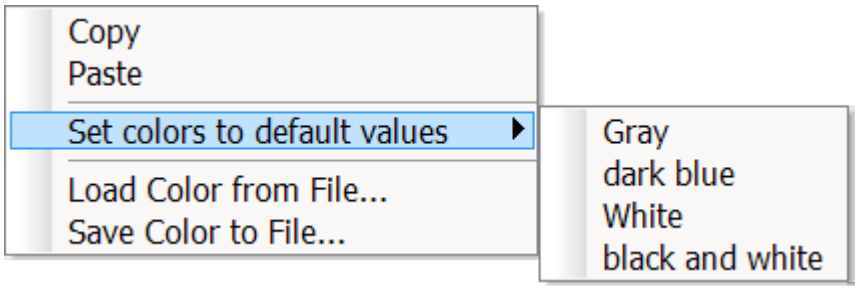
A dialog appears in which the colors can be defined. This dialog contains a list of the graphic elements in the curve window.

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:



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

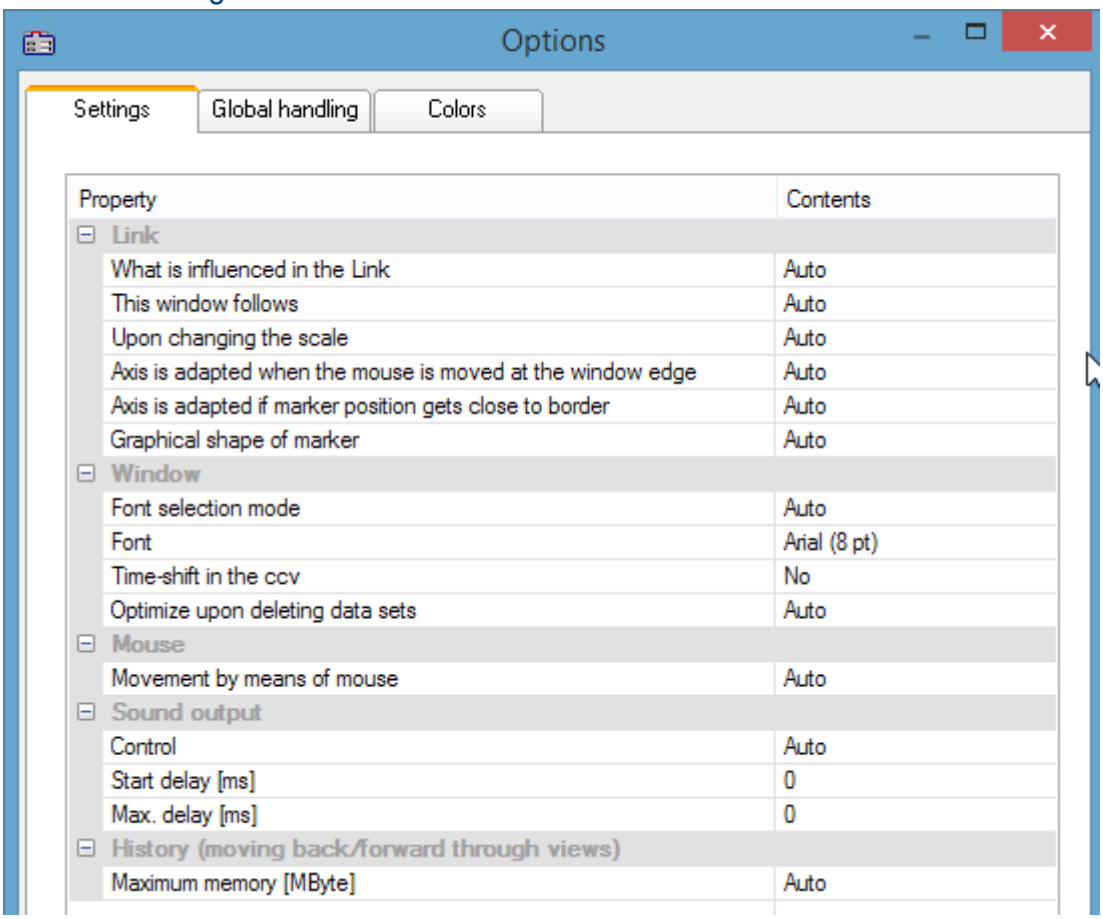
- None of the colors produced by mixing and blending may be selected for lines and texts. If other color intensities are selected, the program will use the closest true color. This also applies to the coordinate system background color.
- It is recommended to select always colors which produce a good 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.

8.6.5.3 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.

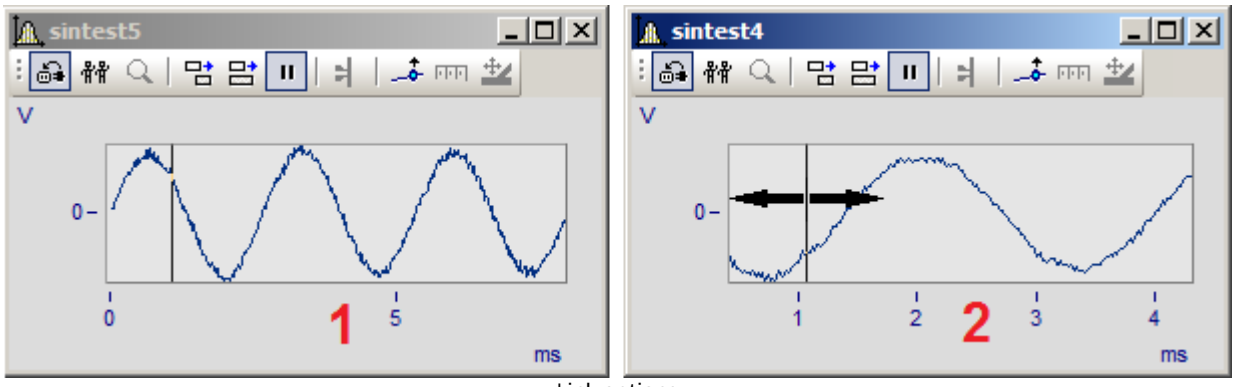
8.6.5.3.1 Settings



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* 985 *x-, yAxis (color map)*: both the x-, and the y-direction are linked 905. *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 maps 985 *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 maps 985 where a circle is used . *Line: Vertical line*



Link options

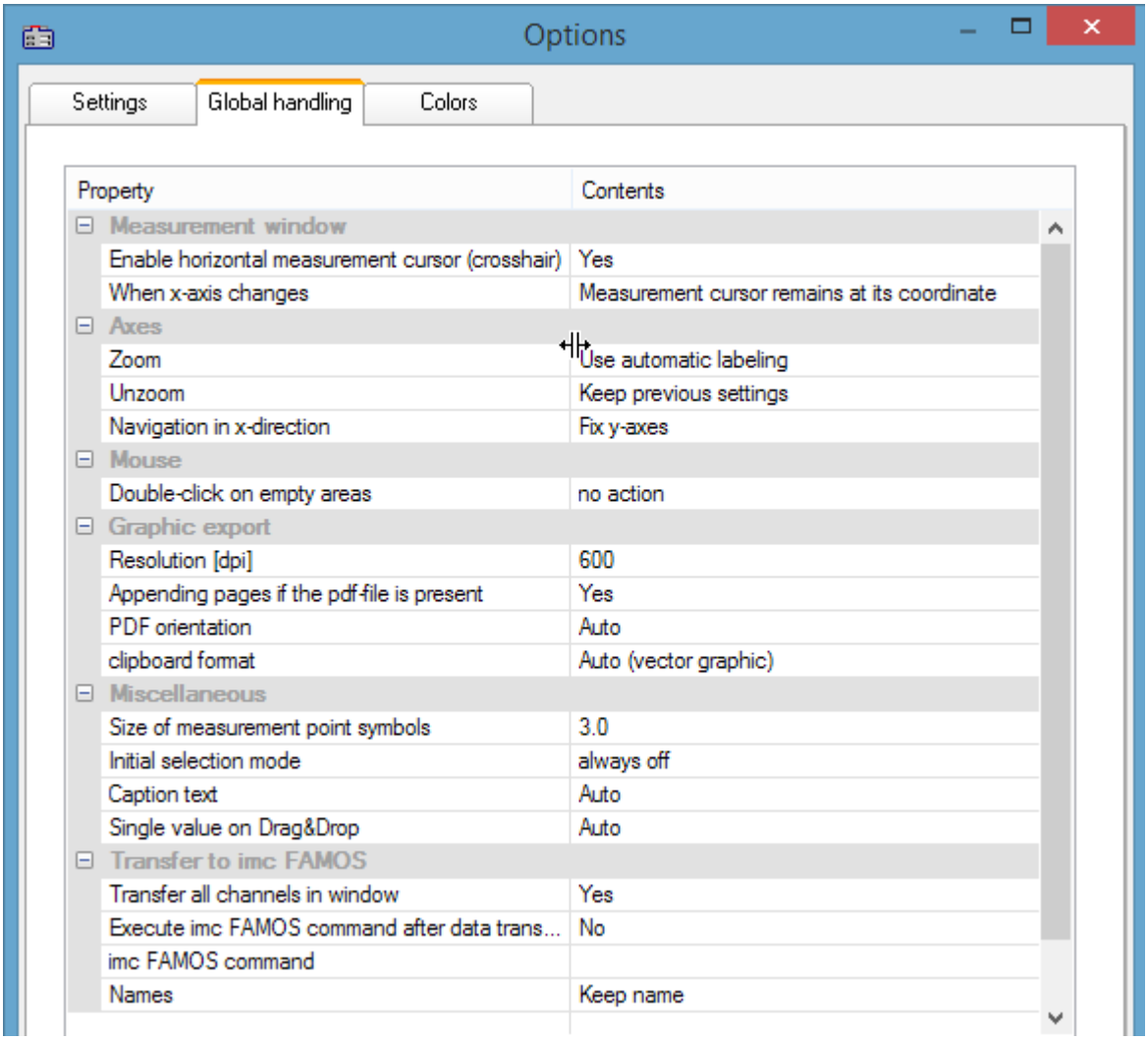
Window: In this section, set the font.	
Font selection mode	*Auto:* the font which is principally used whenever the curve window is opened. *Individual for this window*: Only affects the current curve window.
Font	All fonts used by Windows are available here.
Time-Shift in the ccv	includes any shift made using the Time-Shift function when saving.
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 1068.

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 904 command.

8.6.5.3.2 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.*

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* 1006
Navigation in x-direction	Determines whether after navigating 1073 the Y-axis is permanent or is scaled automatically.

Mouse	
Double-click on empty areas	Enables *Select-mode* 881 to be activated by double-clicking on an empty space in the curve window.

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

Miscellaneous	
Size of measurement point symbol	*0.5 -10* mm. Size of the symbols 1019 representing the measured points, e.g. squares, circles...
Initial selection mode	Default setting for *Select-mode* 881.
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 1057.	
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 1050

8.6.5.3.3 Transfer Options

Waveforms displayed in the curve windows can be transferred directly to imc FAMOS. It is often useful to transfer data from imc DEVICES, 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 870 enables signal portions located between measurement cursors to be copied. Note that there is a way (Name 1058) 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 1057.

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:

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* 870.

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.

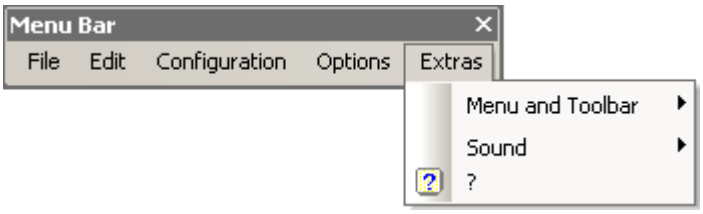
8.6.5.4 Printer preview

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* 1050-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.

8.6.6 "Extras" menu

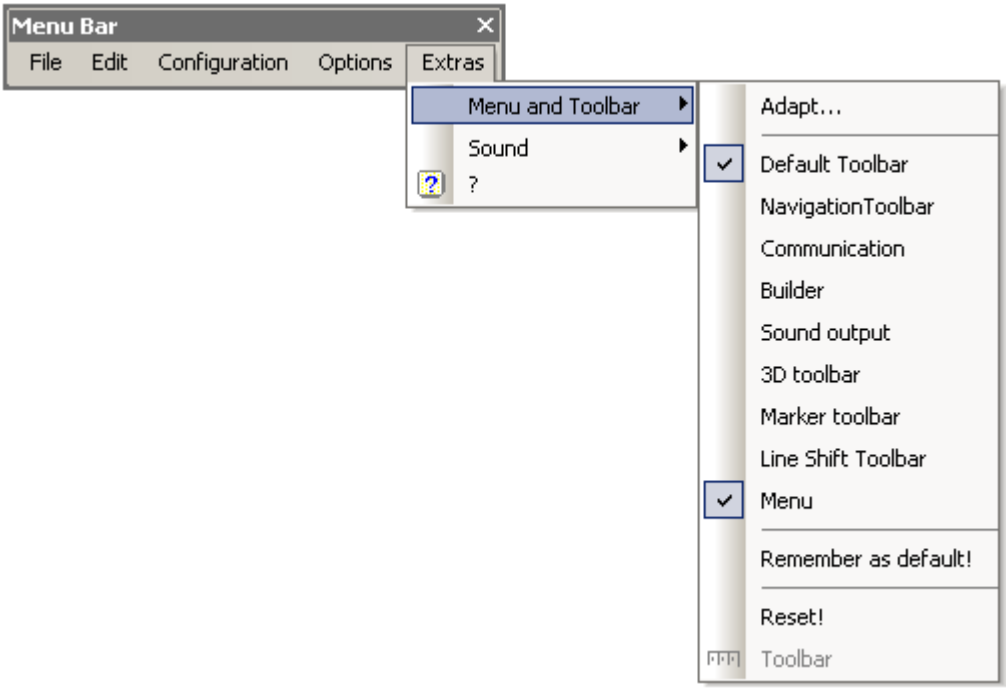


Menu and Toolbar 1060: Options pertaining to the menu and toolbar.

Sound 1064: Activates the sound output and access to the functions of the Sound-Toolbar.

?: Displays the curve window's Help.

8.6.6.1 Menu and Toolbar



Adapt... 1061

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!

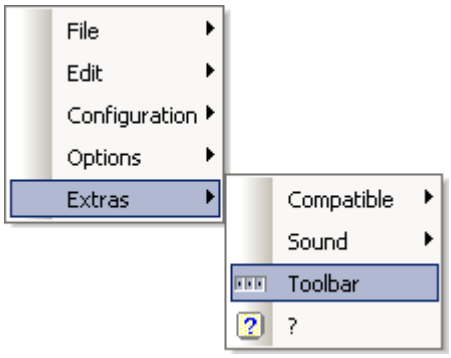
The arrangement of the menu and the toolbar is saved as the valid default arrangement. In this way, the same settings are used even for a different, new curve.

Reset!

The menu and toolbar are reset to their initial, factory-set state.

Toolbar

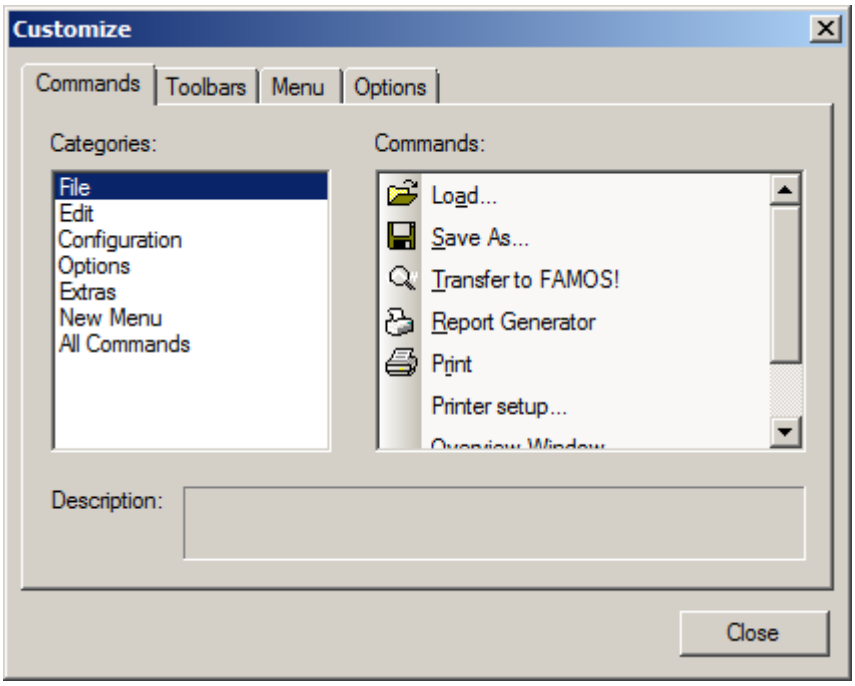
As a curve window integrated in the Data-Browser's Report view, the entire Toolbar only appears when a curve window is selected. To cancel this behavior so that the Toolbar does not appear, the function "Toolbar" can be selected. This is only available from within the Data-Browser. To make the toolbar visible again, select "Toolbar" under "Extras" in the curve window's context menu.



8.6.6.1.1 Adapt / Customize

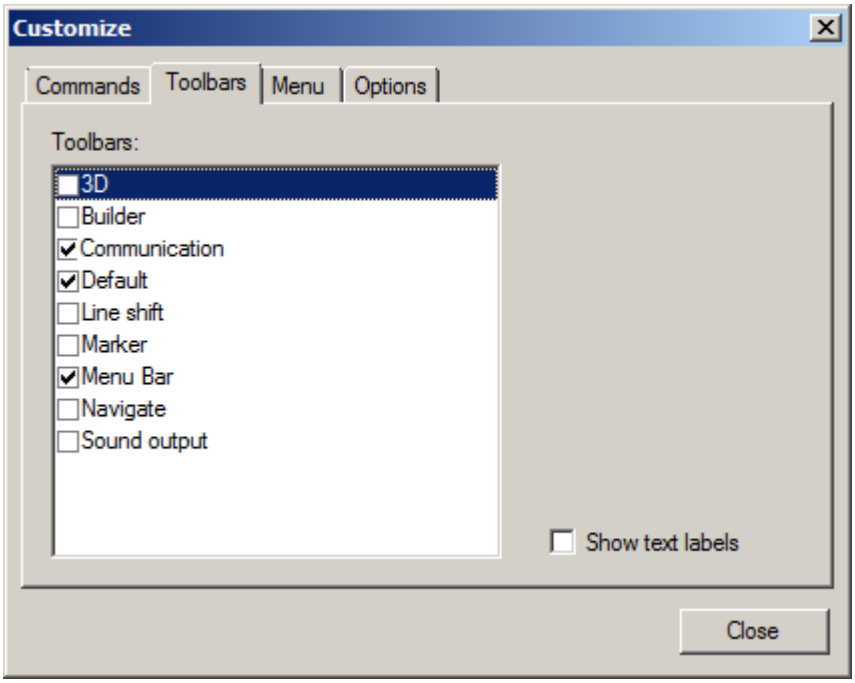
The curve window's menu and toolbar can be adapted. Right-clicking in either the menu or the toolbar opens the context menu. Alternatively, select the menu item *Extras / Menu and Toolbar*.

The menu item *Adapt*... opens the following dialog:

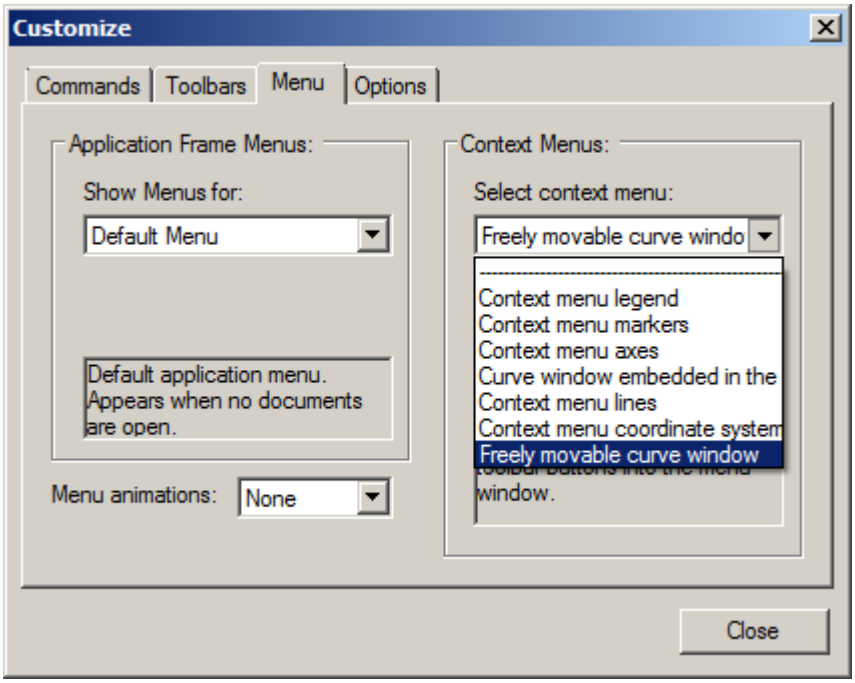


A command can be dragged from thee list at right and dropped on a menu or toolbar. Pop-down menus open by themselves when the mouse is moved over them. The list at left contains the different command topics.

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:



Context menus differ by whether they belong to a normal curve window which is free-floating, movable and can be maximized/ minimized, or whether they belong to a curve window without a title bar which is embedded in a dialog (child window). Select which type you wish to edit.

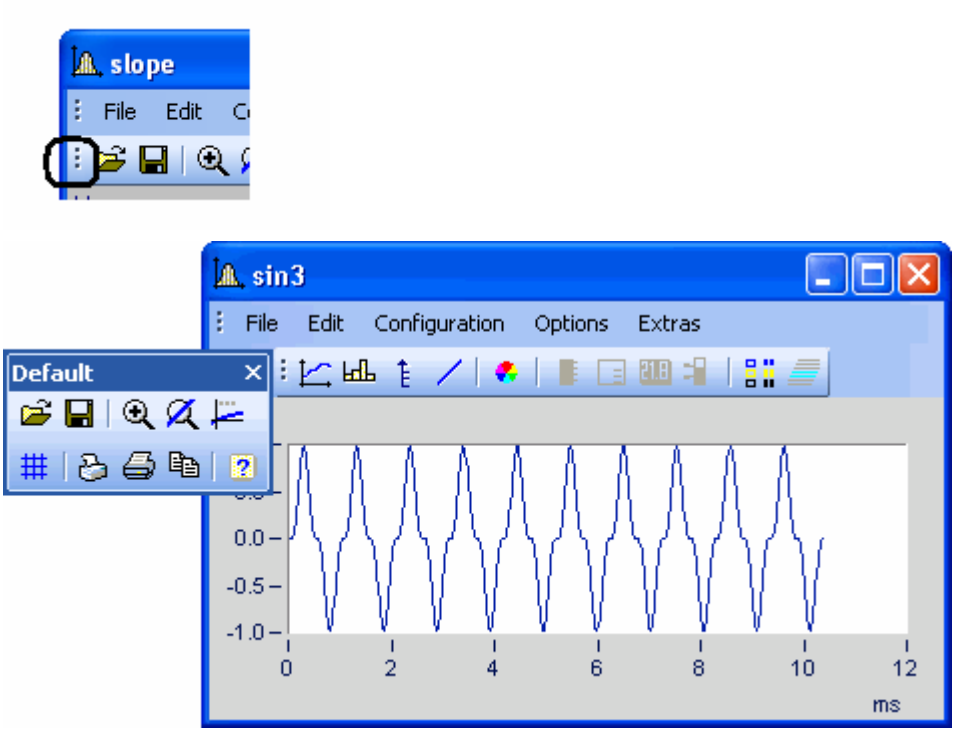
While editing menus, the curve window can't be operated.

While the dialog *Customize* is open, all menu items as well as toolbar elements can be moved around by the Drag and Drop technique. Note that moving an element to the "outside" means deleting the element.

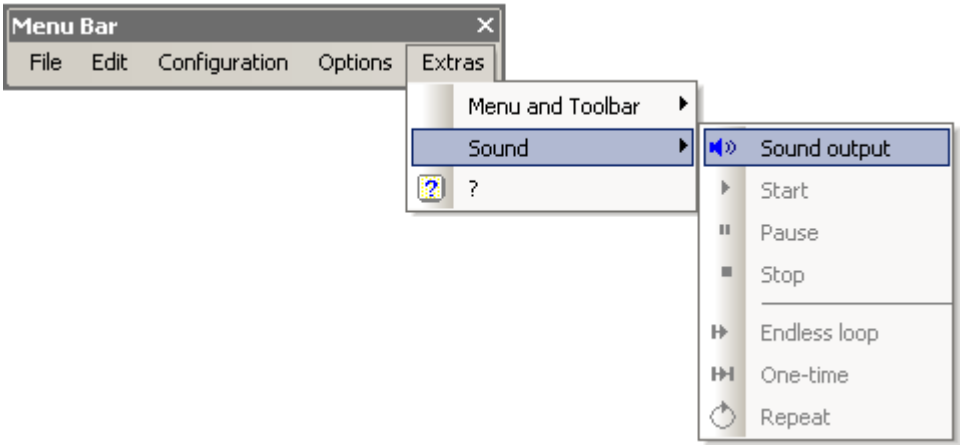
Changes of the menu are permanently saved when the dialog Customize is closed. The particular toolbar arrangement saved depends on the display type:

- 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:



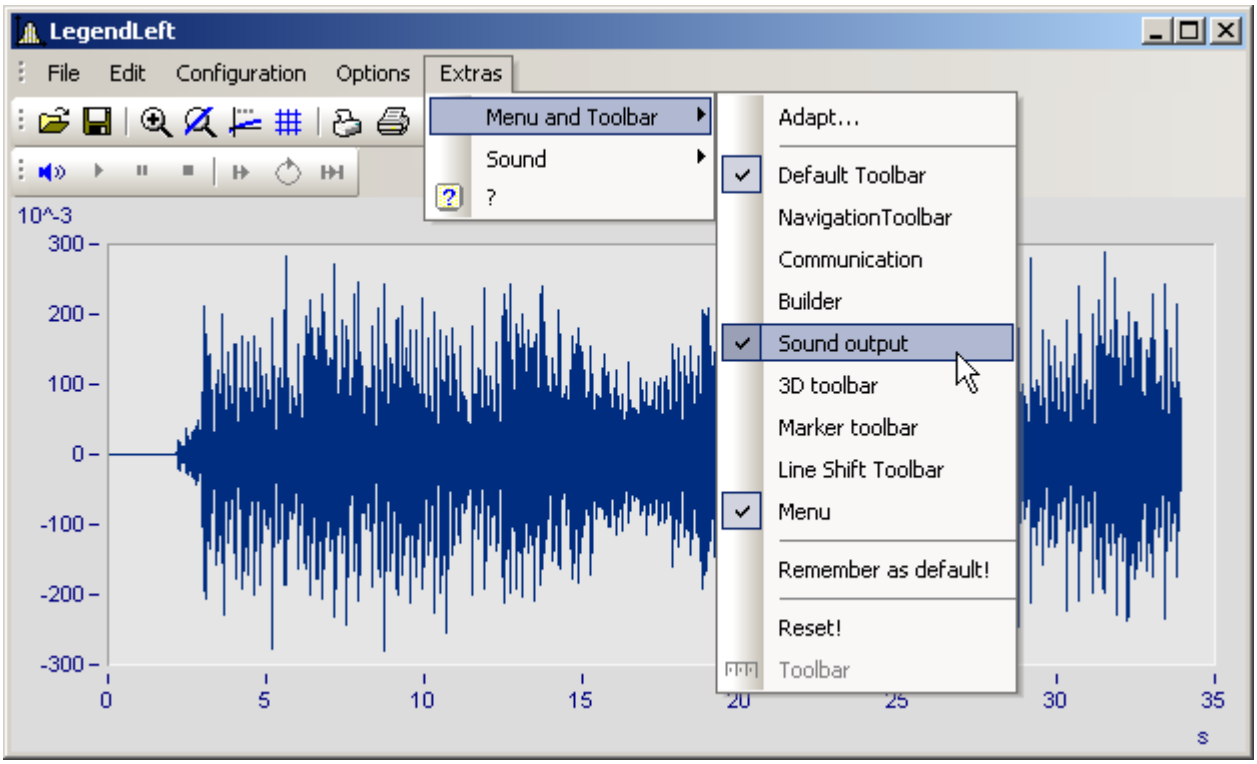
8.6.6.2 Sound



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.

8.6.6.2.1 Sound Toolbar

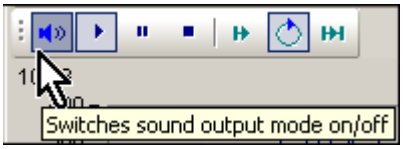
For working with the sound output, you should select the associated toolbar from the menu *Extras / Menu and Toolbar / Sound output*.



Sound output toolbar

Under Sound Output, the following functions are available:

On/Off

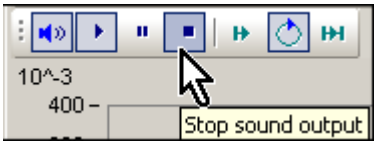


This function lets you switch the sound output on and off. A vertical bar marks the current playback position within the curve window (playback marking), and a horizontal bar denotes the selected playback range.

Start

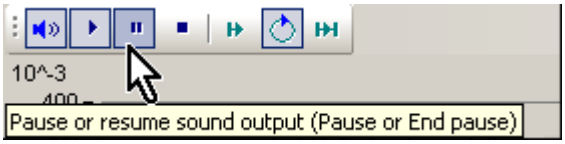
With this function you start output of the sound, and the selected range of the data set is played back in real time. The playback's starting and ending points can be freely selected and the position of the resulting playback time window can be moved as desired (see Cut Sound 1067).

Stop



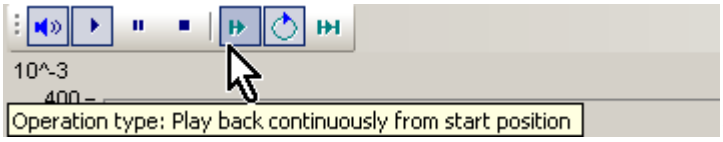
This function stops playback of the data set. The playback marking is set back to the start of the selected playback region.

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

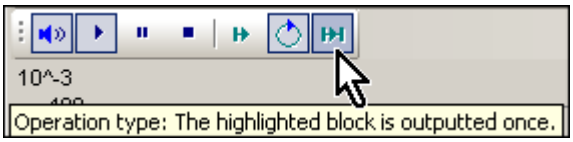


When you choose this operation mode, the final position for the playback is automatically set to the end of the waveform; see Cut sound 1067)

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.

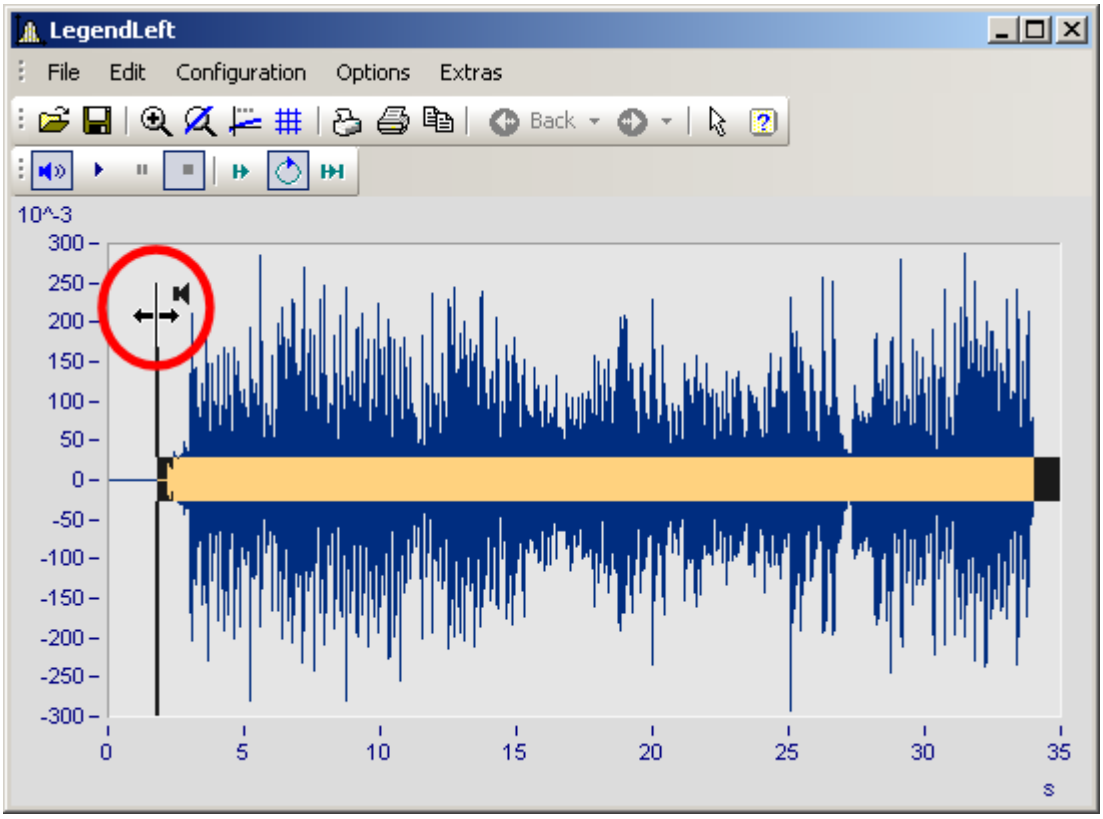
When the volumes of various signals are to be compared, all y-axes must be set to the same scaling. If the y-axes are scaled automatically, all signals have the sound card's maximum level.

The Sound Kit functions have not settings options for volume.

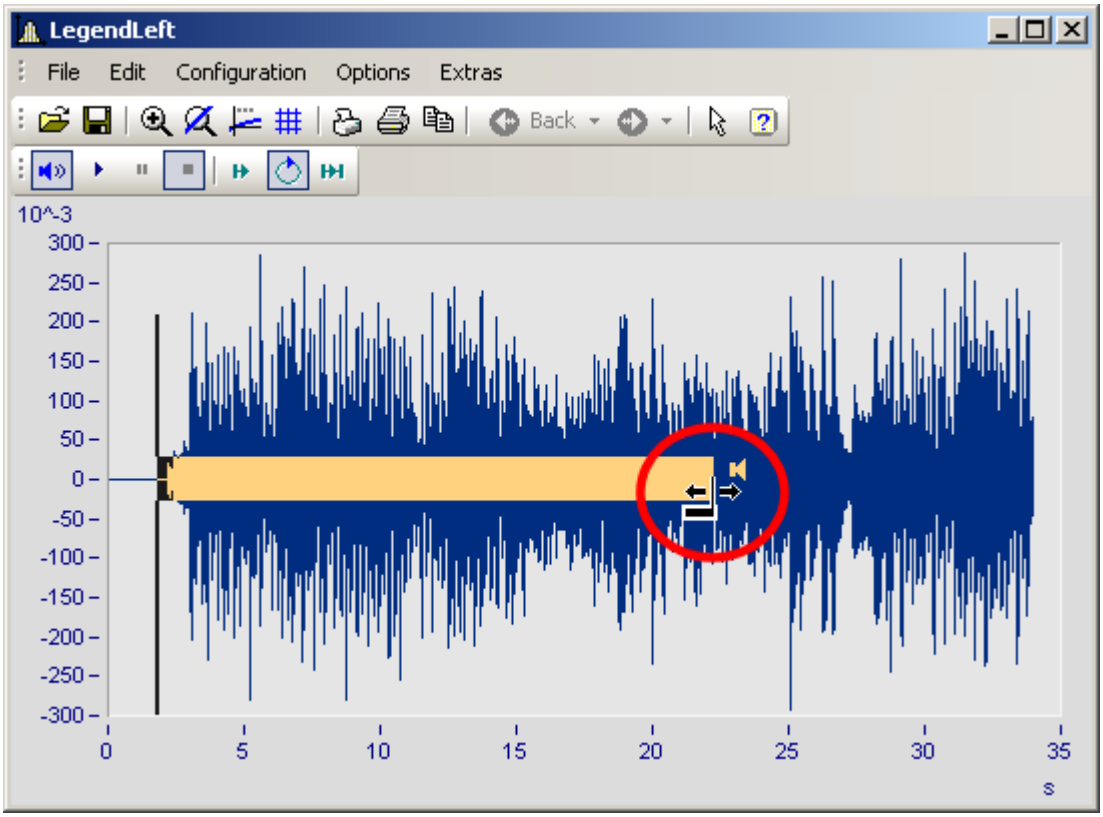
8.6.6.2.2 Sound clip

In sound output, it is possible to freely select (take a clip from) the playback region.

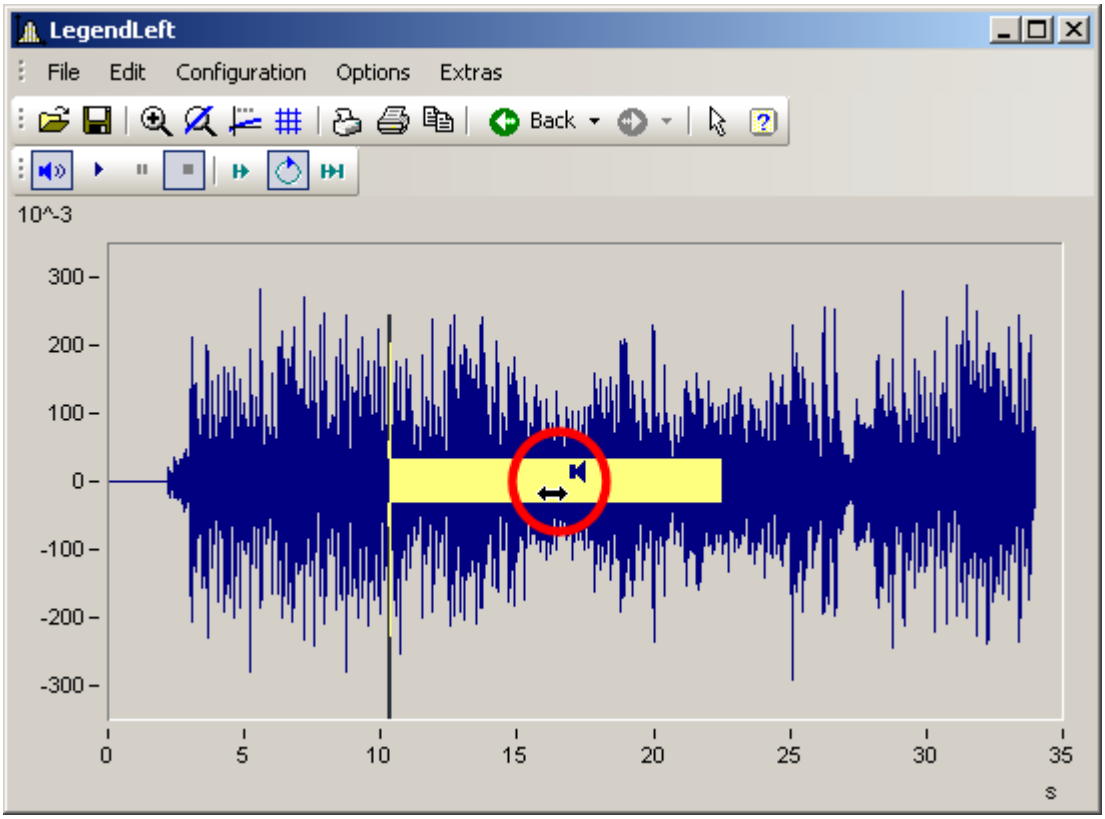
Grab the start position at the beginning of the Playback strip and move it to the position desired.



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.



8.6.6.2.3 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 1055, 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.

8.6.7 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

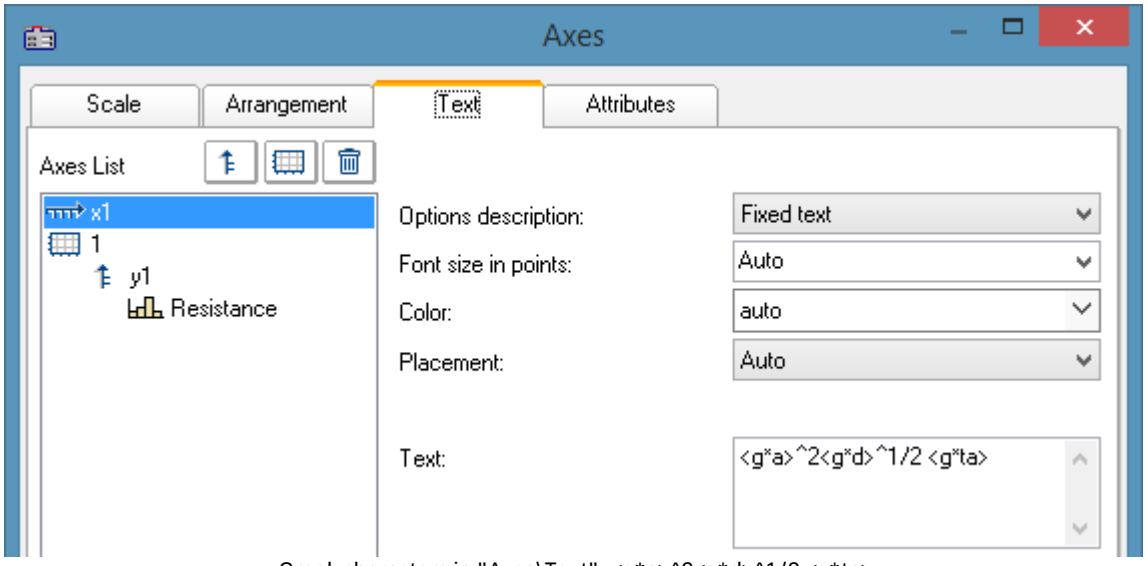
<g*a> ^ <g*a> is displayed as α^{β}.

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

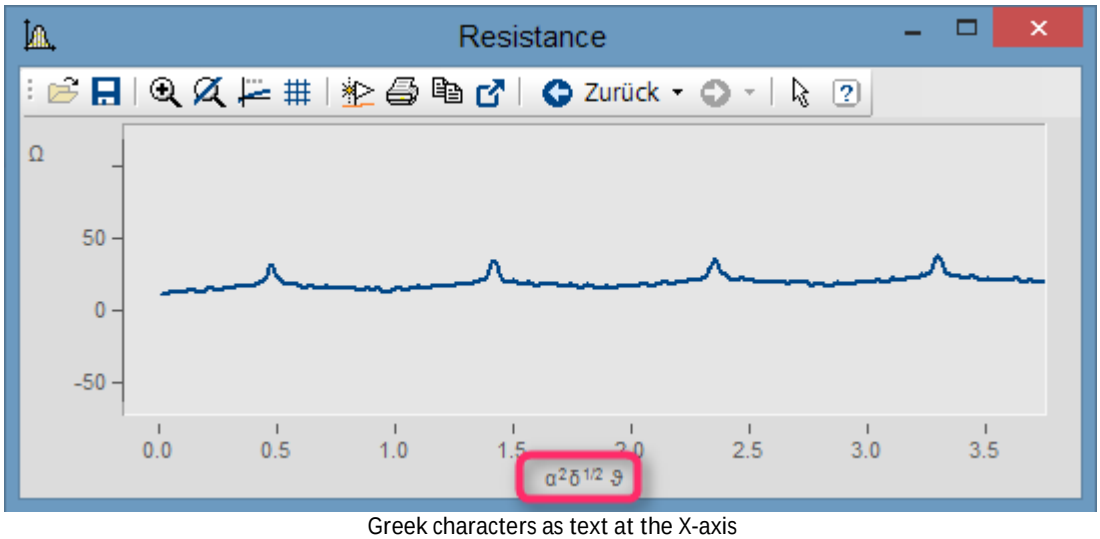
Example: curve window

with



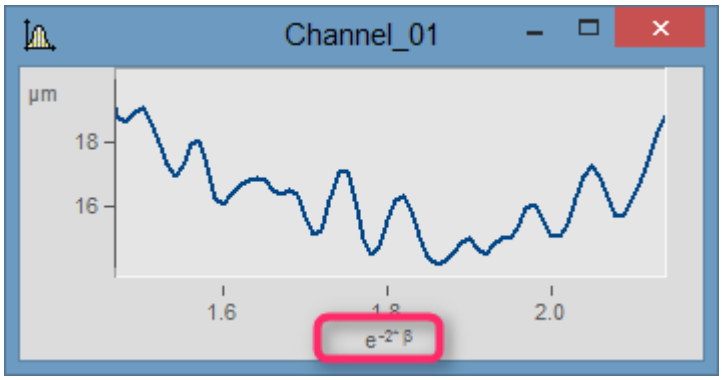
Greek characters in "Axes\Text": <g*a>^2<g*d>^1/2 <g*ta>

the result is:



Greek characters as text at the X-axis

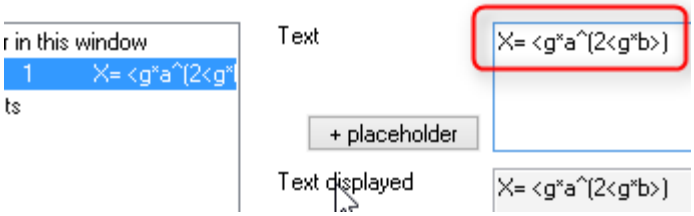
Exponent with multiple characters. From e^(-2*<g*b>) we obtain:



Exponent of placeholder in parentheses: e^(-2*<g*b>)

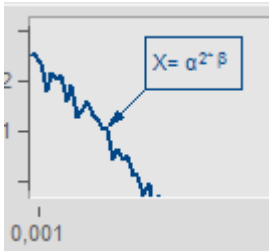
Example: markers

from



Greek characters in marker-definition:

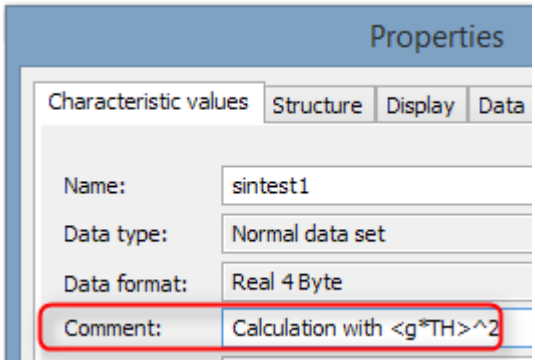
the result is:



Greek characters in markers

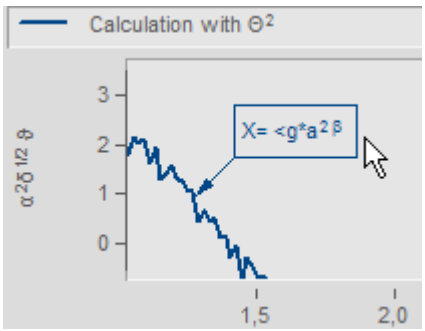
Example: comment on variables

from



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* 1009*"* under *"Scale"*.

8.6.8 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.

8.6.9 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 (child-windows/widgets in the Panel).

8.6.10 Drag and Drop, mouse wheel

Moving waveforms

Drag and 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 and 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 DEVICES or imc STUDIO), they are copied to imc FAMOS.
- With maximized curve window display 920 (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 and 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

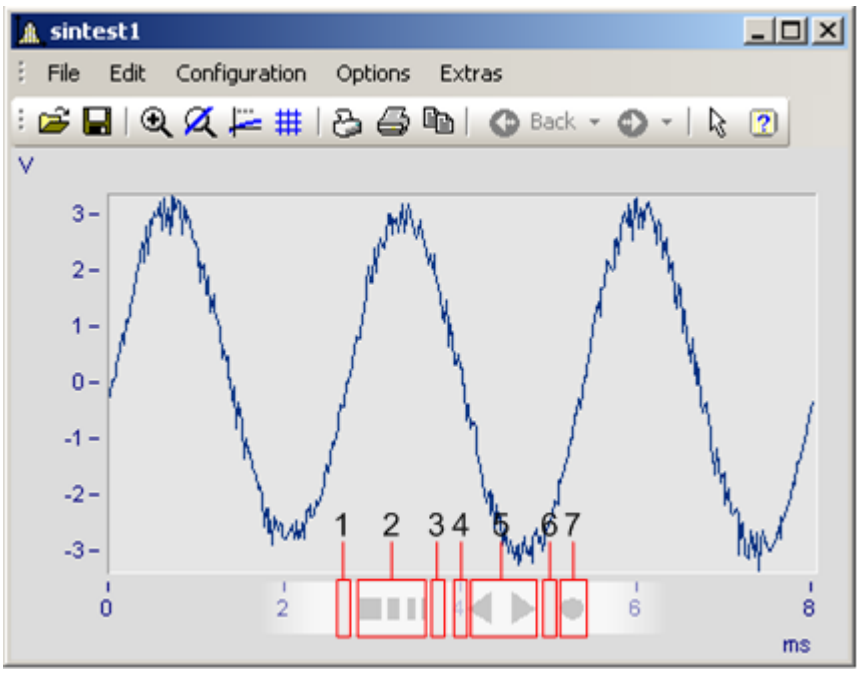
- If the mouse is released over a y-axis, the data are displayed on this axis. Release the mouse over a coordinate system, and the data are displayed with a new y-axis for this coordinate system. Otherwise, a new coordinate system is opened.

- 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.

8.6.11 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 from 2 to 1:

Magnification

Dragging from 2 to 3:

Shrinking

Dragging from 5 to 4:

Move left

Dragging from 5 to 6:

Move right

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.

8.6.12 Last Changes

8.6.12.1 LastC_x_y

Geben Sie hier den Text ein.

8.7 Special Widgets

Here, special Widgets are described.

8.7.1 Text input for report channel

This Widget enables you to write pre-defined texts to a report channel. During the running measurement, it is also possible to add freely defined texts to the channel along with predefined texts.

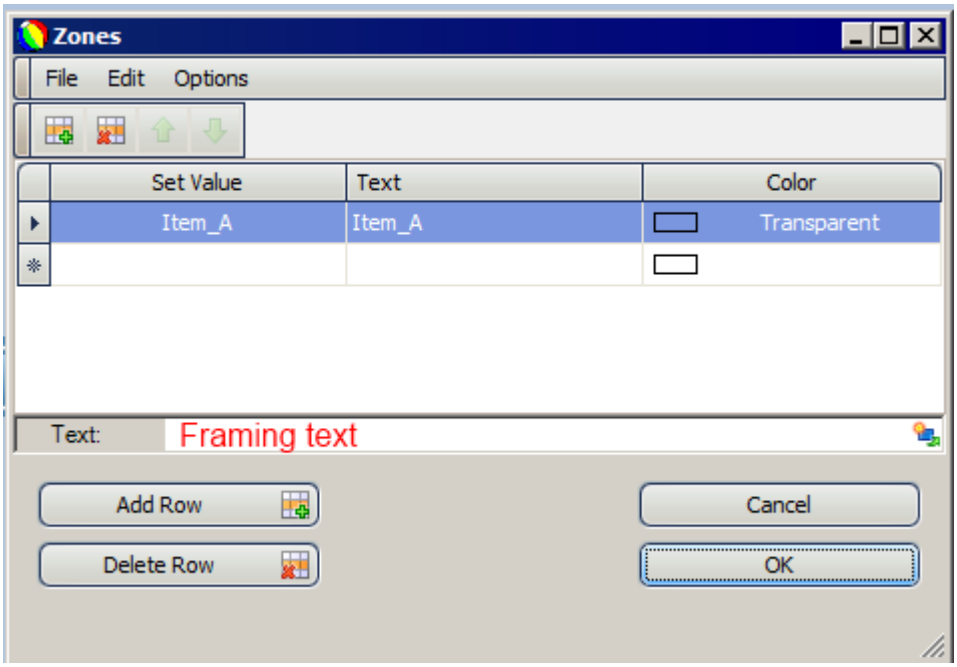
In order to be able to use this Widget, link it with a report channel. In the chapter User-defined variables, there is a description of how to create a report channel. Note that a report channel can contain either text or audio comments, but not both. How a channel is linked with the Widget is described in the section Variable linkage 1089.

Defining the texts

In order to define the text, open the Zones-dialog in the Properties window. This dialog contains the columns Text and Set Value.

- Text: This text is displayed in Widgets selection list
- Set Value: This text is written in the report channel if the Widget is activated

Below the table, a framing text can be entered which is written to the channel each time it is activated, no matter which text is currently set. As soon as something is entered here, the text entered under Set Value is no longer written to the channel. However, it is possible to incorporate the value from the table into this text by means of the placeholder <CONTROL.VALUE>.



Textinput for report channel, zone dialog

Further information on this dialog is presented in the chapter Zone dialog 847.

Example

Two entries are created in the table:

- Text: *It is too hot*, Set Value: *Too hot*
- Text: *It is too cold*, Set Value: *Too cold*

Frame text (text below the table): *Error message: <CONTROL.VALUE>*

If one now selects the first entry in the Widget, then the *Error message: Too hot* is written in the channel.

Writing text in the channel

To write the predefined texts in the channel, there are a variety of possibilities:

- Select the desired entry from the list. For this purpose, just selecting is adequate, additionally clicking on the Widget is not necessary.
- If the entry is already selected, click on the Widget in order to add the entry to the channel; the Widget works like a button.

In order to add a freely defined text to the report channel during the measurement, left-click the mouse once over the value being displayed currently. Then you will be able to edit it. To write the value to the channel, conclude the entry by clicking on Enter. If a frame text is set in the Zones dialog, it is ignored.

Note: Where are entries inserted

- The entries are added during the measurement at the current moment.
- It is possible to add the entries at the position of the Navigation bar's slider; for information on this, see the section Post-processing mode 1101.

Context

Report channels offer the possibility of saving a context along with the text entries. This context is typically the channel (or channels) where the event occurred, about which the comment is to be made.

The context is stated in curved brackets {} in the frame text.

In order that the context (channel) can be added during running measurement, use the placeholder <SELCONTROL.VARS> here. If you add a text during measurement (release of the Widget), you subsequently have time to click the mouse on the Widget, with which the channel to receive the comment is linked. During this time, the Widget flashes. You can set the time in the Widget's properties (display mode: *All Properties*).

Zones dialog:

- Two entries are created, *text1* and *text2*
- Frame test is: *Test: <CONTROL.VALUE>{<SELCONTROL.VARS>}*

Channels:

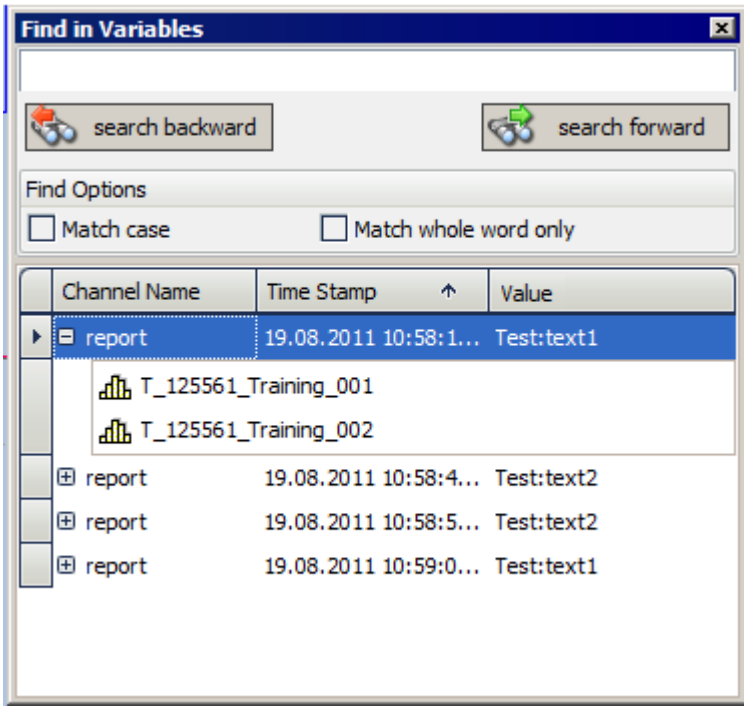
- Two active measurement channels, *T_125561_001* and *T_125561_002*
- Report channel for Text entries: *report*

Panel:

- Widget *Text input for report channels* linked with *report*
- A curve window linked with both measurement channels, curve window linked with Navigation bar
- Report channel *report* linked with the Navigation bar

During measurement, each time after the Widget was released, the curve window with the two channels was clicked on. Thus as the context of the text etries, the two measurement channels were captured, since <SELCONTROL.VARS> always includes all channels/variables, which are linked with the Widget on which you click.

In the Navigation bar's Search box 1099, the context is displayed below the channel:



Display of the context in the Search box

Since the search includes all columns (Channel Name, Time Stamp, Value), it is also possible to search for a specific context. You also have the option of displaying an additional column *Context*.

8.7.2 Audio

This Widget enables you to save audio commentary in a report channel. In order to be able to use this Widget, link it with a report channel. In the chapter User-defined Variables, there is a description of how to create a report channel. Note that a report channel can contain either text or audio commentary, but definitely not both. How to link a channel with the Widget is explained in the section Variable linkage 1089.

Operation

Audio-Widget

- 1: Stop recording/playback
- 2: Start playback
- 3: Start recording

To write audio commentary to the report channel, left-click the mouse over *Start recording.* To conclude the comment, click on *Stop recording/playback*.

Note: Where are entries inserted

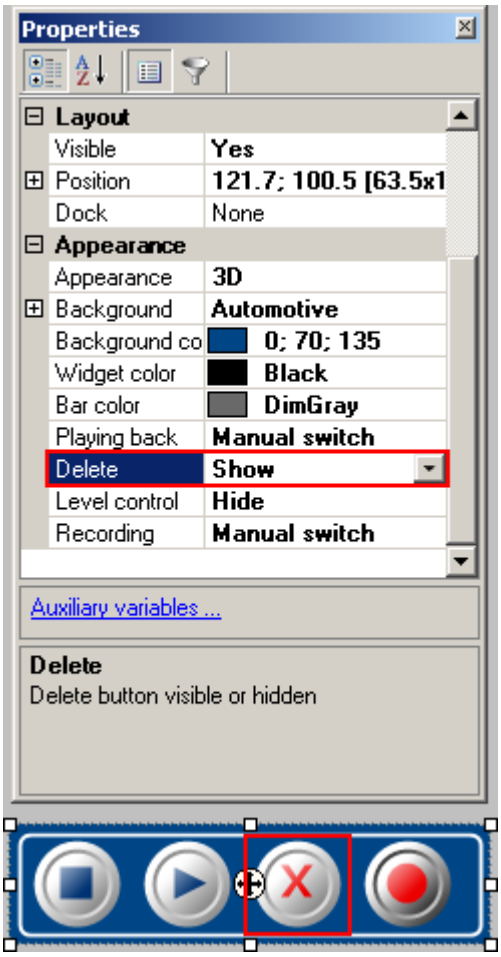
- The entries are added during the measurement at the current moment.
- It is possible to add the entries at the position of the Navigation bar's slider; for information on this, see the section Post-processing mode 1101.

To hear the last recording, click on *Start playback.*

Setting buttons

In the Widget's Properties window, you can set the buttons (Stop recording/playback, Start playback, Start recording).

You can also add buttons, e.g. *Delete*. These are needed if you wish to delete a previously added comment (see the section *Report channel* in the chapter User-defined variables).



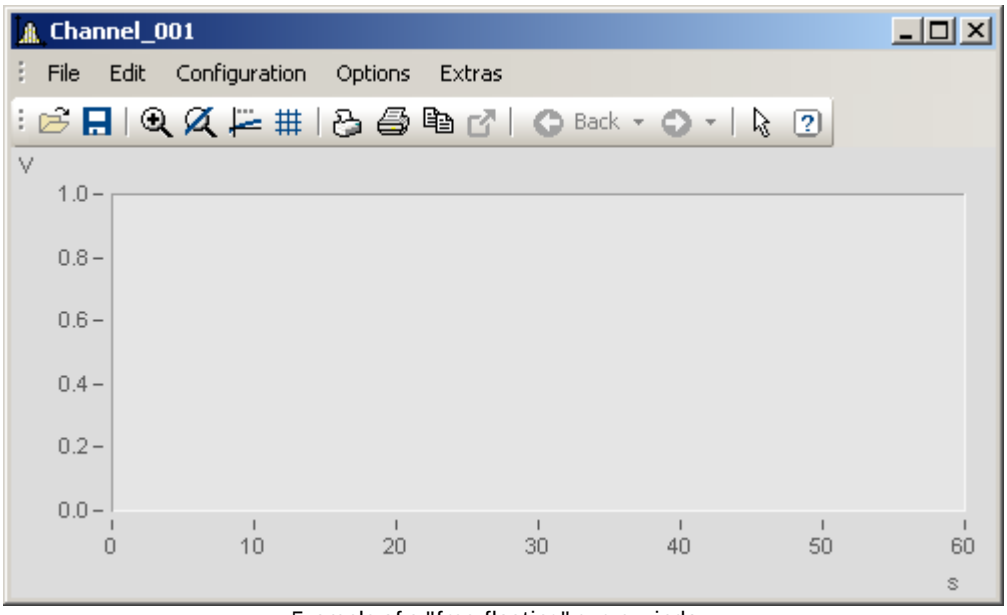
Widget Audio: Properties window

8.7.3 Curve Window - "Free-floating"

The curve window comes in two varieties:

- "free-floating" 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 of a "free-floating" curve window

The embedded curve windows are only visible if the associated Panel page is opened.

The configuration of the windows is described here: Curve Window 854.

Opening a "free-floating" curve window

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* 824-button (/) in the symbol bar of the tool window Data Browser.

Reference: See also

Setup - Device Configuration > "Import of an imc DEVICES Experiment 257": How do I get the free-floating curve windows from imc DEVICES into the imc STUDIO Panel pages?

8.8 Pages

8.8.1 Insert - Dialog / Report

To add a page, there are multiple methods:

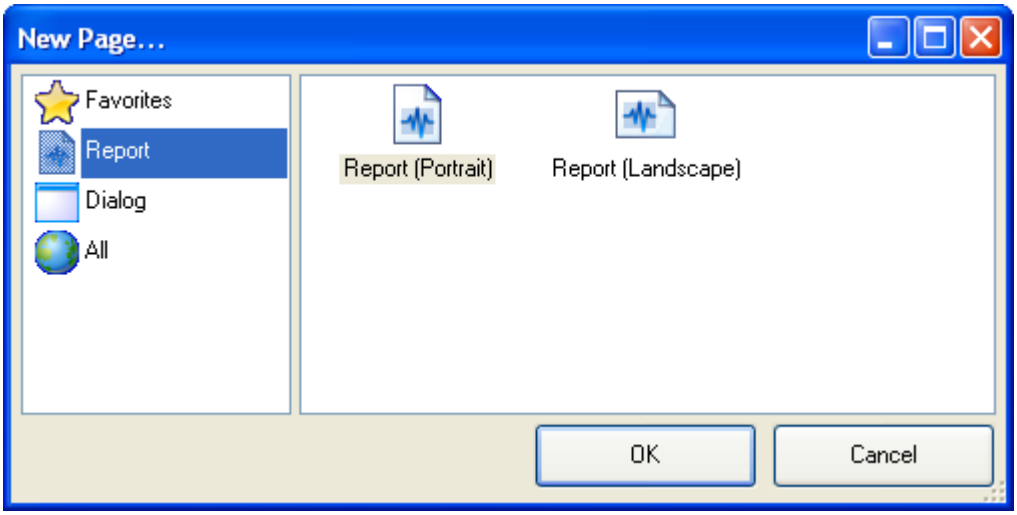
- Context menu 829 of the page-title
- Symbol (or) from the ribbon
- Drag&Drop from the Page templates 828 tool window to the page tabs
- Double-click on a template from the Page templates 828 tool window

Note: Replacing a page

If you use Drag&Drop to take a template from the Page templates 828 tool window to an empty page, the page will be replaced.

A page can be created as a Dialog page or as a Report page:

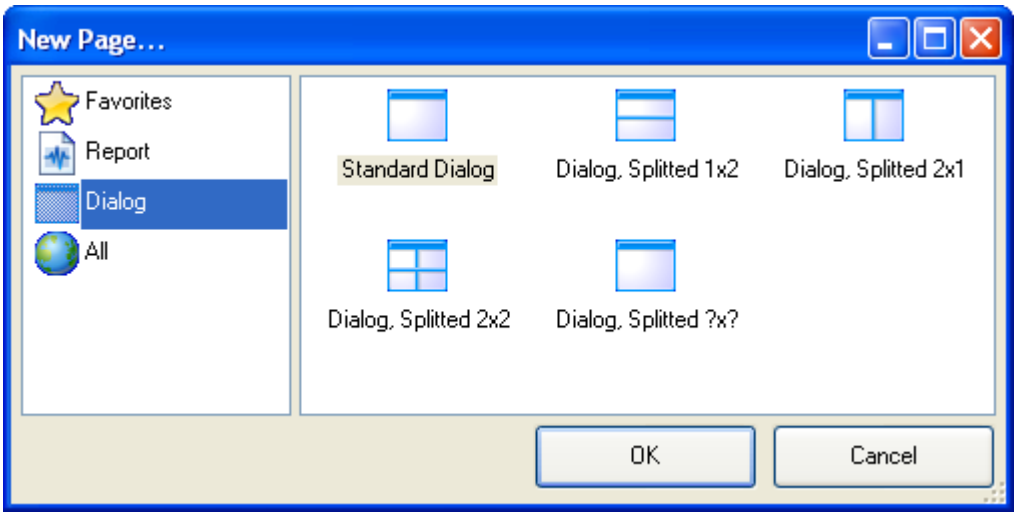
Report page



New page, Report page

The Report page is optimized for printouts. On the screen, it is distinguished among other ways by "side margins". The pages are generated by default in a fixed size (see Properties (Page) 1081).

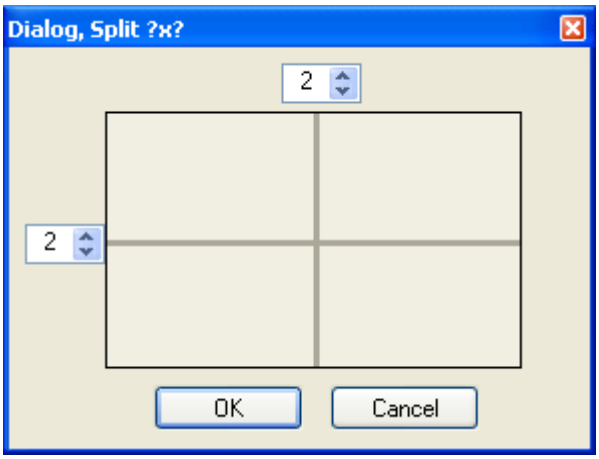
Dialog page



New page, Dialog page

The Dialog page is optimized for screen display. By default, the pages are generated in a constant aspect ratio (see Properties (Page) 1081).

With the template *Dialog, ?x? (Free Format)*, you can set your own layout:



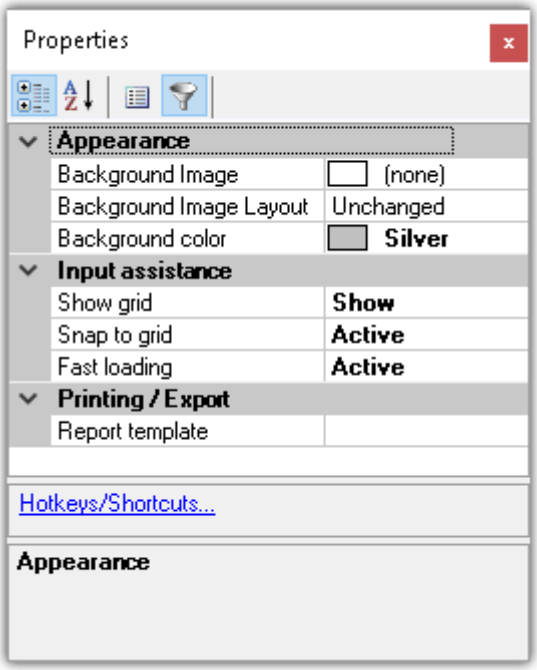
Split format

After the program starts, the Panel is assigned a Standard Dialog page.

8.8.2 Properties - Pages

With this tool window, you can set the page's properties. The Panel must be in Design Mode 832 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.

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>)
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]`".
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	For purposes of export or printing, it is possible to define another 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 page the Widget: "Print Preview" appears. If a page is printed on which the "template page" is set as the "Report template", the Widget is filled with the page to be printed.

Input assistance

Property	Description
Grid size	The distance between the grid points.
Show grid	Hides/shows grid (see "Aligning to grid 1085")
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.

8.8.3 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.

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 1085.

Note: all pages

The functions can be applied either to the current page or to all loaded pages simultaneously.

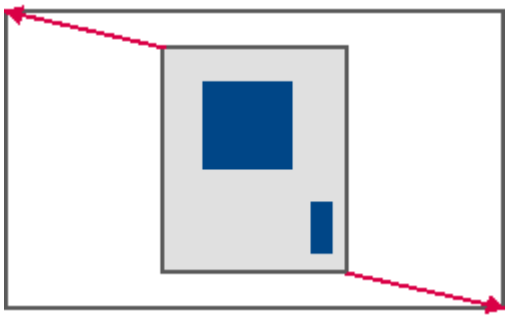
The following pages are not affected by the functions applied to all pages:

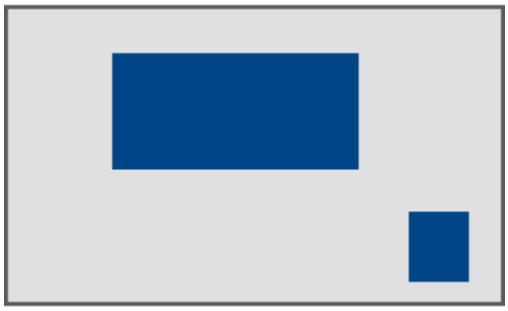
- all pages in which the property "fast loading" was deactivated, unless they are already open currently.

Adapt: 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)



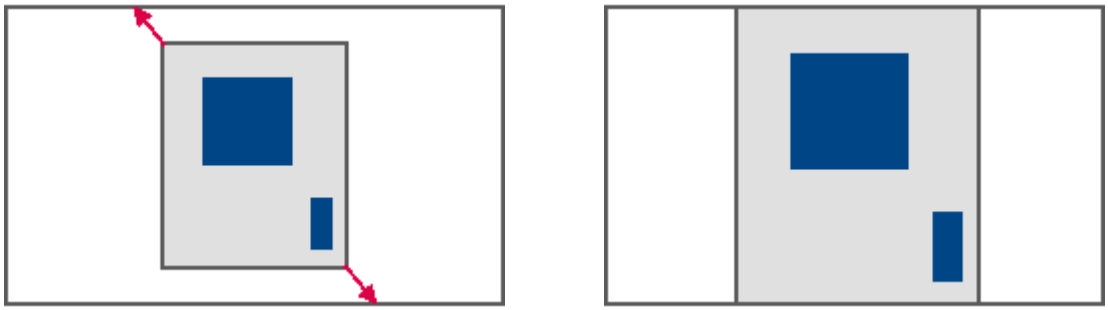


Result

Adapt: 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).

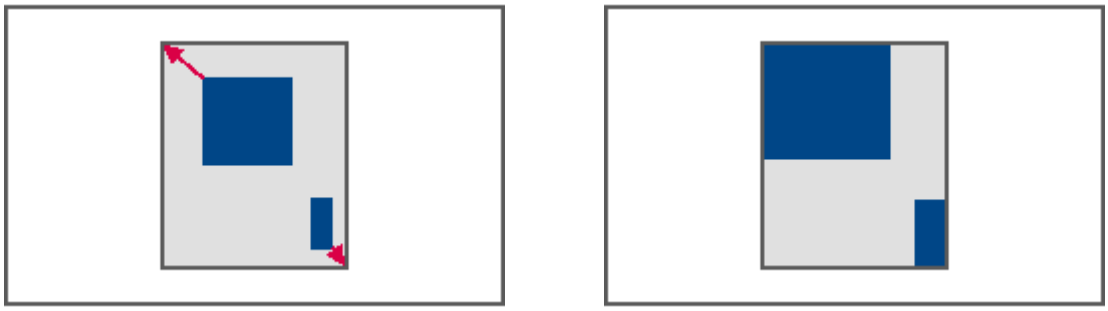


Result

Adapt: 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

8.8.4 Lock and unlock page

Also in Design Mode, individual pages can be protected against changes.

Open the context menu on the page title (command: *Lock/Unlock current page*) or click on the corresponding icon in the ribbon ( ) in order to change the status:

The associated symbol appears next to the page-name.

In contrast to the deactivated Design Mode 832, not even Widgets can be placed on a locked page by means of Drag&Drop.

8.8.5 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* ( ):

	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)

8.8.6 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)* ( ):

	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.

8.8.7 Fullscreen

Panel pages can be displayed 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

The possibilities of fullscreen mode

1. 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 located in the background (so it covers the user interface of a monitor). Thus, you could observe the Sequencer on the main monitor, while the second monitor shows measurement readings on a Panel page in fullscreen mode.
   - To do this, open the context menu of the Panel page tab.
   - Select *Show Page on Monitor* ( ):

	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.

Automatic assignment by command: Browse in workspace

When the command "Browse in workspace 1599" 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.

2. Panel in fullscreen mode, so that the imc STUDIO user interface is no longer accessible (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) .

Menu ribbon	View
Start > Panel Fullscreen ( )	Standard, Compact
Panel-Control > Panel Fullscreen ( )	Complete
Panel-Design > Panel Fullscreen ( )	Complete

Notes: Protection against changes, and use of guided pages

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: 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 1599".

Automatic launch in Fullscreen mode

Using the command-line parameters 43, 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 1599" 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 818 to be shown.

Additional functions can be accessed via the context menu or via the button in the expanded menu.

	Menu item	Description
	Login 77	Logs in a user
	Logout 77	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

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 automatisch anpassen 1083.

8.9 Variable Linkage

To link a Widget with a variable, there are two methods:

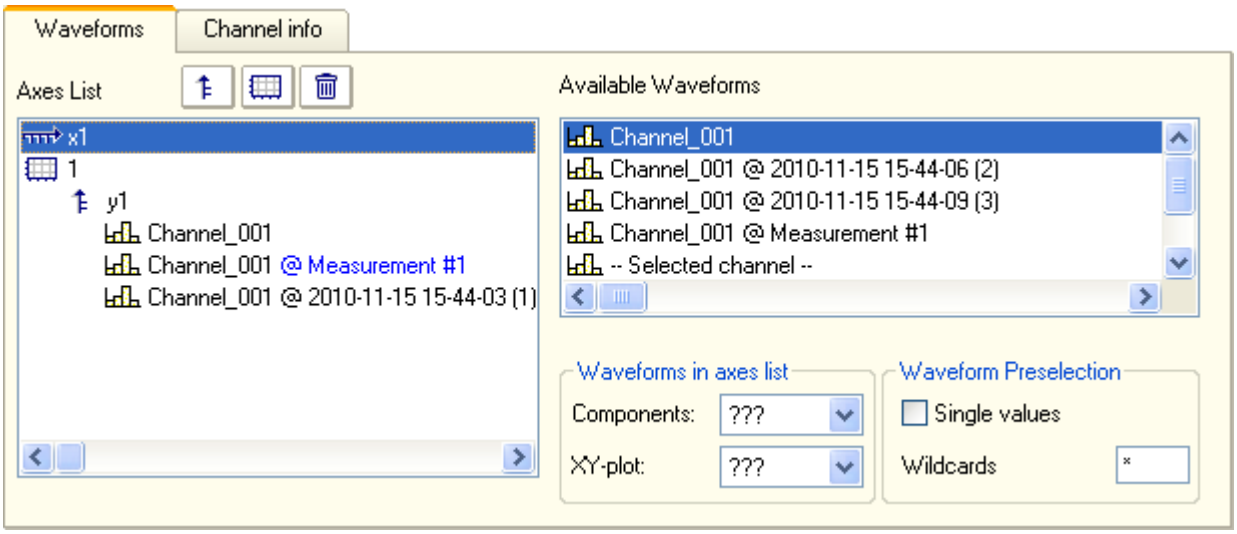
- move the variable using Drag&Drop from the tool window Data Browser 818 to the Widget or the Panel page (see: Variable linkage by Drag&Drop 1090)
- open the Properties 843 window of the Widget and there click on the *Variable* (see: Variable linkage by Widget Properties 1091)

Even without existing variables, it is possible to set up a linkage (see: Variable linkage to not yet existing variables 1092).

Fixed name, symbolic name / measurement number (Measurement#<x>)

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 or Channel_001@Current measurement In this way, the Widget always indicates the current measurement. or Channel_001@2010-11-15 15-44-03 (1) 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.



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 821).

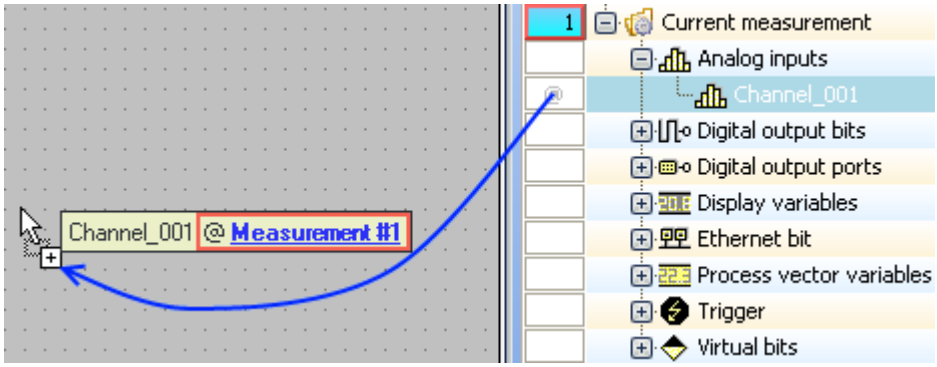
Note on the clocks:

The clock can be linked to any desired channels (no single values). Then the time of the last samples is shown.

Variable linkage by Drag&Drop

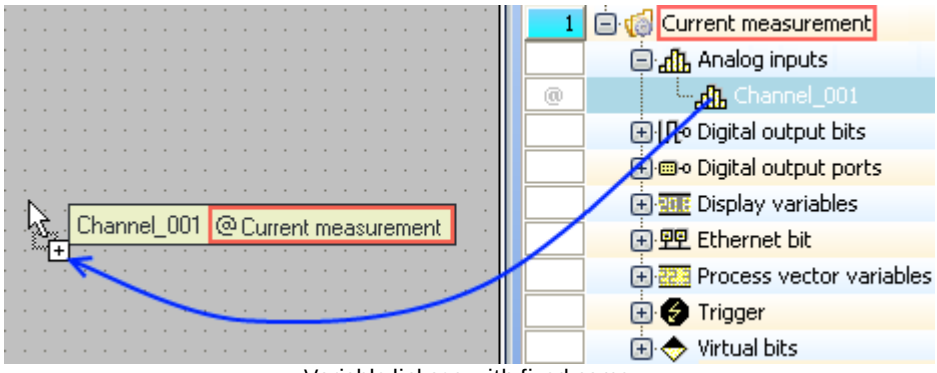
If you have opened a measurement in the Data Browser, the "@" Symbol (@) appears in the Numbering column.

- To link a variable with a Widget by its symbolic name, start Drag&Drop from this "@" symbol.



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

- To link a variable with a Widget by its fixed name, start Drag&Drop from the entry in the Names column.



Variable linkage with fixed name.
Example: "Channel_001@Current measurement"

- 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 854.

You can establish the link with the variable by means of the Widgets's properties.

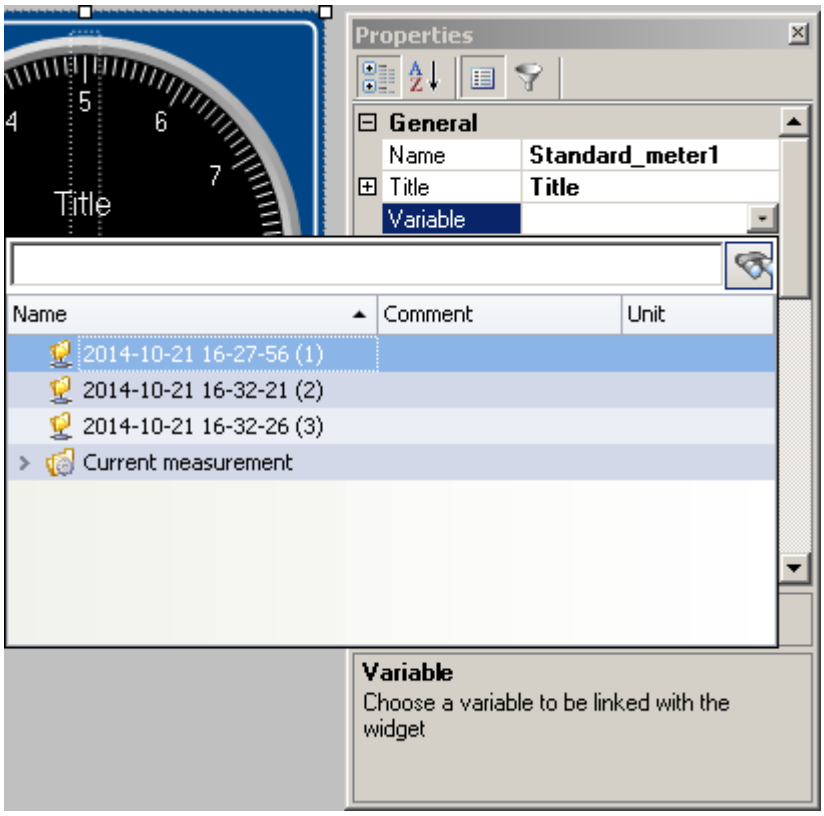
- Open the properties of the Widget
- There, click on the entry *Variable*

There are then two ways to make the link:

- Variable linkage by dropdown list
- Entry of the variable's name

Variable linkage by dropdown list

- Click on the dropdown button ( )
- A Data Browser opens

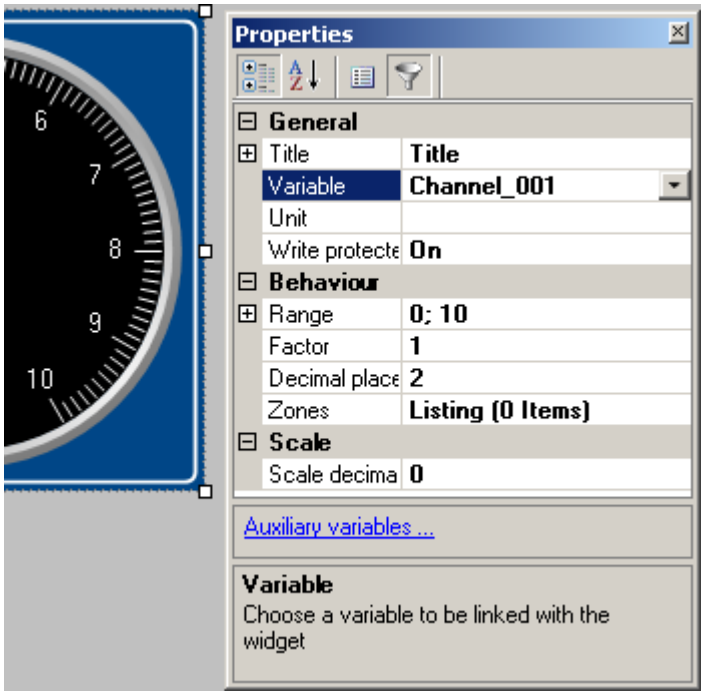


- Select the desired variable

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@2010-11-15 15-44-03 (1)
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 1091)

8.10 Navigation Bar

The *Navigation bar* facilitates navigation across time in one or more Widgets. The Navigation bar can be shown/hidden in Design Mode by means of the icon . In the context menu of the Panel page, the Navigation bar can also be shown/hidden by means of the menu item *Show Navigation bar.* If the Design Mode is deactivated while the Navigation bar is visible, the Navigation bar remains visible.

If you move the mouse pointer to over the Navigation bar or click on it, the Navigation view launches:

Navigation bar: Navigation view

Opens the Navigation bar menu 1094

Tool for linking Widgets with the Navigation bar 1095

The black lines at left and right indicate the boundaries of the visible area of all linked curve windows.

The white bar shows the visible range. If varying zooming levels for the linked curve windows are used, this bar is shaded grey and indicates the maximum visible range.

Changes the zoom level of the linked curve windows: Move the mouse over the icon, click on its left button and while holding it down, move the mouse pointer left in order to zoom out, or right to zoom in.

Slider 1096

Navigation left/right: Click on the arrows multiple times in order to move forwards in individual steps, or hold the mouse button down and drag the mouse in the direction desired.

Rezoom of X-axis: Displays all data in the linked curve windows

Pins the Navigation bar down or loosens it again

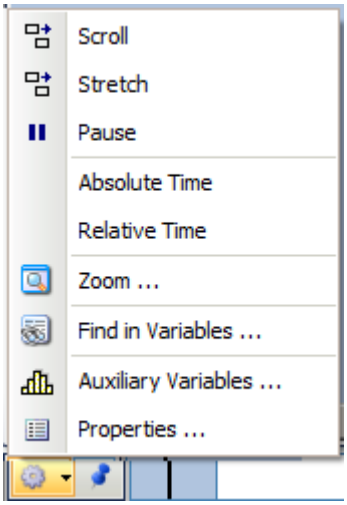
The functions used here (Zoom, Rezoom and Navigation left/right) correspond to the ones which are also available for navigation in the individual curve window.

Note

- If the curve windows linked with the Navigation bar are in either Stretch or Scroll mode, then the position of the slider changes when navigating right or left.
- Conversely, if the linked curve windows are in the Pause mode, the slider stays in the same position in the curve window, and the x-axis moves to the right/left.

8.10.1 Menu

To open the Navigation bar menu, click on the cogwheel icon all the way on the left of the Navigation bar.



Navigation bar menu

Scroll	In *Scroll mode*, the time window displayed always has the same width. The data set thus scrolls across the time window.
Stretch	In *Stretch mode*, the entire data set from the first to the last value is displayed. As long as the measurement is running, the time axis range continues to extend, so that the signal plot appears increasingly compressed.
Pause	In *Pause mode*, all curve window's linked with the Navigation bar are frozen. Measurement continues in the background without being displayed.
Absolute Time	Sets the X-axis of all linked curve windows to absolute time.
Relative Time	Sets the X-axis of all linked curve windows to relative time.
Zoom	Opens a window which enables the zoom to be entered in seconds. This is used on all linked curve windows.
Find in Variables	Searches for entries among variables linked with the Navigation bar, see also the segment Find in Variables 1099
Auxiliary Variables	see section Linking variables 1097
Properties	Post-processing mode 1101 shows the Navigation bar's tools and time readout

In order to use the zoom settings and to change the position using the Navigation bar as well as the slider on all Panel pages simultaneously, select the Panel page's context menu item *Options/Navigate through all Panel pages.*

Properties

Widgets Navigation Mode

Option	Description
Widgets navigation mode	One can choose which value the Widgets shall show. There exist following possibilities: • Automatic: During the measurement is running or while working on the current measurement, the Widgets show the current value. They show the value at the slider's position, whenever one is working on saved measurement data. • Always show navigation value: The Widgets show always the value at the slider's position. • Always show current value: The Widgets show always the current value.

Post-processing Tool

see part Post-processing mode 1101

Time Controls

By means of the menu item *Properties*, you can choose to display time controls in the Navigation bar:

Navigation bar with time controls

The times indicated are the time points corresponding to the left and right edges of the white bar. Using the time readout it is possible to navigate left or right, and to Rezoom the X-axis.

There exist, as for the navigation tools in the middle, two possibilities to navigate left and right:

- Click on the arrows multiple times in order to move forwards in individual steps.
- Move the mouse pointer over a time control, hold the mouse button down and drag the mouse in the direction desired.

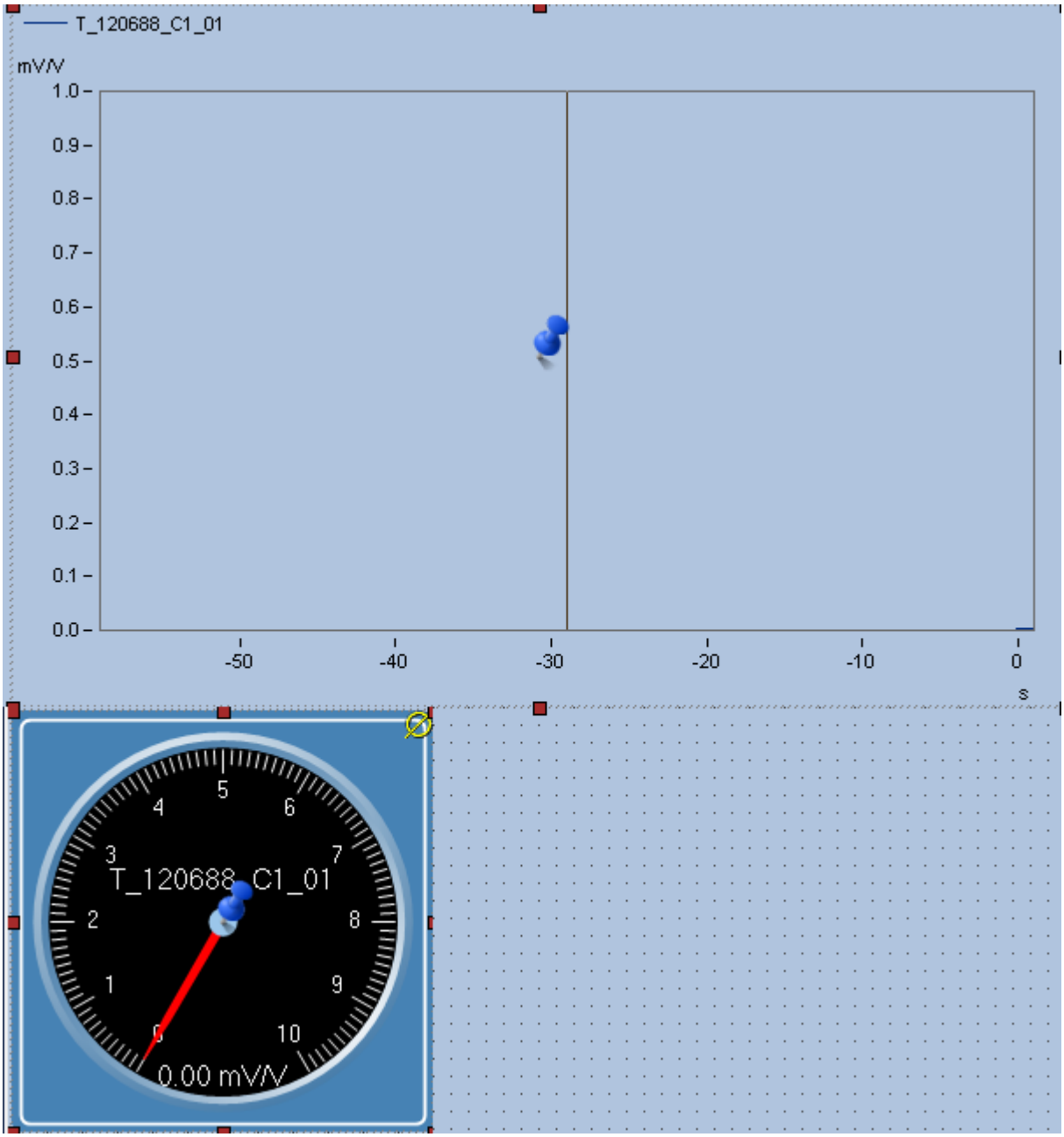
In contrast to the navigation tools in the middle of the Navigation bar, these tools apply to the left/right edge of all linked curve windows. Illustrated by the example of the Navigation tools for the left time point:

- Rezoom of the X-axis changes the zoom of all linked curve windows in such a way that the earliest time is visible in the curve window.
- In navigation to the left and right, the right edge does not change

The navigation tools for the time point on the right side apply analogously.

8.10.2 Linking a Widget

In order to link a Widget with the Navigation bar, click on the *Link*-icon ( ) in Navigation bar. Note that for this purpose, the Design Mode must be activated. The mouse pointer changes into a pin symbol. Next, click on the Widget which you wish to link with the Navigation bar. All linked Widgets are indicated by a blue pin symbol in the center:



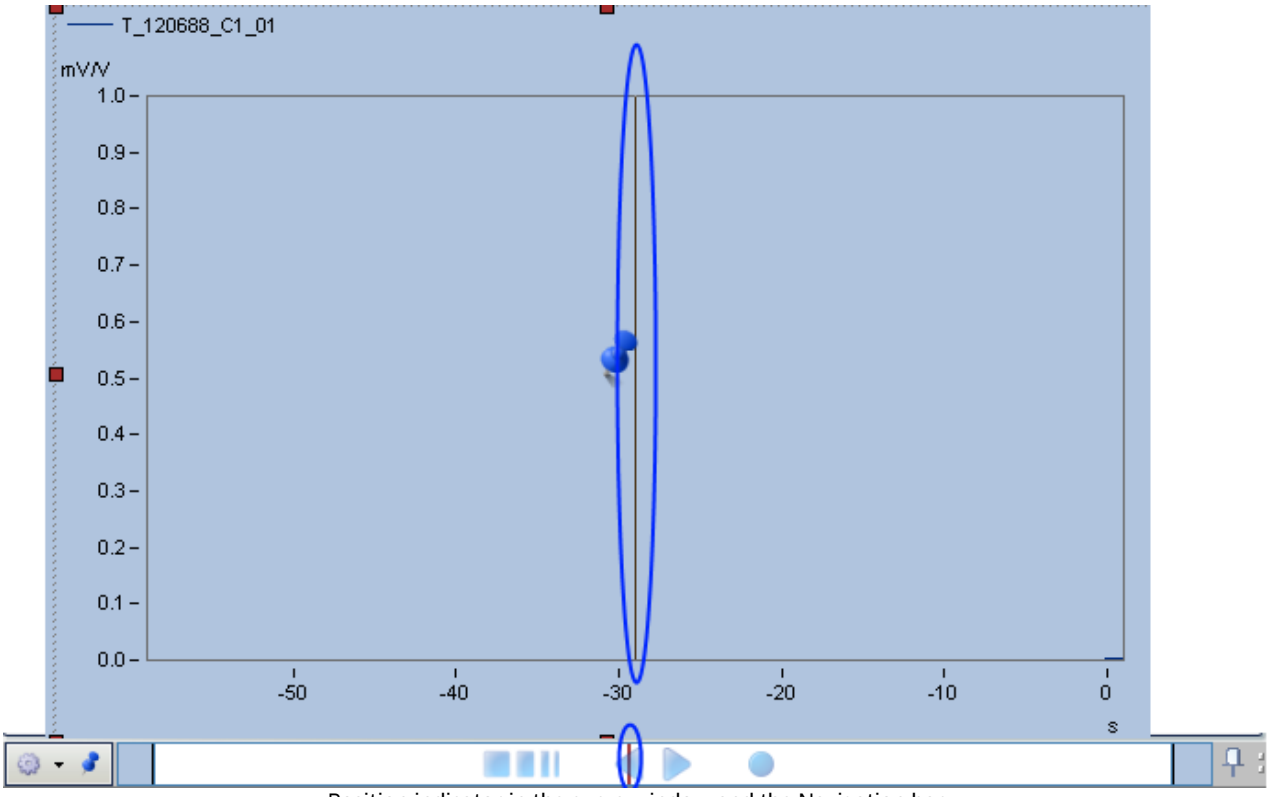
Widgets linked with the Navigation bar

To cancel the link, click on the *Link*-symbol ( ) in the Navigation bar, so that the mouse pointer is again displayed as a pin symbol. Then click again on the linked Widget whose linkage you wish to cancel. The pin on the Widget disappears and the linkage is canceled.

In order to link new Widgets with the Navigation bar automatically, select in a Panel page's context menu the item *Options/Link new Widgets to navigation bar*.

8.10.3 Slider

If a curve window is linked with the Navigation bar, a black line is displayed in the curve window. This black line is a slider, corresponding to the red line displayed in the Navigation bar. In the image below, the two lines are circled in blue.



Position indicator in the curve window and the Navigation bar

If multiple curve windows are linked with the Navigation bar, a black line of this type is displayed in each of the curve windows.

There is an additional black line displayed in the Navigation bar as a slider. To change the position, move the mouse pointer over the line, which changes the mouse pointer to turn into a double-arrow. Then move the mouse left or right to the desired position while holding down the mouse button.

The slider's position is changed in all linked curve windows. Therefore you can use this function to compare different curves at a certain location, for instance. The slider is independent of the individual curve windows' zoom setting.

If the slider is not visible, a red arrow is shown at navigation bar's left or right boundary. There are two ways to get it into the desired curve window:

- Move the mouse pointer over the desired curve window and press the key combination Shift+ left mouse button. the slider is fit in at the mouse pointer's position.
- Move the mouse pointer over the desired curve window's left or right boundary, which changes the mouse pointer to turn into a double-arrow. Press and hold the left mouse button and move the double-arrow to the desired position in curve window, then release the mouse button.

8.10.4 Linking variables

It is possible to link variables with the Navigation bar. This is needed in particular for report channels in order to be able to navigate through entries in the report channel.

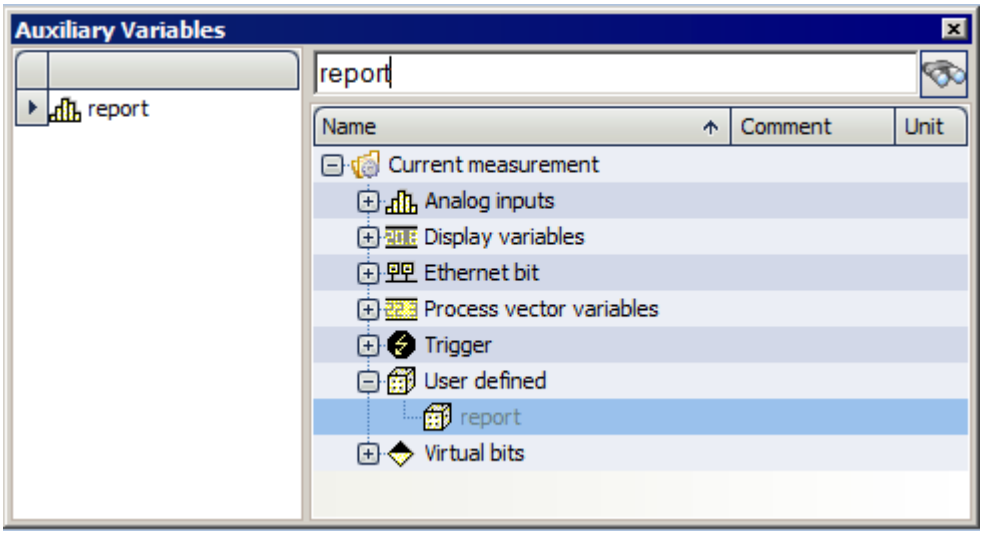
Note

The variables belonging to linked Widgets are not automatically linked with the Navigation bar.

There are two ways to create the link:

- Move the desired variable from the Data Browser to the Navigation bar by means of Drag&Drop.

- Use the Navigation bar menu ( ) item Auxiliary Variables and select the desired variable there.



"Auxiliary Variables" dialog

In this dialog window, variables which have already been linked are shown in the left column. On the right side, available variables in the experiment are shown, sorted the same way as in Data Browser. Above that is a search input box to use for finding variables.In order to link the desired variable with the Navigation bar, use the Drag&Drop technique to move it from the right side to the left side.

In order to cancel the link, highlight the desired variable in the left column and select the context menu (called by clicking the right mouse button) item *Delete*.

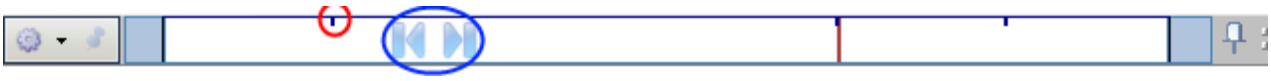
8.10.5 Navigation by marker

If a variable (e.g. a report channel) is linked with the Navigation bar, the Navigation bar displays stylized markers on its top edge. Markers are set at the following positions:

- Text entry (day)
- Audio-entry (trigger time + offset for entry)
- Trigger time (+ offset) of all linked measurement channels

By means of these markers (circled in red in the image below), it is possible to navigate.

If the Navigation bar is in the focus (the mouse is either on the Navigation bar or the mouse had clicked on the Navigation bar), then the Navigation mode can be switched by clicking in the Alt-key. Then it is possible to navigate by means of the keyboard (left/right) arrow keys or the mouse:



Navigation bar, navigating by means of markers

To skip with the mouse from one marker to the next, use the symbols on the Navigation bar, which are shown above circled in blue.

In addition to this way of navigating, you can also search for and jump to targeted markers:

- With the key combination Ctrl+F
- Using the Navigation bar menu ( ) item *Find in Variables*

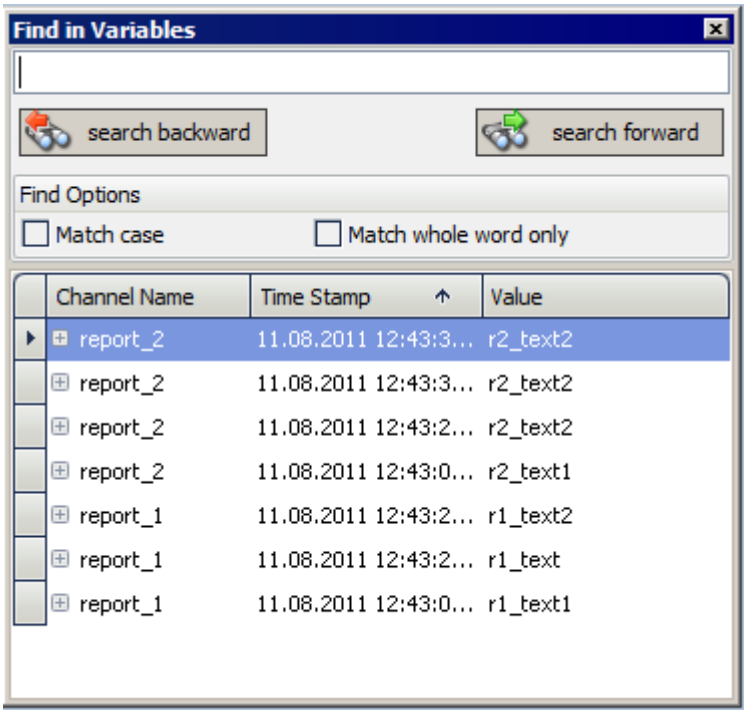
8.10.6 Find in Variables

With this function, you can target certain markers for a search, or skip straight to them. There are two ways to open the search window:

- With the key combination Ctrl+F
- Using the Navigation bar menu ( ) item *Find in Variables*

A dialog window opens, which lists all markers with a time stamp and the associated channel. By double-clicking on an entry, or by selecting one using the Enter key, it is possible to navigate to the desired location.

The text entered in the search box is compared with all the entries in the list at the bottom. Thus, it is also possible to search for a time stamp, for example.

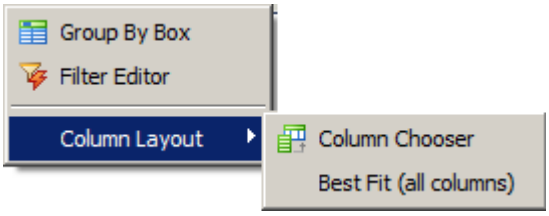


"Find in Variables" dialog, here with two report channels

In the image above, two report channels which contain text inputs are linked with the Navigation bar. Report channels containing audio comments have a button in the column *Text*, on which the loudspeaker symbol ( ) is shown. Click on this button to hear the comment.

Adapting the layout

The search window's columns can be adapted. To delete a column, click on the title in the column, keep the left mouse button down and drag the mouse down until a black cross appears. Then release the mouse button and the column is deleted. Also, you can add columns: right-click the mouse over the left corner of the title bar to open the table's context menu:



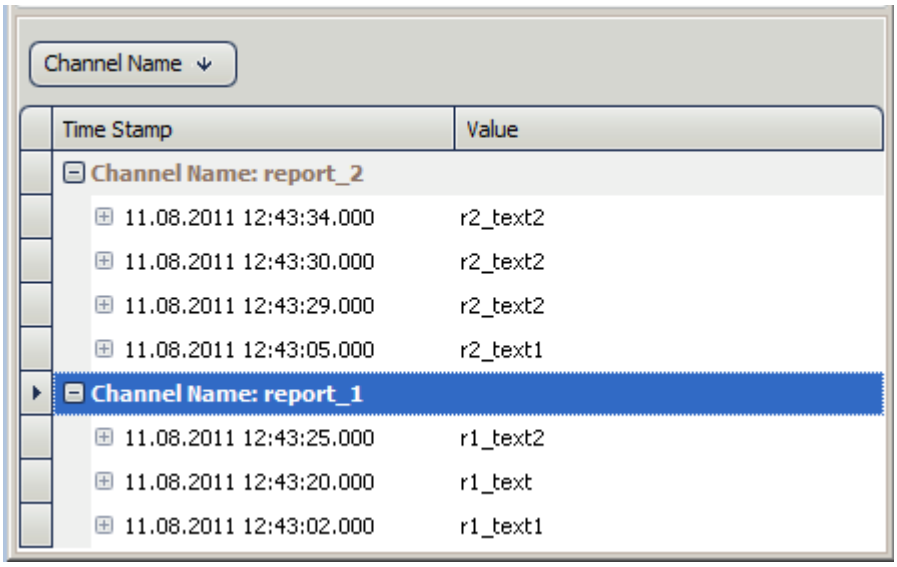
Context menu of the "Find in Variables" - table

Here, if you select the item *Column selection* under *Column layout*, you get the columns which are available to adding to the table by means of Drag&Drop. The available columns also include the columns you have already deleted.

Search

In this dialog, there are various ways to find your desired entry:

- In the upper input box, you can enter a search string. You can have a search for it going down through the list (search forward , F3 key) or upwards in the list (search backward ,key combination: Shift + F3).
  - Under *Search options*, you can additionally select whether to match upper- and lower-case spelling and or whether to search for the entire string.
- By clicking on the column title, the search is according to the corresponding column, i.e. according to *Channel name, Time stamp* and *Value.*
- Filtering entries (table context menu item: *Filter Editor*)
- Grouping (table context menu: *Show grouping box*): For example, you can group according to the column *Channel name*, in which case you get a branching for each channel, under which the respective channel's entries are listed.

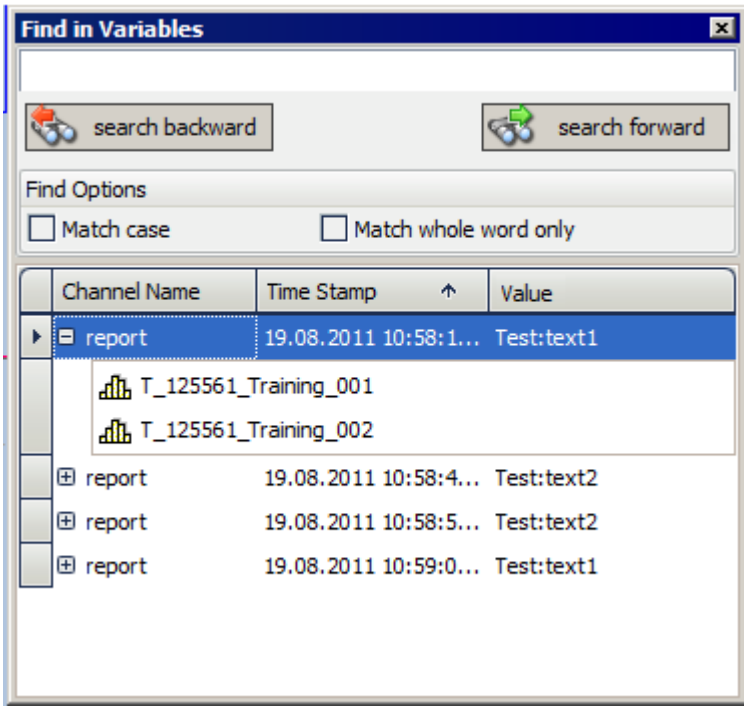


Example above, grouped according to channel name

Initially, the entry found is only highlighted. To navigate to the entry, hit the Enter key or double-click on the entry.

Context

If a context was specified with an entry to a report channel, it is displayed below the channel name. Since all entries in the columns are taken into account, it is also possible to enter a context in the search box. Additionally, the column *Context* can be displayed; for information on this, see the section *Adapting the layout.* Use of the context is described in more detail in the section Text input for report channels 1074.



Displaying the context in the Search window

8.10.7 Post-processing mode

As a rule, entries to report channels (both audio and text commentary) are written to the channel at the current point in time. Entries added after the end of the measurement are written to the position of the slider control. Detailed information on this topic is presented under Special Widgets 1074 as well as in the chapter user-defined variables in the section *Report Channels*.

The Post-processing mode offers the ability to write comments to the channel at the position of the slider even during a running measurement. For this purpose, the curve window in which the slider is to be moved to a certain position must be linked to navigation bar and must be in Pause mode. Further, the report channel's Widget must also be linked to the navigation bar, see also part Linking a Widget 1095.

You can activate Post-processing mode by selecting the sub-item *Post-processing mode as default* from the context menu of a Panel page under the item *Options*. If you select the same item again, Post-processing mode is then deactivated.

You can also use the dialog *Properties* to set a button to be shown, by which means the Post-processing mode can be activated and deactivated. The symbol appears at the left in the navigation bar. If it is shaded blue, then Post-processing mode is active.

9 Automation

imc STUDIO Automation is the imc STUDIO plug-in with which you can generate *Tasks* 1141 (routines) for open- and closed-loop control operations. With the help of a graphical interface, you can construct a state-based logic. The imc measurement device captures the running task's states and evaluates them.

You can generate up to five *Tasks* 1141 for one device.

With the plug-in Automation's graphical interface (Automation Editor 1140), it is easy to impose a clear structure on a complex routine. You generate states which govern how the system proceeds. This has the advantage of avoiding the need for complex programming language for capturing the states.

A variety of control structures are available for the tasks (For 1121, Do 1121, While 1121, If 1123). For the purpose, of evaluating measured data, you can insert imc FAMOS* sequences.

The Task has four components:

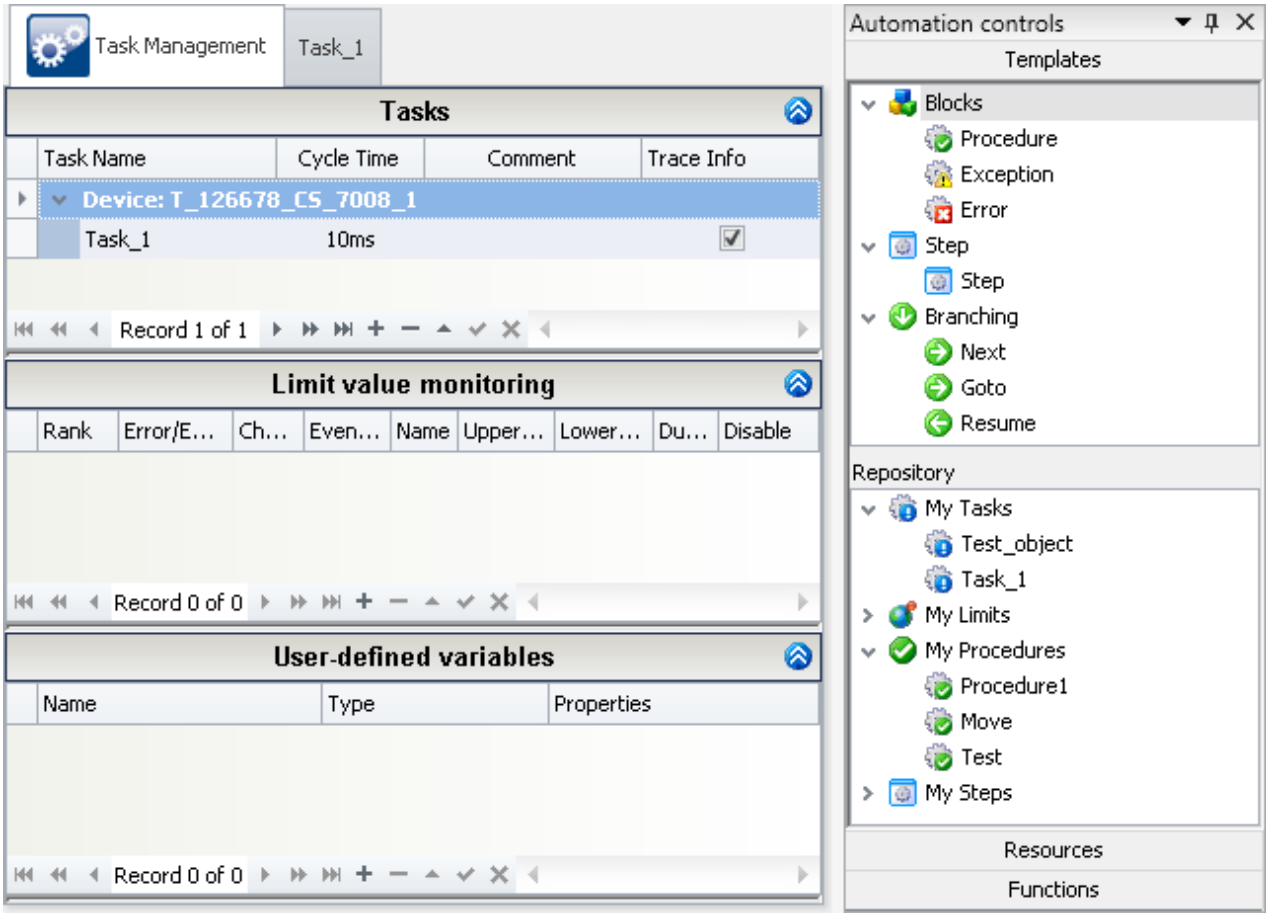
- *Initialization / Terminate* 1151, here, the Task is started (initialized) and ended.
- *Procedures* 1152 represent the actual routing working.
- *Exception handlers* 1153: With these, you can remedy certain errors in a controlled way, for instance if value limits 1143 are violated during the running of the task.
- *Error handlers* 1153: With these, it is possible to end the task in a controlled way if serious errors (ones which can not be remedied) occur. The test object may, for example, be placed in a safe position before the system is shut off.

In order to observe the results on the PC or to intervene in the procedure during a running task, use the plug-in Panel. On this topic, see the separate Documentation on the plug-in Panel 813.

A prerequisite for operation of the plug-in Automation is an imc measurement device with imc Online FAMOS Professional installed.

* the product imc FAMOS is sold and installed separately (see Additional imc Software Products (optional) 1702).

9.1 Structure of the plug-in Automation



Components of the plug-ins: Automation

In the picture at left, you see the Automation Editor 1140 in which the tasks (flow controls) are created.

In the tool window Automation controls 1113 at right, process elements, functions and access to the input/output parameters (resources) are avialable. These include: Templates 1113, Resources 1137 and Functions 1138.

9.2 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 plug-in 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

Within a task it is possible to create multiple blocks 1115. A block comprises multiple steps 1118, which in turn comprise multiple states 1119, loops 1121 and branches 1123.

First create the complete tasks using the blocks and then proceed in more detail step-by-step.

- A Task represents a complete routine
- A Block represents a portion of a routine
- A Step represents one concrete action to be taken. This is filled with the actual action elements: states, loops and branches.

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. Afterward, it is possible to start all over.

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:

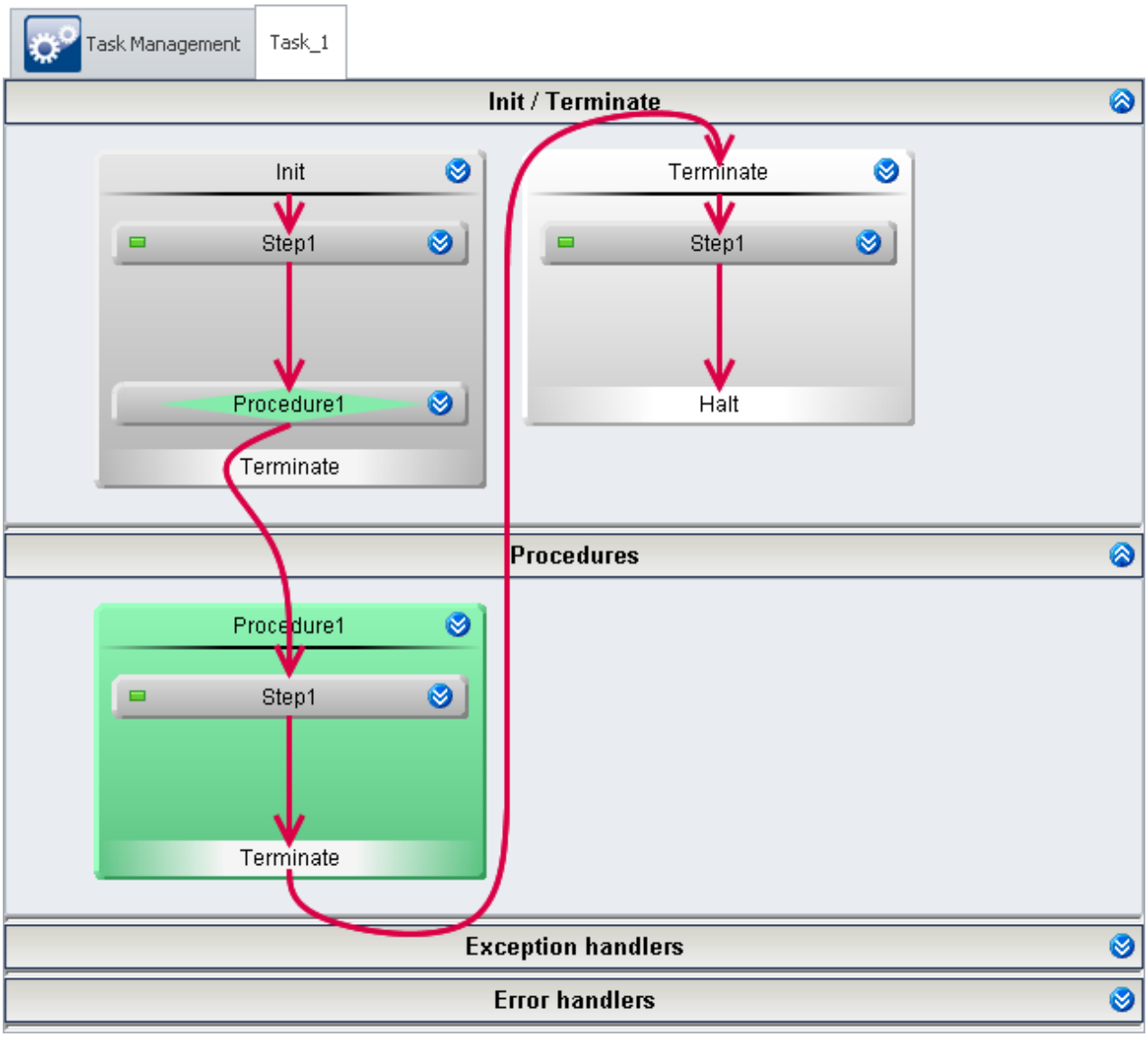
- Start movement
- End movement once height reached

9.3 How does it work?

A task begins at the start of the measurement.

After the start, the Initialization (Init) 1151 is called. You must specify how matters proceed next. By default, the initialization contains a Next branching 1116. There, the task skips back to the first procedure 1152. At the end of the task, the block Terminate 1151 is performed (as long as no error has occurred). This concludes the task with *Halt*.

Below is a simple example:

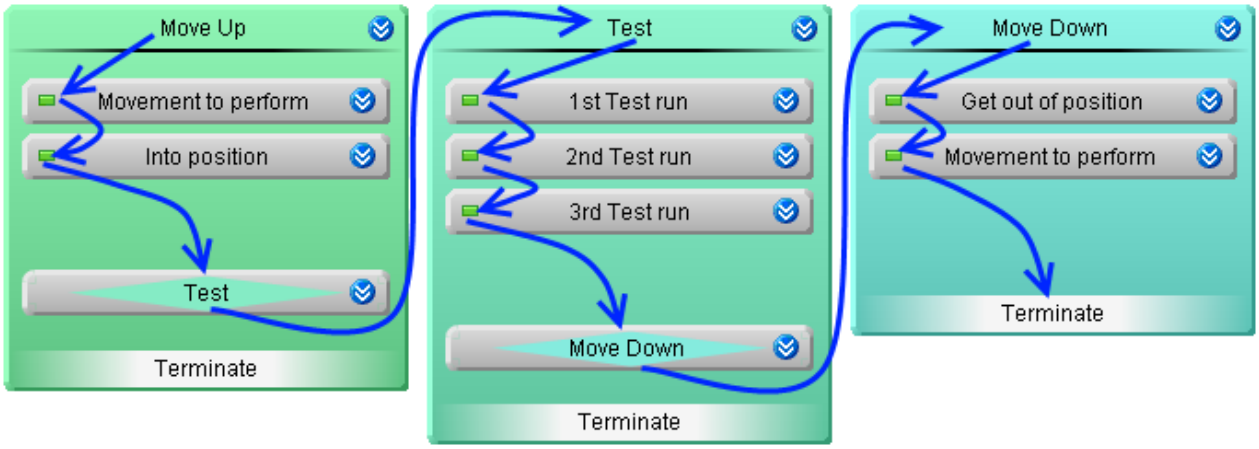


Minimal task without any funtion

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.

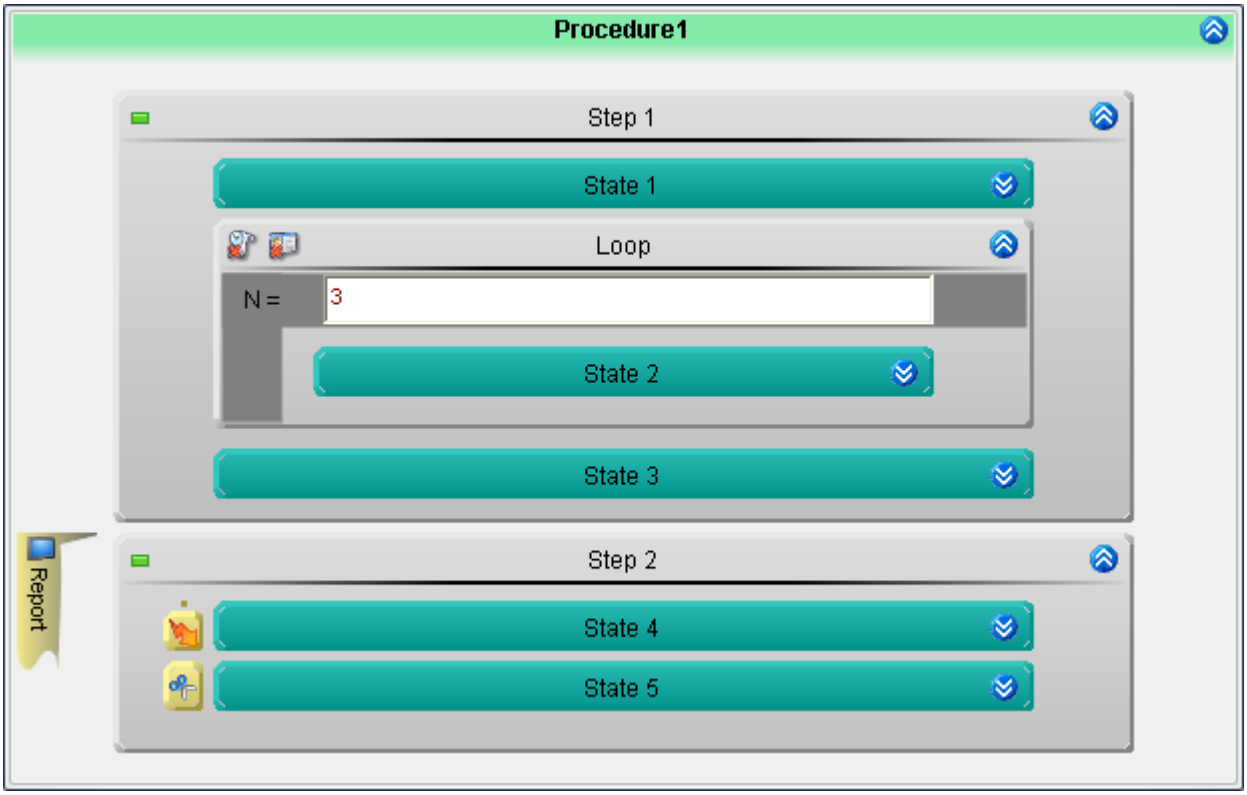
Below, an example with three steps is shown:



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. Below is an 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 chapter: State 1119).
- 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.

Note

Evaluation of the For-loop and of the jump to the next step or block 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.

9.4 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.

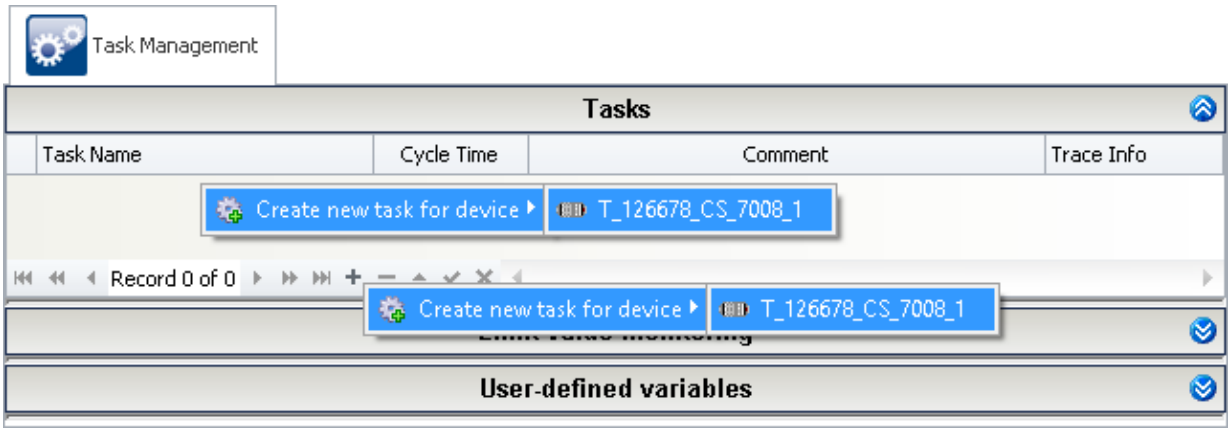
- Double-click on the area title
- Click on the double-arrow (or), or:
- select either the item *Open* or *Close* from the context menu.

9.4.1 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.

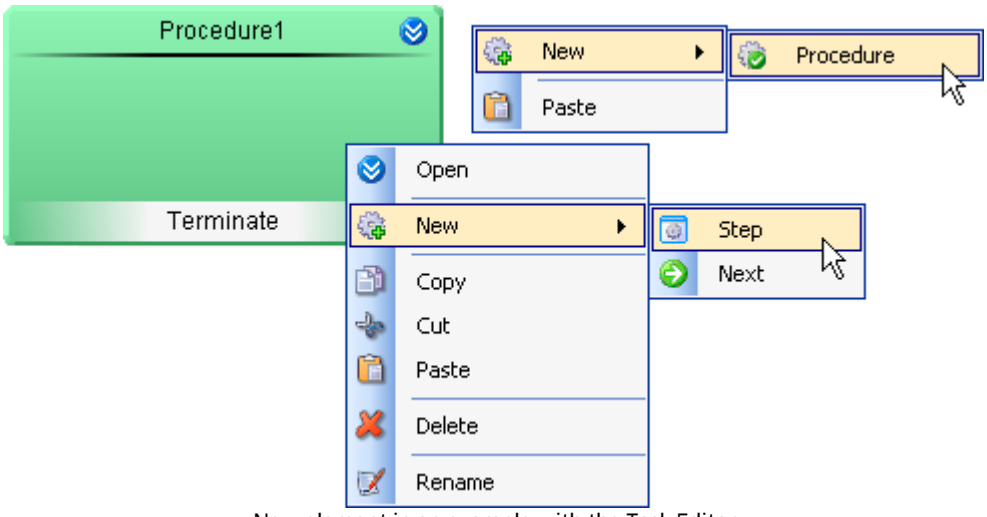
Exporting an element to a different device

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.

9.4.2 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.

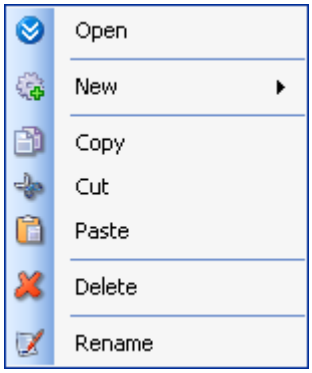
The element is created at the selected location.

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

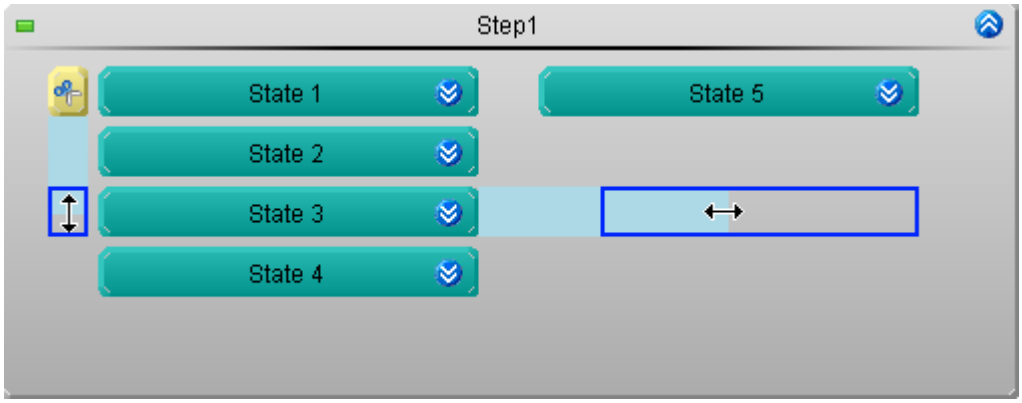
Some elements in the Task Editor can change in size and thus also change their association with other elements.

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

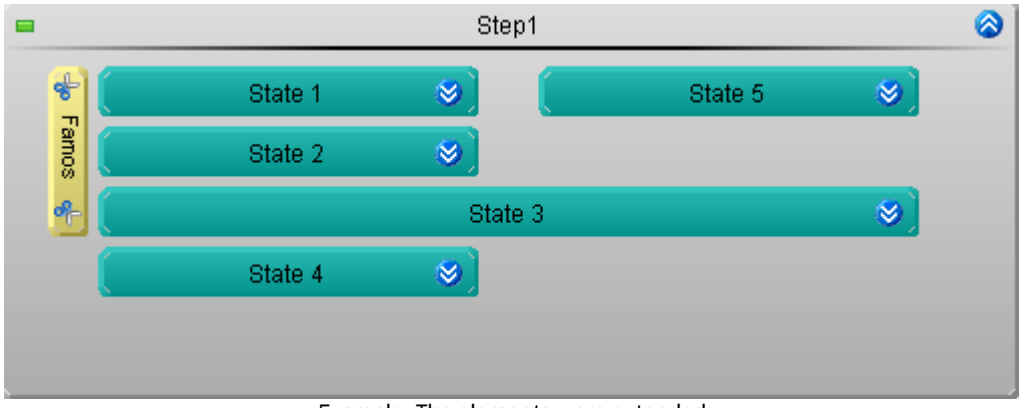
Example: *State 3* is intended to synchronize across two tracks, and a imc FAMOS evaluation is intended to "cut" three states.

- Grab the edges of the element and release then within the marked space.



Example: Changing the extent of the elements

The elements are extended.



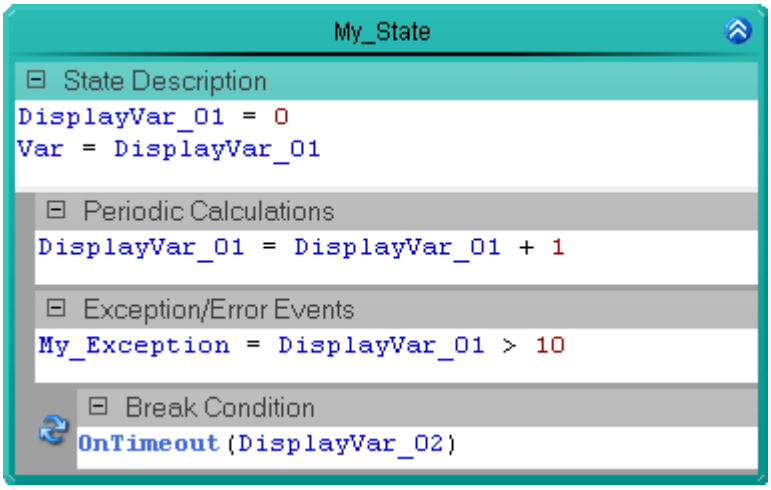
Example: The elements were extended

9.5 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.

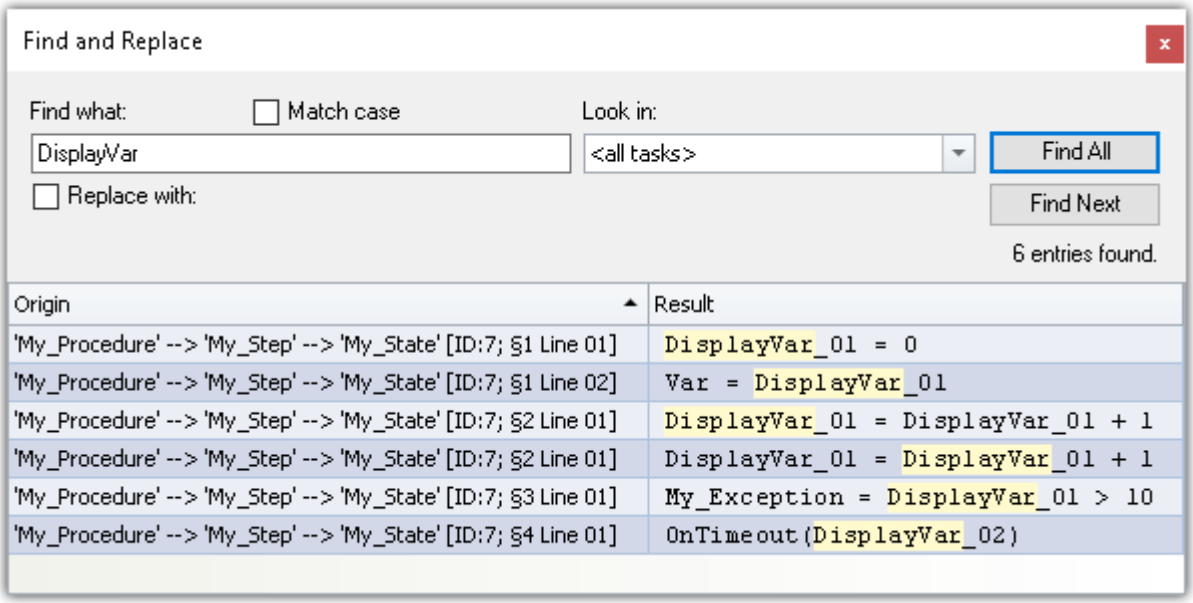
To find a text, open the dialog Find and Replace ( ) (Ribbon *Edit* > *Find and Replace*).

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

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

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

9.6 Trace Info

The trace info displays during a tasks execution which procedure and which step is currently being processed.

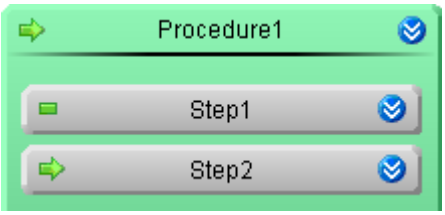
To activate/deactivate the trace info for the task, click on the corresponding checkbox in the Task Management's area Tasks. By default, this option is active.

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

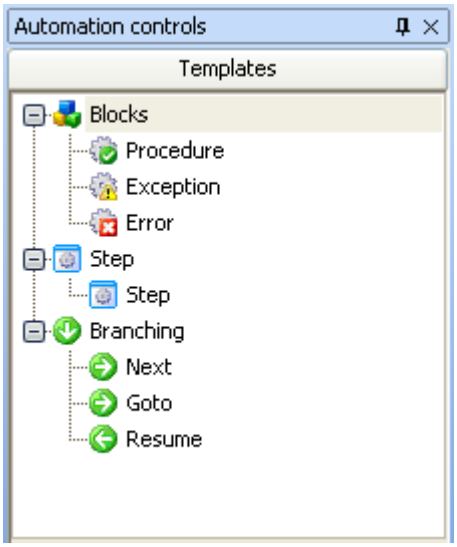
9.7 Tool window Automation controls

The plug-in *Automation* includes the tool window Automation Elements.

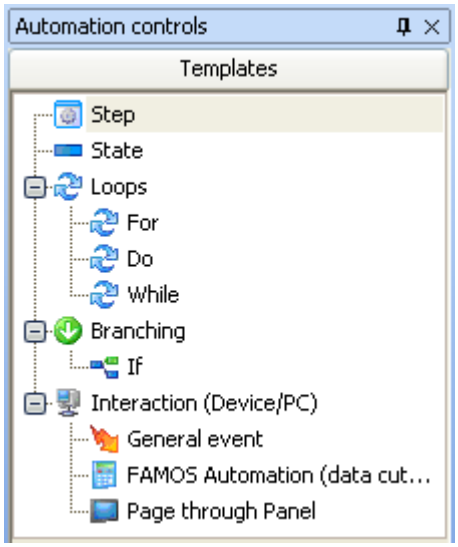
The tool window is divided into three groups:

- Templates 1113 and Repository 1137
- Resources 1137
- Functions 1138

9.7.1 Templates



Templates (Task Editor)



Templates (Block Editor)

The group Templates contains 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 chapter Operation 1107).

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

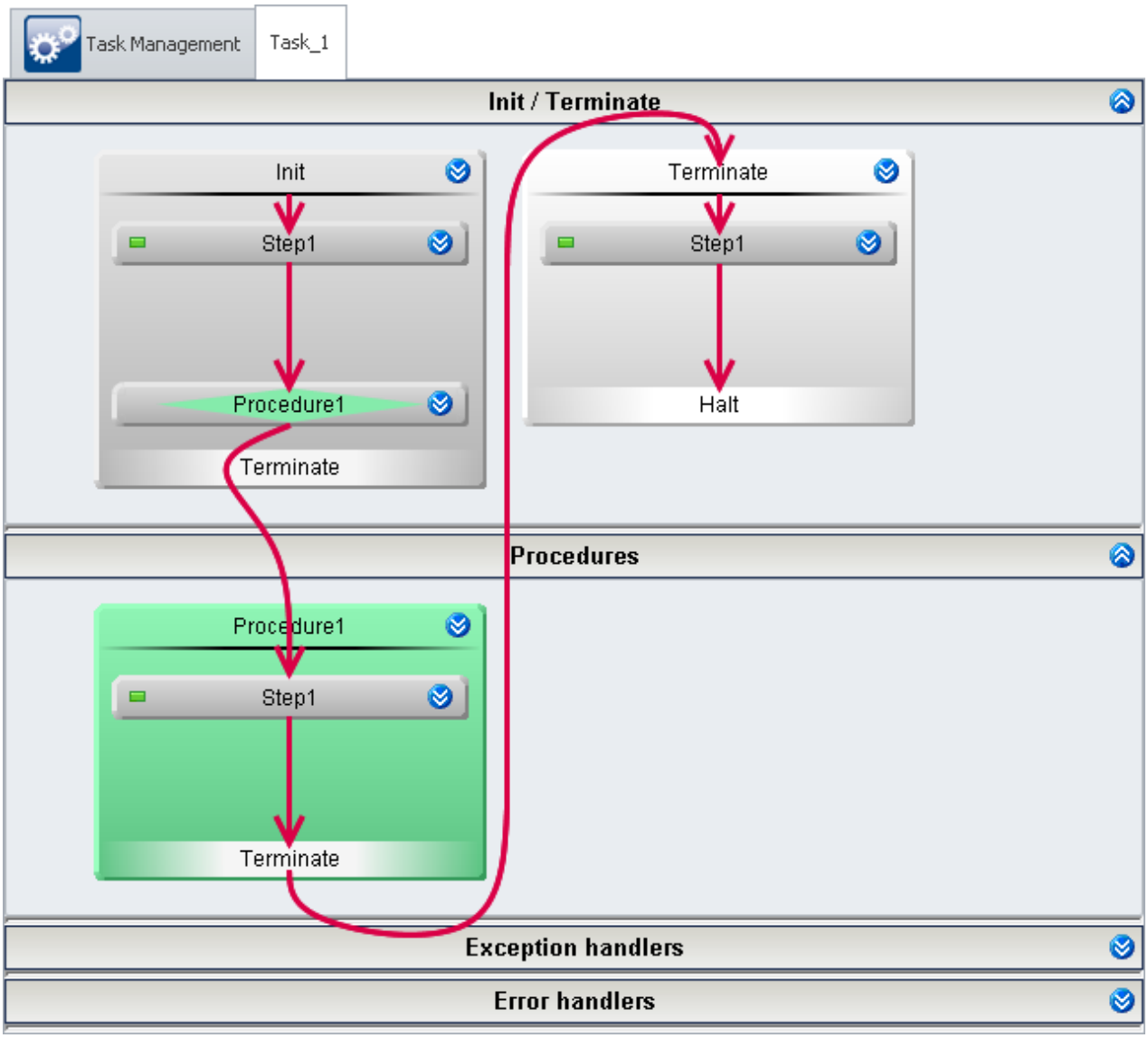
- Blocks 1115
- Steps 1118
- Branchings 1116

Block Editor templates

- Steps 1118
- States 1119
- Loops 1121
- Branchings 1123
- Interaction (Device/PC)

9.7.1.1 Block

A task consists of at least three blocks: an *Init* 1151, a *Procedure* 1152 and a *Terminate* 1151. A block combines multiple steps, which in turn are a combination of multiple other elements.



Minimal task without functions

Five different blocks are available:

- Init(ialization) 1151
- Terminate 1151
- Procedure 1152
- Exception handlers 1153 and
- Error handlers 1153

In each task there can only be one Init and Terminate block each.

You may add any desired amount of blocks of *procedures* 1152, *exception handlers* 1153 and *Error handlers* 1153.

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:

- *Steps* and
- *Branches* (not in Terminate or error handling blocks).

Execution

When the block starts, it is executed from top to bottom.

The End-branch is not an obligatory *branch*. You can insert a *branch* to jump beforehand to any desired *procedure* (not to Terminate or Error handling)..

9.7.1.2 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

- Next
- Resume
- Goto

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".

Next

This branch is used in the Init and in procedures.

Next-branch open and closed

When the branch is called, there is a test of whether the condition is met.

- If it 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.

The branch's color adapts to the target.

Resume

This branch is used in exception handlers.

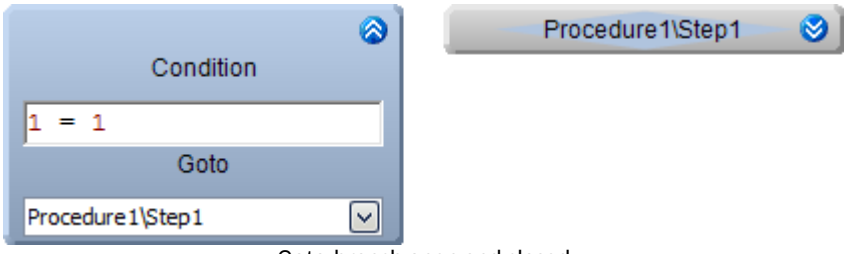
Resume-branch open and closed

When calling the branch, a test is made of whether the condition is met.

- If it is met, then the Task is resumed from the point at which the first exception handler was started. The state is resumed.
- If it is not met, the block's next item is run. This can either be another branch or the block's termination.

Goto

This branch is used in exception handlers.



Goto-branch open and closed

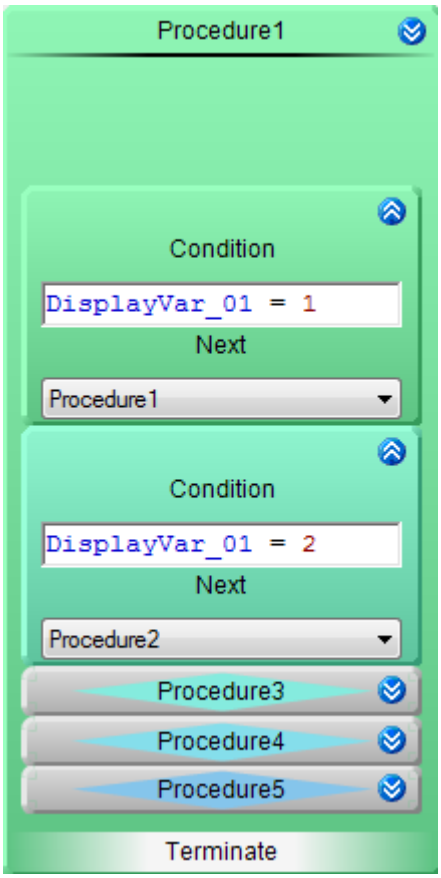
When calling the branch there is a test of whether the condition is met.

- If it 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.

The branch's color adapts to the target.

Multiple branches

Multiple branches can be placed in succession. If the conditions of the first branch are met, then the next one is tested.



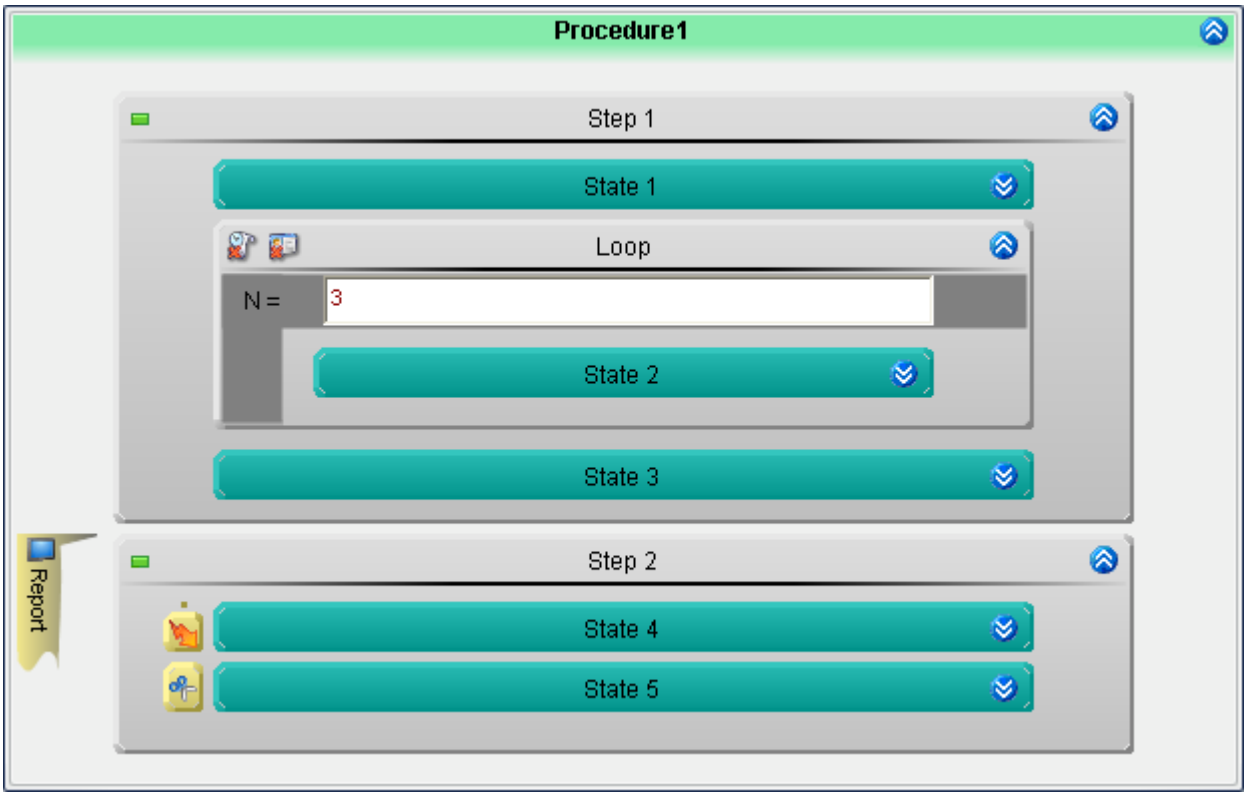
Multiple branches in succession

9.7.1.3 Steps

Empty step

Each block can contain a number of steps. Steps can be created here in the Task Editor or in the Block Editor.

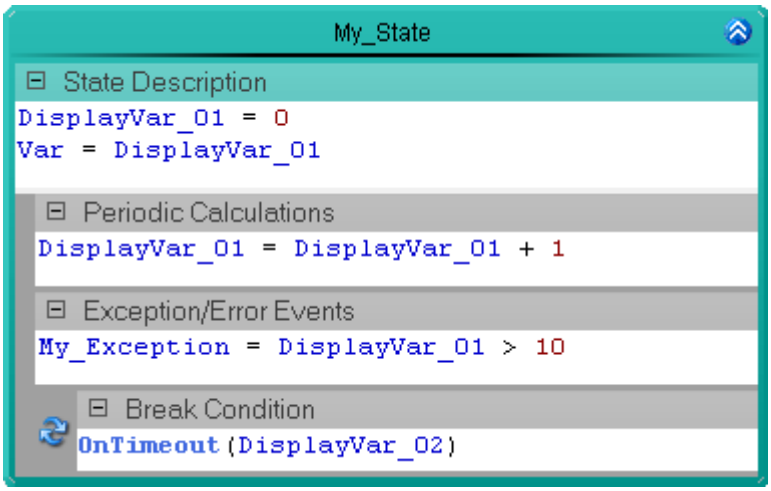
A step comprises multiple states, loops and If-branches.



Example: Two steps
A step can contain various elements

9.7.1.4 State

The element State puts the measurement system in a certain "state" or "condition. For this purpose, a variety of commands are available.



State

Positioning

A state can be located within a step, loop or If-branch.

Structure

A state consists of multiple input boxes:

- 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: — This input box serves the purpose of monitoring measurements. If a condition defined here occurs, the respective handling begins.
- Break condition: — 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 (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

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)

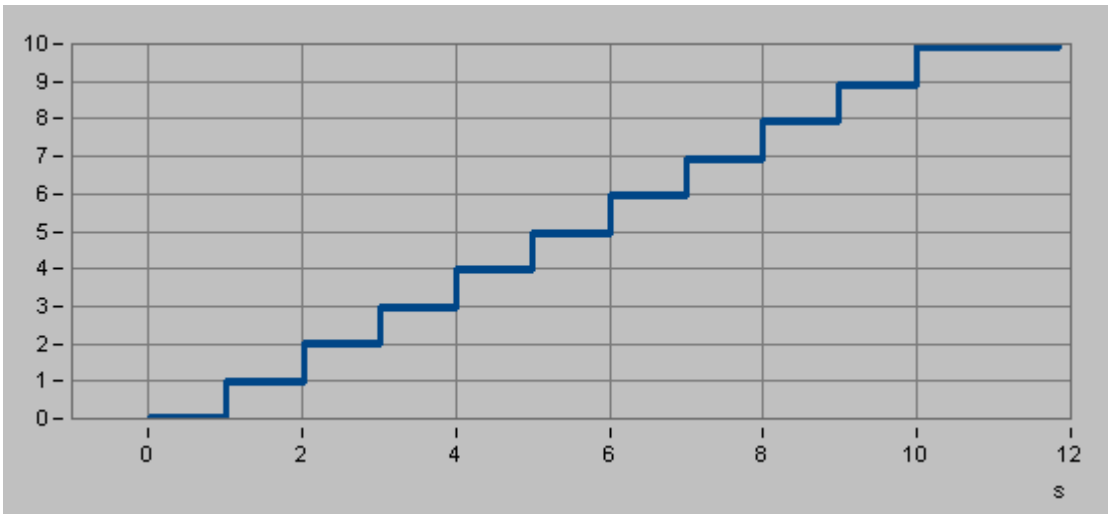
This continues until DisplayVar_01 = 10

t = 9s:

- Cyclical calculation DisplayVar_01 is set from 9 to 10

- Break condition is true (10 = 10)
- The state is concluded

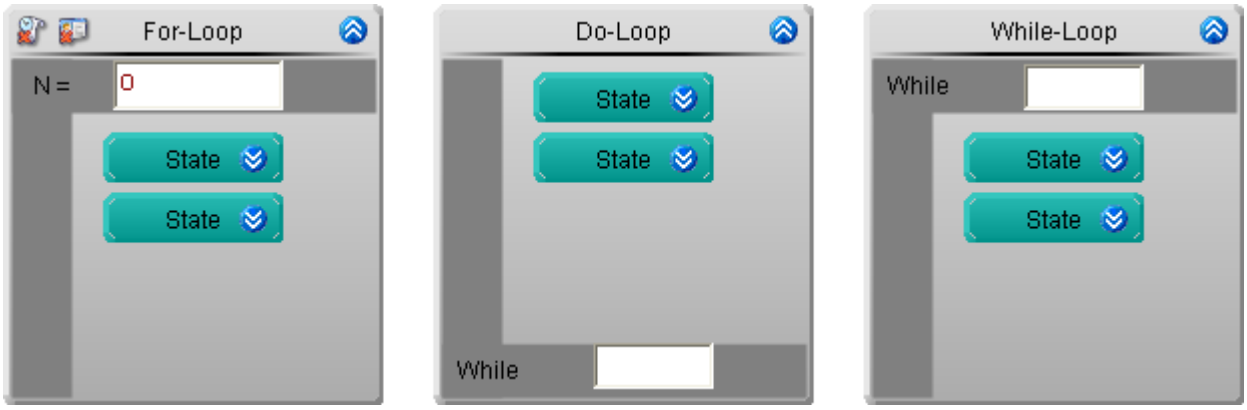
t = 10s: the next state starts



Example: Counter
Plot of the variable: DisplayVar_01

9.7.1.5 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:

- For-Loop — counting loop (for additional properties, see chapter: For-loop 1122)
- Do-Loop — post-test loop
- While-Loop — pre-test 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.

Execution

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.

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.

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.

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.

9.7.1.5.1 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*.

No variable is created.

Note

The variable created is only visible within the task.

Persistent/Non-persistent loop counter

If the element For-loop is run through multiple times in succession, the loop counter always starts counting from "0". 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.

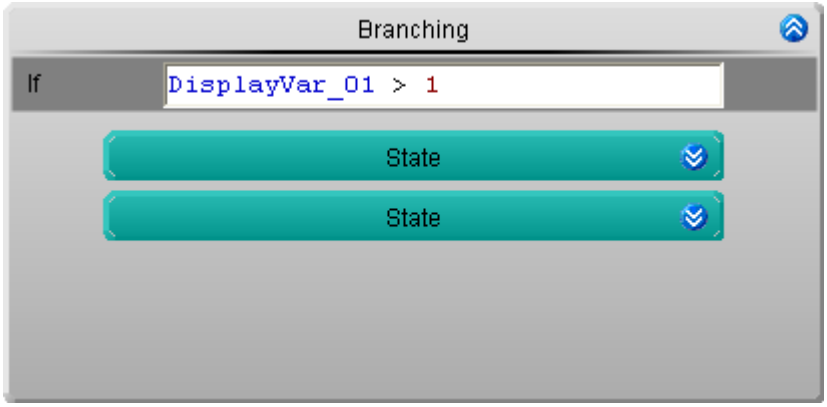
The counter always starts at 0.

Note

- 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.

9.7.1.6 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

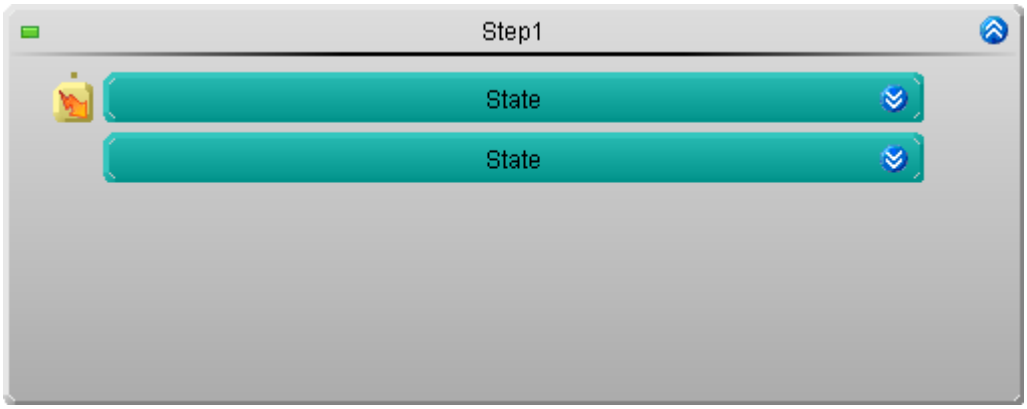
A test box is available for the loops, in which a condition can be entered. This box can contain numbers, variables or other commands.

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.

9.7.1.7 Interaction (Device/PC)

9.7.1.7.1 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

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.

Execution

When the specified event occurs (beginning or end), the command sequence is started.

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.

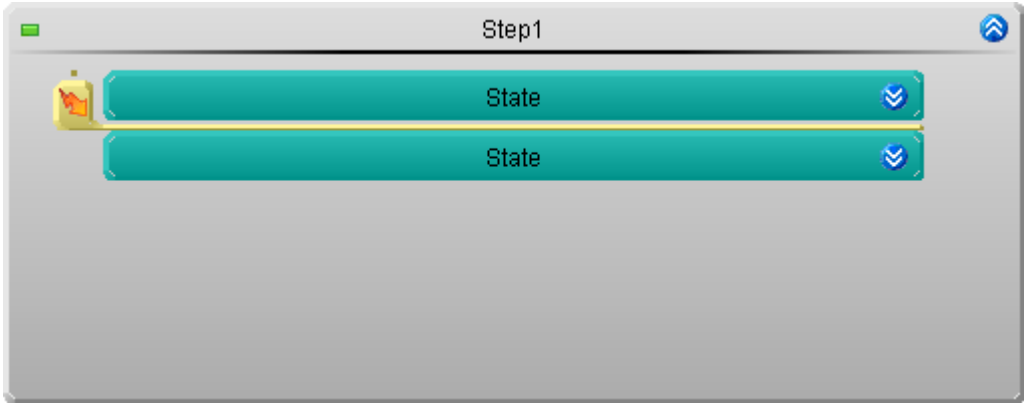
Synchronization

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.

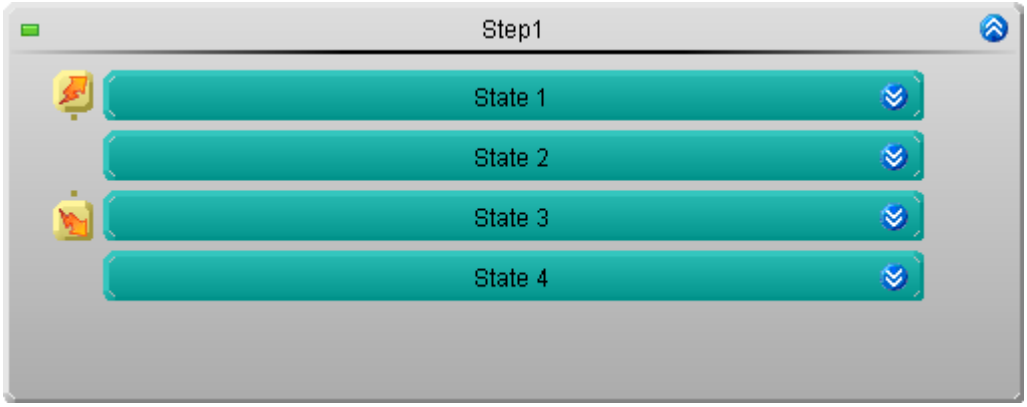
Execution at the beginning or at the end

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:

- *Before* (execute at beginning)
- *After* (execute at end)

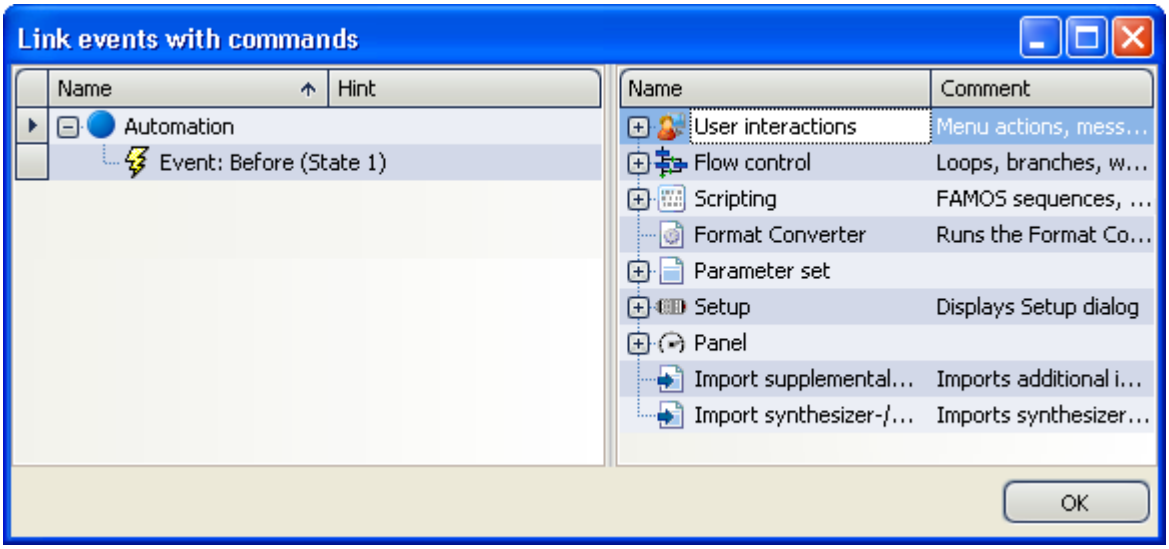
Example: Execute at end



- 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 chapter Command Reference 1592 beschrieben.
- A sequence of commands can be set up in a variety of imc STUDIO plug-ins and is described separately (see chapter Command Sequence 1650).

9.7.1.7.2 imc FAMOS automation (data cutting)

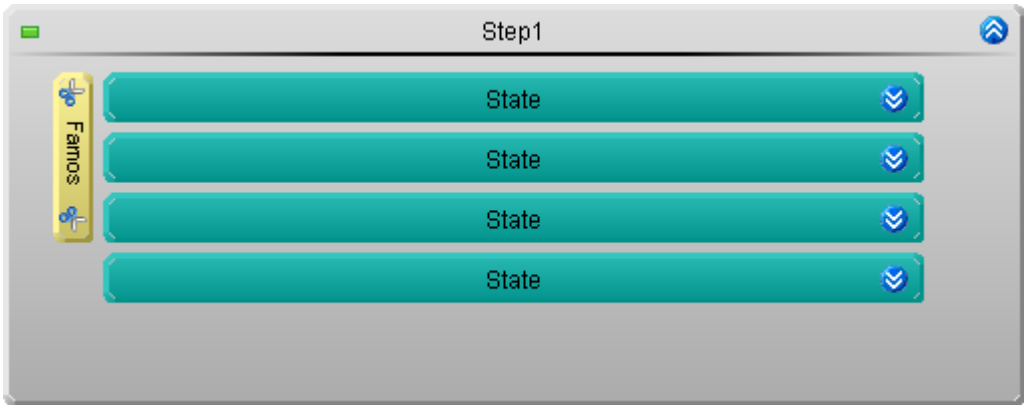
From within imc STUDIO it is possible to transfer variable currently being measured to imc FAMOS and to process them with a imc FAMOS sequence there. The results of the imc FAMOS sequence can be transferred back to imc STUDIO and displayed in a Panel page, for instance.

Note

The prerequisite is installation of imc FAMOS on the same (see *Technical data sheet* > *Additional imc Software Products* 1702).

Reference

Information on imc FAMOS' scope of functions is presented in the separate imc FAMOS manual



Analysis with imc FAMOS

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 are possible.

Execution

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*.

Note

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.

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.

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.

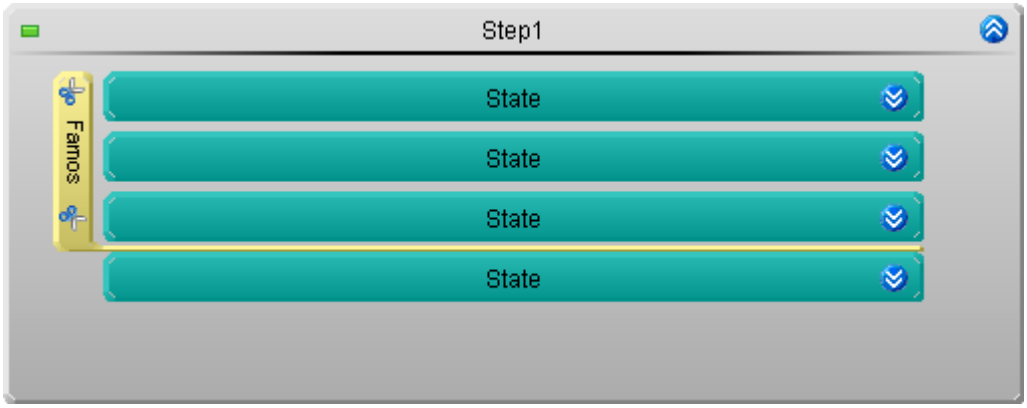
Synchronization

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

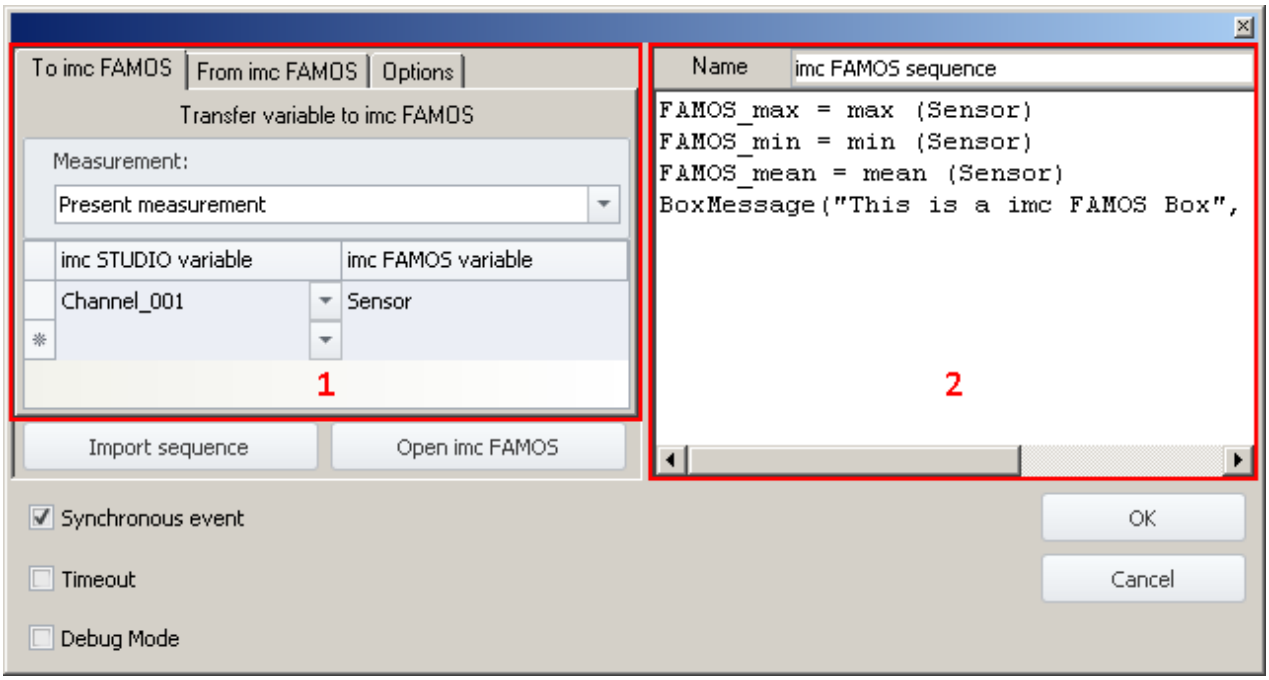
The duration of the analysis depends on the sequence and the computer. In this case, it is not possible to say when the next state will start.

Opening the imc FAMOS dialog and making settings

To open the imc FAMOS Dialog 1129, double-click the left mouse button over the element.

9.7.1.7.2.1 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 1132 (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 1132 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 1132 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*.

Checkbox: 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.

Checkbox: 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.

Checkbox: 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.

Note

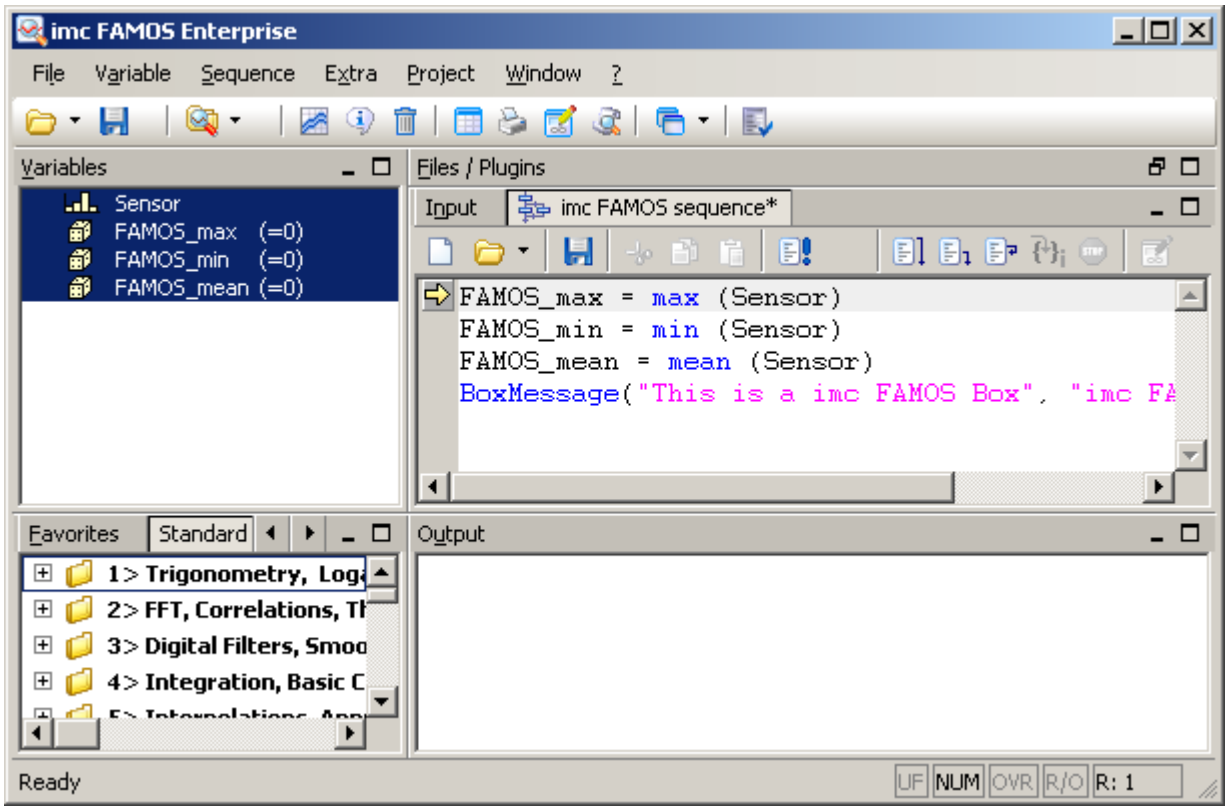
- Please note that in Debug Mode, the command needs longer to execute:
  - With the option: *Synchronized event* active: The subsequent commands wait until the imc FAMOS-command has concluded.
  - With the option: *Synchronized event* deactivated: Required results may not exist in time since the command 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.

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 1131 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 1213 .

Note: Variable not present

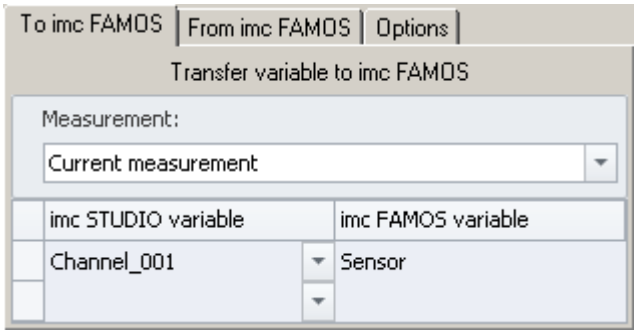
If a variable does not exist in imc STUDIO when it is to be transferred to imc FAMOS, an "empty" variable is created in imc FAMOS as a substitute.

- Data type: Normal data set
- X-Delta: 1
- Total size: 0

9.7.1.7.2.2 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 1131).

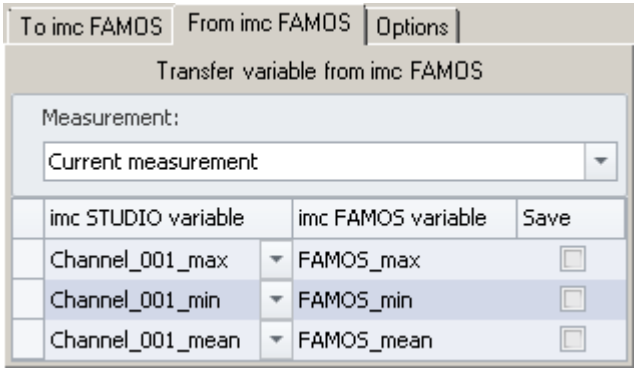
In the example shown, the measured variable *Channel_001* is passed to the imc FAMOS sequence as the variable *Sensor*.

Measurement: Selecting the source

All variables defined are transferred from the measurement selected.

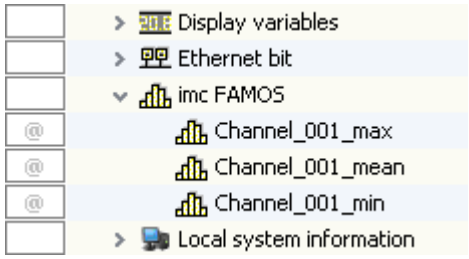
- Current measurement: The current measured data are transferred. Note that in this case, only the amount of circular buffer memory set for the curve window display is available for use. This may be only a fraction of the entire measurement.
- Last concluded measurement: If the measured data are saved, the last measurement saved is loaded automatically. The associated variables are passed to the imc FAMOS sequence.
- Measurement#<No>: The variables belonging to the measurement with the respective measurement number are passed.

Receiving the sequence's results ("From imc FAMOS")



The variables calculated by imc FAMOS are transferred to imc STUDIO along with the transfer table.

The results appear in the Data Browser:



Without saving:

The result appears in the Data Browser, but is not saved.

Measurement: Selection of the target

- 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 concluded measurement and Measurement#<Nr>: The result appears temporarily in the Data Browser along with the last/selected measurement. The result is not saved and is deleted as soon as the measurement is closed (un-loaded).

With saving:

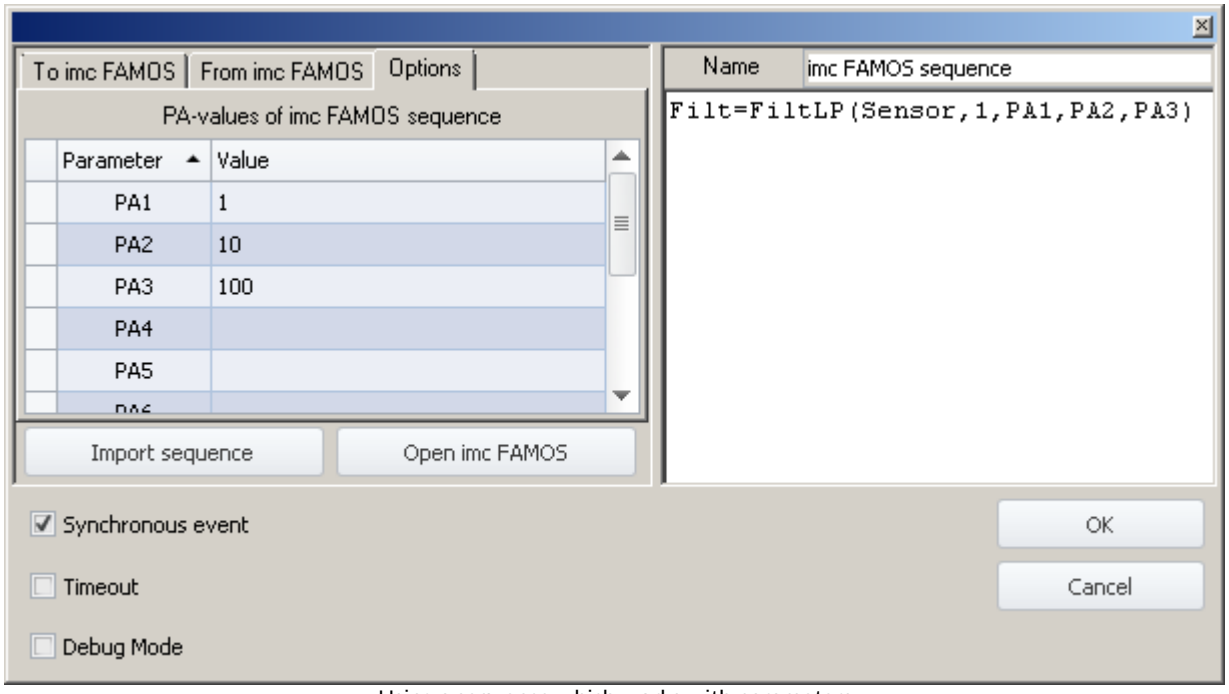
The result appears in the Data Browser and is stored as a.dat-file.

Measurement: Selection of the target

- 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.
- If Current measurement is selected as target (not recommended), 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.



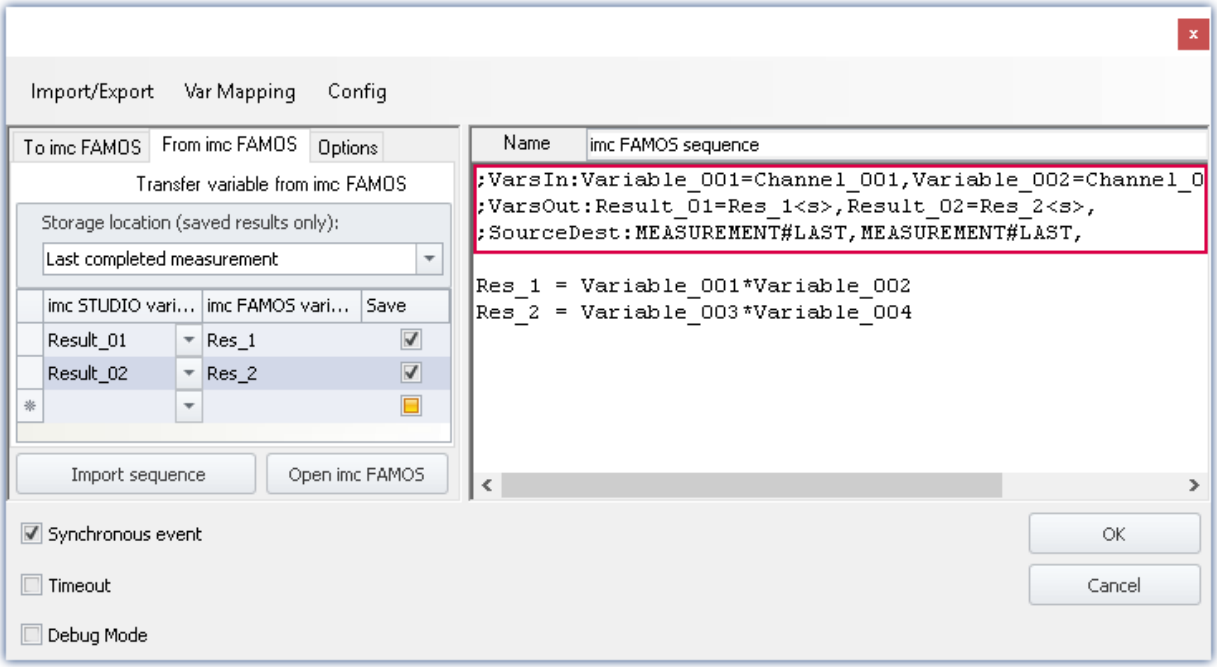
Using a sequence which works with parameters

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):

Variables: To imc FAMOS:

;VarsIn:`Variable_001`
`=Channel_001`
`,Variable_002=Channel_002,Variable_003=Channel_003,Variable_004=Channel_004,`

Description	Syntax
Start	;VarsIn:
First variable name	Name in imc FAMOS
Assignment character	=
Second variable name	Name in imc STUDIO
Separator from next assignment	,

Variables: From imc FAMOS:

;VarsOut:`Result_01=Res_1<s>,Result_02=Res_2<s>,`

Description	Syntax
Start	;VarsOut:
First variable name	Name in imc STUDIO
Assignment character	=
Second variable name	Name in imc FAMOS
Activation of data storage (optional)	<s>
Separator from next assignment	,

Source and data storage location:

;SourceDest:`MEASUREMENT#LAST,MEASUREMENT#LAST,`

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:

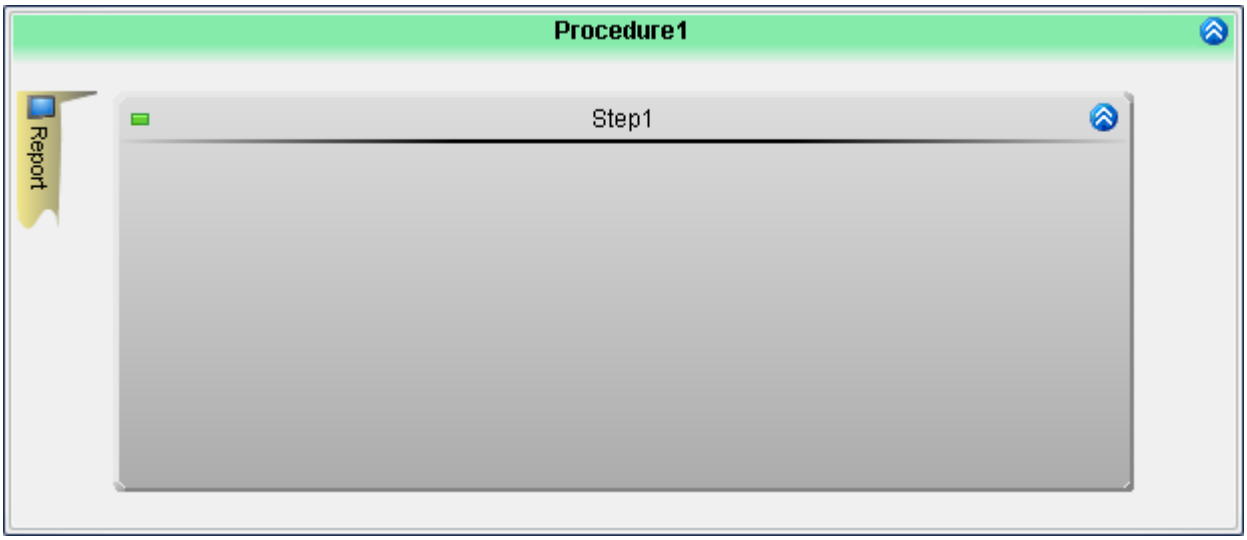
Source or data storage location	Syntax
Last concluded measurement	MEASUREMENT#LAST
Measurement number 3	Measurement#3
Current measurement	empty, so only ","
Fixed measurement name (as in the Data Browser)	2017-02-08 16-42-41 (1)

Examples:

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

9.7.1.7.3 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

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 *Page through Panel* is added, it is not linked with any Panel page.

For linking *Page through Panel* 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 *Page through Panel* 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

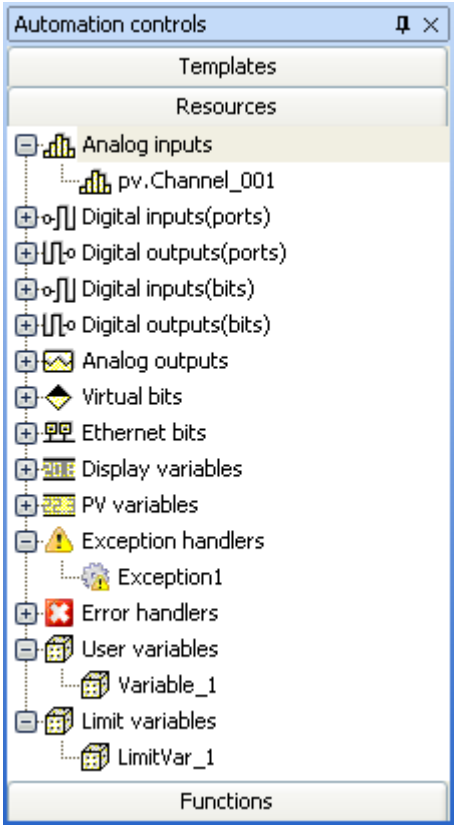
The description of the Panel pages is presented in the description of the plug-in Panel 813.

9.7.2 Repository

You can copy certain elements such as steps and procedures to the clipboard, or use ones from the clipboard. To do this, move the element to the desired position by means of Drag&Drop. When an element is imported from the Repository, the variables used must be confirmed or edited.

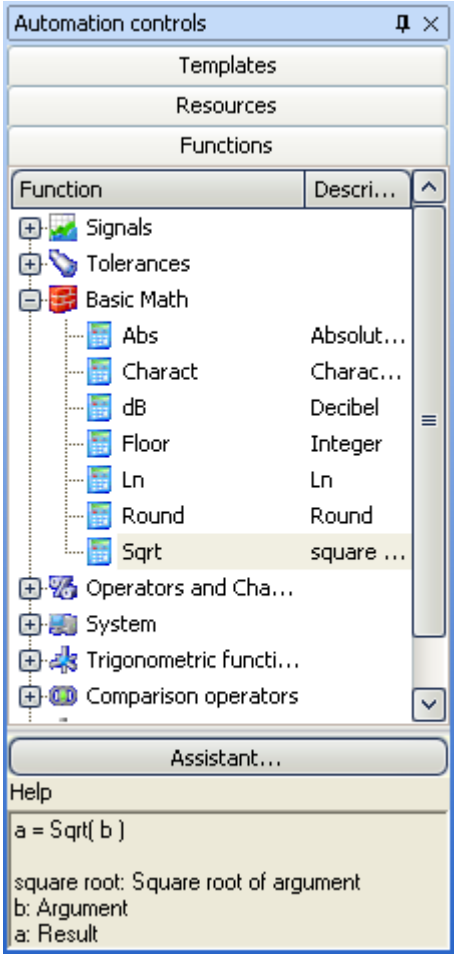
9.7.3 Resources

The *Resources* window displays all active channels. The channels can be displayed as either a tree diagram or a list. To add a channel to the active element either doubleclick on the channel or use Drag&Drop.



9.7.4 Functions

The Functions window consists of three parts and served as an aid in entering States, Loops and Branchings.

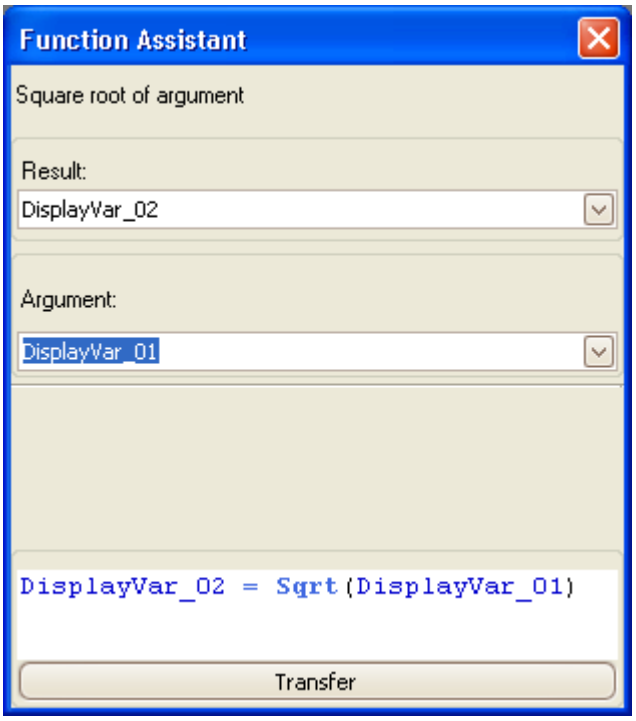


Functions

Here, all functions which can be used in an automation are listed. The functions can be displayed and sorted in either a tree diagram or a list view. By double-clicking on the mouse or using Drag & Drop, it is possible to place the commands into the active element (e.g. a State).

Function Assistant

In the Function Assistant, whole command lines can be created. To insert the command line to the active element. click on the *Transfer* button.

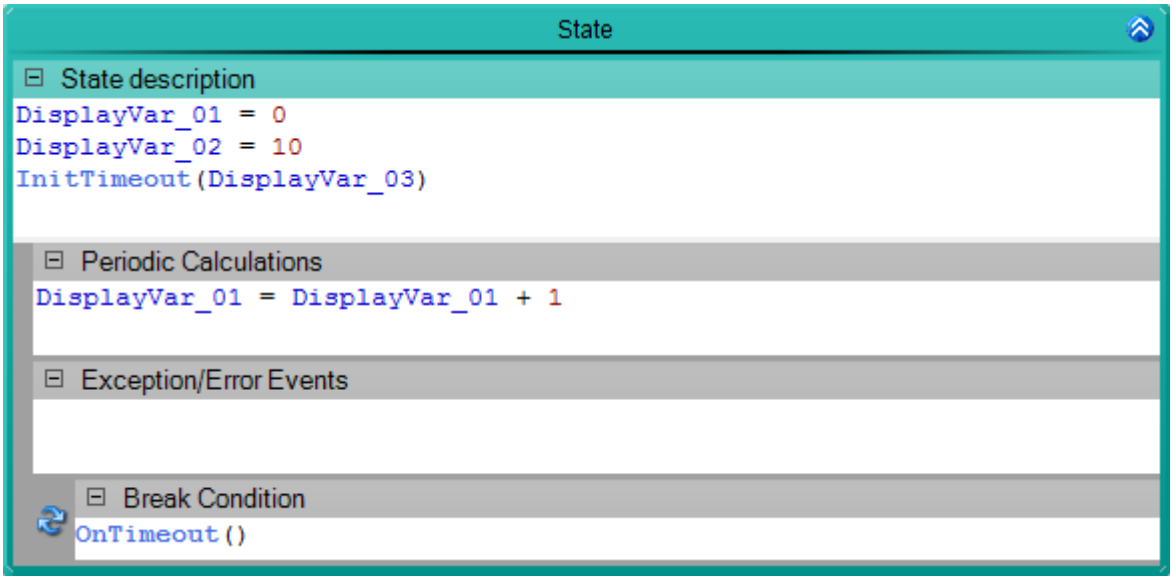


Help

The Help window displays a brief description of the selected command.

9.7.4.1 User Timeouts

The function *InitTimeout()* that can be used within the program for a state's initial actions. It accepts variable names as well as constant floating point values. The timeout interval is always measured in seconds. This new feature enables the user to adjust the timeout period even while it is elapsing.

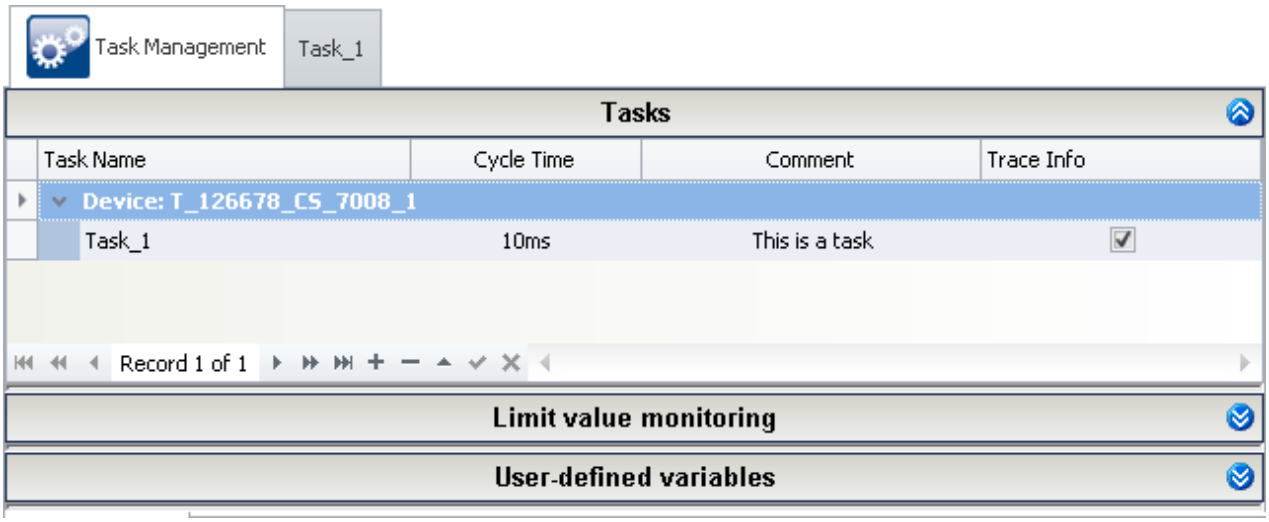


9.8 Automation Editor

Automation Editor contains three different views. All views are needed in order to program your tasks.

- *Task Management* 1140 — Here you define the tasks and their environment
- Task Editor 1149 — Here you create the outlines of your tasks' flow.
- Block Editor 1155 — Here you set the exact procedures of your tasks' individual *Steps* 1118.

Only after you have added a task you can open the Task Editor and the Block Editor.



Automation Editor with Task Management opened

In the image above, you see the *Task Management* with the task: "Task_1".

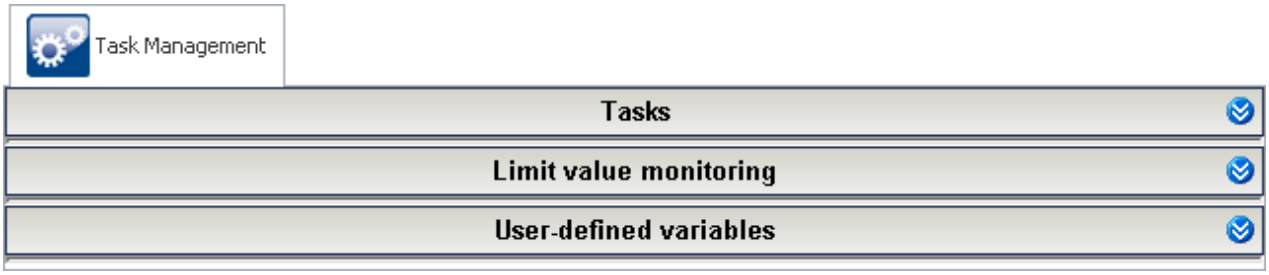
For each task, a new tab appears next to the *Task Management* .

Tab list

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.

9.8.1 Task Management



Task Management (Fields minimized)

Task Management consists of three fields:

Tasks 1141	Here you define the tasks.
Limit value monitoring 1143	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 1147	Here you define the variables for every task, which are used in the Task Editor 1149, among other places.

Each task has its own combination of *Limit value monitoring* and *User-defined variables*.

9.8.1.1 Tasks

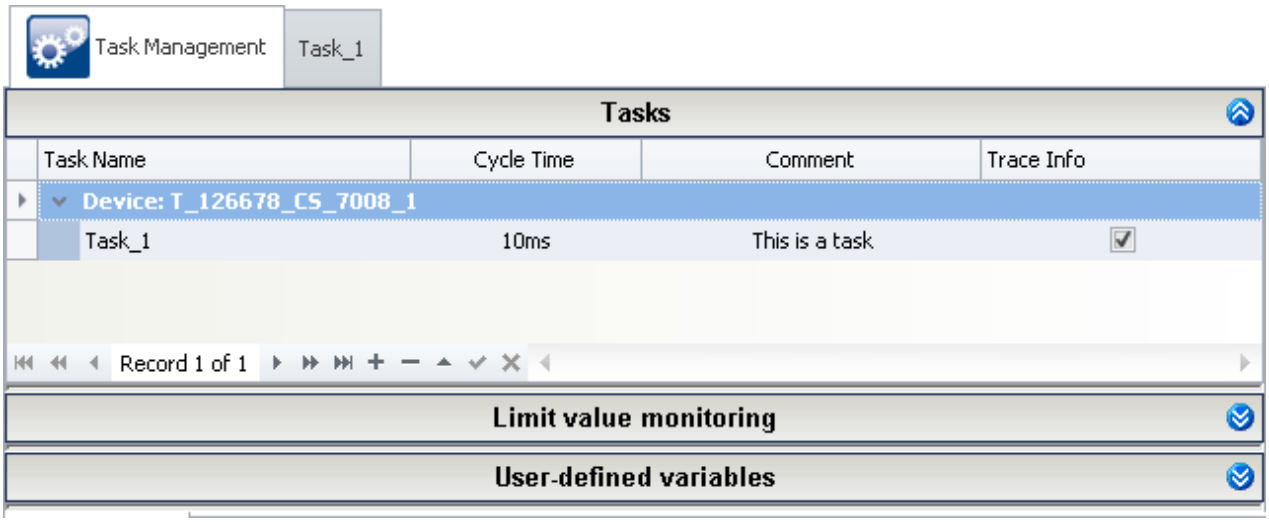
A Task is a complete routine of various commands performed in real-time. Functions to be performed in the task are defined exactly.

For this purpose, various elements are available to the task:

- Value limit monitoring — See chapter Task Management 1140
- User-defined variables
- Blocks — See chapter Task Editor 1140

For each device, up to five tasks (routines) 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

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 1113	Activation/deactivation of monitoring of the tasks' current status.

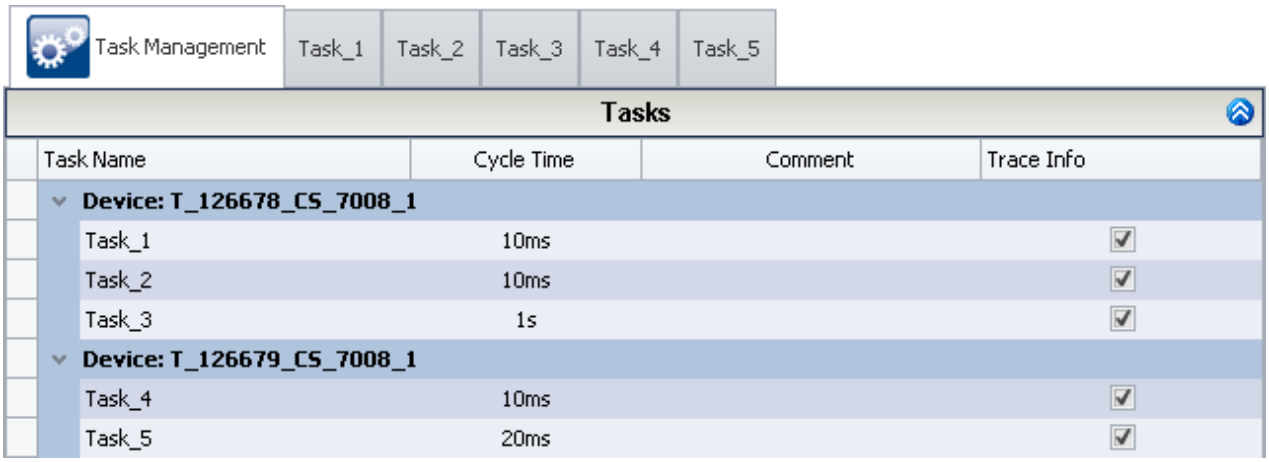
For each task, a new tab appears next to the *Task Management* 1140: (here) "Task_1". When you click on this, you come to the Task Editor 1149 with which you can create the flow of the selected task.

Tab-list

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 chapter State 1119.

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

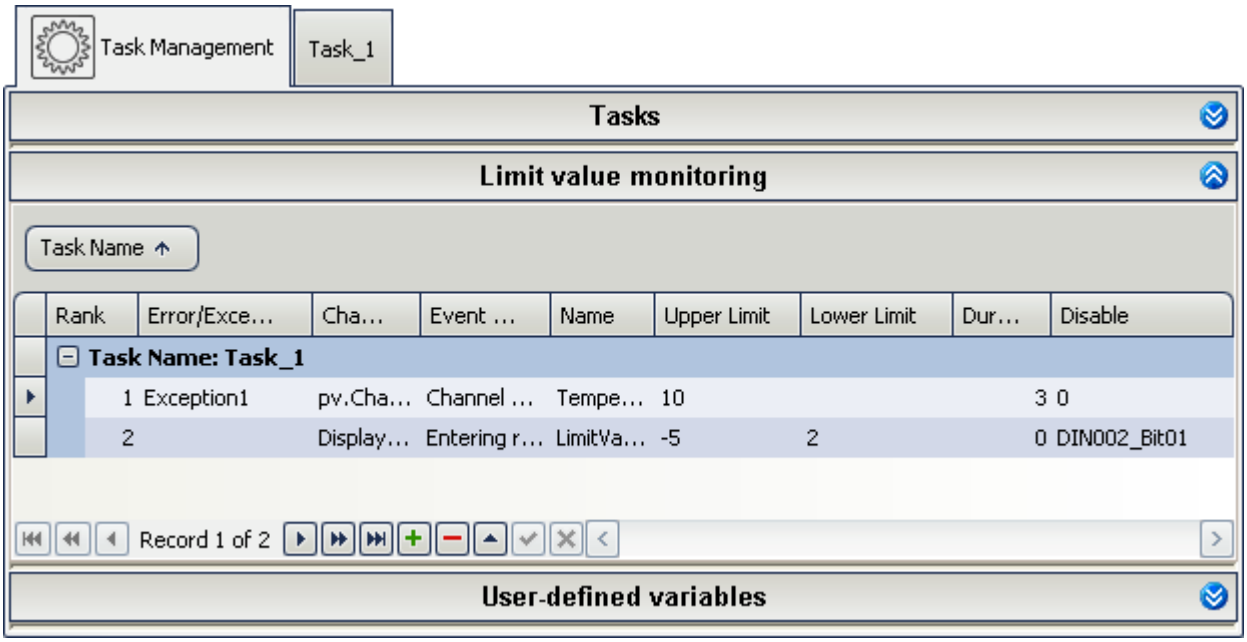
Trace Info

While the automation is running, you can observe which block and which *Step* 1118 are currently being processed. For more information, see the chapter Trace Info 1113.

9.8.1.2 Limit value monitoring

In the field Limit value monitoring you can define events. Events can be assigned directly to an Exception handler 1153 or an Error handler 1153. 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 chapter Exception and error handling 1153.



New limit value monitoring

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* 1153 or *error handler* 1153 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.
Disable	Here you can deactivate the Limit value monitoring.

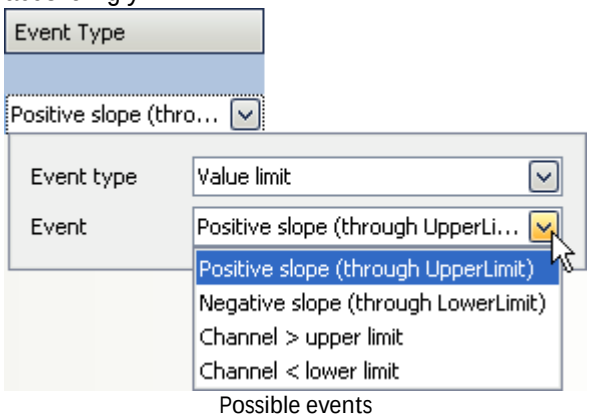
Events

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 handlings 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

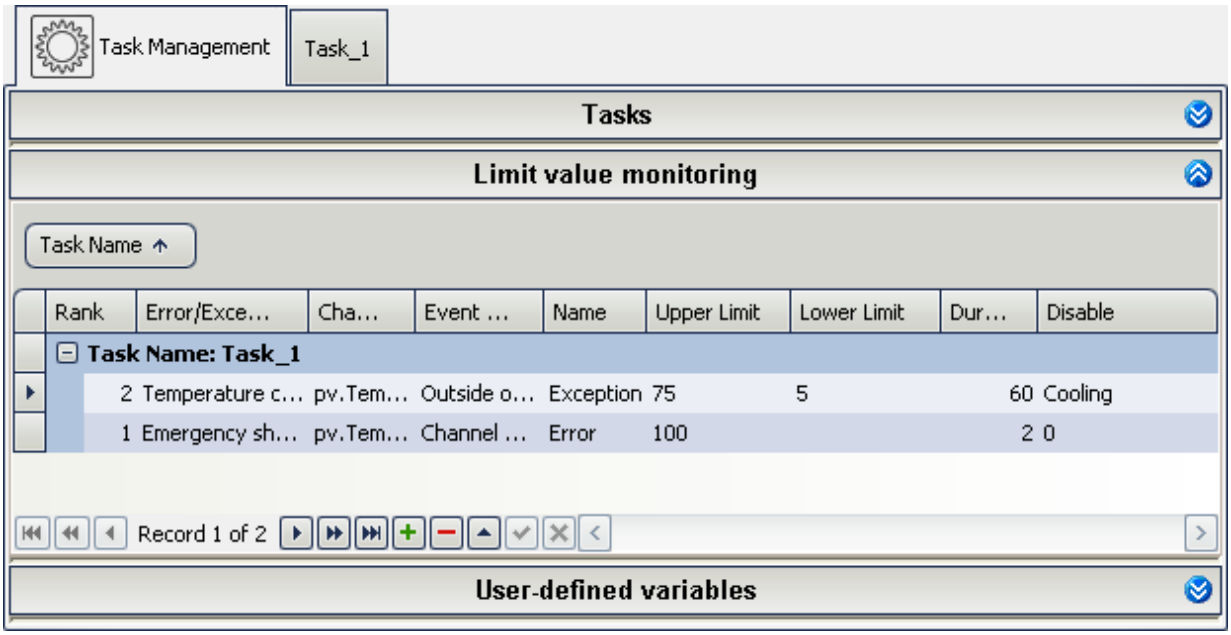
Event	Description
Entering range	If the monitored channel enters a range from either above or below, the condition is met.
Exiting range	If the monitored channel exits a range from either above or below, the condition is met.
Inside of range	As long as the monitored channel is within the range, the condition 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 1147).



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 1147).

- 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

The conditions

If all of a value limit monitoring's conditions are met

- its variable is set to 1 and
- a handling action is started if one was selected.

Note

- If no handling action 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 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.

Examples

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

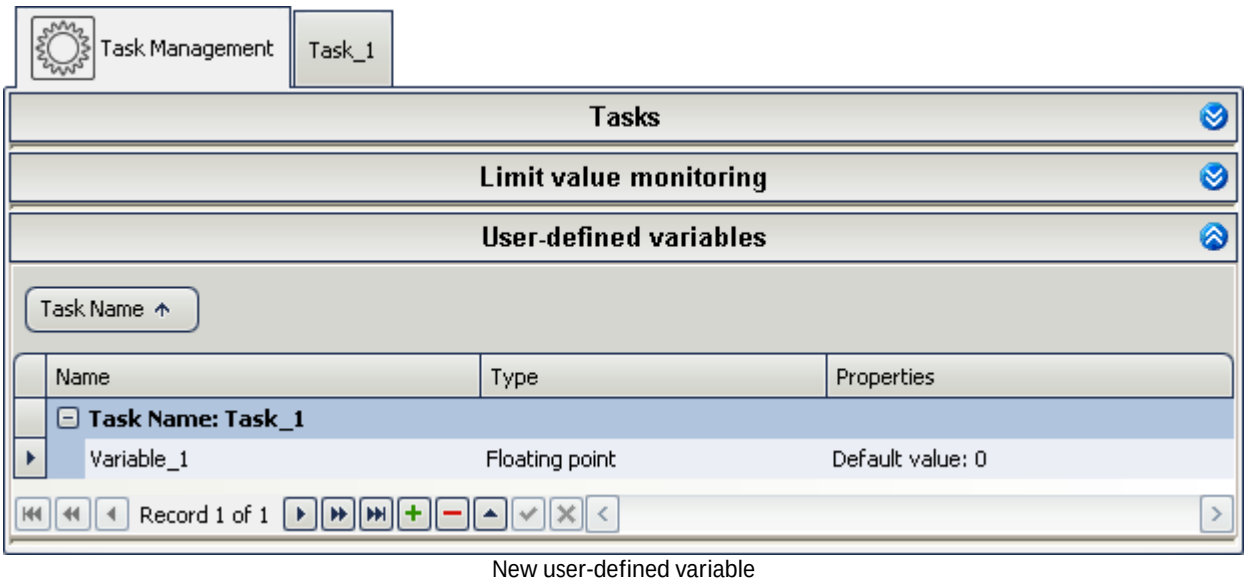
Note: Occurrence of the value limit monitoring's event while the Terminate block is in process

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 than the exception handling and is carried out the conclusion.

9.8.1.3 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	Variable-type
Default Value	Here you can assign an initial value to the variables. The recorded value is written to the variable upon reconfiguration 93 (or when the configuration downloading is performed). 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".
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.

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.

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 download.
  When the experiment is first downloaded once the device has started, the value is 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:

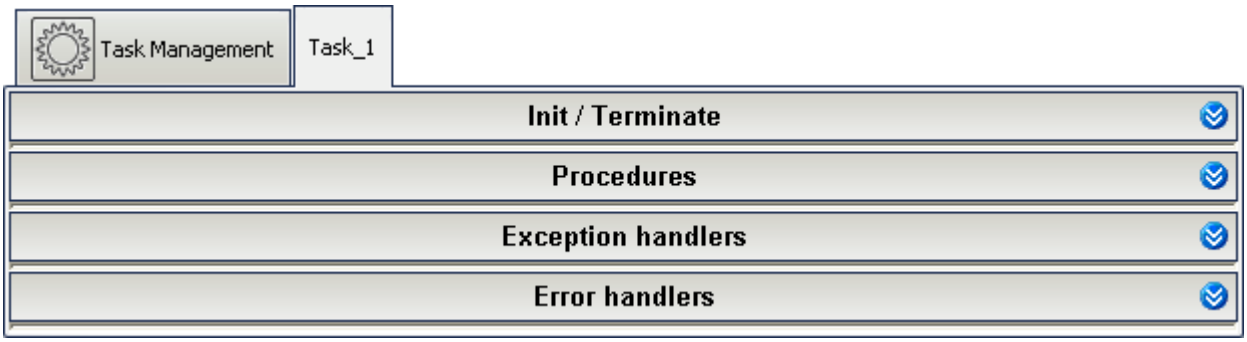
<table>
<tr><th colspan="3">Parameter</th><th colspan="3">Action</th></tr>
<tr><th>Persistent</th><th>Process-vector variable</th><th>Default value</th><th>Reconfiguration</th><th>Measurement start (after reconfiguration)</th><th>Switch device off and back on
Start measurement</th></tr>
<tr><td>no</td><td>no</td><td>5</td><td>5</td><td rowspan="7">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 Deleting saved values on the Setup page: "Devices". You may need to first show the column 133.

When the function is selected, all persistent values are deleted upon the next start and the device is prepared with the "Default values".

9.8.2 Task Editor



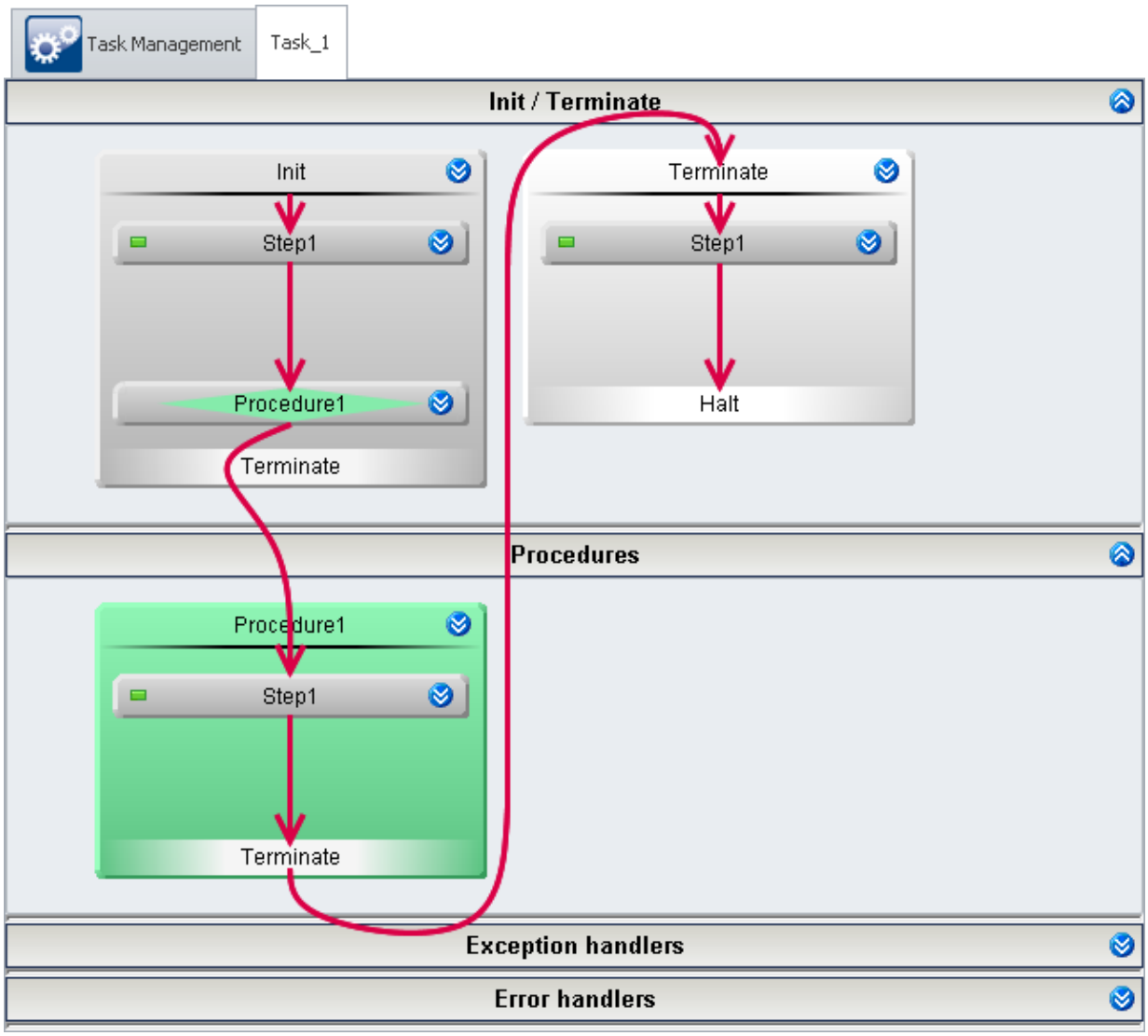
Task Editor

The Task Editor is displayed in the Automation Editor 1140 and in normal cases consists of four fields:

- *Init / Terminate* 1151 — Here, the beginning and end of the Task are defined.
- *Procedures* 1152 — Here is where the actual flow is created.
- *Exception handlers* 1153 — Here, possible exceptions are defined in order to be able to remedy them.
- *Error handlers* 1153 — Here, processes are created which enable controlled closure of the test in case of major errors.

Blocks

Each of these fields contains individual blocks. A task consists of at least three blocks: an *Init* 1151, a *Procedure* 1152 and a *Terminate* 1151. A block comprises multiple steps, which in turn are collections of multiple other elements.



Minimaler Task without function

You can add any amount of blocks consisting of *procedures* 1152, *exception handlers* 1153 and *error handlers* 1153.

Execution

- After the task starts, initialization is called. You specify the subsequent process.
- 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*.

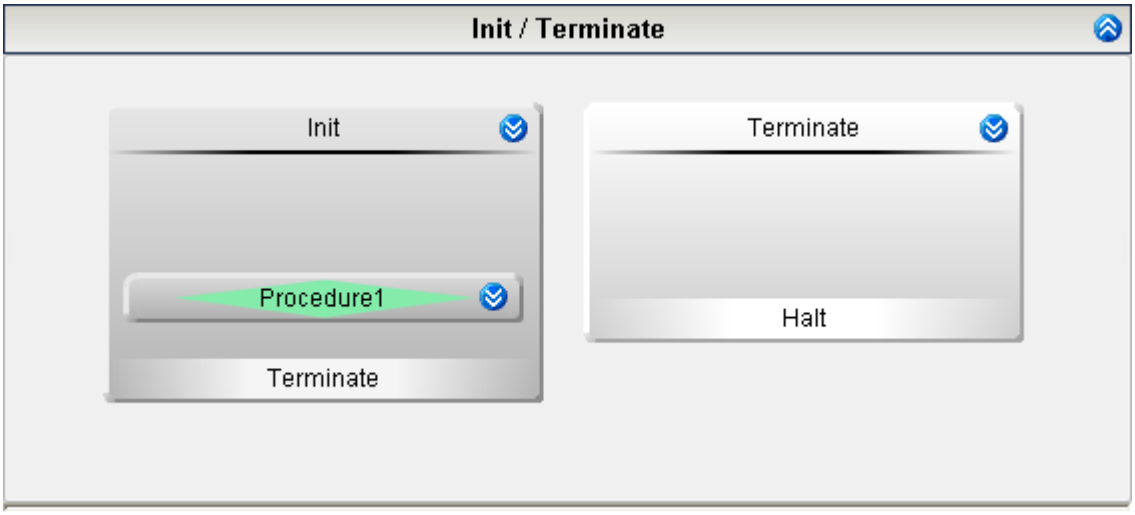
If the measurement concludes with the stop button ( ), then the block *Terminate* is called automatically, as long as no handling is running.

The End-branch is not an obligatory *branch*. You can insert a *branch* beforehand to any desired *procedure* (not in Terminate or Error Handling).

9.8.2.1 Init / Terminate

The field *Init / Terminate* 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

Init

With the help of the Init, the measurement system is placed in a defined initial state. For this purpose, you are provided with

- Steps and
- Next-branchings.

End of the Init

At the end of the Init, there are two possibilities:

- *Next*-branching: The task skips to the procedure defined in the branch.
- No branching: the Terminate block is started.

For more information on the branchings, see the chapter *Branching (Task Editor)* 1116.

By default, the Init contains a *Next*-branch. From here the Task, skips to the first *procedure*.

Terminate

Terminate brings the measurement system to a defined conclusive state. For this purpose, you are provided with

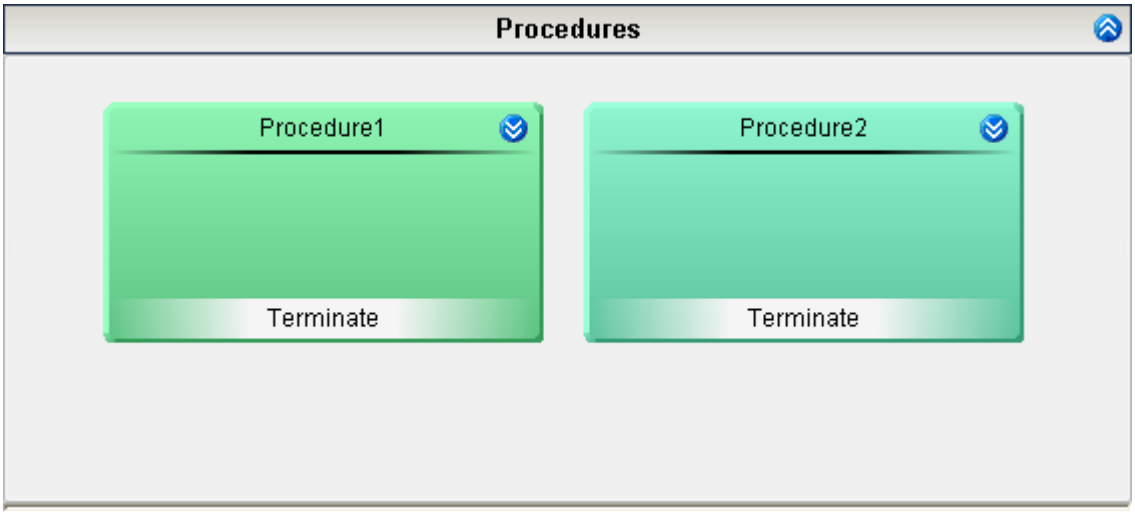
- Steps.

Halt

At the end of Terminate, the task is concluded with *Halt*. Skipping back to a procedure is not possible.

9.8.2.2 Procedures

The field Procedures contains the actual algorithm.



Procedures

The procedures are used to create the outlines of the algorithm. For this purpose you are provided with

- Steps and
- Next-branchings.

End of a procedure

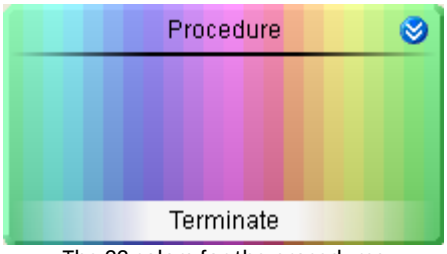
At the end of a procedure there are two possibilities:

- *Next*-branch: the Task skips to the procedure specified in the branch.
- No branch: the Terminate block is started.

For more information on the branchings, see the chapter *Branching (Task Editor)* 1116.

Colors

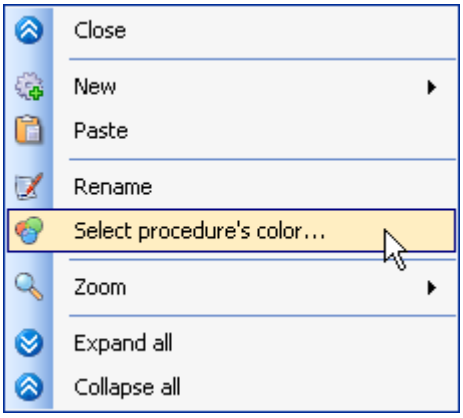
Each procedure is distinguished by its name and a *color*. When a new procedure is created, it is assigned a new color. There are 22 pre-defined colors.



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...*, as shown below.



Subsequently, the color selection window appears.

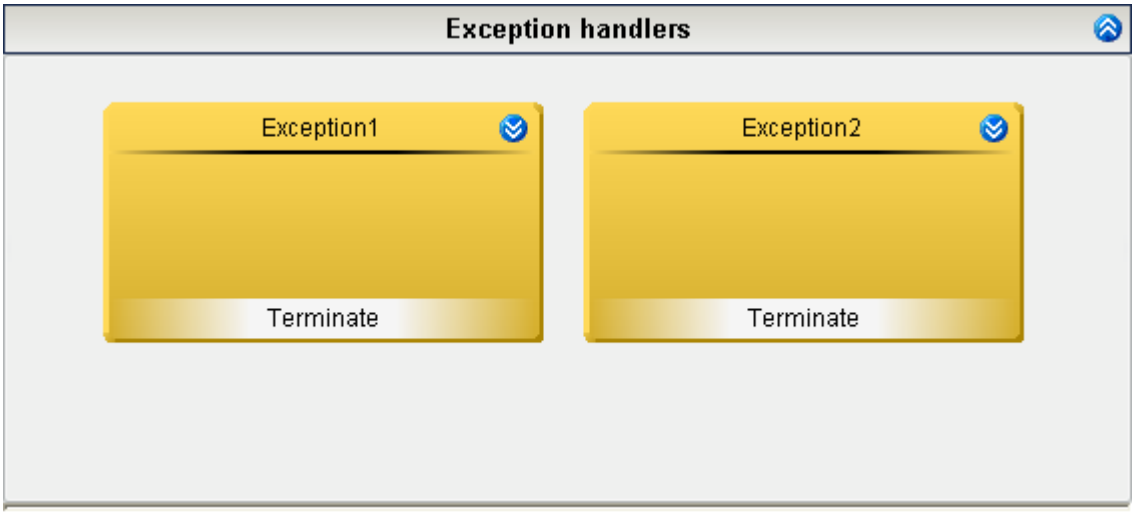
- Select the desired color and click on the *OK*-button.

9.8.2.3 Exception and error handling

Handlers are blocks for treating any disturbances or errors. If a disturbance occurs, the current *procedure* is stopped and the handler is run.

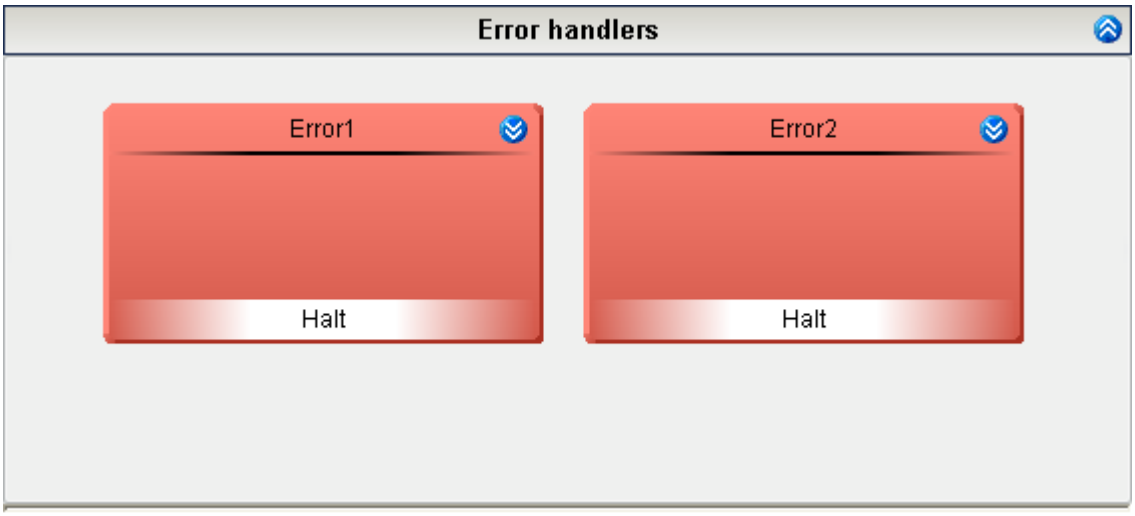
There are two types of handlers:

Exception handler



Exception handlers

Error handler



Error handlers

A minor error can be remedied in a controlled manner using an exception handler. In the case of major errors, you can end the running task in a controlled manner with an error handler. For this purpose, the following elements were available.

- Steps and
- for the Exception handler additional Resume- and Go to-branchings.

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 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.
- For more information on defining the limit monitoring, see the chapter *Limit value monitoring* 1143.

Defined in a *State*. This only applies to the *state*.

- The handling routine is started if a disturbance occurs during the state.
- For more information on this definition of the exception and error handler, see the chapter State 1119.

The procedure running is interrupted in such a case.

End of a handling routine

At the end of an error handler, the *Halt*-state is set and the task is concluded.

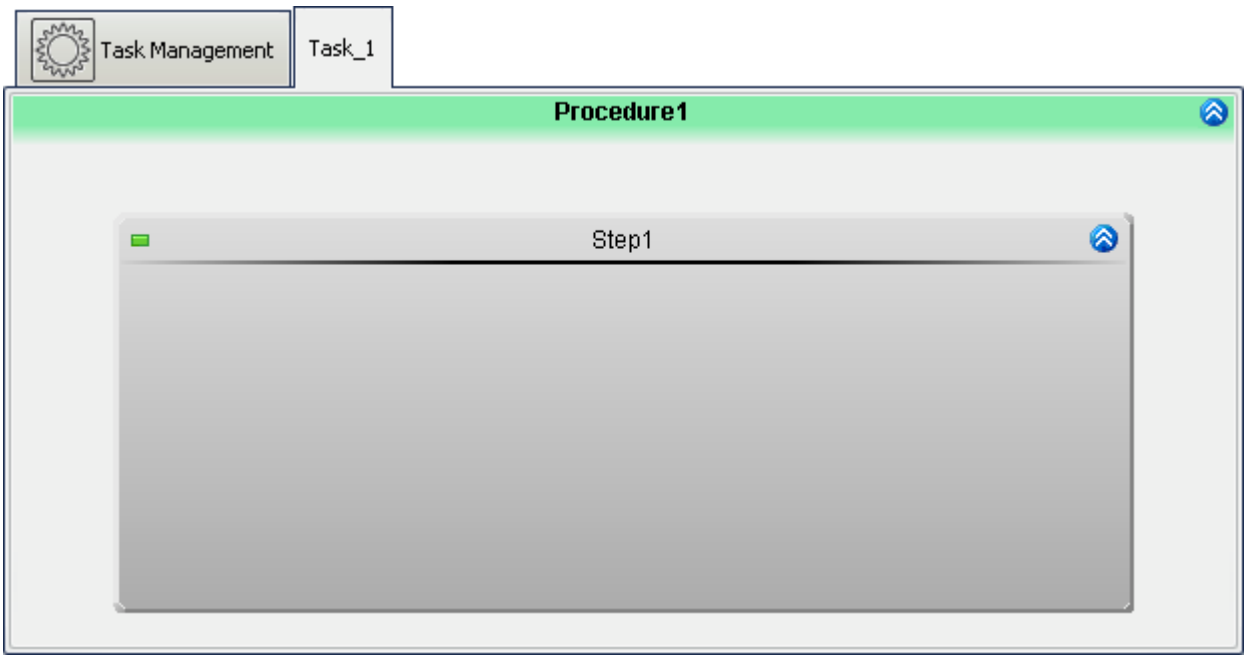
At the end of an exception handler routine there are multiple possibilities:

- *Resume*-branching: the Task resumes from the point at which the first exception handler was started. The *state* is restored.
- *Go to*-branching: the Task skips to the step specified in the branch.
- No branch: the Terminate block is started.

For more information on the branchings, see the chapter *Branching (Task Editor)* 1116.

9.8.3 Block Editor

The Block Editor is displayed in the Automation Editor 1140. It amounts to a close-up of a block. Here, the exact behavior of the block's individual *steps* 1118.



Block Editor with a procedure step created

In the Block Editor, all of the opened block's *steps* 1118 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:

- States 1119
- Loops 1121
- If-branches 1123

Interaction between the devices and the PC can be set up by means of the following elements:

- General Event 1124
- imc FAMOS Automation (data cutting) 1127
- Page through Panel 1136

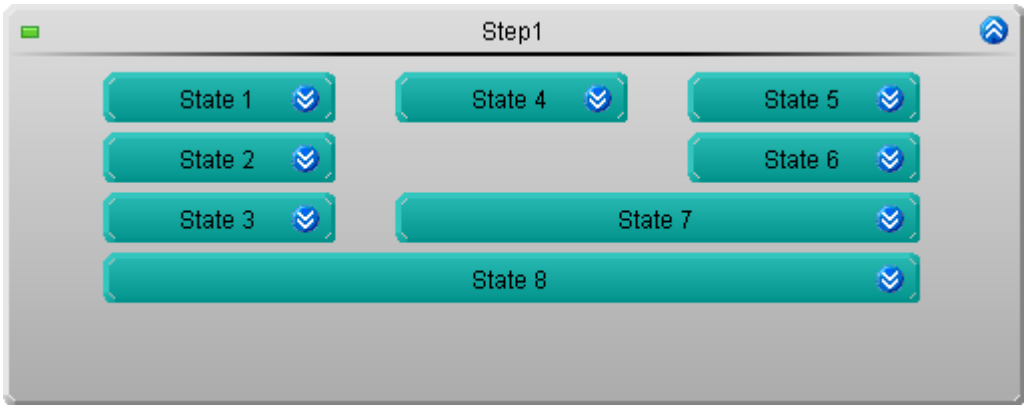
Execution

- 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.

More information on execution of the block is presented in the chapter: How does it work? 1105, or, in reference to the individual elements, in the chapter: Templates 1113.

Example of a procedure

9.8.3.1 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.

Note

This also works with loops and branchings.

Descriptions of how to operate the elements are provided in the chapter: Operation in the Task and Block Editor 1108

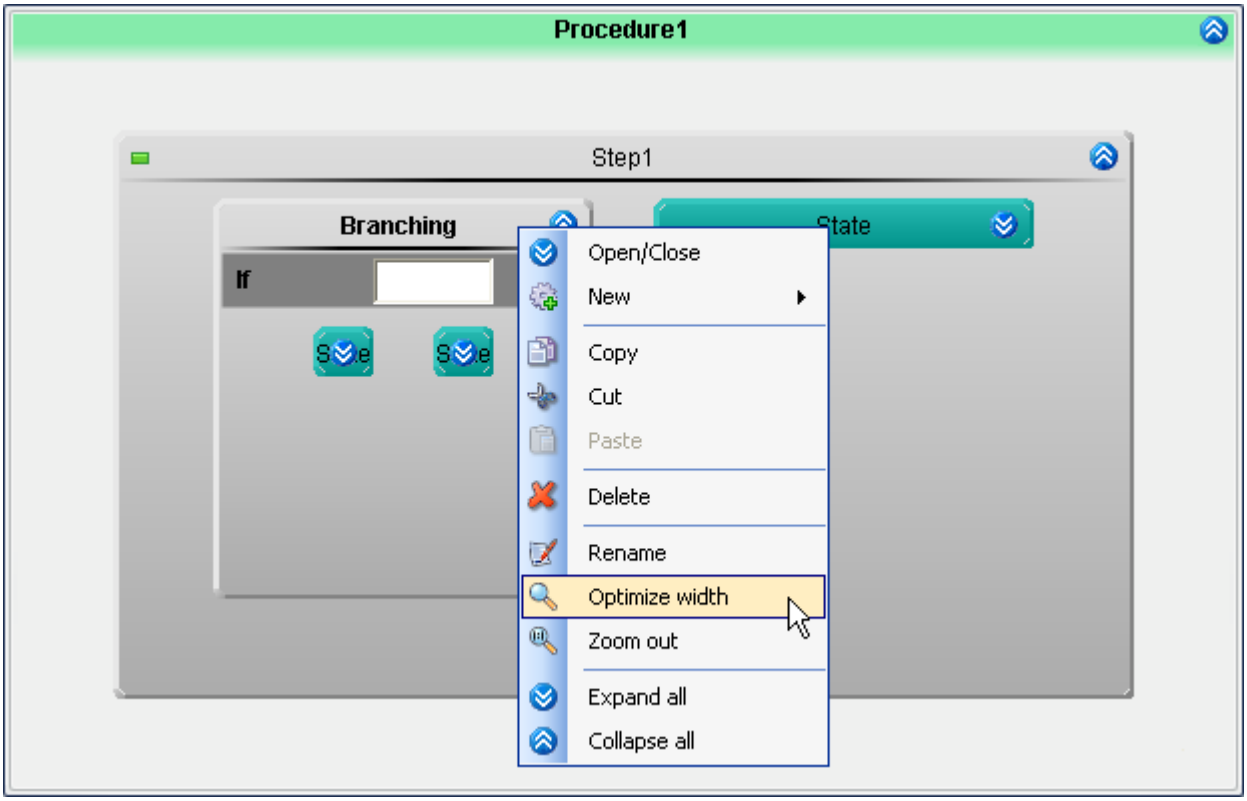
9.8.3.2 Zoom - Adapting an element's size

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.

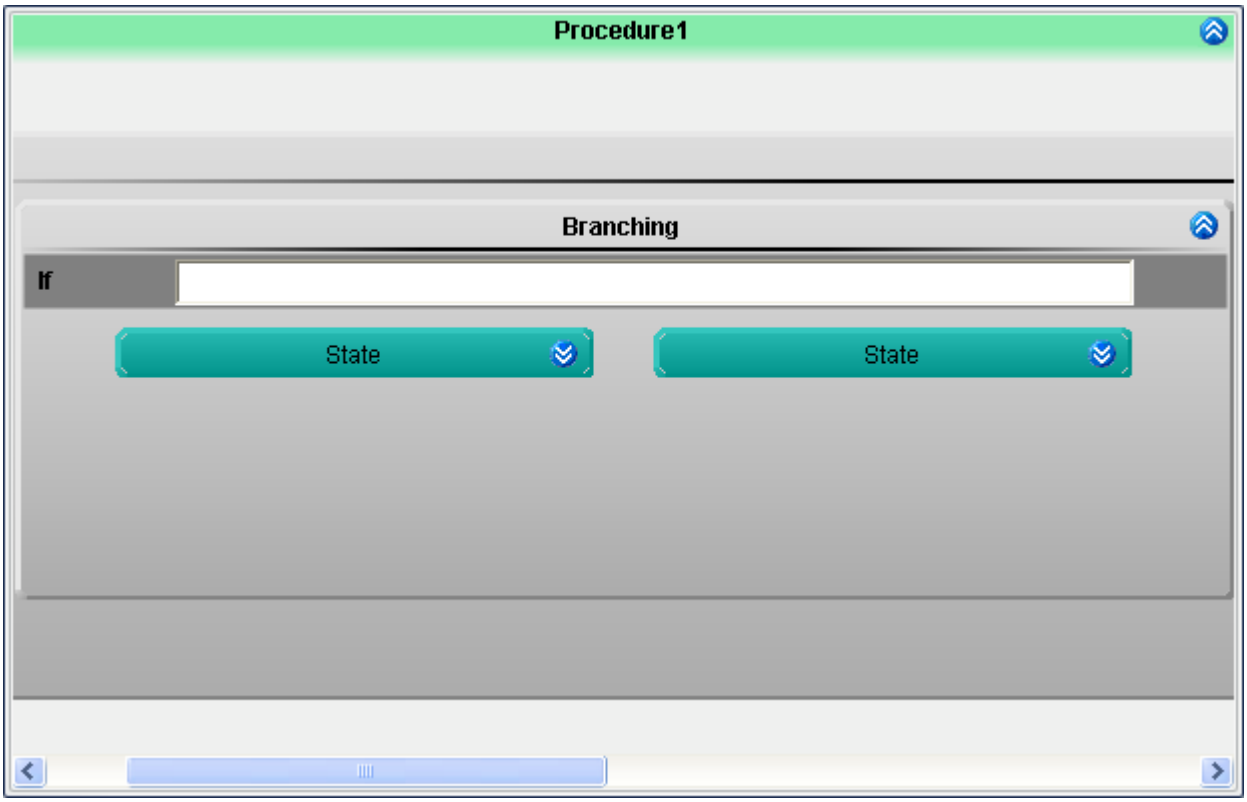
For this reason, in order to keep a clear overview, it helps to assign descriptive names to the nested loops, which reflect what they do.

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*

9.9 Tutorial

Below we present some examples of the plug-in Automation. The examples assume that the plug-ins Setup 87 and Panel 813 are installed.

A prerequisite for operation of the plug-in Automation is an imc measurement device with imc Online FAMOS Professional installed.

9.9.1 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 39.

9.9.1.1 Setup - Settings

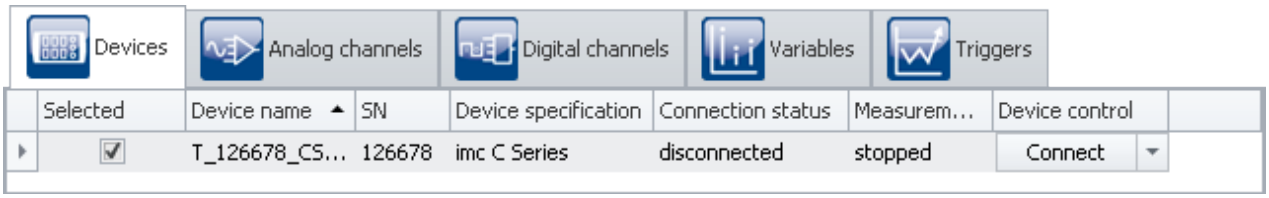
To begin, the measurement device is selected and configured.

- Open the plug-in 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, then perform a device search 139 ( ) (Ribbon *Home* (or *Setup-Control*) > *Search for Devices*). Subsequently, the list will display all devices found.
- Click on the Checkbox-symbol ( ) in order to select the device.



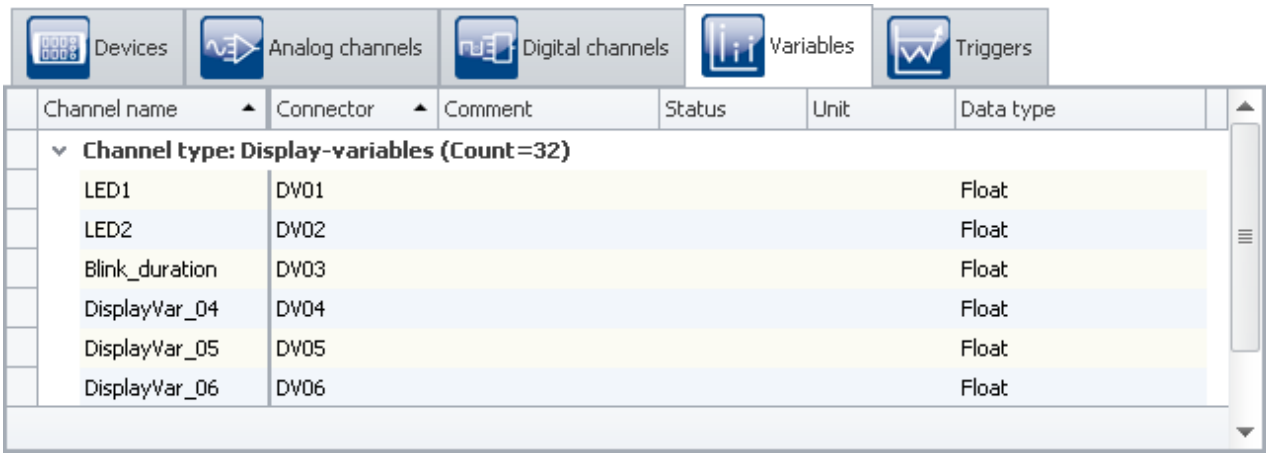
Selecting a device

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 139 ( ) (Ribbon *Home* (or *Setup-Control*) > *Process*).

By this means, the device settings are available to other plug-ins.

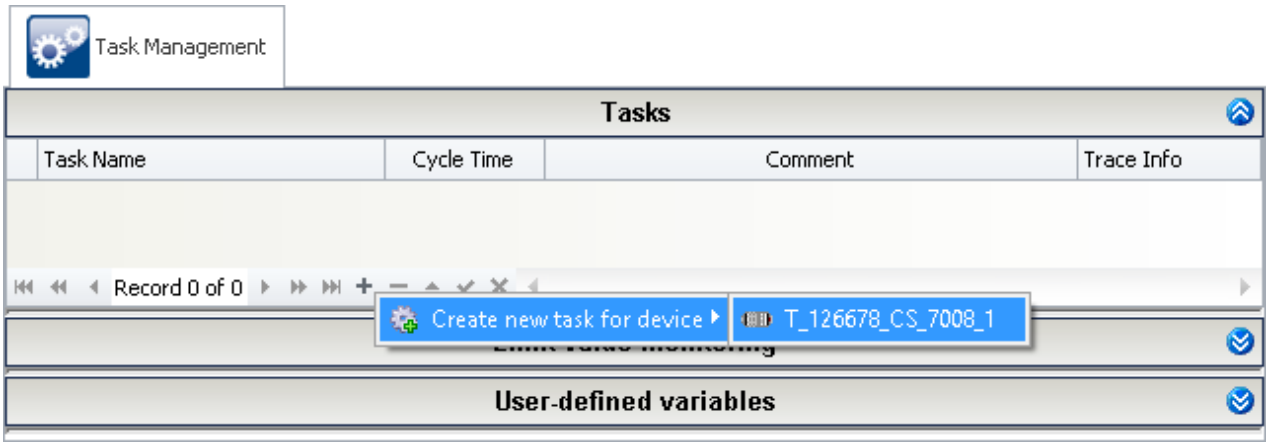
9.9.1.2 Automation - Settings

Now it is possible to set up the Task.

- Open the plug-in Automation.

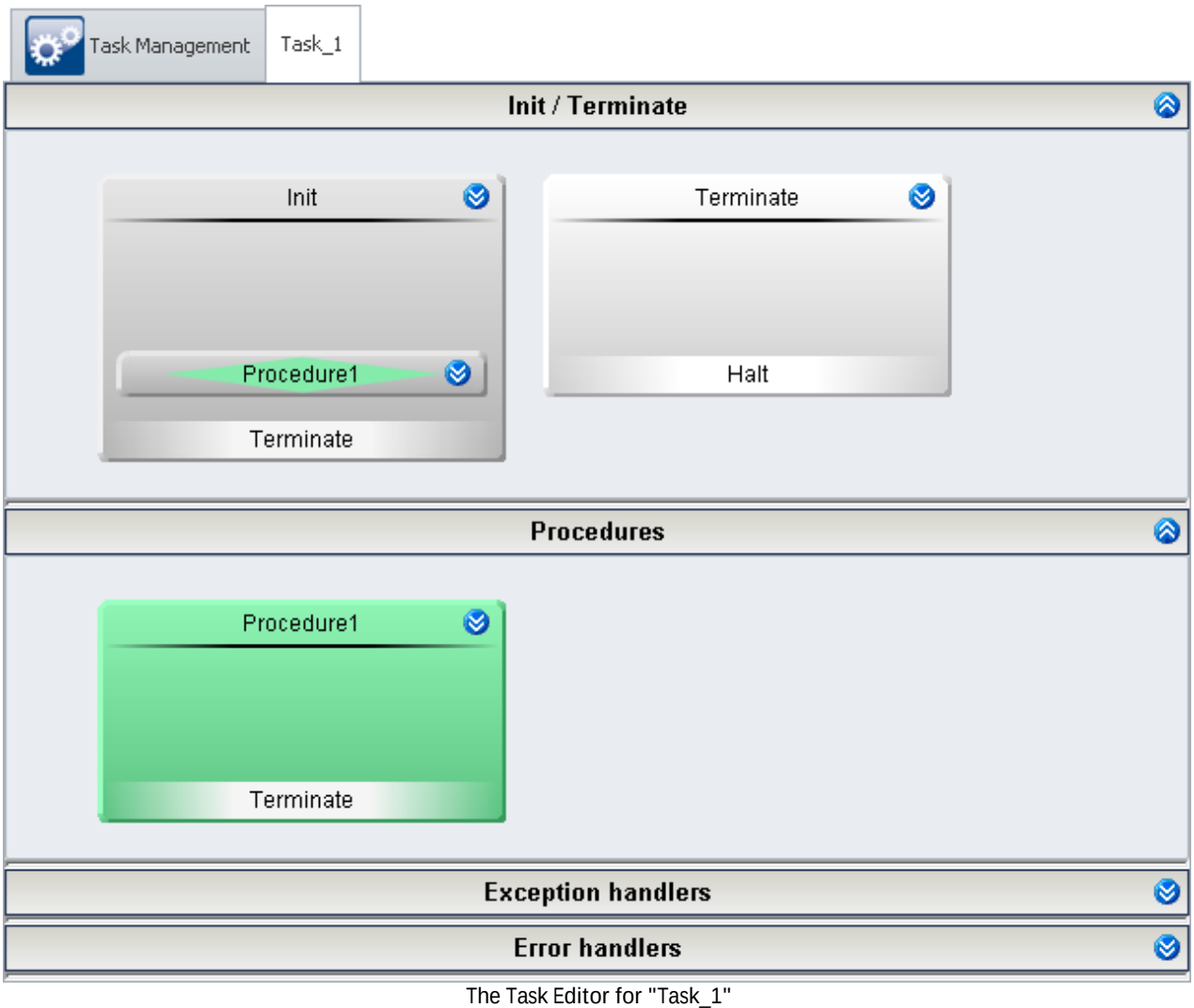
You are now in the Task Management interface. Proceed with creating a Task.

- 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*.

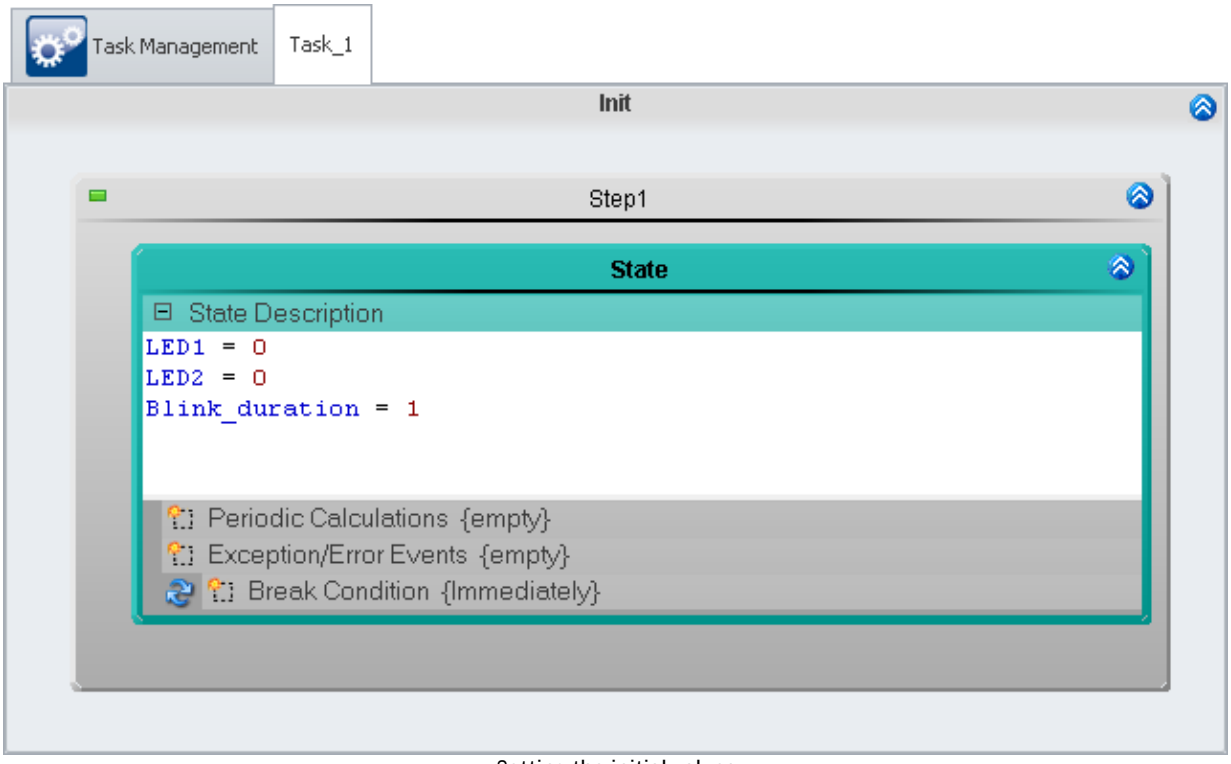
Open this Step by clicking on the double arrow ( ). You arrive at the Block Editor.

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.

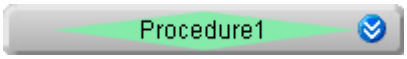
```
LED1 = 0
LED2 = 0
Blink_duration = 1
```



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.

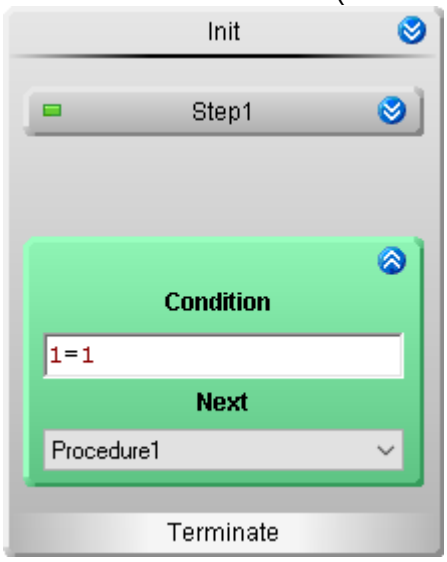


Branching to Procedure1

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).



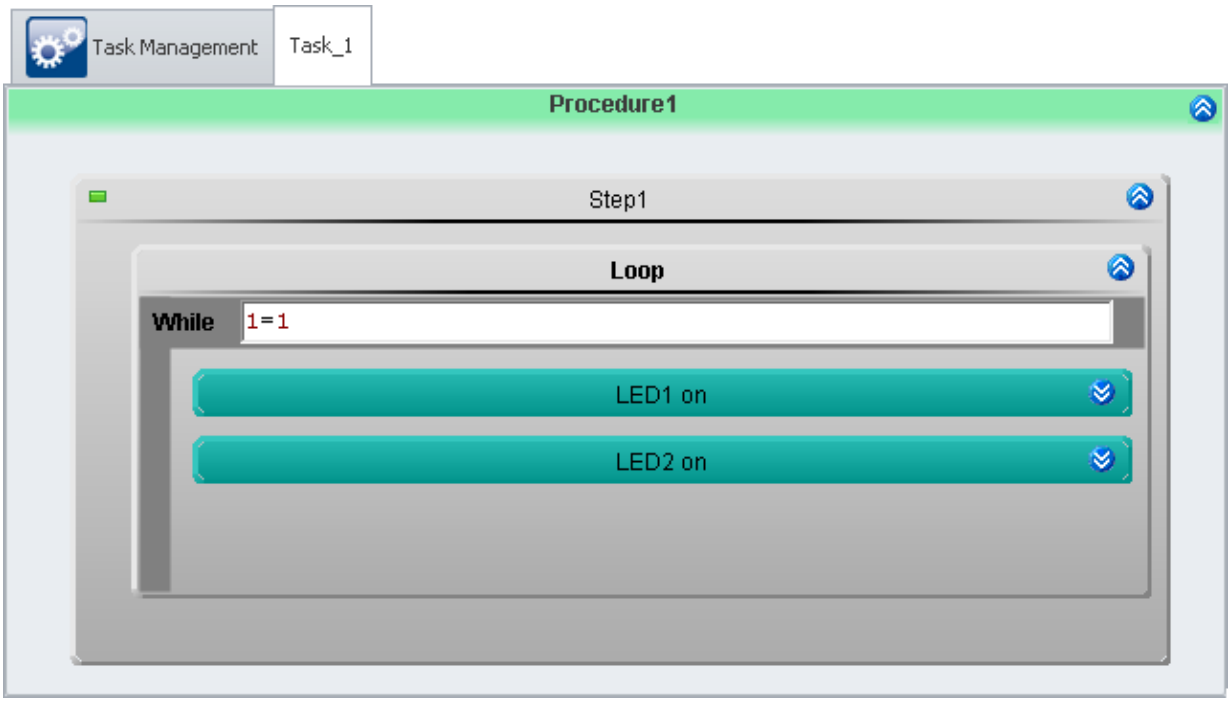
Skip from block "Init"
to the block "Procedure1"

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:

```
1 = 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.

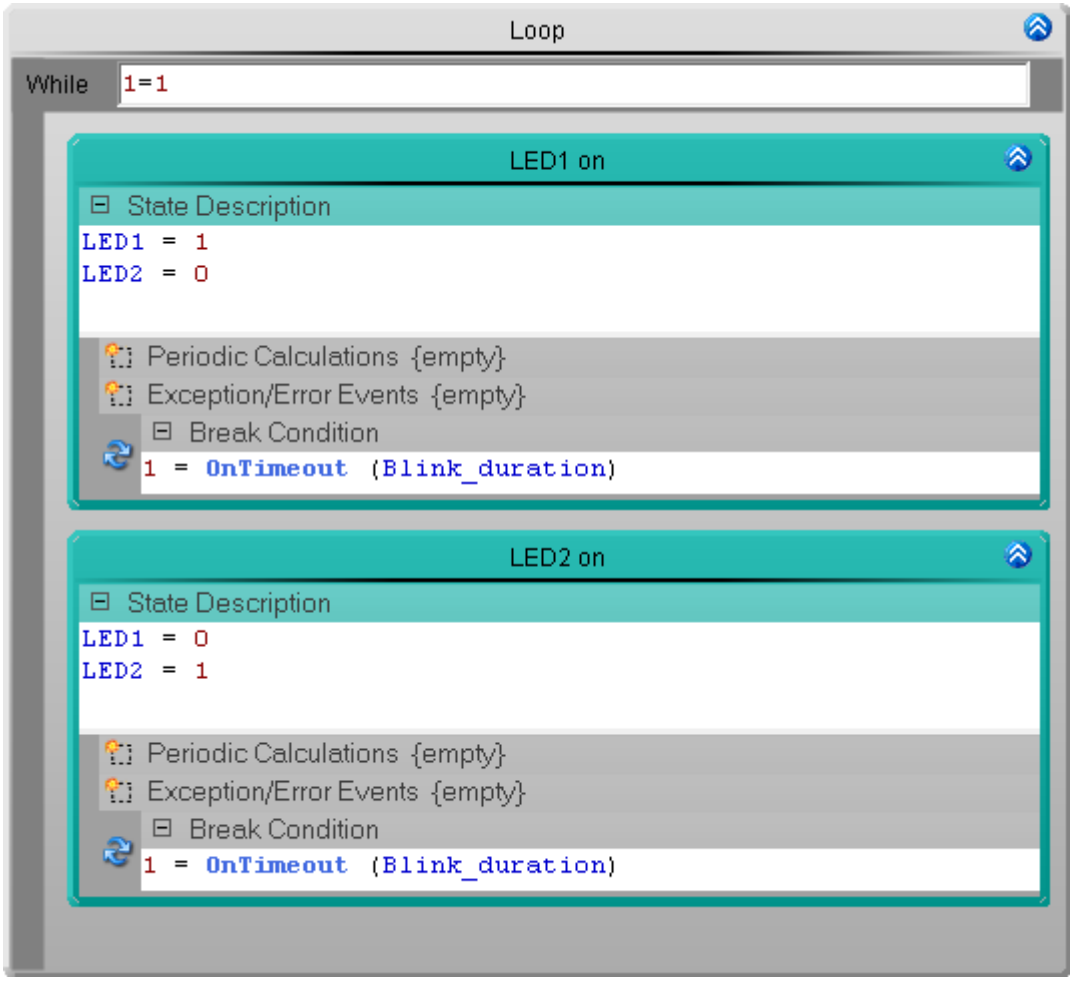
The State should be exited once the *Blink duration* set has elapsed.

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*.

```
1 = 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).

Once the entire Task has been set up, the syntax of the program should be checked.

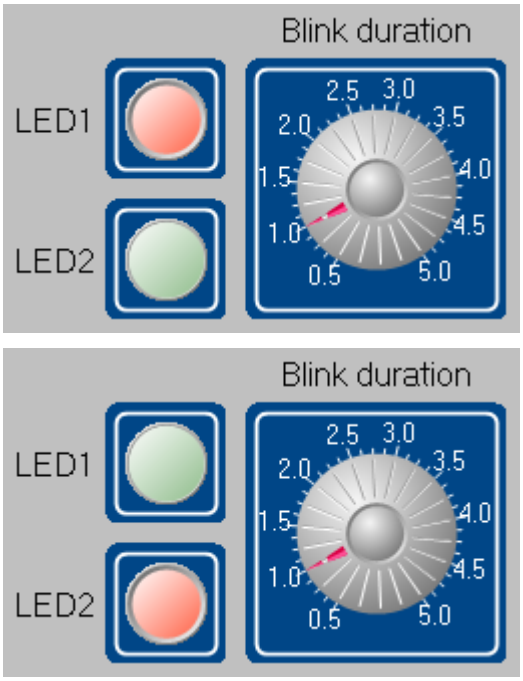
- Check and apply the changes (✔) (Ribbon *Home* (or *Setup-Control*) > *Process*).

9.9.1.3 Panel - Settings / Start measurement

Now the Panel page can be set up.

- Open the plug-in 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 139 ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Start*).

Test the blinking behavior for various settings of the *Blink duration*.



Panel- Blinking LED using timer (example)

9.9.2 Blinking LED using Cycle Time

Assignment:

Set up the flashing LED from the previous example Blinking LED using timer 1159 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.

9.9.2.1 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 1159.

9.9.2.2 Automation - Settings

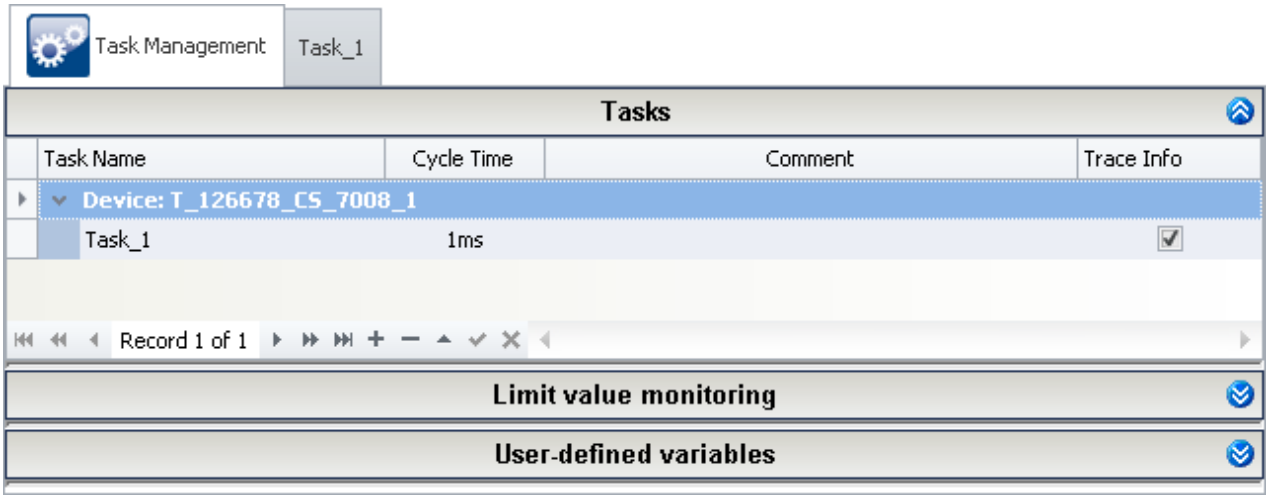
Now it is possible to set up the Task.

- Open the plug-in 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.

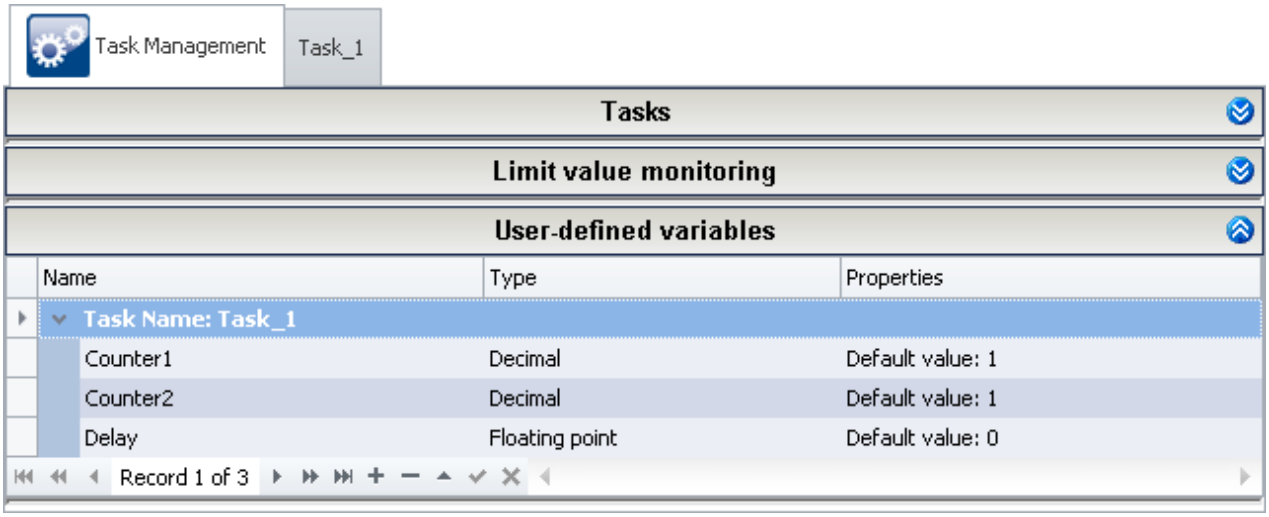
- Enter 1ms as the Cycle Time.



Task with a 1 ms Cycle Time

- Create three User-defined variables.

Name	Type	Properties
Counter1	Decimal	Default value: 1
Counter2	Decimal	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 ( ).

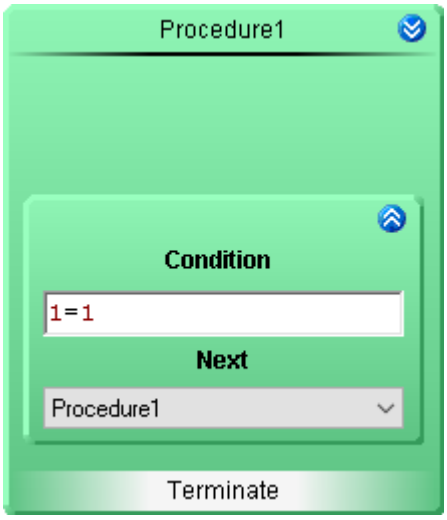
- Enter `1=1` for the condition.

Once the initialization has been run, *Procedure1* is performed automatically.

Procedure

In the example Blinking LED using timer 1159, a continual repetition is carried out by means of a While-loop. This can also be achieved by a Next-branch.

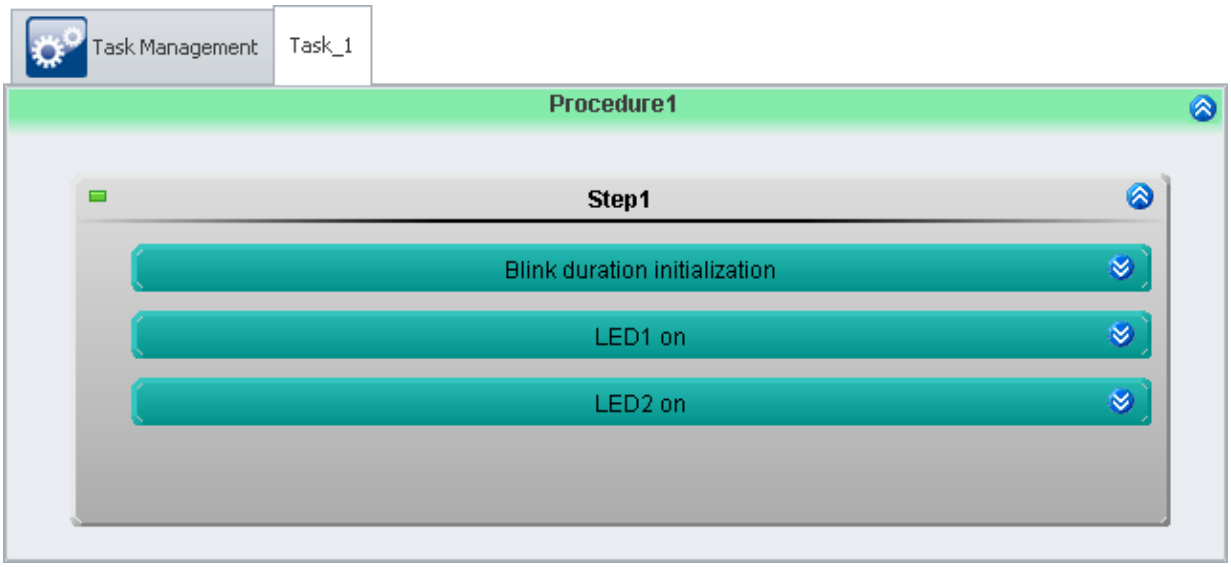
- Add a Next-branch into *Procedure1.*
- Open the Next-branch
- Set the condition to `1=1`
- The procedure is is own target (Next).



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
```

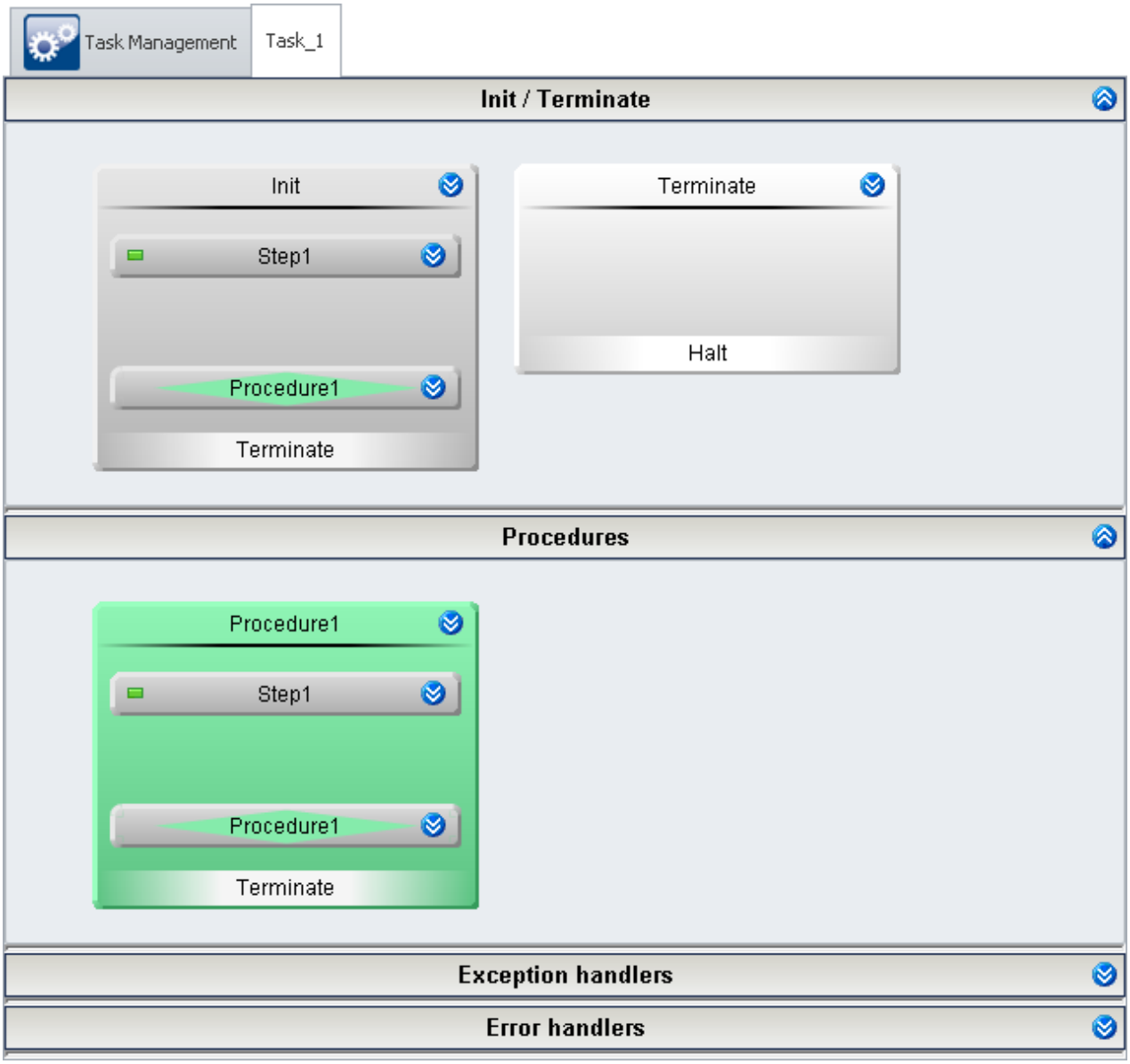
Exception/Error Events {empty}

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.



Once the entire Task has been set up, the syntax of the program should be checked.

- Check and apply the changes (✔) (Ribbon *Home* (or *Setup-Control*) > *Process*).

9.9.2.3 Panel - Settings / Start measurement

Now the Panel page can be set up.

- Open the plug-in Panel.

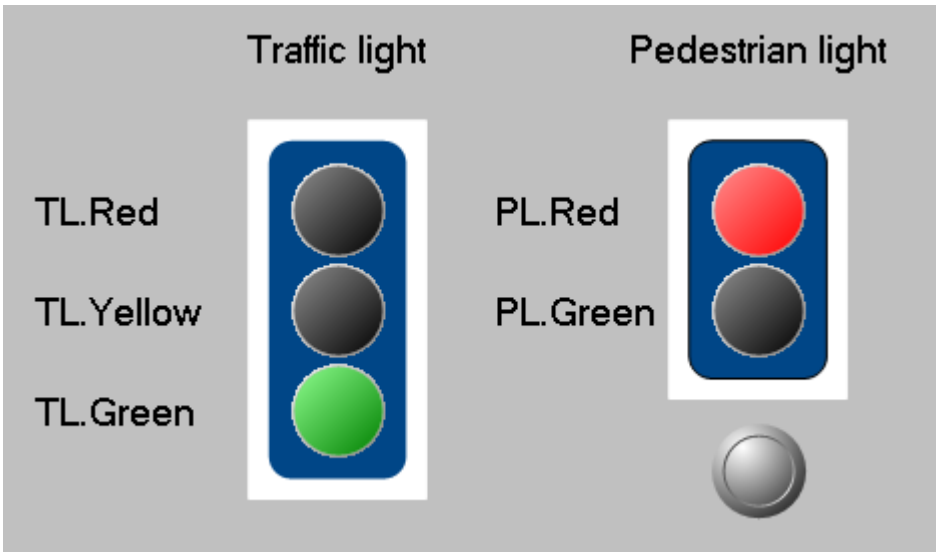
Proceed as described in the previous example Blinking LED using timer 1159.

9.9.3 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.

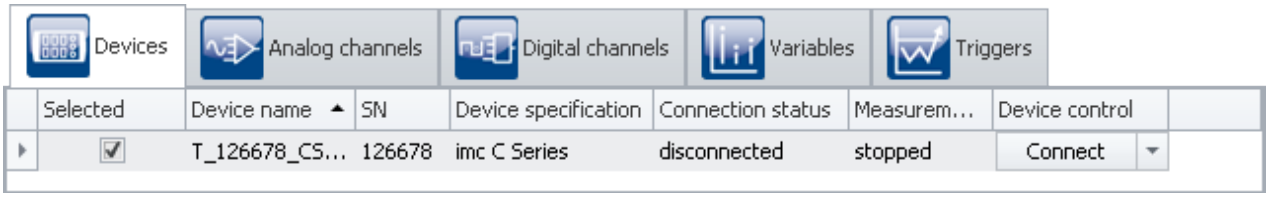
9.9.3.1 Setup - Settings

To begin, the measurement device is selected and configured.

- Open the plug-in Setup.
- Go to the tab: Devices .

Select your device:

- If the device desired does not appear in the list, then perform a device search 139 ( ) (Ribbon *Home* (or *Setup-Control*) > *Search for Devices*). Subsequently, the list will display all devices found.
- Click on the Checkbox-symbol ( ) in order to select the device.



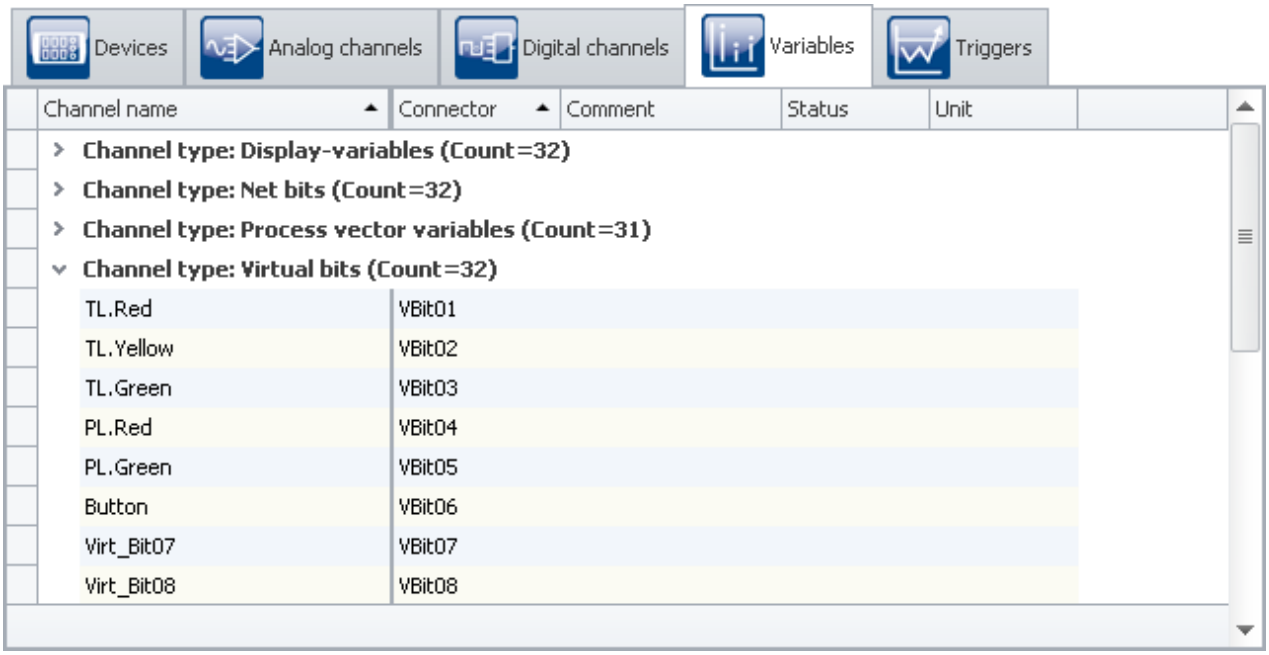
Selecting a device

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 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 139 (✔) (Ribbon *Home* (or *Setup-Control*) > *Process*).

By this means, the device settings are available to other plug-ins.

9.9.3.2 Automation - Settings

Now it is possible to set up the Task.

- Open the plug-in 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.

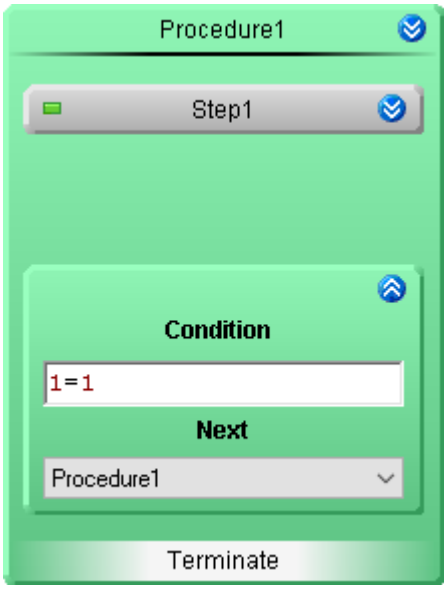
Open the existing Next-branch by clicking on the double-arrow ().

- 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 Break 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

```
1 = 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.

Once the entire Task has been set up, the syntax of the program should be checked.

- Check and apply the changes (✔) (Ribbon *Home* (or *Setup-Control*) > *Process*).

9.9.3.3 Panel - Settings / Start measurement

Now the Panel page can be set up.

- Open the plug-in 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.

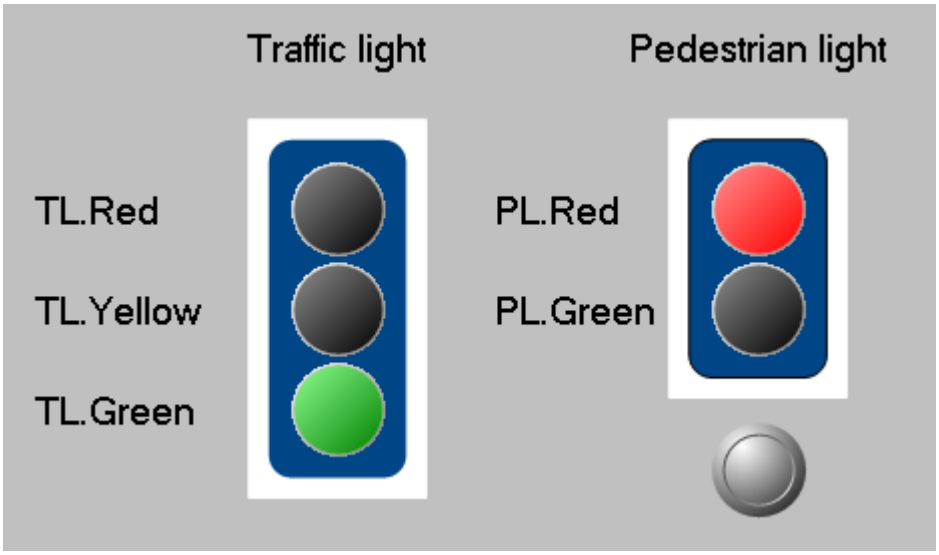
- If you have not reset *Button* with the state *B* in the task, then select the *Push-button behavior*
- If you reset it by program control, the switch does not need *Push-button behavior*

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 139 ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Start*).



Panel - Traffic light (example)

9.9.4 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.

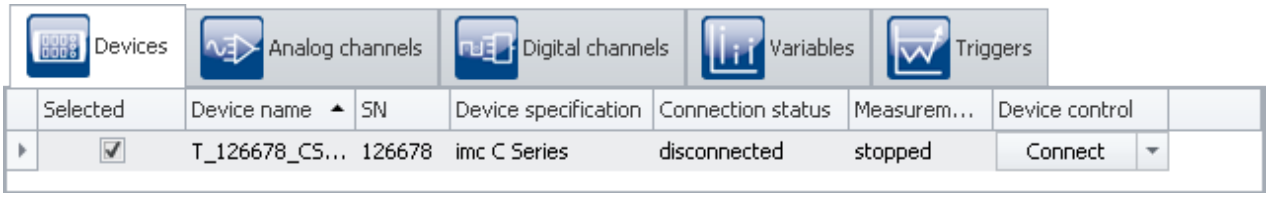
9.9.4.1 Setup - Settings

To begin, the measurement device is selected and configured.

- Open the plug-in Setup.
- Go to the tab: Devices .

Select your device:

- If the device desired does not appear in the list, then perform a device search 139 ( ) (Ribbon *Home* (or *Setup-Control*) > *Search for Devices*). Subsequently, the list will display all devices found.
- 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 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 139 ( ) (Ribbon *Home* (or *Setup-Control*) > *Process*).

By this means, the device settings are available to other plug-ins.

9.9.4.2 Automation - Settings

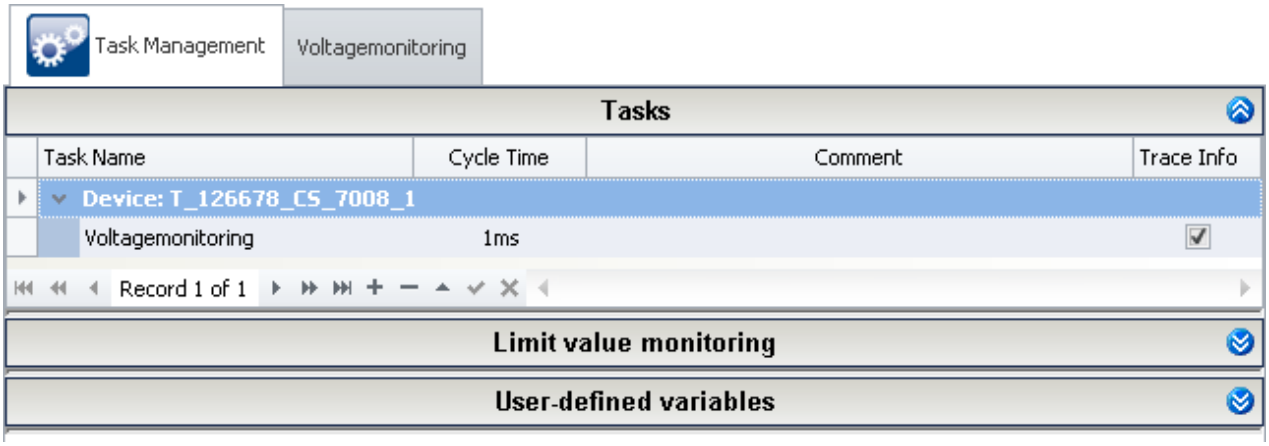
Now it is possible to set up the Task.

- Open the plug-in Automation.

You are now in the Task Management interface.

Create a task with the name *Voltagemonitoring*

- Change the Cycle time to `1ms`, 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*

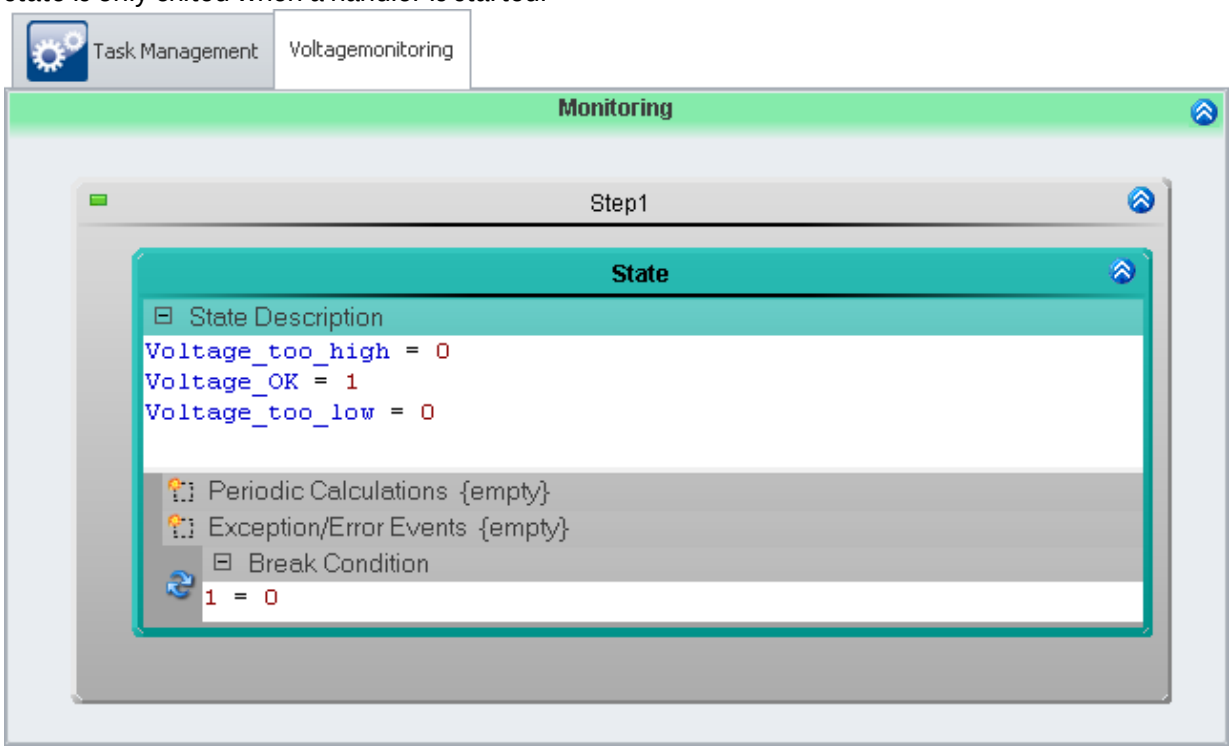
Open this block by clicking on the double-arrow ( ). This takes you to the Block Editor.

- There, create a step
- In this step, add a state.

In the State description, enter the following instructions:

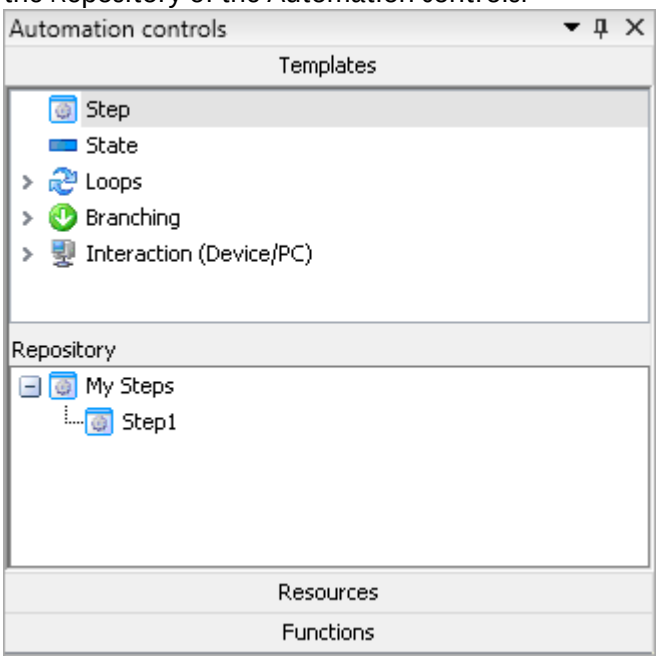
```
Voltage_too_high = 0
Voltage_OK = 1
Voltage_too_low = 0
```

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.

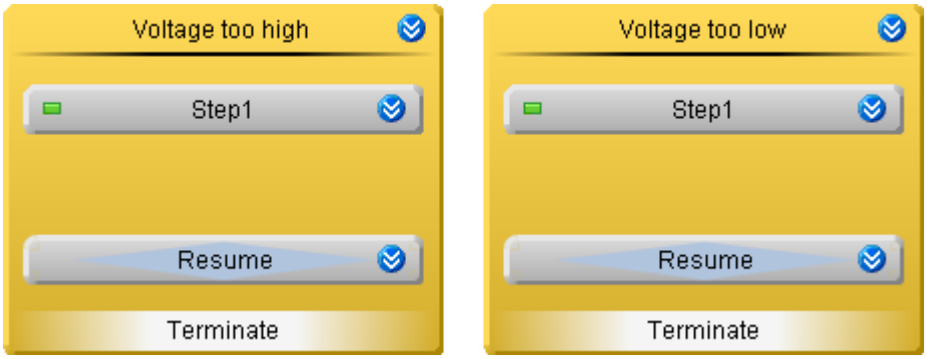
Exception handling

Create two exception blocks in the Exception handling.

- *Voltage too high*
- *Voltage too low*

These can be used in the Limit value monitoring.

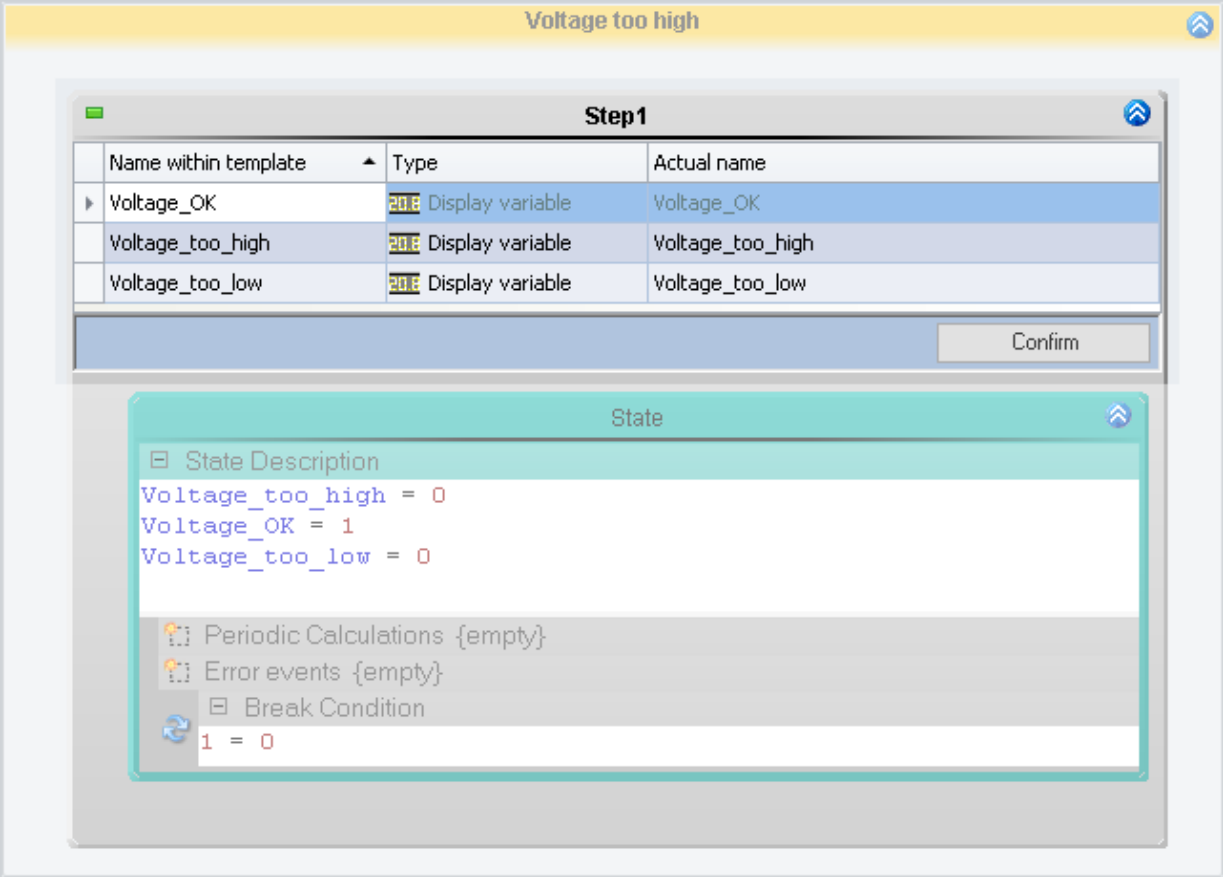
- Use Drag&Drop to copy the previously added step *Step1* to both blocks
- Add to each block one Resume-branching having the condition: `1=1`



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*.

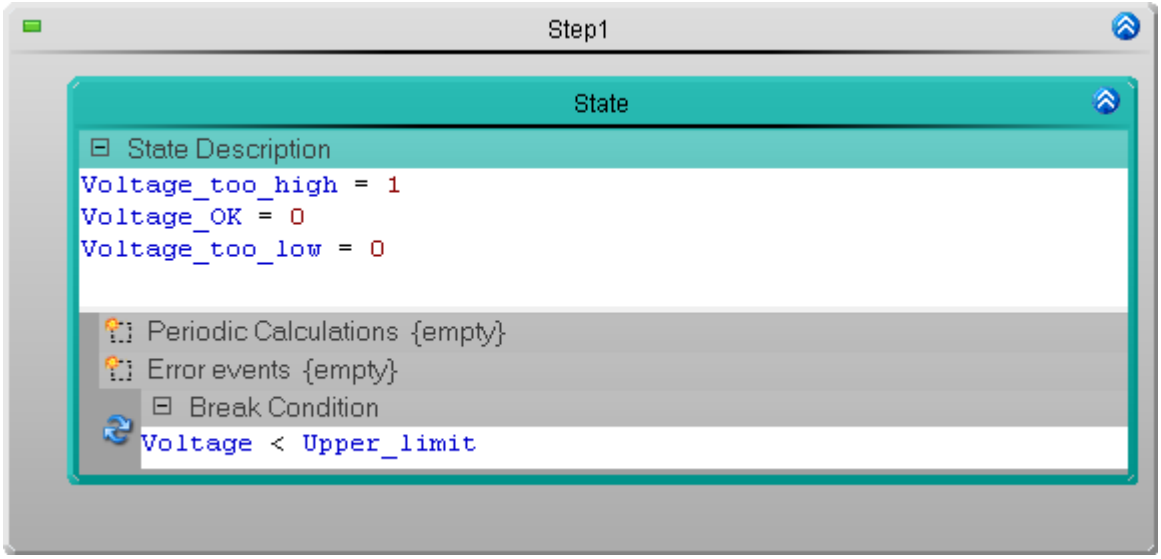
Change the values in the State description:

```
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*.

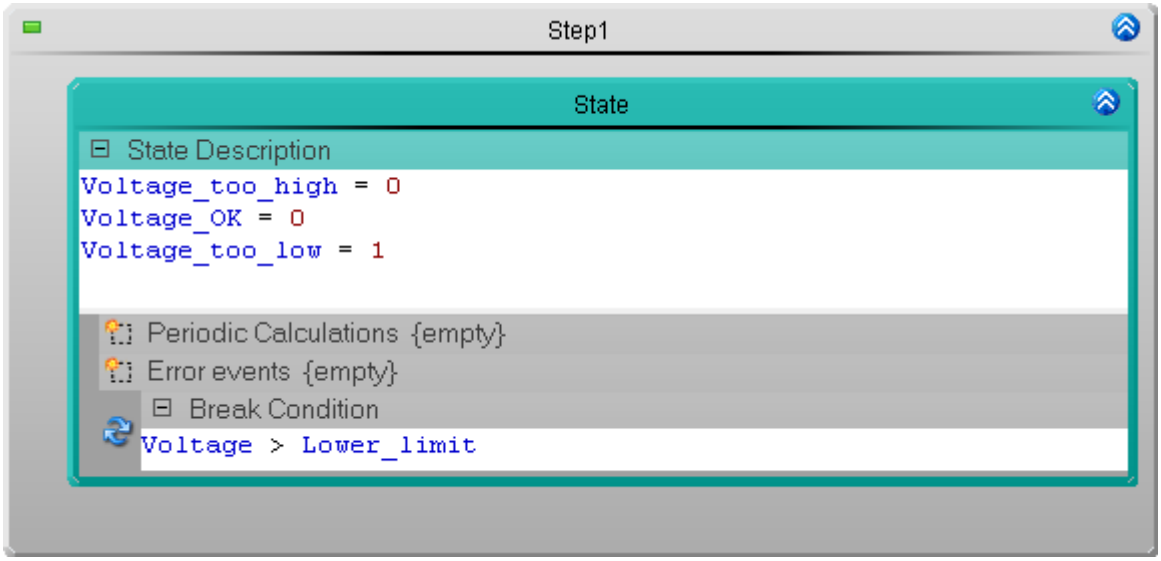
Change the values in the State description as follows:

```
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*.



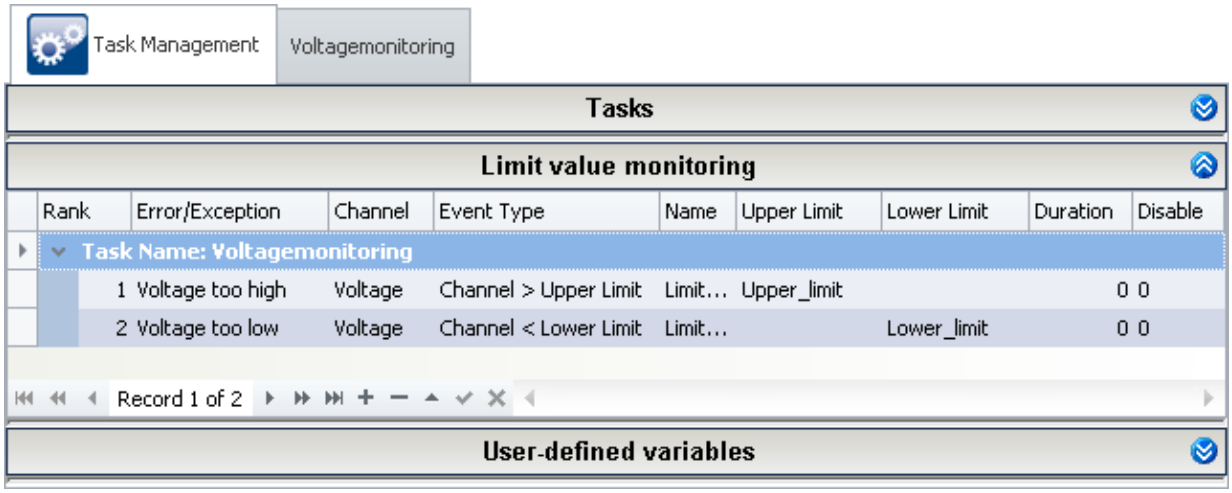
Source code: Voltage below the specified interval

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.

Once the entire Task has been set up, the syntax of the program should be checked.

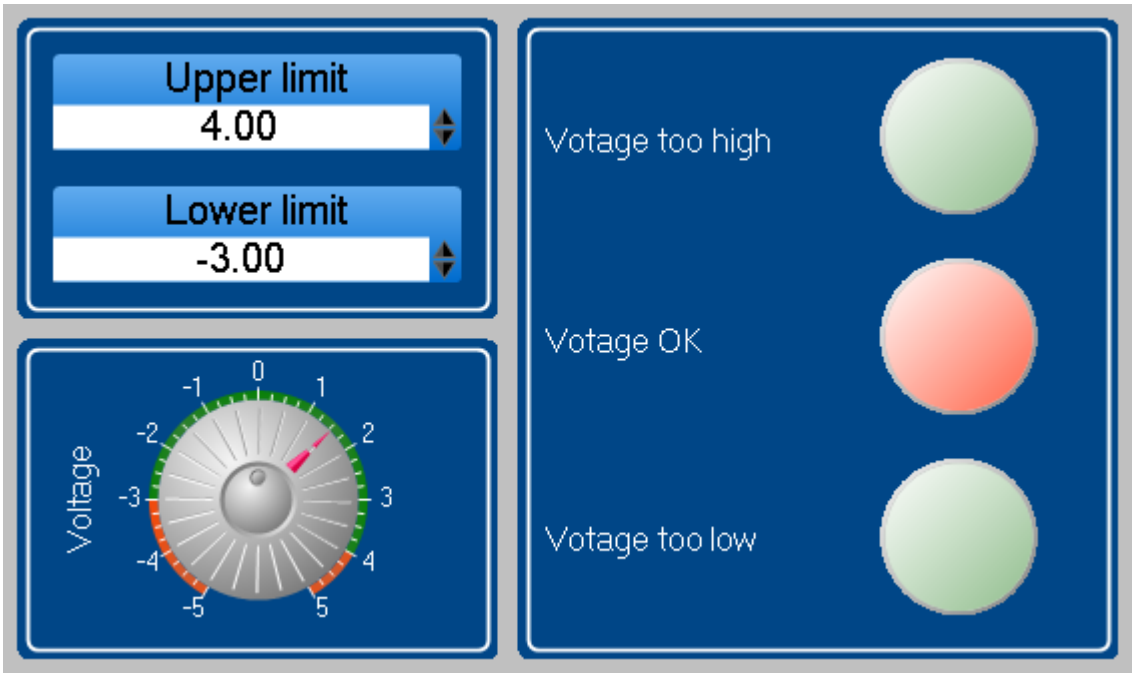
- Check and apply the changes (✔) (Ribbon *Home* (or *Setup-Control*) > *Process*).

9.9.4.3 Panel - Settings / Start measurement

Now the Panel page can be set up.

- Open the plug-in 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 139 ( ) (Ribbon *Home* (or *Setup/Panel-Control*) > *Start*).

Test the monitoring function by adjusting the *voltage* to higher than one of the *limits* by means of the potentiometer.

10 Sequencer

imc STUDIO Sequencer is the imc STUDIO plug-in with which automated measurement routines are created. 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 1191. The commands appear there in a tree diagram. The available commands can be found, among other places, in the Commands 1190 tool window.

Reference: Command Reference

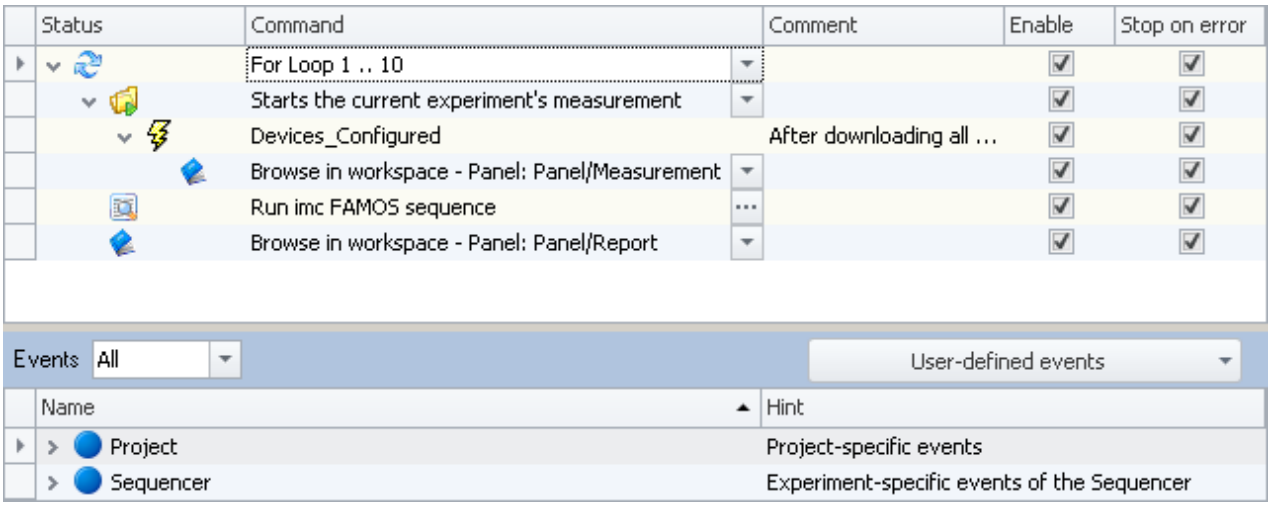
How the individual commands operate is described in the chapter Command Reference 1592.

Further, you can link the commands to events. This assignment is performed by a graphic technique in the so-called event handling 1197.

To open the plug-in Sequencer, click on the entry representing its title or use the Navigator 67.

The content of the plug-in Sequencer is saved 83 as part of the experiment .

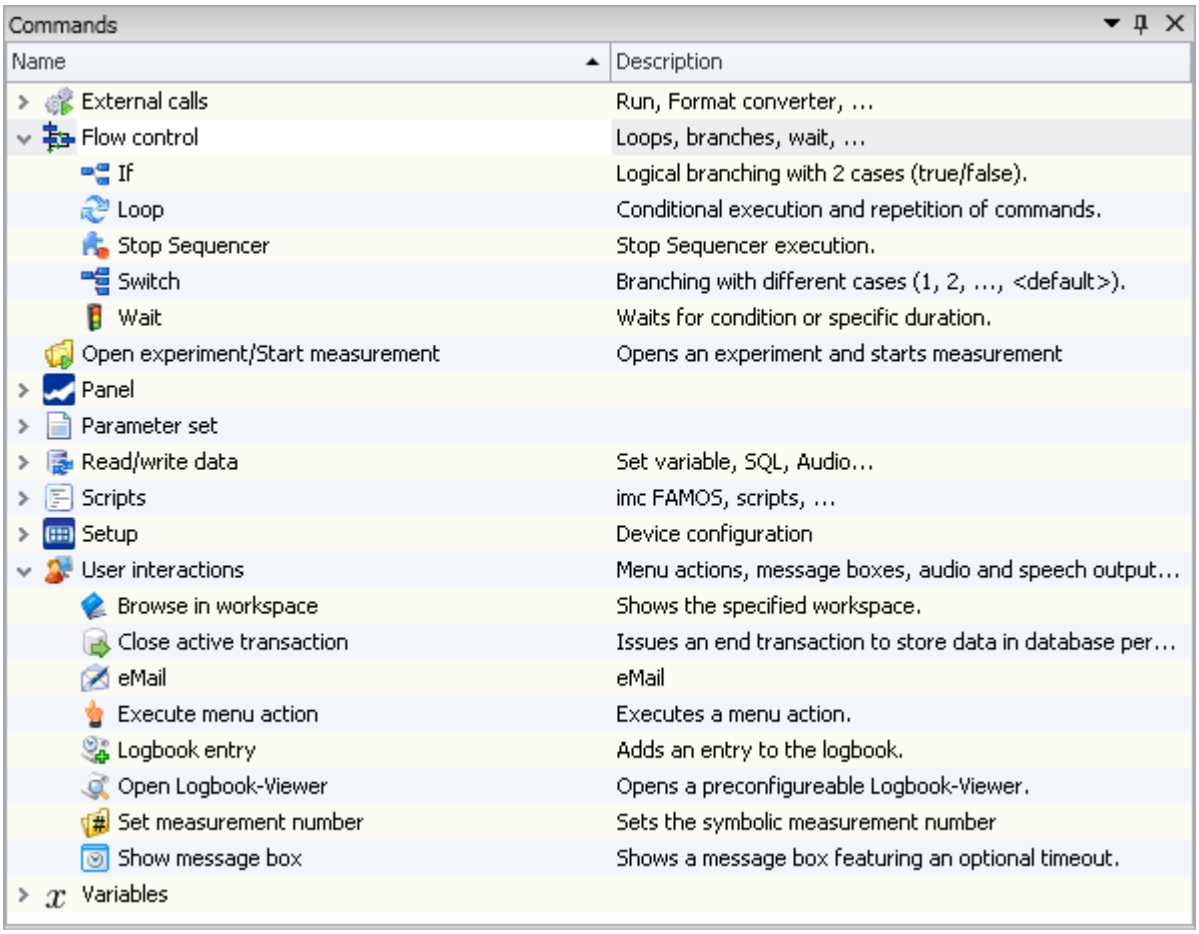
The image below shows an example of a sequence table 1191 with multiple commands and the event handling 1197.



Sequence table with routine of the plug-in "Sequencer"'s commands

10.1 Tool window Commands

The plug-in Sequencer has a tool window called *Commands*. This comprises commands with which the sequence is created.



Name	Description
External calls	Run, Format converter, ...
Flow control	Loops, branches, wait, ...
If	Logical branching with 2 cases (true/false).
Loop	Conditional execution and repetition of commands.
Stop Sequencer	Stop Sequencer execution.
Switch	Branching with different cases (1, 2, ..., <default>).
Wait	Waits for condition or specific duration.
Open experiment/Start measurement	Opens an experiment and starts measurement
Panel	
Parameter set	
Read/write data	Set variable, SQL, Audio...
Scripts	imc FAMOS, scripts, ...
Setup	Device configuration
User interactions	Menu actions, message boxes, audio and speech output...
Browse in workspace	Shows the specified workspace.
Close active transaction	Issues an end transaction to store data in database per...
eMail	eMail
Execute menu action	Executes a menu action.
Logbook entry	Adds an entry to the logbook.
Open Logbook-Viewer	Opens a preconfigureable Logbook-Viewer.
Set measurement number	Sets the symbolic measurement number
Show message box	Shows a message box featuring an optional timeout.
Variables	

Commands tool window in Sequencer plug-in

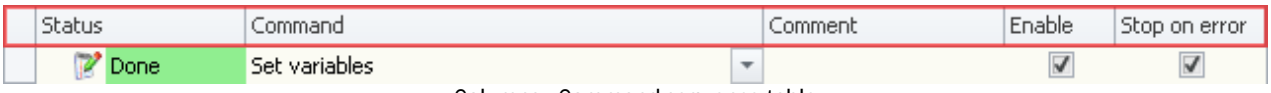
It is possible to access the commands from various locations in imc STUDIO. For this reason, they are described separately.

Reference: Command Reference

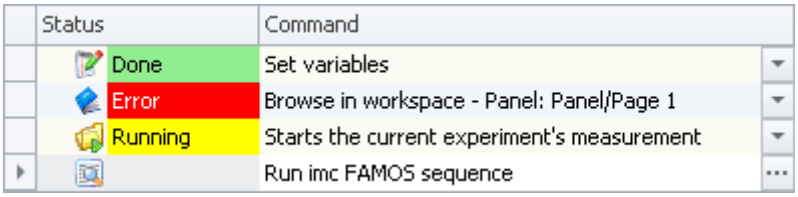
How the individual commands operate is described in the chapter Command Reference 1592.

10.2 Sequence table

The sequence table contains multiple columns.



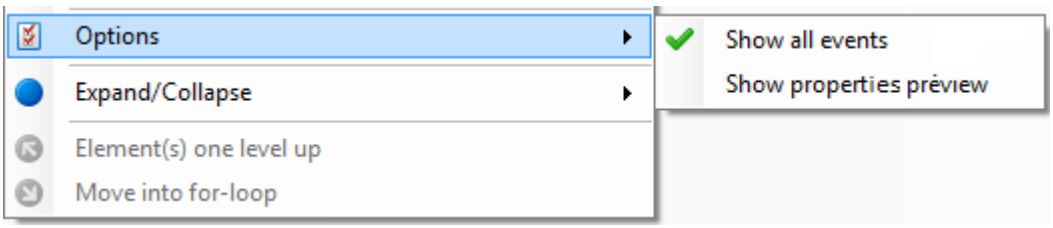
Columns - Command sequence table

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.
Activated	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	This stops the sequence if an error occurs in a command. In order to resume the sequence at the next step, select the command Start (from selected command to end of group) 1196.

10.2.1 Display in the sequence table

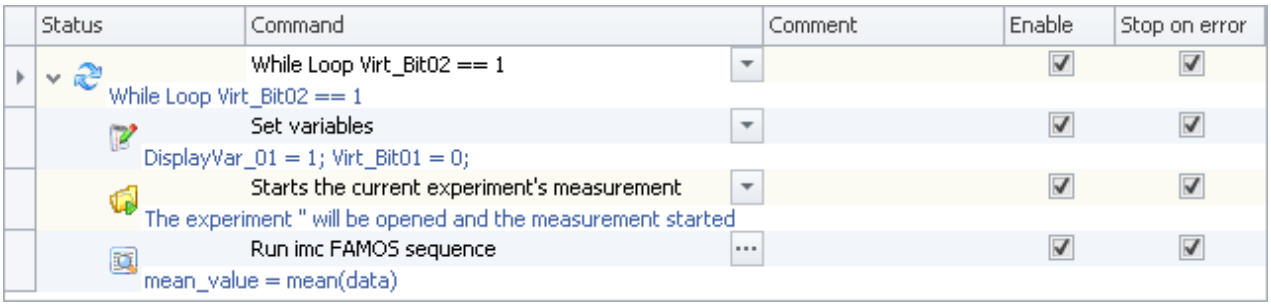
Preview of properties:

Some commands come with extra information which is displayed if you select the context menu items *Options* > *Show all events* and *Show properties preview*.



Context menu item Options > Show all events

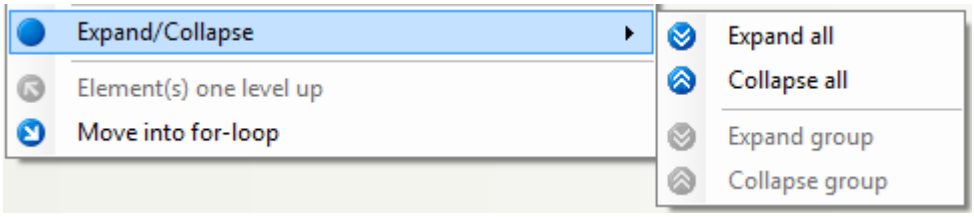
The sequence table then shows important information on the commands:



Properties preview

Expand/Collapse

Conditional commands, as well as commands in loops, can be displayed or hidden under their respective headings.



Expand/Collapse of conditional subroutines

10.2.2 Groups

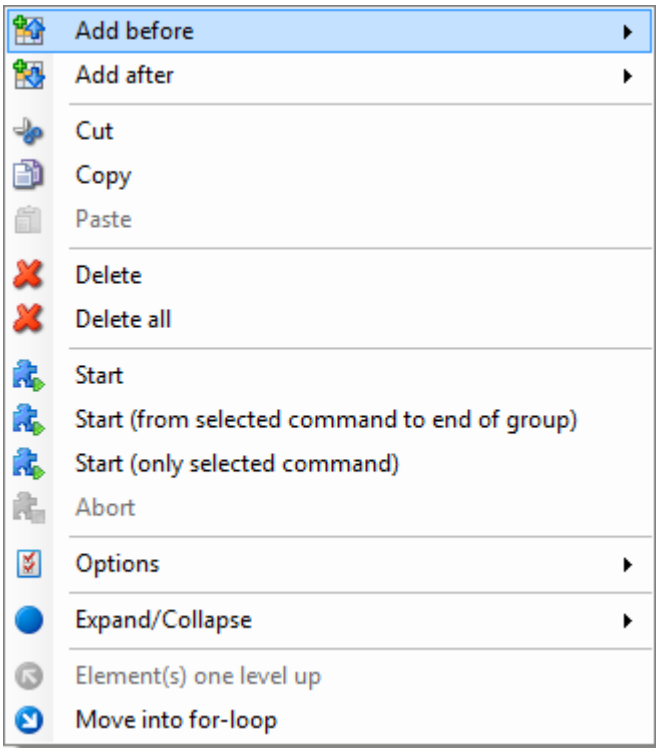
By placing an command under a loop (Loop) or a conditional branch (If, Switch), you create a so-called group. This means the sequence has groups and levels and branches, which can be represented by a tree diagram.

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.

10.3 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.



Test routine context menu

The individual commands are described here:

- Add before/after 1194
- Cut 1195
- Copy 1195
- Paste 1195
- Delete 1195
- Delete all 1195
- Start 1196
- Start (from the selected command to the end of the group) 1196
- Start (only the selected command) 1196
- Stop 1196
- Options 1191
- Expand/Collapse 1191
- Element(s) 1195 one level up 1195
- Move to For-loop 1195

10.4 Creating a sequence of commands

To add a command to the sequence table, there are two options:

- by Drag&Drop 1194
- by the context menu 1194

You can place a command at different positions.

	Before the current position
	Below or within the current position (groups or events)
	After the current position

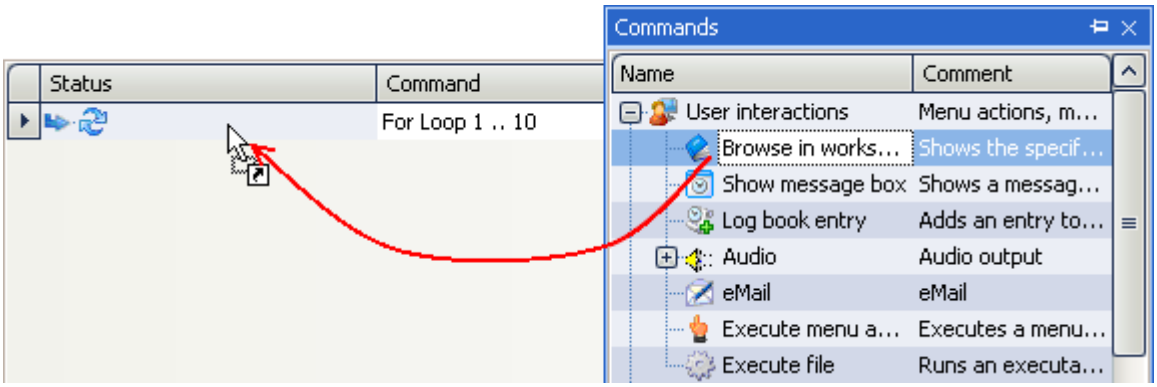
10.4.1 Adding commands by means of Drag&Drop

Commands can be added by using the Drag&Drop technique to bring them from the Commands tool window.

Use Drag&Drop to move the command to the desired position.

Adding commands to a group

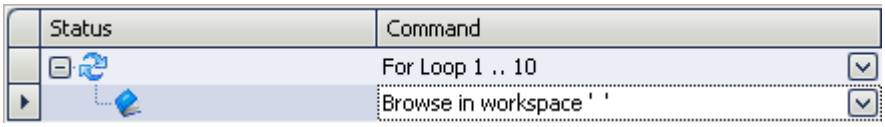
- Use Drag&Drop to place the desired command in a group, as seen below:



Example: Using Drag&Drop to add the command:
"Browse in workspace" to the loop

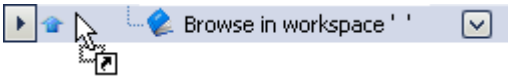
To the left of the even the position is specified ( )

- Release the mouse button. The command is added to the group

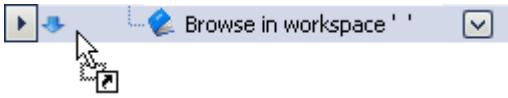


Adding commands before or after existing commands

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"

- Release the mouse button. The command is added *before/after* the existing command.

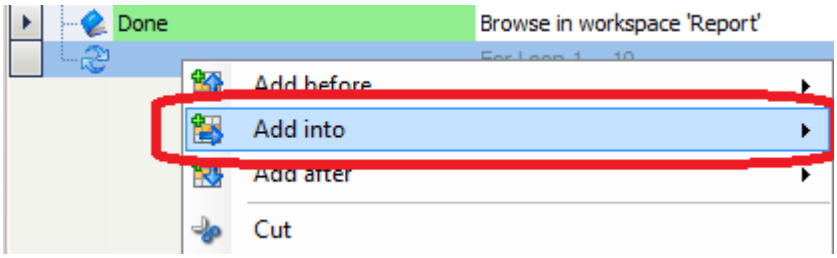
10.4.2 Adding a command by means of the context menu

Commands can be added at various positions by means of the context menu.

Open the context menu of the desired position and select a command.

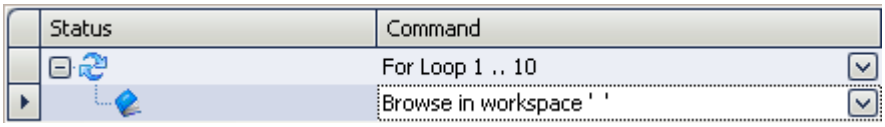
Adding commands to a group

- Open the context menu of the desired position, as shown below:



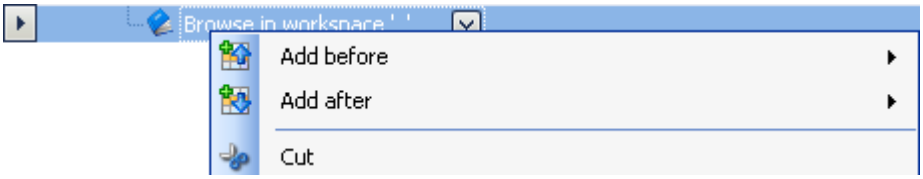
Example: Adding the command: "Browse in workspace"
to the loop by means of context menu

- Select the position *Add into* ( )
- Select the desired command. The command is added below the group



Adding commands before or after existing commands

Commands can be added either *before* or *after* an existing command. To do this, open the command's context menu, as shown below:



Inserting before or after the command:
"browse workspace"

- Select the position *Add before/after* (/)
- Select the desired command. The command is added *before/after* the existing command.

10.4.3 Deleting, copying, moving commands

Deleting a command

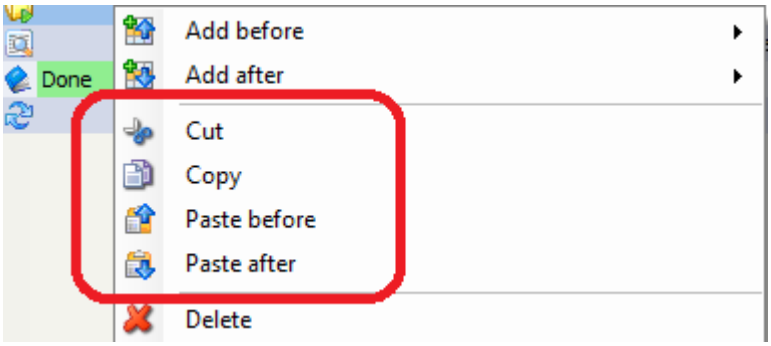
To delete selected commands, highlight them and select the context menu item *Delete*.

To delete the entire sequence, select the command *Delete all*.

Cutting (moving), copying, pasting

To cut or copy commands and paste them somewhere else, you can use the context menu, or use Drag&Drop to move the command to the desired location:

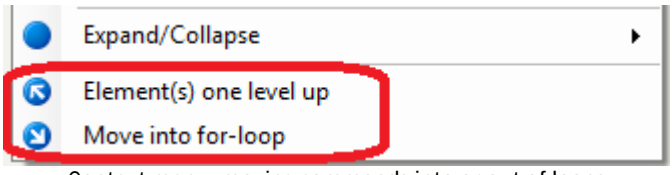
- Drag&Drop: Cut + Paste
- Drag&Drop + <CTRL>: Cut + Paste



Context menu: "Cut, Copy, Paste"

Movement into loops

Commands can be moved into or out of loops.



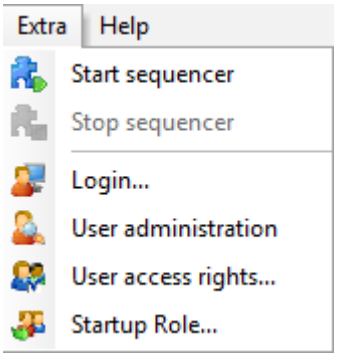
Context menu: moving commands into or out of loops

10.5 Running Sequencer

Execution

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.

Starting

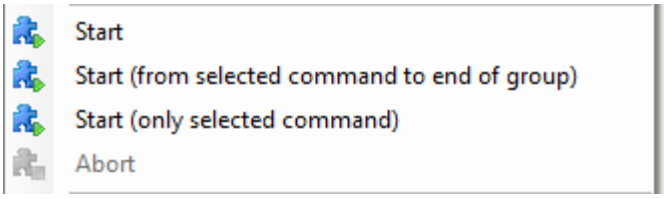


Extras menu

Start Sequencer by clicking on the toolbar button . Alternatively: Menu item Extras -> *Start Sequencer* or the context menu item *Start*.

The measurement devices are prepared and started by the command Open/Run experiment.

Targeted command execution



Excerpt from the context menu

You can run single lines in the sequence table by means of the context menu: *Start (only selected command)* .

The context menu command *Start (from selected command to end of group)* runs the Sequencer from the selected line up to the end.

Stopping

A running Sequencer can be interrupted using the button . Alternatively: menu item Extras -> *Stop Sequencer* or the context menu item *Stop*. The context menu can be opened only under some circumstances (the Sequencer plug-in may not be open).

If the Sequencer is currently running the commands Open/Start experiment, this stops the measurement.

10.6 Events

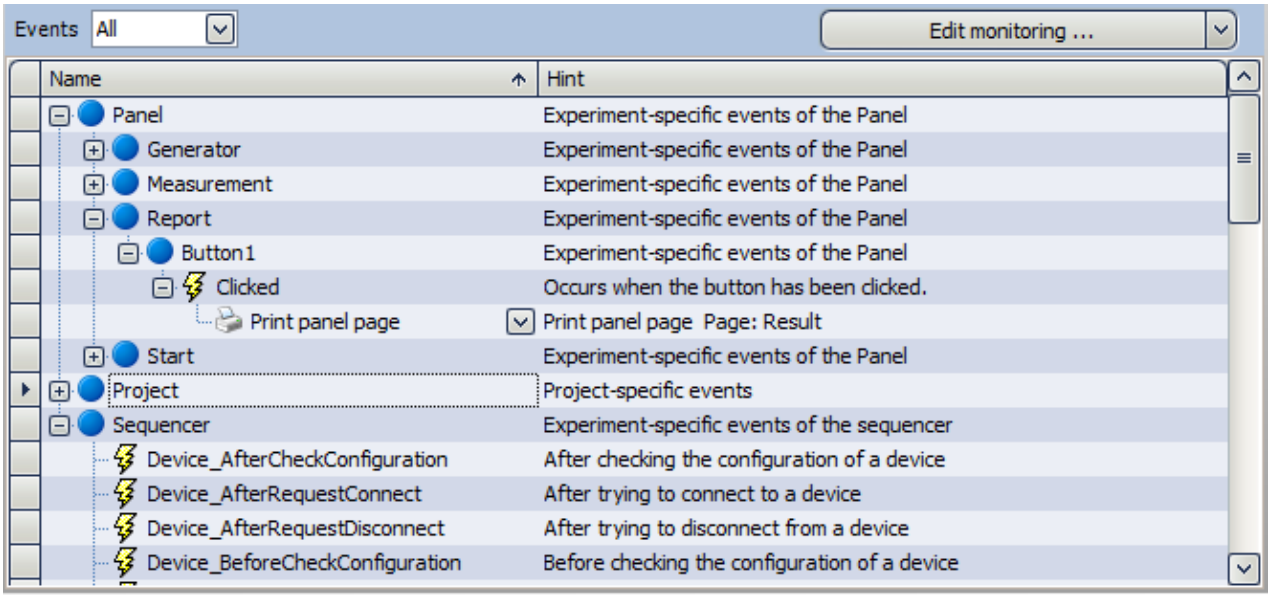
The *event handling* is the *Sequencer* plug-in's second main menu. Here, you can use Drag & Drop to link commands with events. The linked commands are run when the event occurs.

- To link an command with an event, see the chapter: Creating a link 1199)

Events can pertain to different areas:

- *Panel:* Events which are assigned to specific Widgets.
- *Project:* Events pertaining to all of a project's experiments
- *Sequencer:* Events only pertinent to the current experiment.

The image below shows events and their assignments to the branches *Panel*, *Project* and *Sequencer*. The *Events list* consists of the names of the events and a brief description (column *Notes*). When a command is linked with an event, then the list expands to a tree diagram.

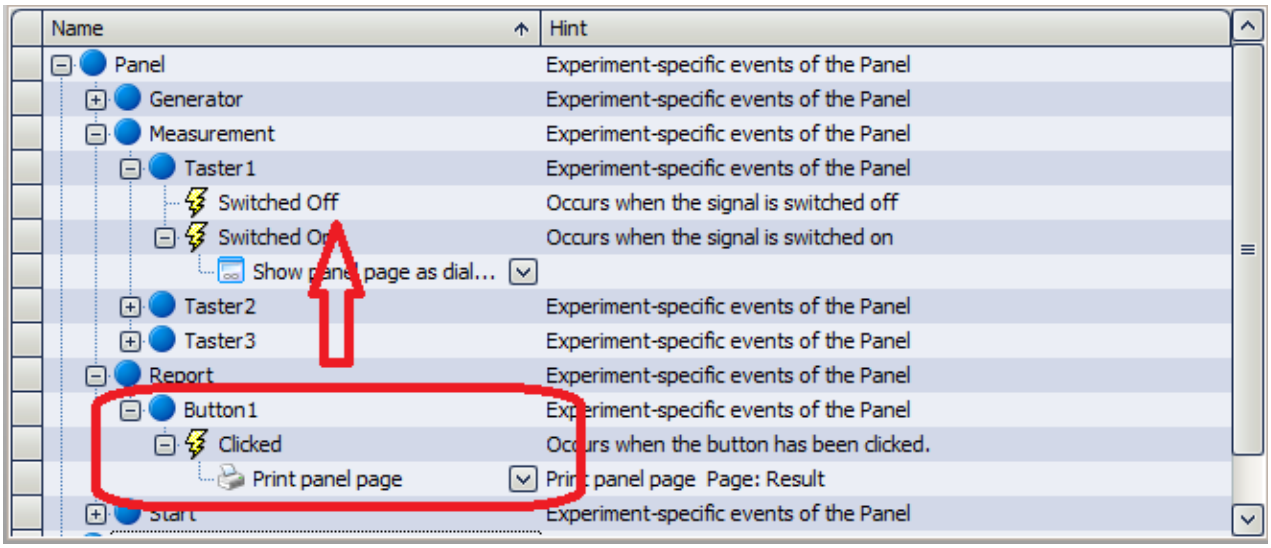


Events on the Sequencer page

10.6.1 Panel

In the branch Panel all Widgets are listed, which possess events. Here you can easily get an overview of which events are triggered by means of the plug-in Panel.

You can also edit the assignments by means of Drag&Drop. For example, you can move or copy (by holding the Ctrl-key) a command to the Widget of a different Panel page. Additional commands can be added from the Commands toolbar.



The event "Print Panel page" can be copied here to the panel "Measurement"'s button by holdoing the Ctrl-key.

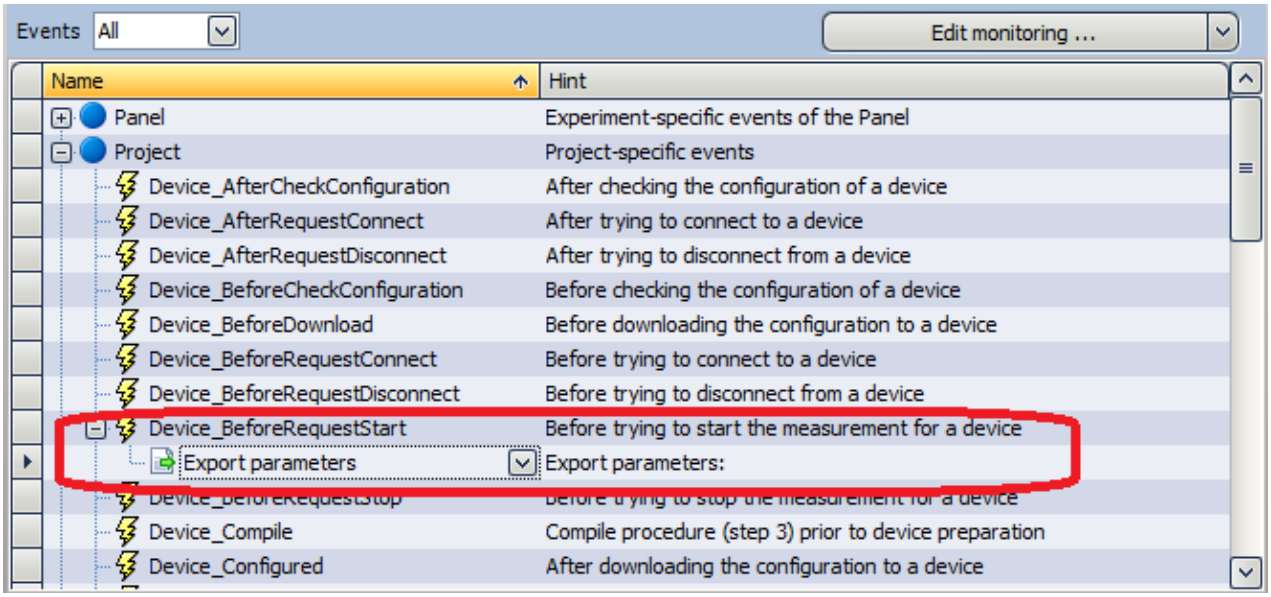
10.6.2 Project

Events in the branch *Project* are valid for all experiments of this project.

The events listed here can, in general, be expanded with certain commands. This provides an easy way for you to adapt the measurement routine to your needs.

Examples:

- Jump from the measurement start to the setup's Experiment page
- Have the user confirm a message that all is proceeding correctly before continuing with the command Device_BeforeRequestConnect (before connectoin to the device).
- Export the current settings as a CSV file once the measurement ends.



Example: Automatic export of settings before every measurement start

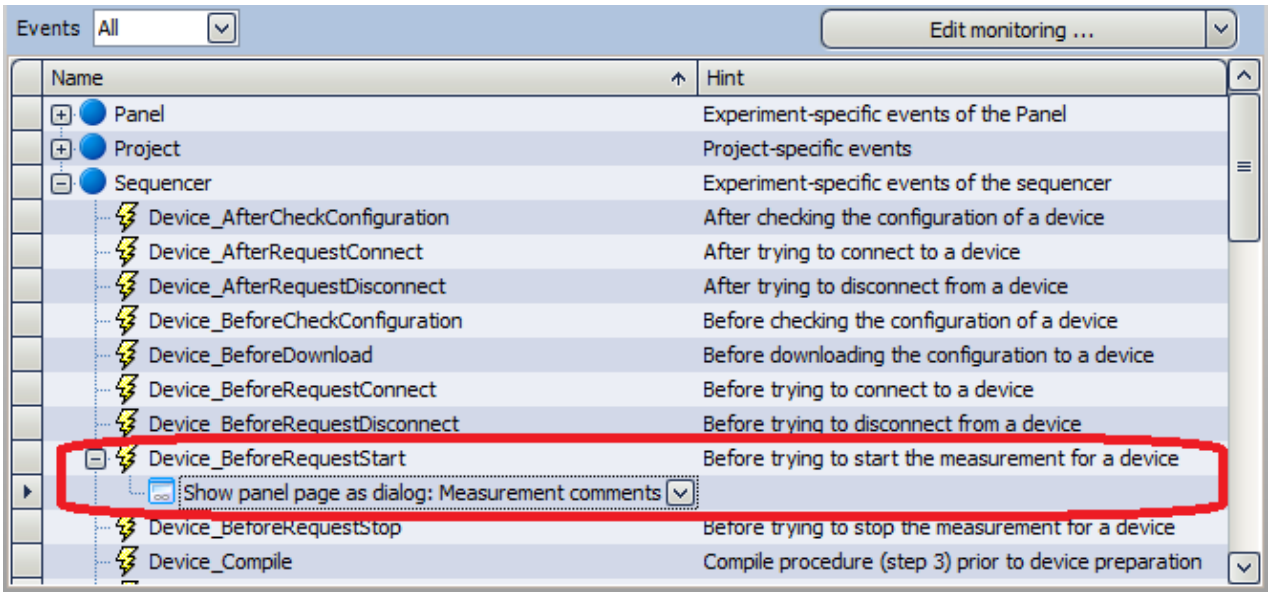
10.6.3 Sequencer

Events in the branch Sequencer are valid for the current (Sequencer-)experiment.

Here you will find the same events as under Project. Since the commands used here are only valid for the current experiment, you can make adaptations here which affect the current measurement process in a targeted way.

Examples:

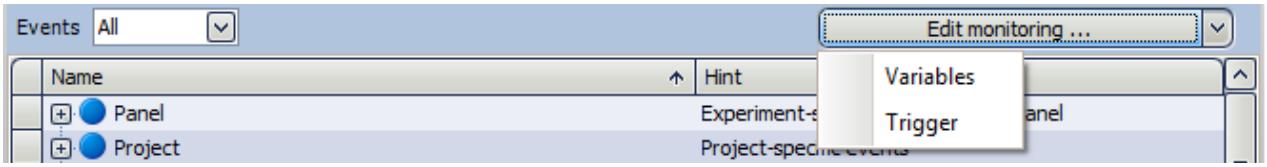
- Create a Panel page for entering special testing information. Open this Panel page as a dialog before every measurement and prompt the user to make certain entries. Only afterwards is the measurement started.
- Start a imc FAMOS sequence for initializing certain pv.Variables which are processed by the device's installation of imc Online FAMOS.



Example: Panel-page as dialog for entering testing information prior to starting measurement

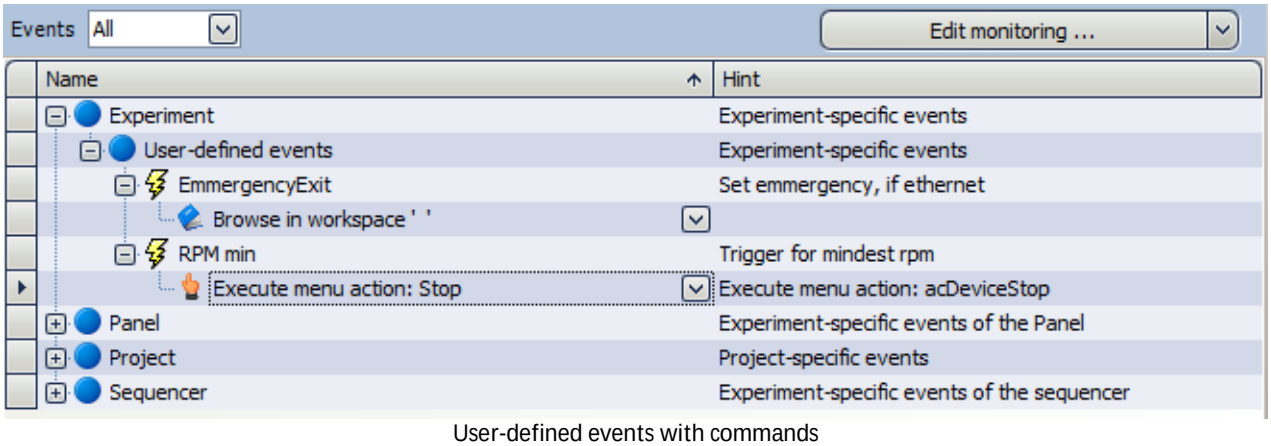
10.6.4 User-defined events

The button *User-defined events* enables creation of user-defined events.



Adding user-defined events

For example, you can monitor the value of a variable and run a command when its value changes.

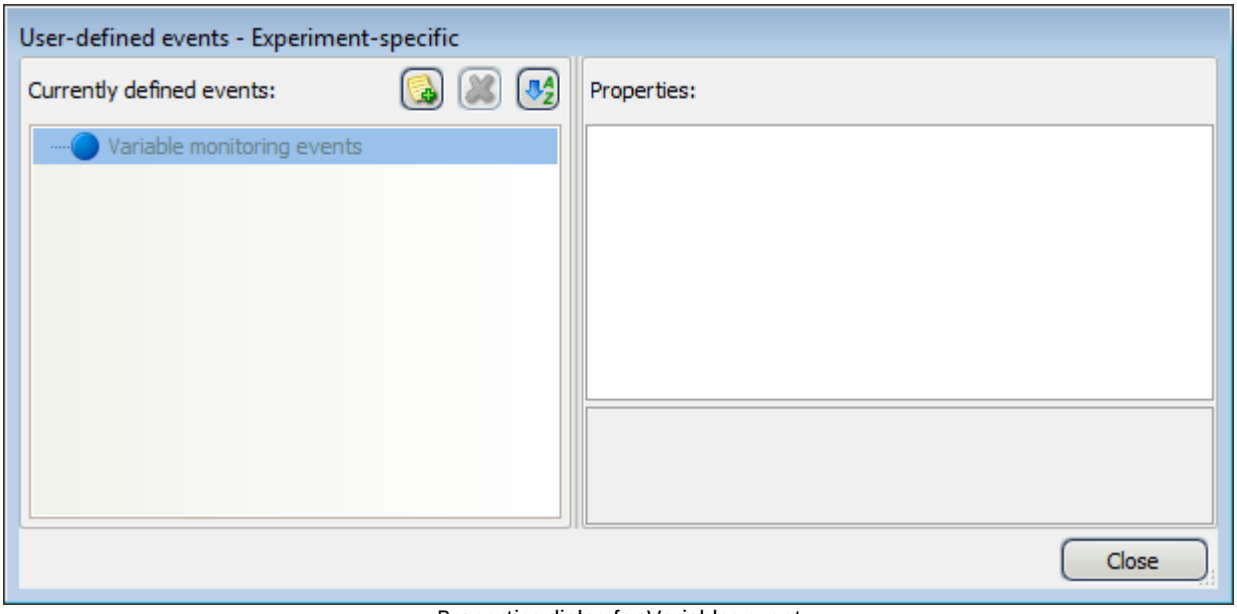


User-defined events with commands

Procedure

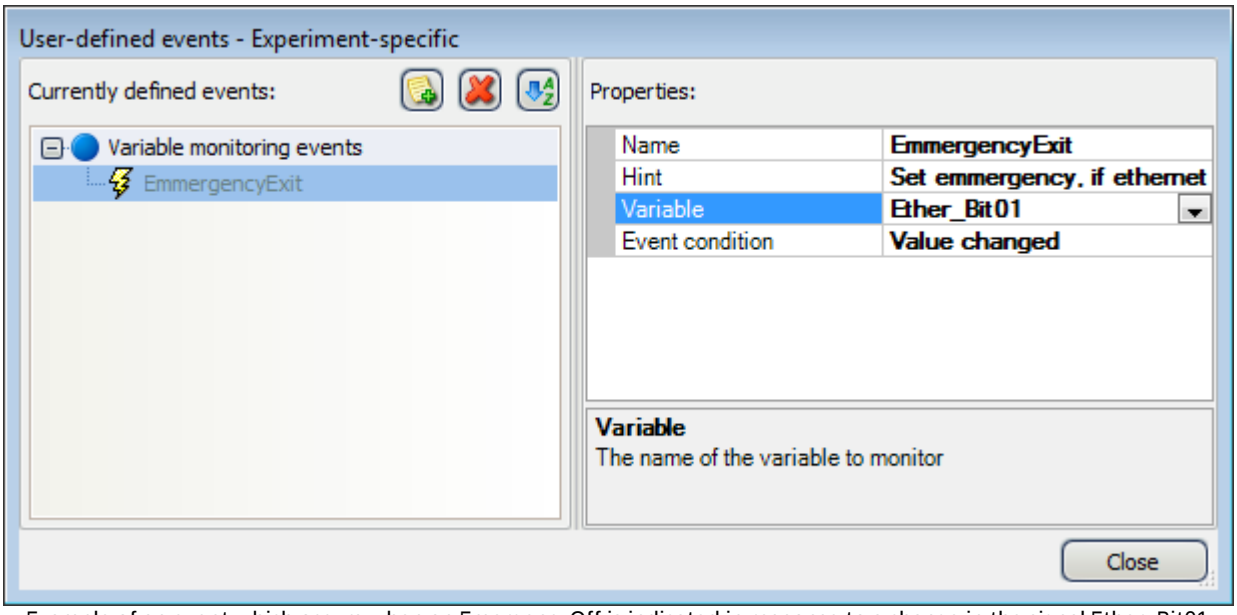
Example: Variables event

Click on the button and select the event for *Variables*.



Properties dialog for Variables event

Add a variable by clicking on the symbol .



Example of an event which occurs when an EmergencyOff is indicated in response to a change in the signal Ether_Bit01.

Name: Name the event

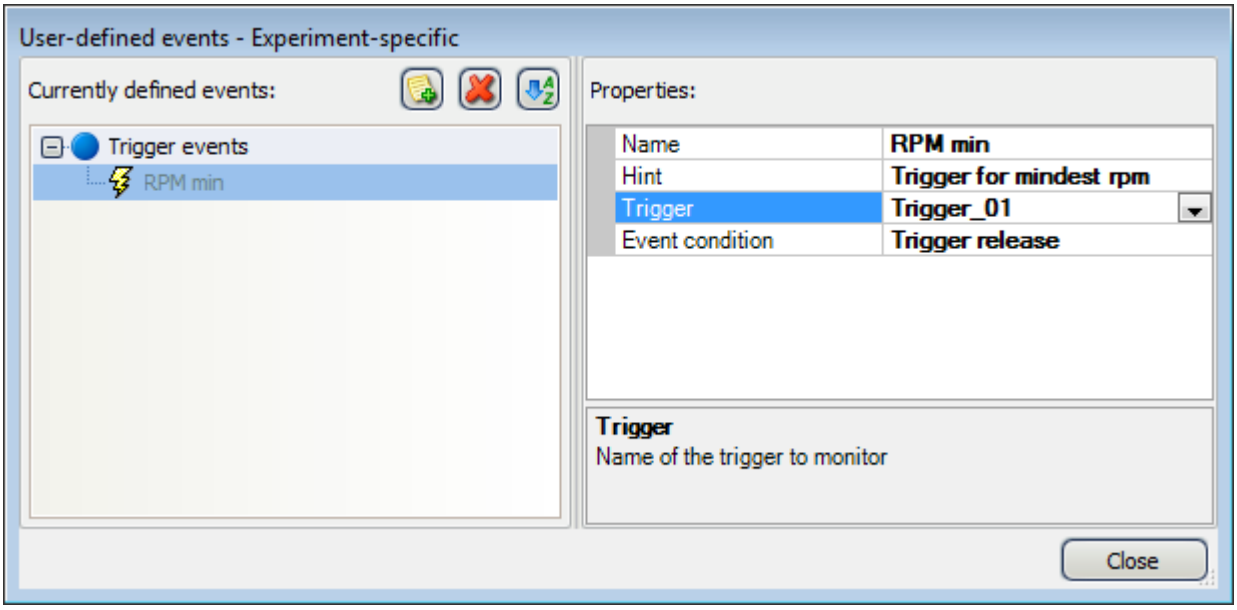
Hint: A note displayed in the Sequencer

Variable: Opens the selection dialog for the variables

Event condition: Event occurs when there is a *value change* in the variables.

Example: Trigger event

Click on the button and select an event for *Trigger*.



Properties dialog for a trigger event

Name: Name the event

Nint: A note displayed in the Sequencer

Trigger: Opens the selection dialog for the triggers

Event condition: Event occurs with the start of the trigger condition (*Trigger released)* or at its end (*Trigger end)*

10.7 Tutorial

The following are examples involving the plug-in Sequencer. The examples use Widgets 833 from the plug-in Panel 813, which may not come with your installation (depending on the product configuration).

10.7.1 Starting the Sequencer

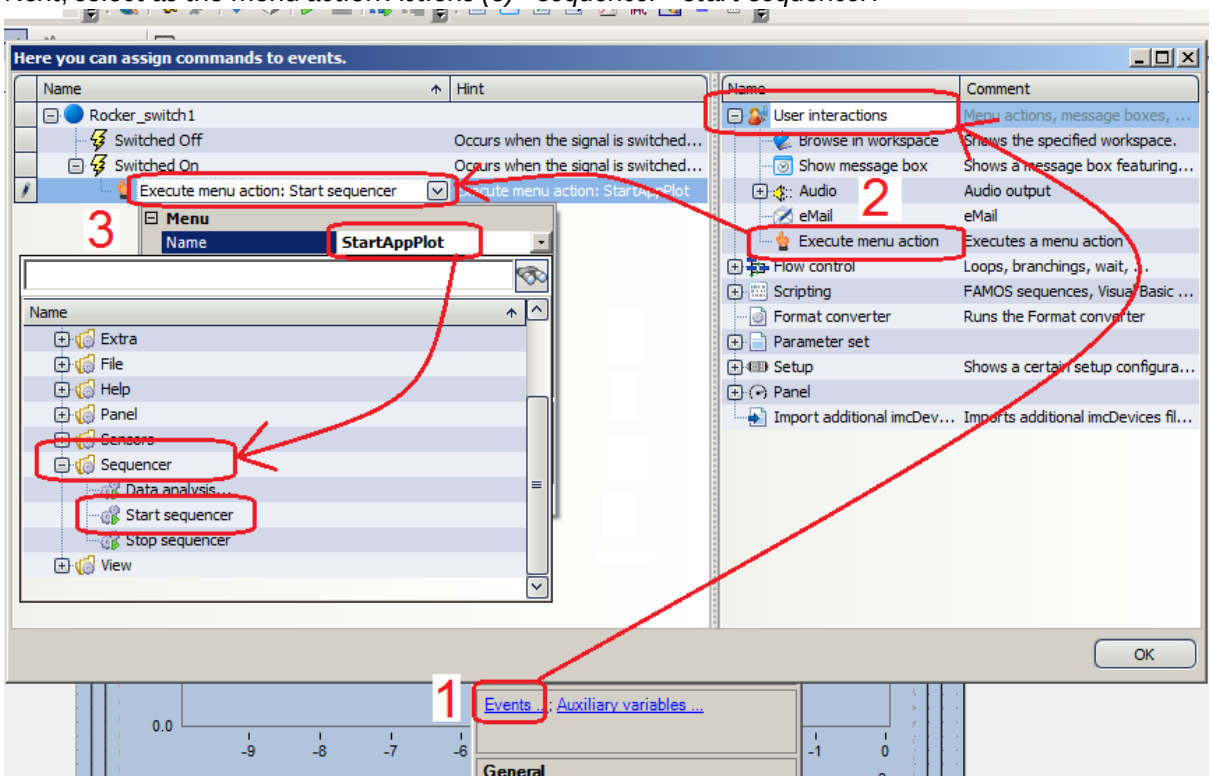
This exercise shows how to start a measurement by means of the Sequencer. At the end, an audible announcement of the measurement's conclusion is emitted by the computer's sound card.

Assignment:	Create a curve window in the Panel page with one channel. The measurement is started by the Sequencer and automatically ends after 10s. The computer announces the measurement's conclusion via the sound card.

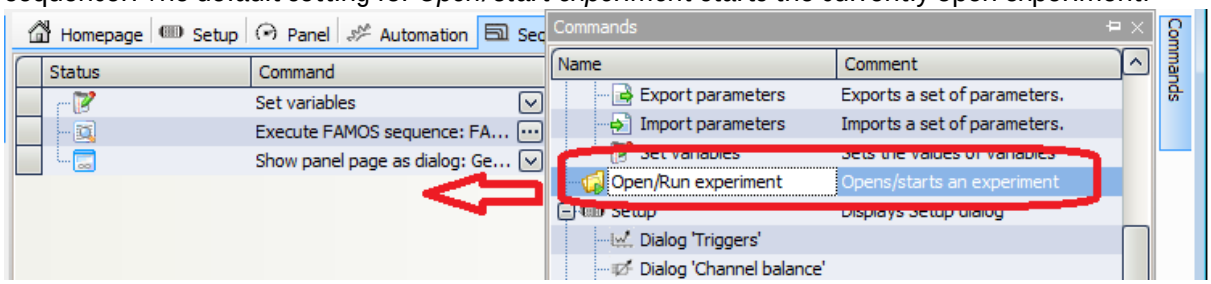
Procedure:

1. Go to the page Setup. Select the desired measurement device and activate a channel. Set the channel's measurement duration to 10s.
2. Go to the Panel page.
3. Open the tool window *Widgets* and select from the group Curve window the Widget Standard. Drag the object to the Panel page. Move the curve window to its center, using the curve window's move handle. Change the shape by dragging the sizing handles, which are the white squares at the corners and sides of the object.
4. Click on Prepare Measurement . The channel then appears in the Data Browser. Drag the channel to the curve window.
5. From *Widgets - Automotive*, drag a *Rocker switch* to the Panel page. Set its *Switching behavior* to *Hold-to-run*.
6. Open the *Rocker switch* properties dialog. The switch is to be labeled with the word *Start*.
7. Click on the link *Events...* in the lower portion of *Properties (1)*. Select the event *Switched On* and assign to it the *User interactions - Execute menu action (2)*.

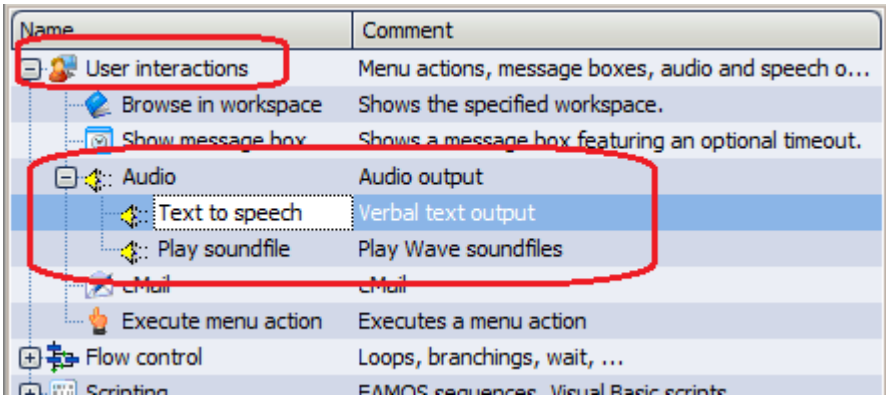
8. Next, select as the menu action *Actions (3) - Sequencer - Start Sequencer*.



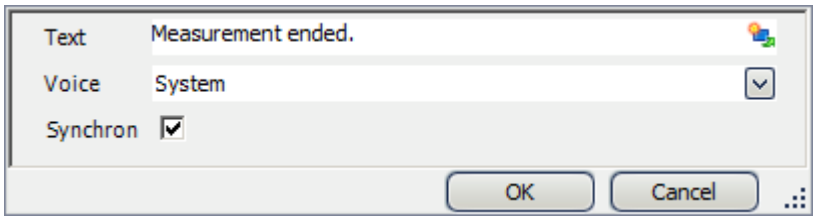
9. Go to the Sequencer and select from *Commands* the item *Open/Start experiment*. Drag it into the Sequencer. The default setting for *Open/Start experiment* starts the currently open experiment.



10. From the commands in the branch *User interactions*, drag the entry *Audio - Text to speech* to the Sequencer.



Enter a text, e.g. "Conclude measurement". Try a different voice as well. The defaults depend on the Windows system. There are usually only English voices, so that any German inputs are pronounced as English words.



11. Next, select instead of voice output the entry *Play soundfile*. Enter as the WAV file a sound from the folder C:\Windows\Media.
12. 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 which you previously created: The measurement starts and finally the synthetically produced voice output is emitted, followed by the sound you selected.
13. Save the experiment under the name "Start Sequencer"

Result: The measurement is started via the Panel. By means of the Sequencer, you can have the user notified when the measurement ends.

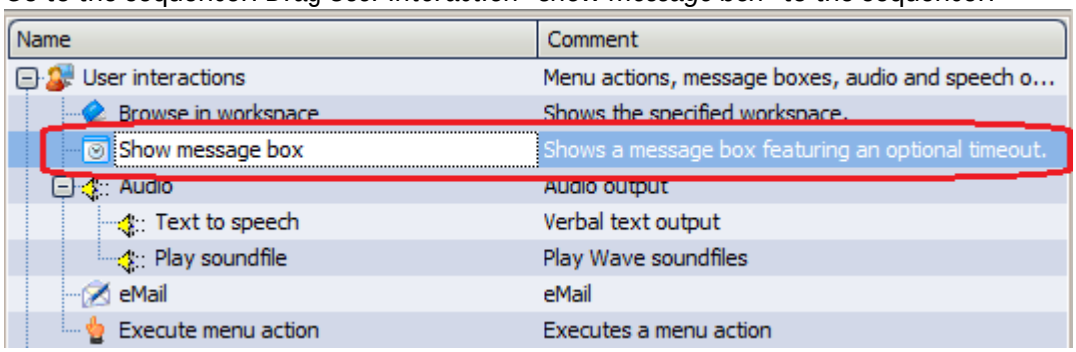
Remark: It is still possible to start the measurement independently of the Sequencer. (Green Start button in the Studio toolbar). In this case, however, the Sequencer does not start with it.

10.7.2 Output box

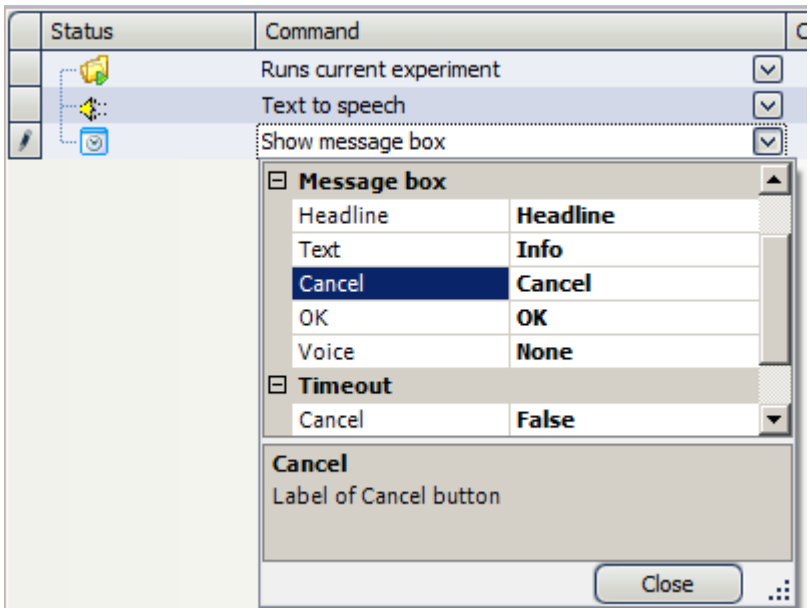
The preceding exercise is expanded to include a message box at the end of the measurement. Subsequently, an additional message box with sound output is displayed.

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.

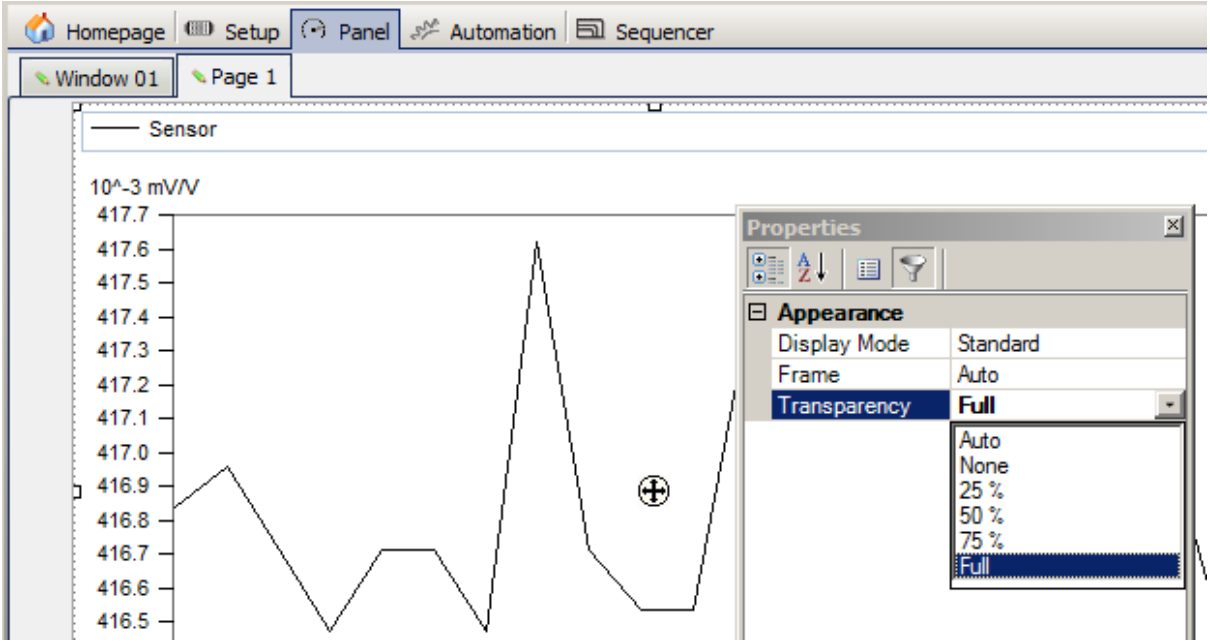
Remark: The voice output can be switched off by the entry Voice: none.

10.7.3 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.

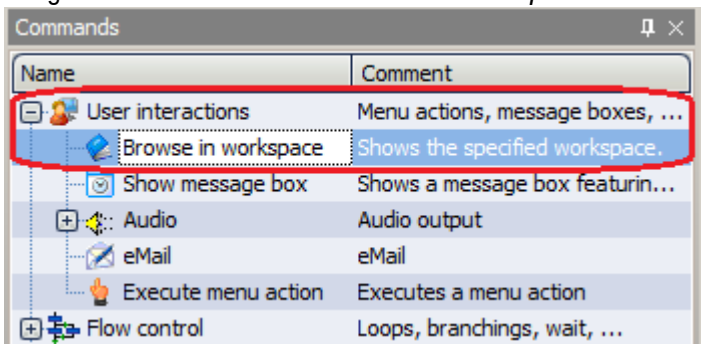
Assignment: Change the example "Start Sequencer" so that at the end of the measurement, a second Panel page is displayed instead of the sound output.

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 format).* To do this, right-click the mouse over the Panel and select *Add Page...* Name it "*Report".*
3. Create a Standard curve window and set the property *Transparency* to *Full*.

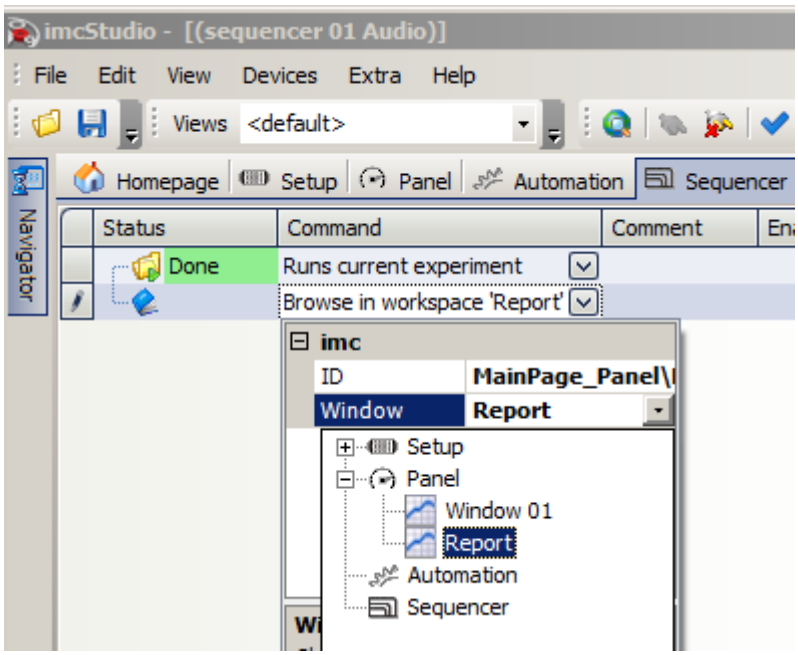


4. Go to the Sequencer. Delete all Sequencer steps except the first step, which starts the measurements. To do this, highlight the steps and right-click the mouse to open the context menu. Select the context menu item *X Delete*.

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.

10.7.4 Dialog

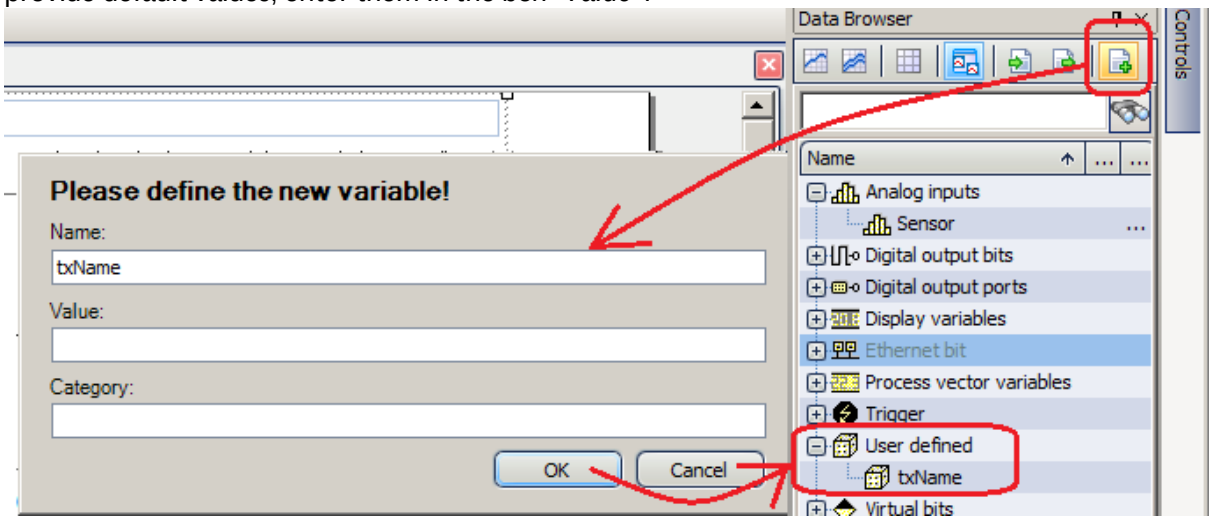
An additional Panel page is displayed as a dialog.

Assignment: Expand the experiment "Scroll". Create a dialog by entering your name and test object number before the measurement.

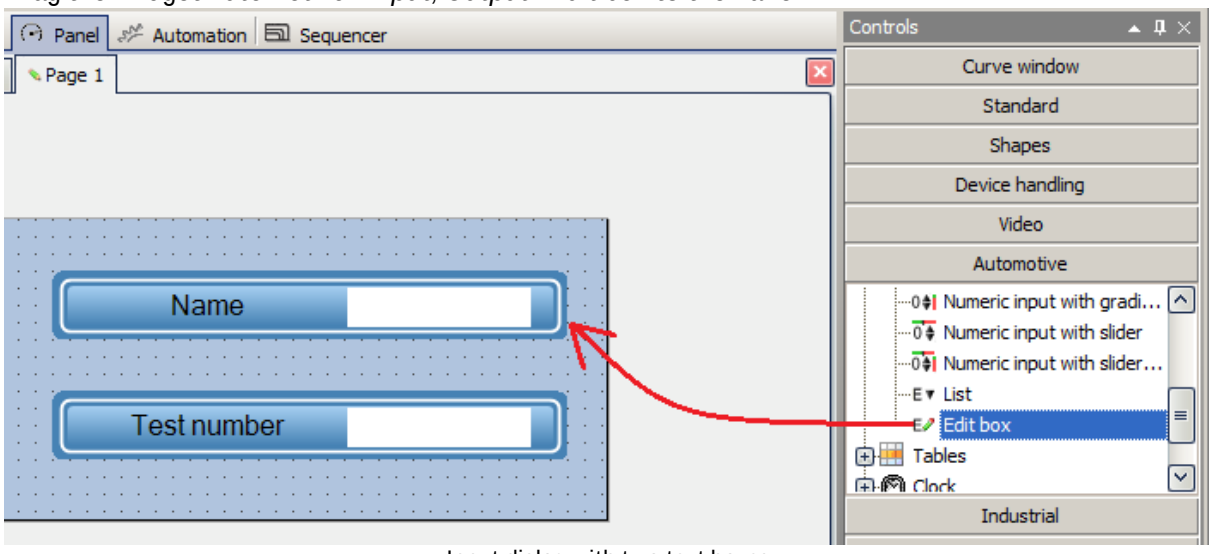
1. Load the experiment "Scroll" if it isn't open yet and save the experiment under the name "Dialog"

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*".



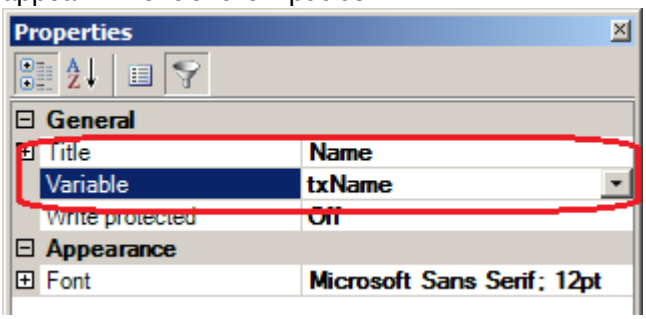
4. Drag the Widget *Automotive - Input, Output - Edit box* to the Panel



Input dialog with two text boxes

5. Set the Widget's properties to :
   Title= Name
   Variable= txName (this is under the 3rd variable created)

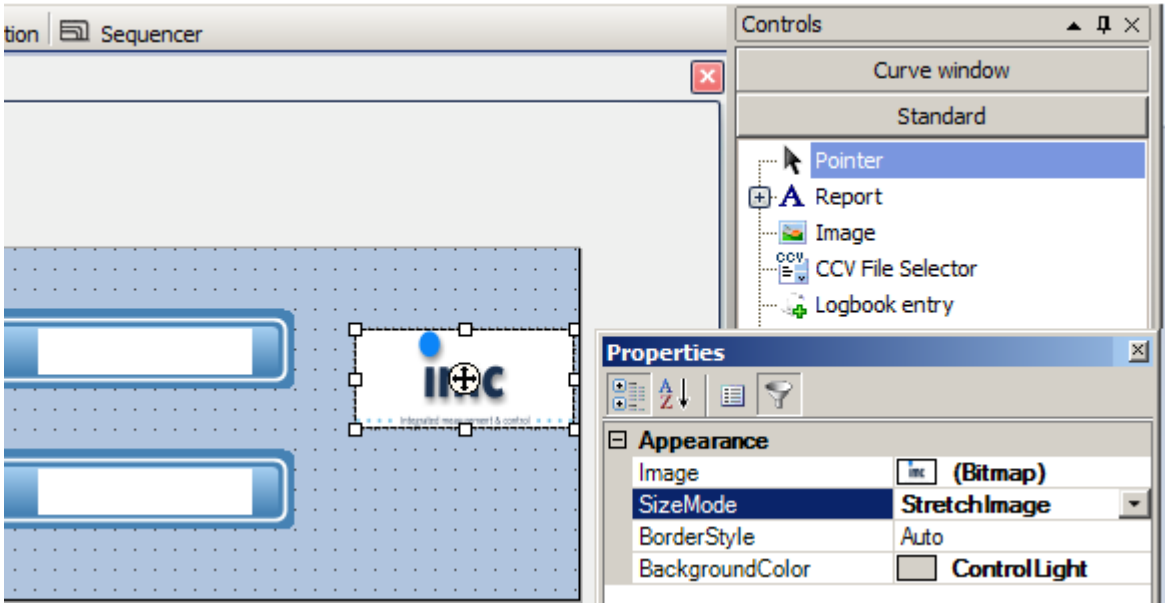
   Click on the small triangle next to the title entry and select the left arrow. Then the name will appear in front of the input box.



6. Repeat the last step with
   Title= Test Object
   Variable= txTestObject

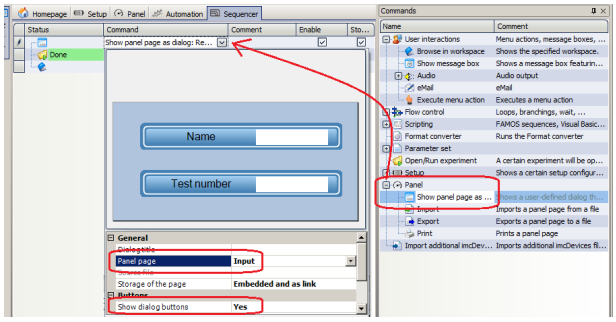
7. Make the Panel page smaller using the sizing handle at the lower right corner.

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*.

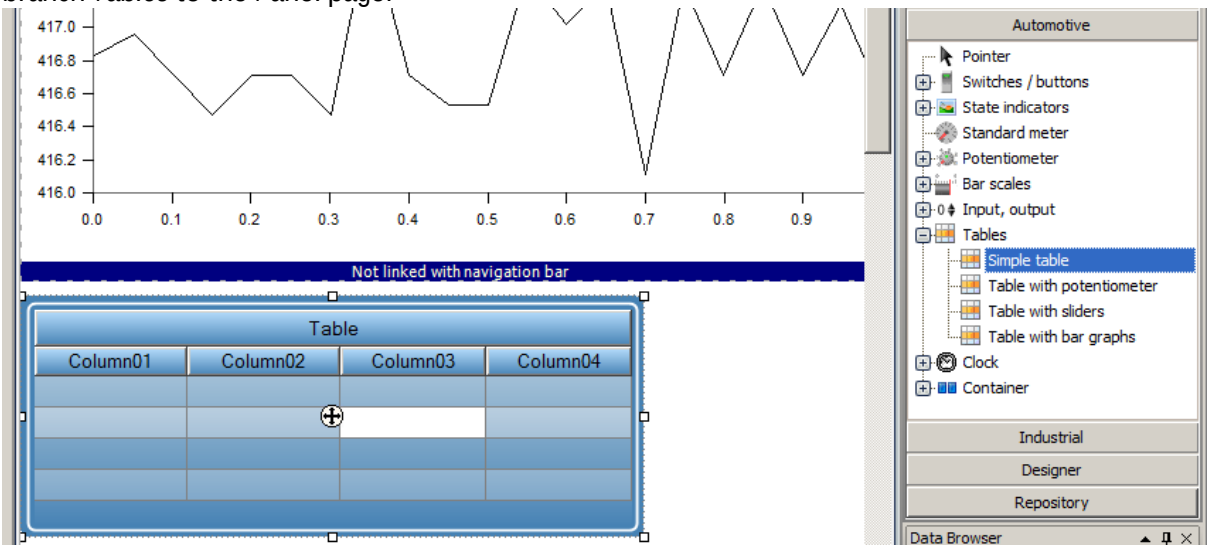


9. Go to the Sequencer

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 *Visible* by entering *Yes*.

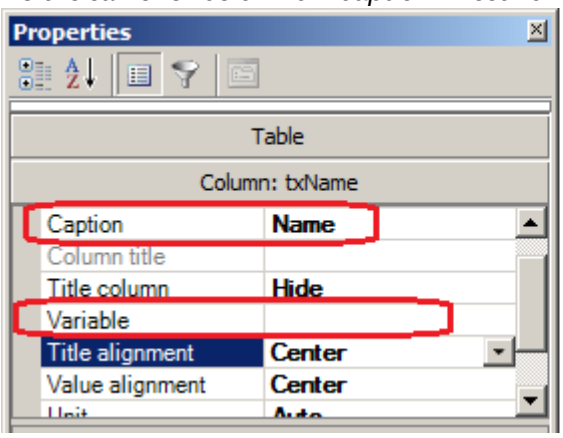


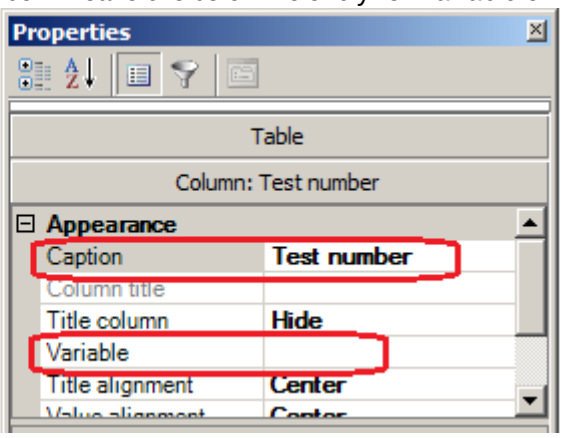
11. Go to the Panel page "Report". From the group *Automotive*, drag the table object *Table* from the branch *Tables* to the Panel page.

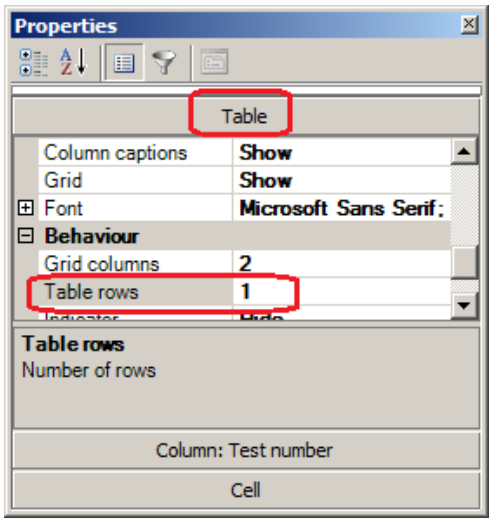


12. Remove *Column03* and *Column04* by pulling them down.
    From the Data-Browser, drag the user-defined variable txName to the first cell in the column (not the column header). Drag the variable txTestObject to the cell in the second column.

    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.

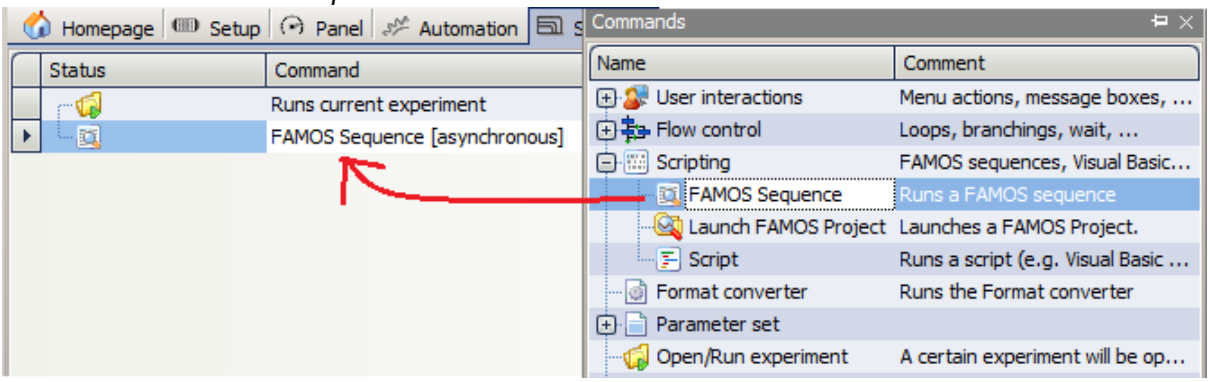
10.7.5 Use of imc FAMOS

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.

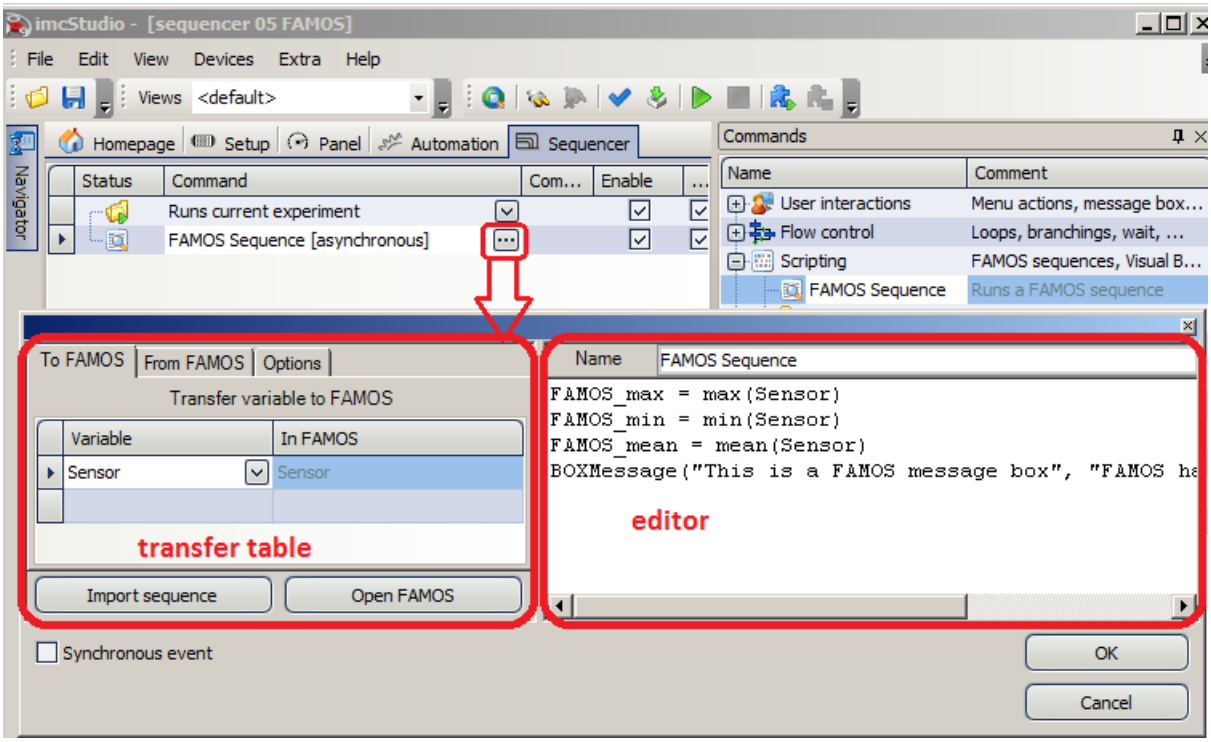
ATTENTION: For this exercise, an installed version of imc FAMOS is required.

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.

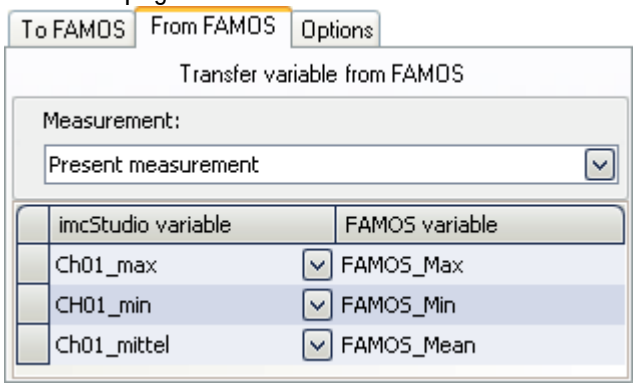
1. Load the experiment "Start Sequencer" if it isn't open already and save it under the name "FAMOS_Analysis".
2. Go to the Sequencer page and select from the commands *Scripting - imc FAMOS Sequence*, and drag it to under "*Runs current experiment*"



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.

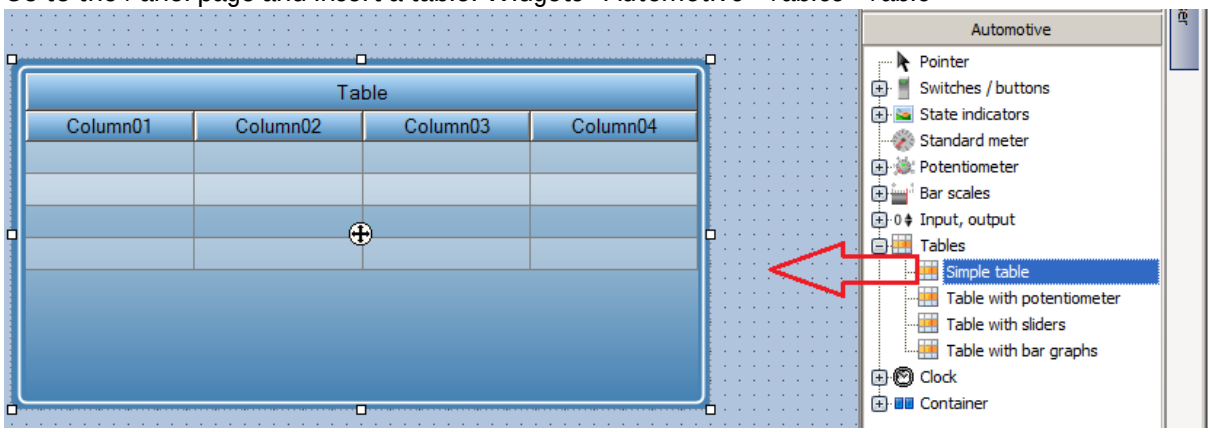


4. Enter the command lines as shown in the picture. As of imc FAMOS 6.1, the language in which the function names are expressed no longer matters.
```
FAMOSMax = Max(Sensor)
FAMOSMin = Min(Sensor)
FAMOSMean = Mean(Sensor)
BOXMessage("This is a imc FAMOS Box","imc FAMOS has done
calculation.","!1")
```
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:

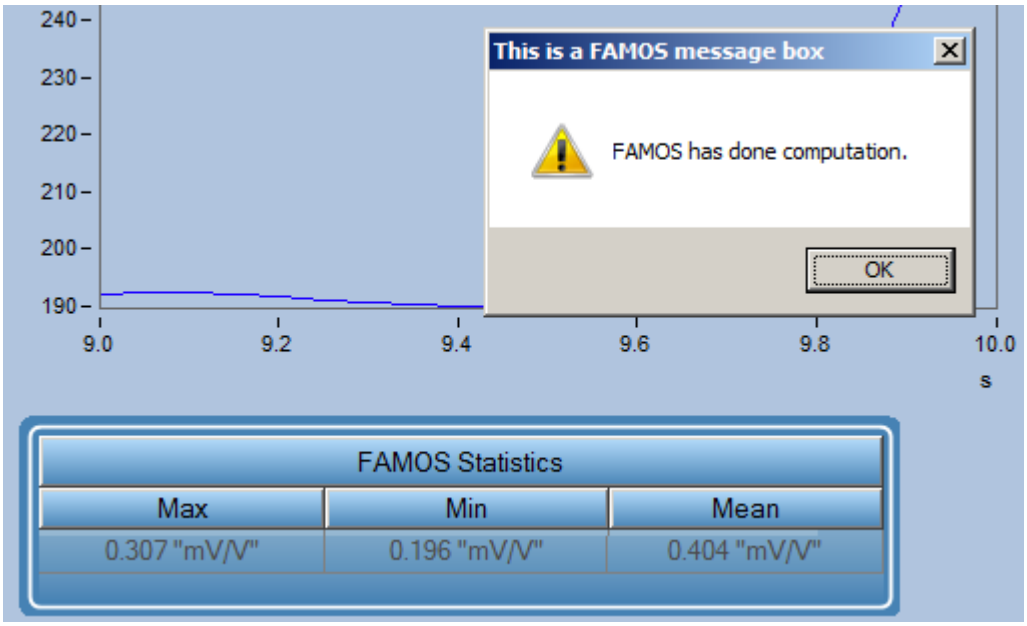


These variable name were already used in the Results table. After the calculations by imc FAMOS, they are automatically applied.

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 and 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.

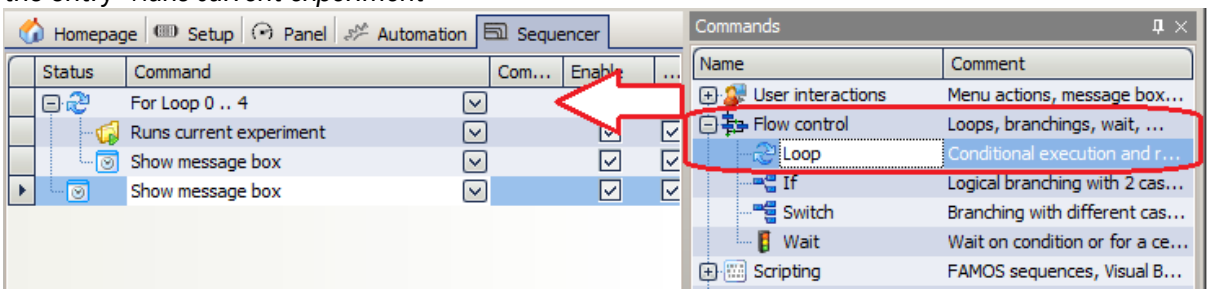
10.7.6 Conditional branches

This exercise shows how the conditional branches (For, While, Switch, If, Else etc.) are used in the Sequencer.

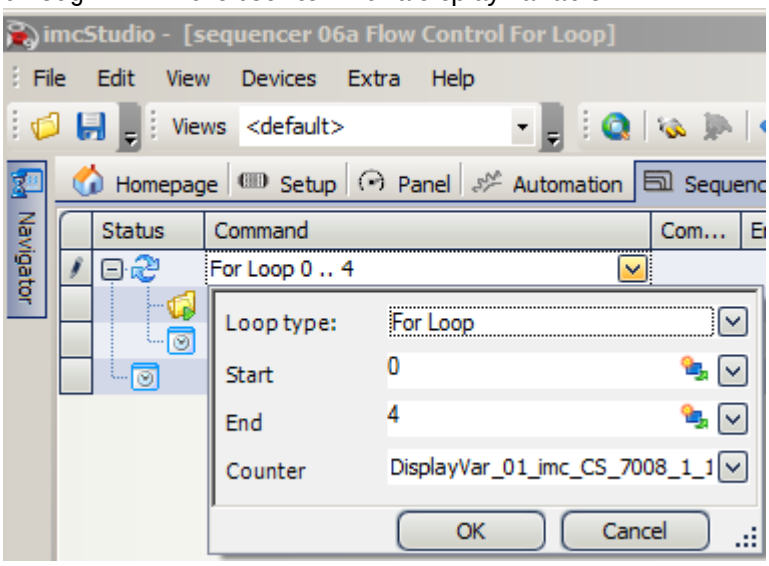
10.7.6.1 For Loop

Assignment: Run the current experiment 4 times in succession. A Seven-segment-displaydisplays the number of runs on a Panel page.

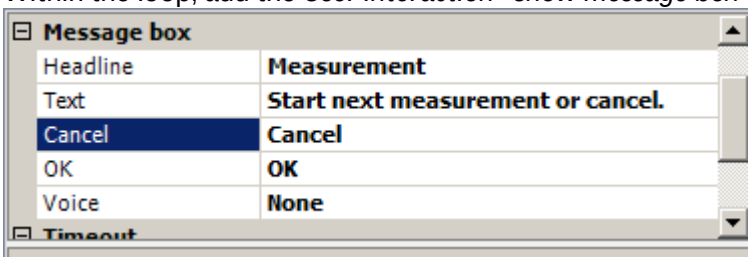
1. Load the experiment "Start Sequencer" and save it under the name "Sequencer_For".
2. Reduce the channel's measurement duration to 3s.
3. Go to the Sequencer and drag from the command group *Flow control* the command *Loop* to before the entry "*Runs current experiment*"



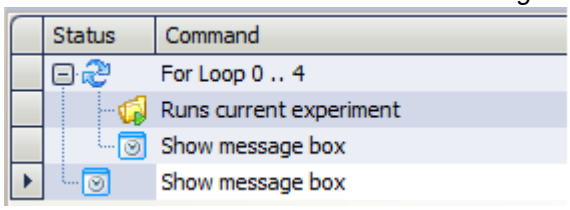
4. Right-click the mouse next to the *For Loop* entry and make specifications for the loop stages 0 through 4. Link the counter with a display variable.



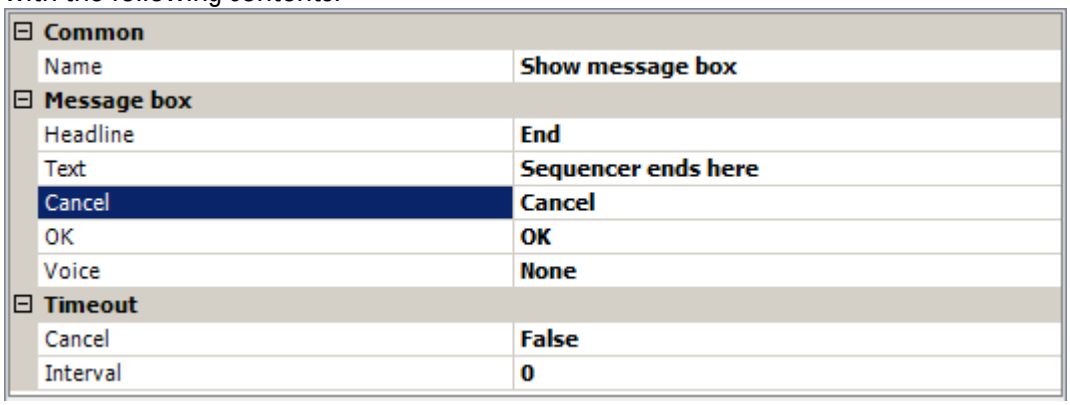
5. Within the loop, add the *User interaction* "*Show message box*". Make the following entries:



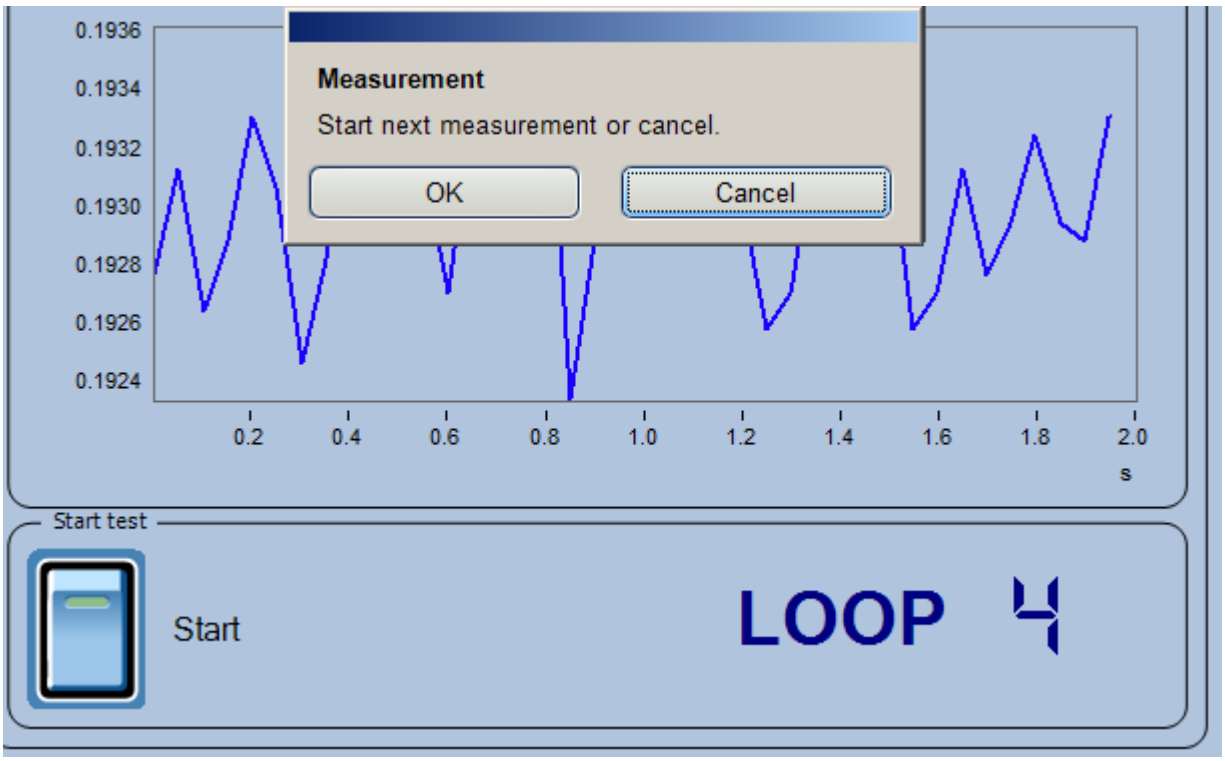
6. Add another *User interaction* "*Show message box*", but after the loop



with the following contents:



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.



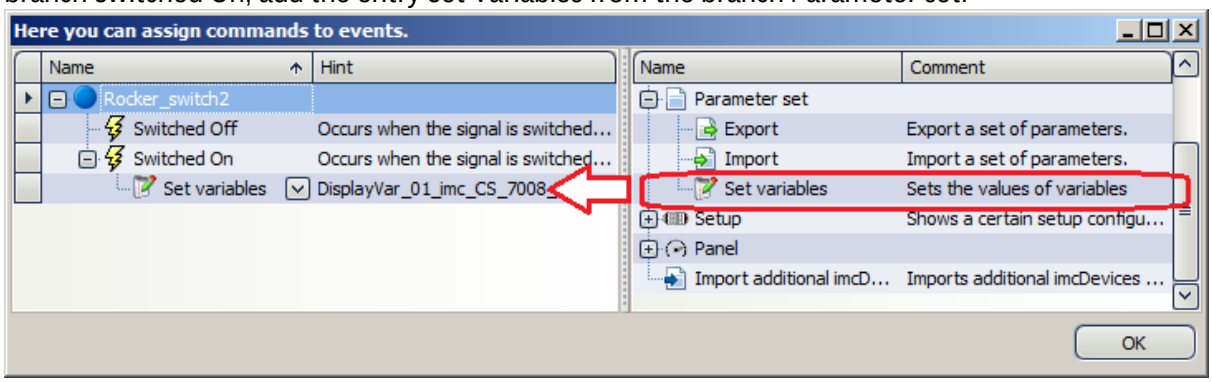
Result: A message appears after each run. The seven-segment-display indicates how many runs have been performed.

10.7.6.2 While Loop

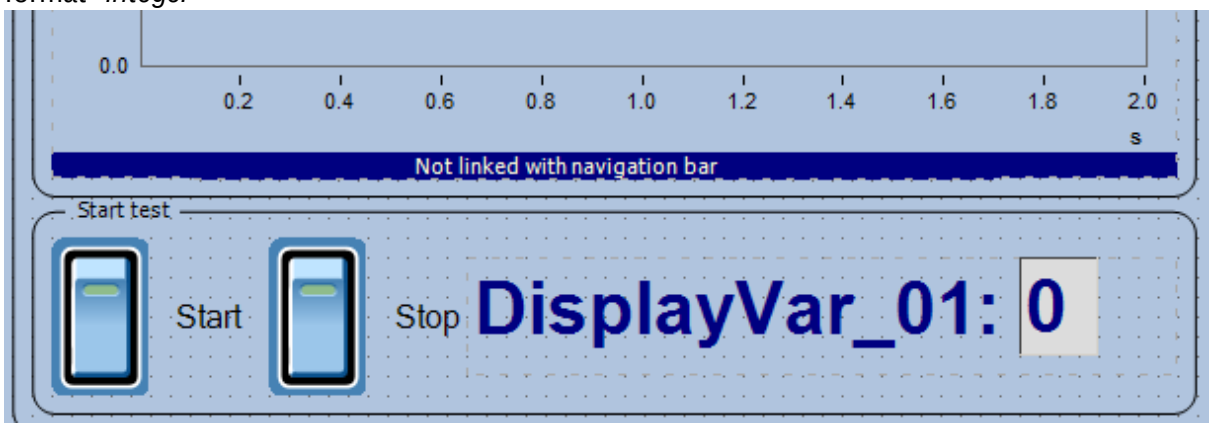
Assignment: Run the current experiment repeatedly until a Display variable contains a nonzero value. Use a pushbutton which assigns a value of 1 to the Display variable when pressed. The Display variable value is displayed on a Panel page.

Based on the experiment "Sequencer_For", an experiment "Sequencer_While" is created.

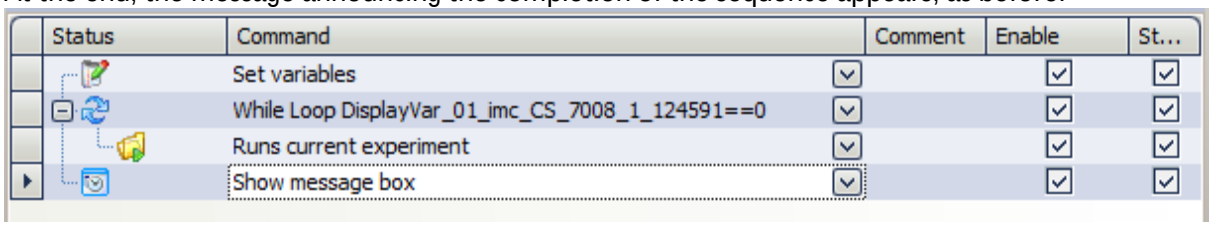
1. Load the experiment "Sequencer_For" and save it under the name "Sequencer_While".
2. Go to the Panel and create a pushbutton. Give it the caption "Stop". Select "Events" and in the branch Switched On, add the entry Set Variables from the branch Parameter set.



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.

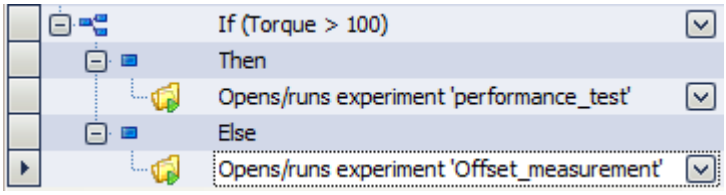
Result: When you click on the Stop button, the Display variable is set to 1 and the measurement does not start again.

10.7.6.3 If - Else

Assignment: The exercise Sequencer_While 1218 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 1213.

One solution could appear as shown below:



10.7.6.4 Switch-Case

Assignment: The exercise Sequencer_While 1218 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 1213.

10.7.7 Example: Wind Power

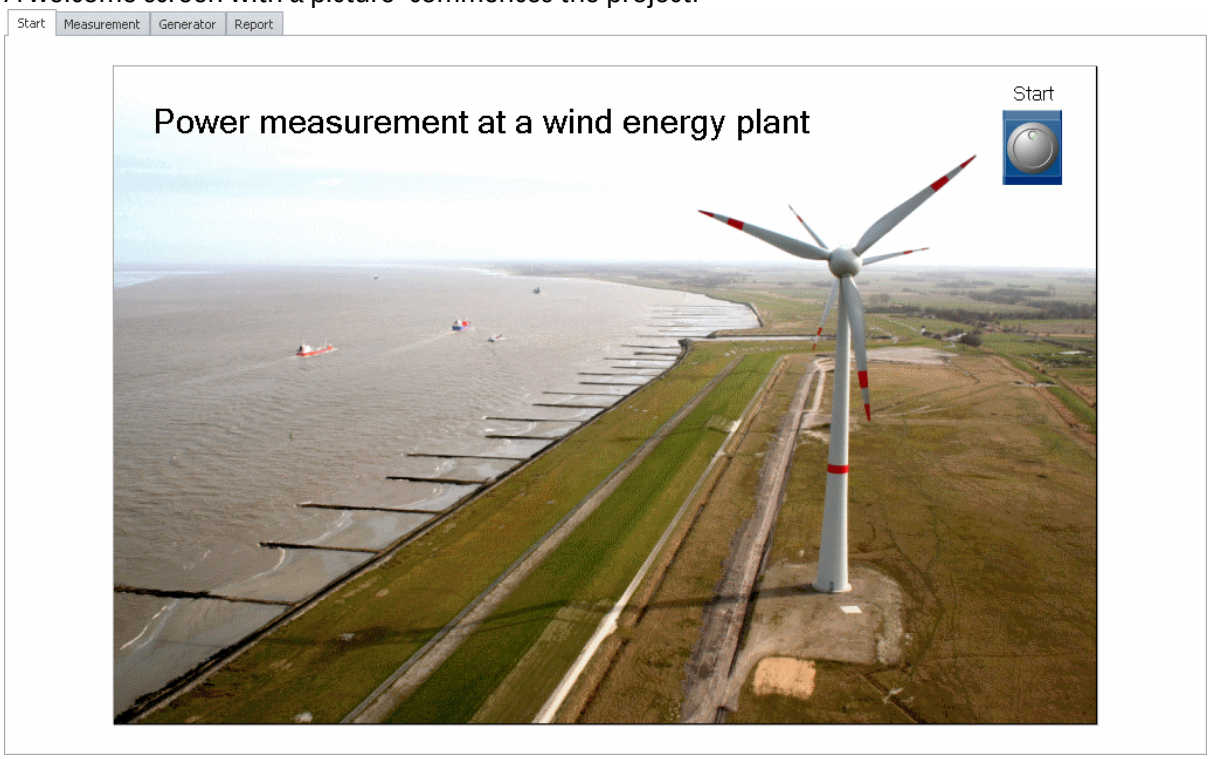
Assignment: Based on the project WEP, an example of skills presented in all the previous exercises working in concert is shown.

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.

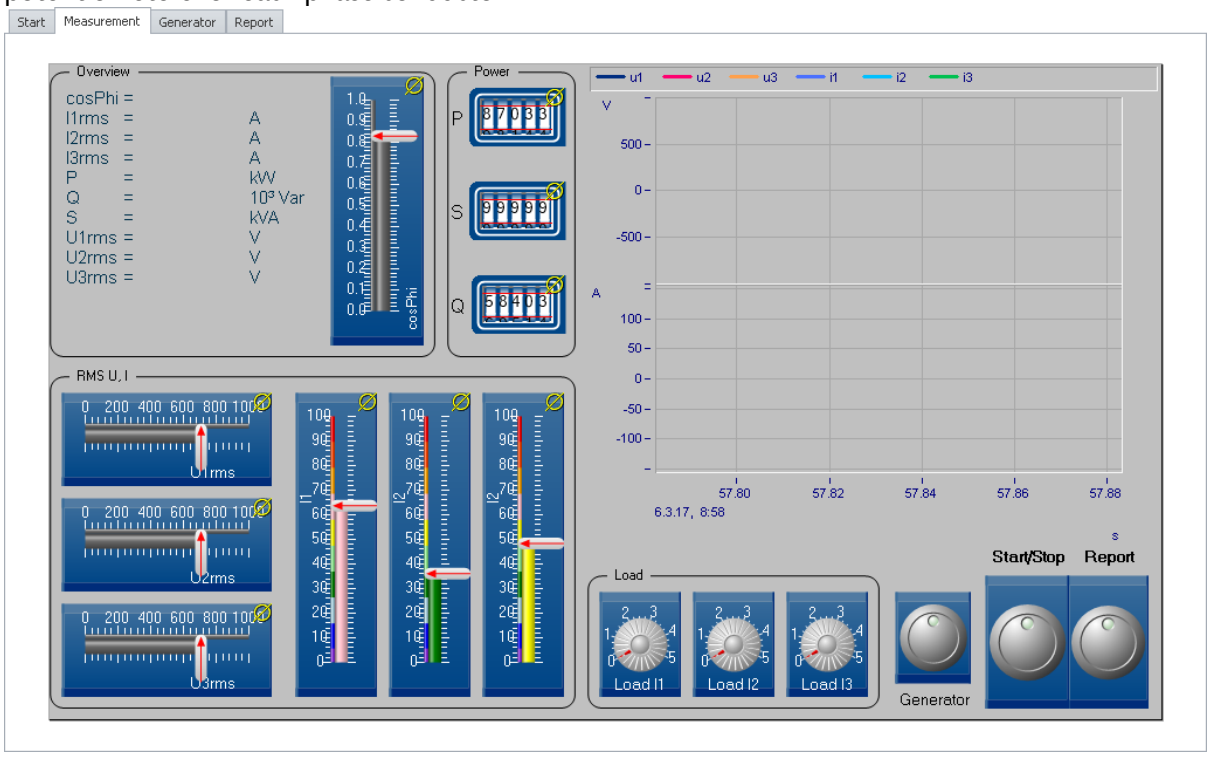
Additionally, a torque (simulated by a tuning fork) is measured, and a temperature.

The following steps are performed:

1. A welcome screen with a picture commences the project.

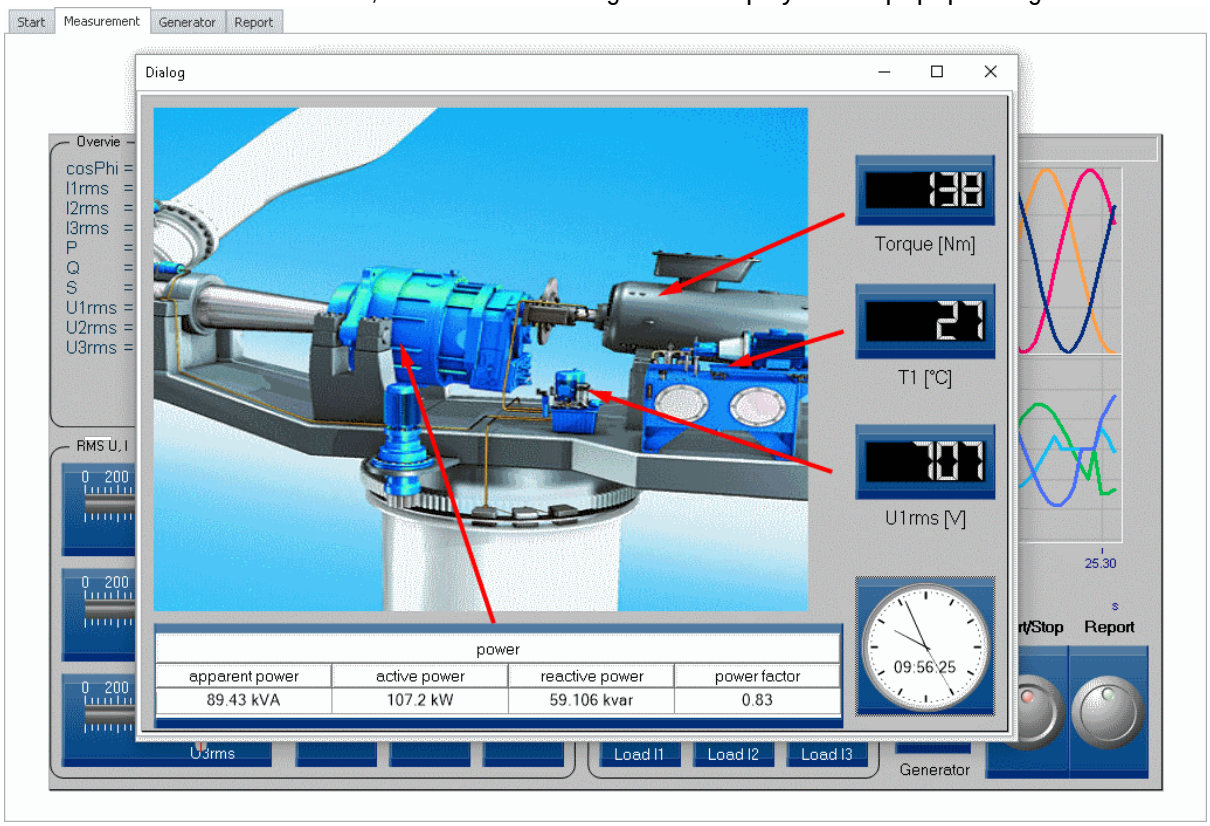


2. On a measurement page, both time and power data are displayed. The load can be set using potentiometers for each phase conductor.

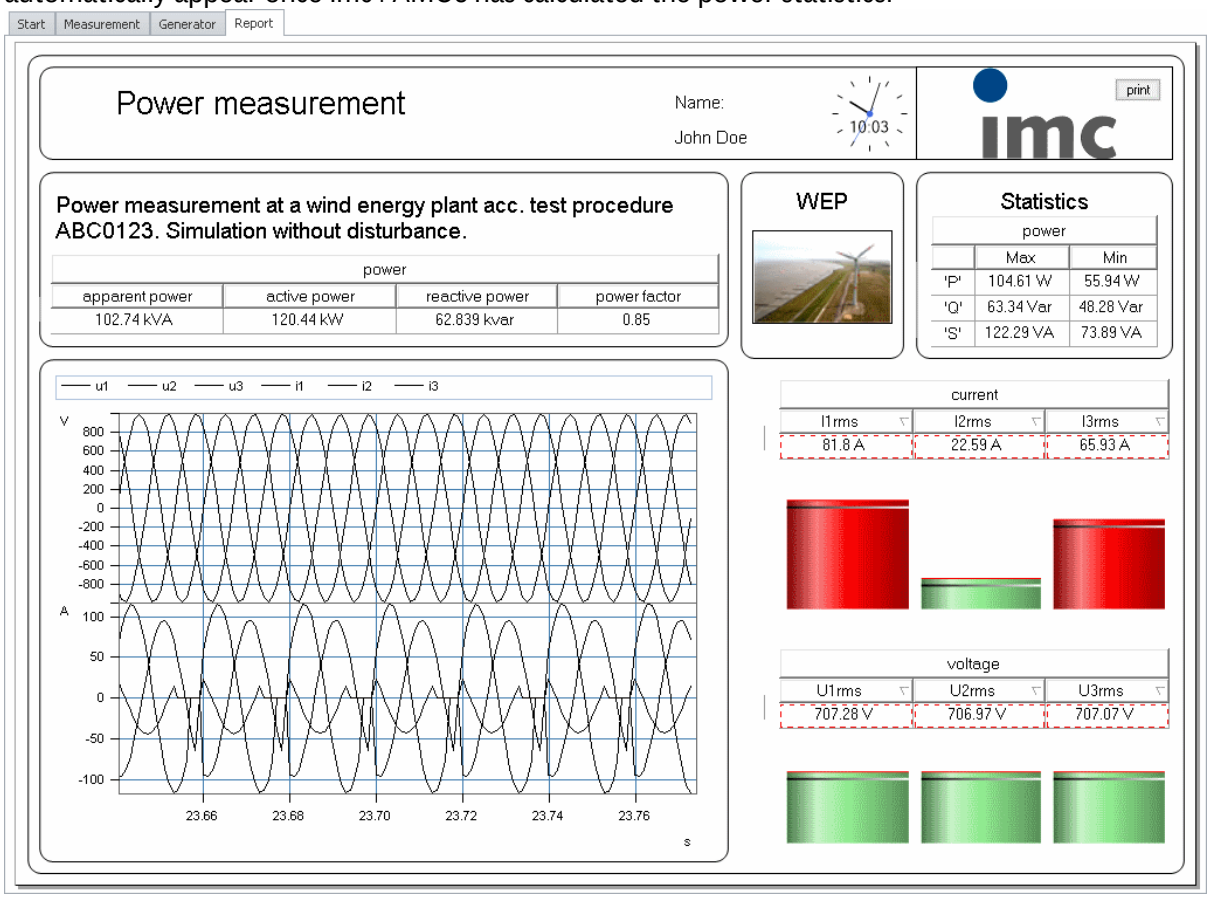


Measurement is started and stopped by a switch. At the end of the measurement, additional results are calculated by imc FAMOS: P max/min, Q max/min and S max/min.

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.



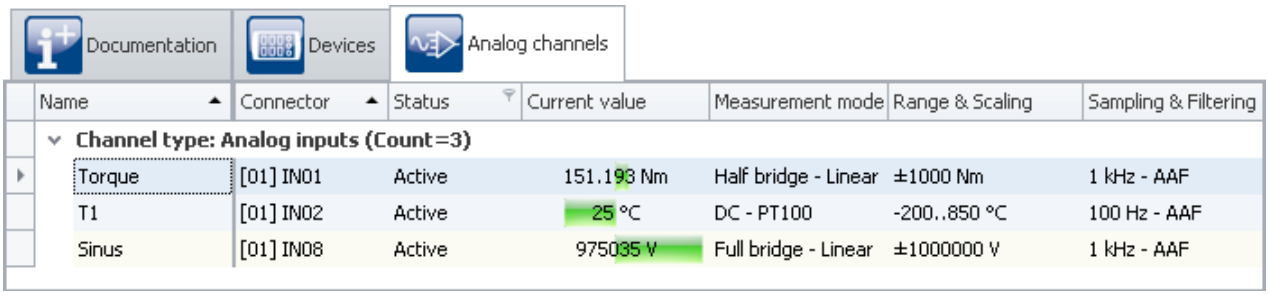
imc Online FAMOS Code: This code is based on the existence of a channel "Sine" which is recorded at a rate of 1kHz.

```
; 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" )
```

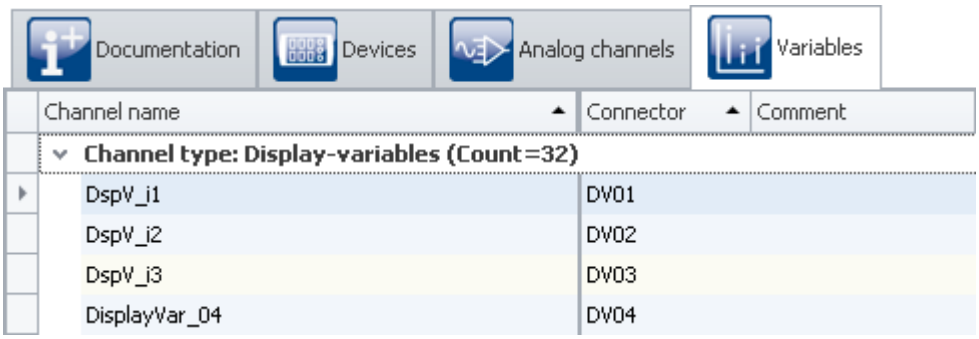
One possible solution is illustrated on the following pages.

10.7.7.1 Setup

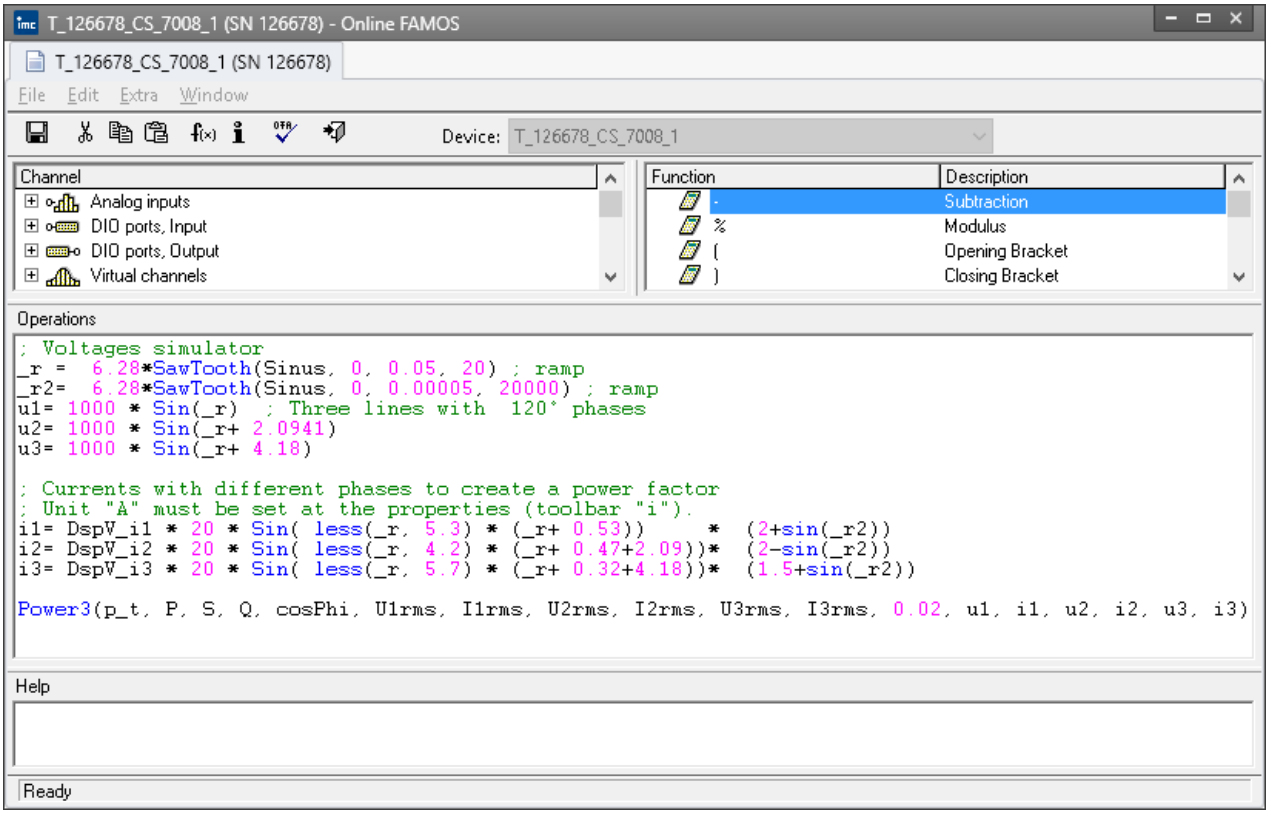
1. Go to *Setup \ Analog channels*
2. First activate three channels. Name one of the channels, which does not need to have an applied signal, "Sine", with a sampling rate of 1kHz.
3. Connect the tuning fork to Channels 1 and 2. Name Channel 1 "Torque" and Channel 2 "T1".
   Settings:
   Torque: Half-bridge, 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.

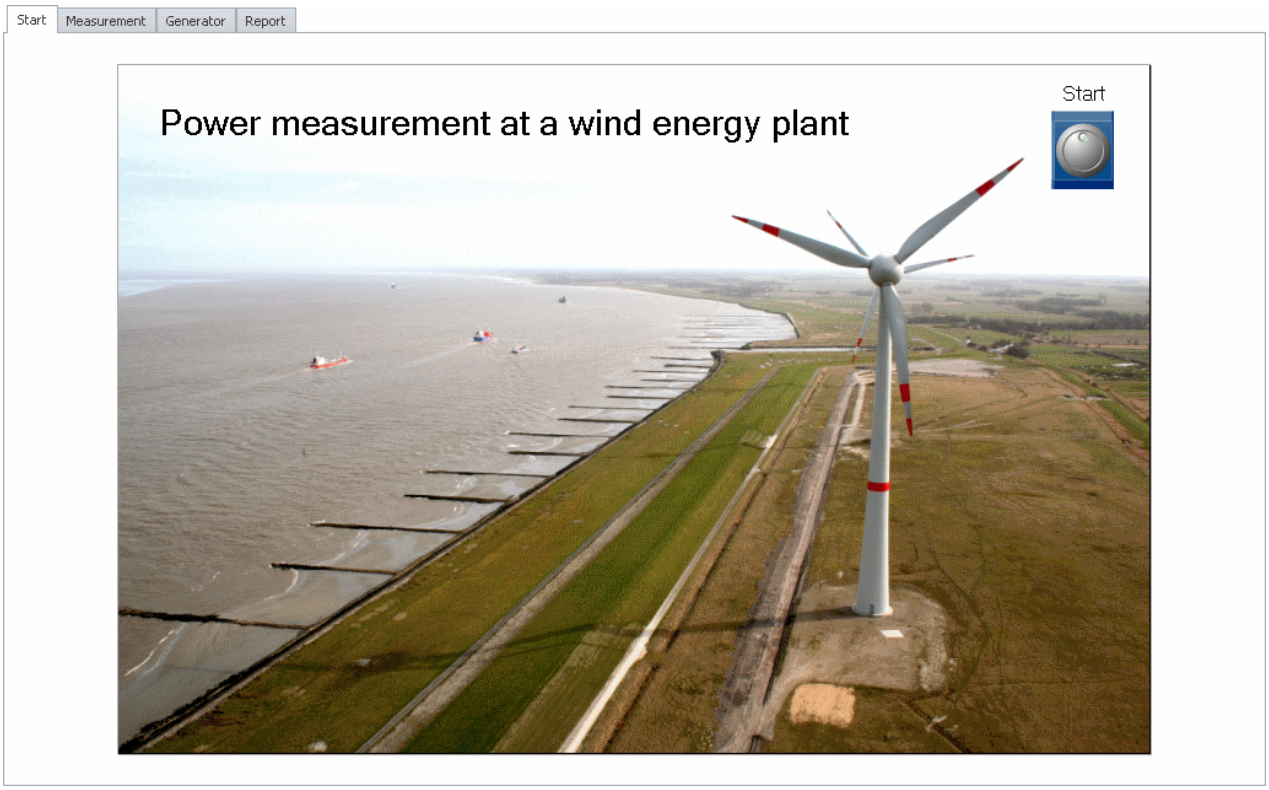
10.7.7.2 Panel pages

All Panel pages in the example were created with the Skin "*Gray color scheme*". 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.

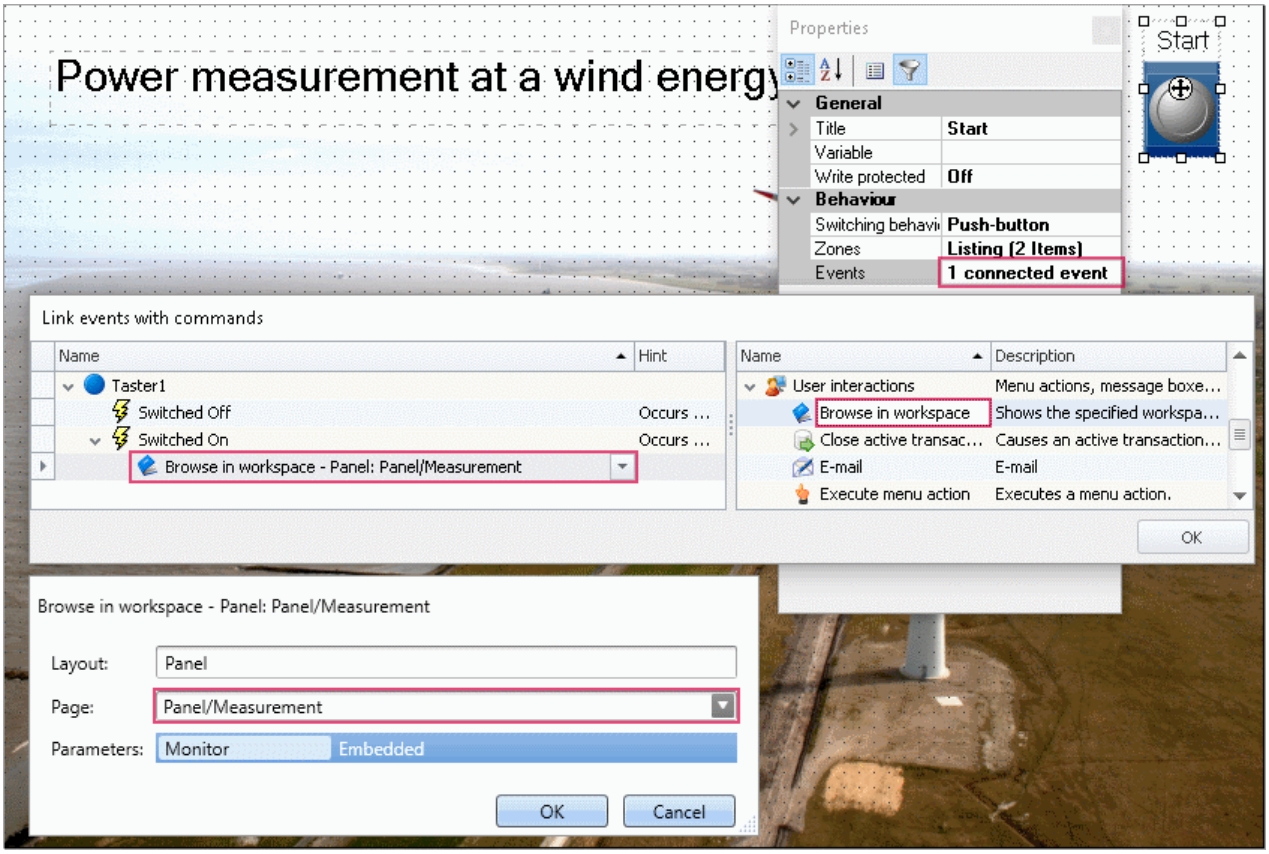
10.7.7.2.1 Panel - Start

The Panel page initially contains only a start button and the picture "Windanlage.jpg".

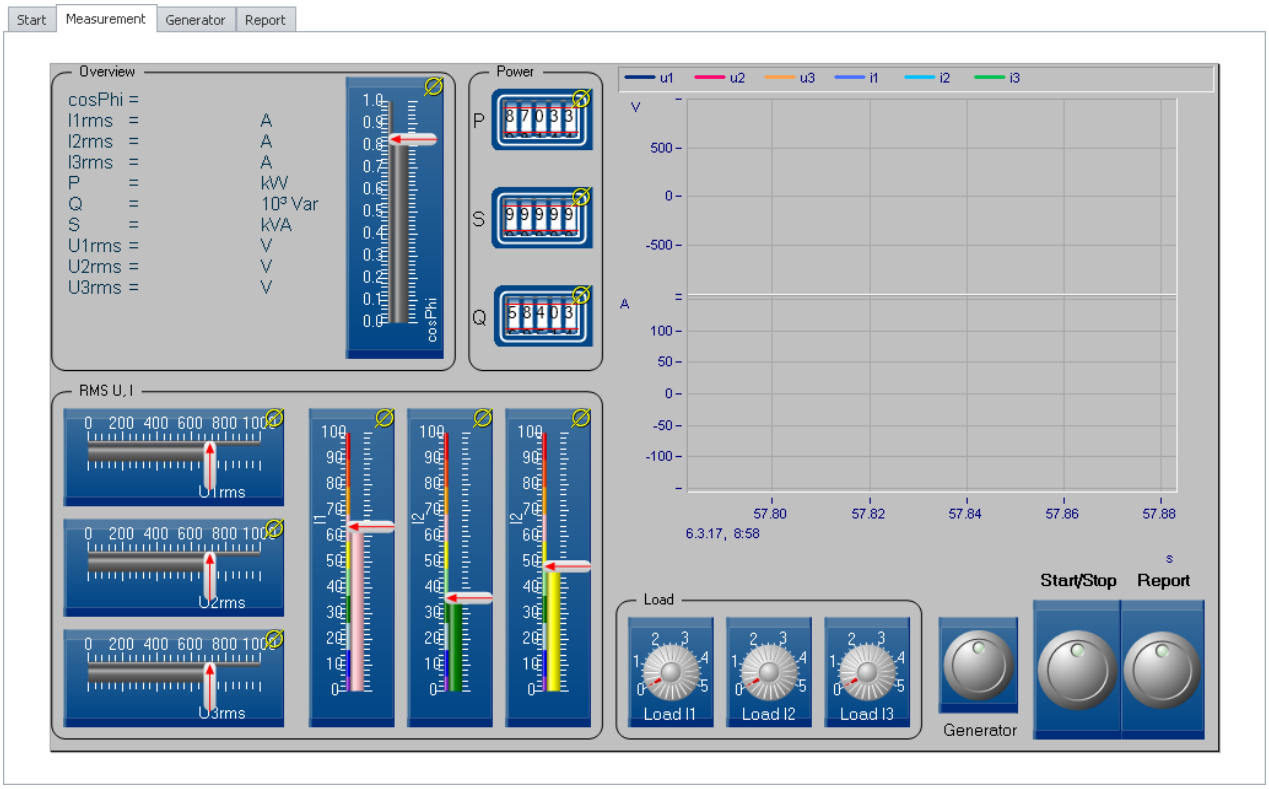


In the example, the Widget "*Industrial - Switch/buttons- Push-button*" is used. Give the *push-button* the caption "Start".

The *push-button* when pressed is to page to the Panel page *Measurement*. This is arranged by setting the *Events* in the button's Properties:



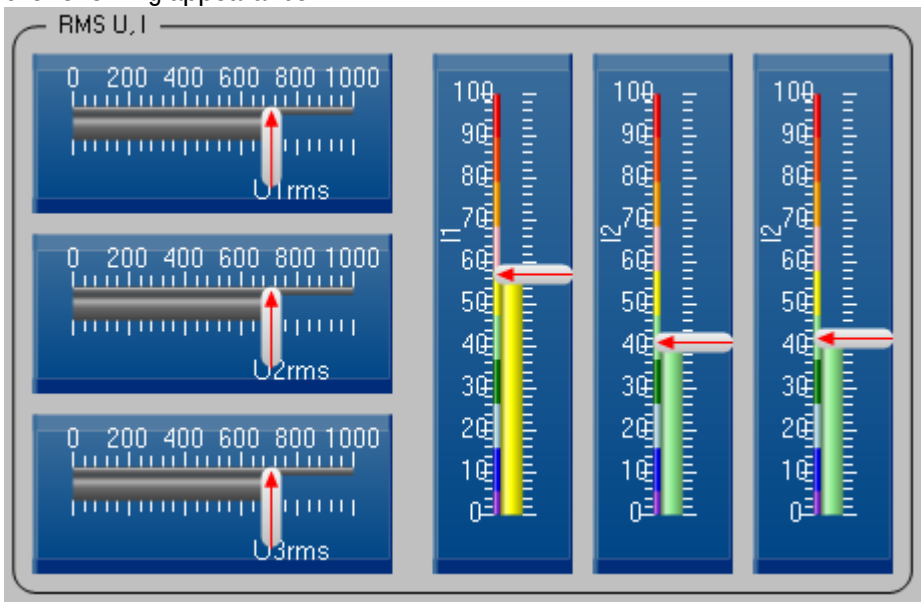
10.7.7.2.2 Panel - Measurement



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.

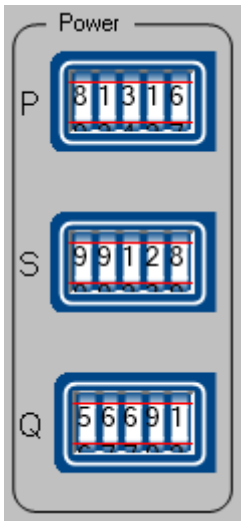
1. Begin with the curve windows. The window at right shows the instantaneous U and I values. Determine the variable. Create two Y-axes by first dragging thre voltages to the Y-axis. Then drag the three currents to under the X-axis. Add a grid.
2. Set up the left curve window. Drag the power signals, 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*.



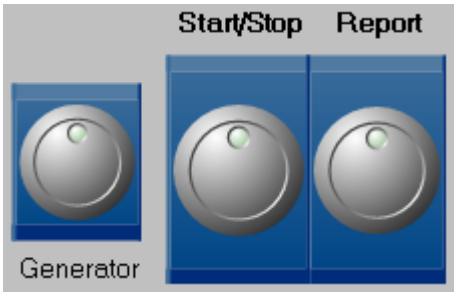
A *GroupBox* is dragged from the group *Shapes* to a position around the power signals.

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.



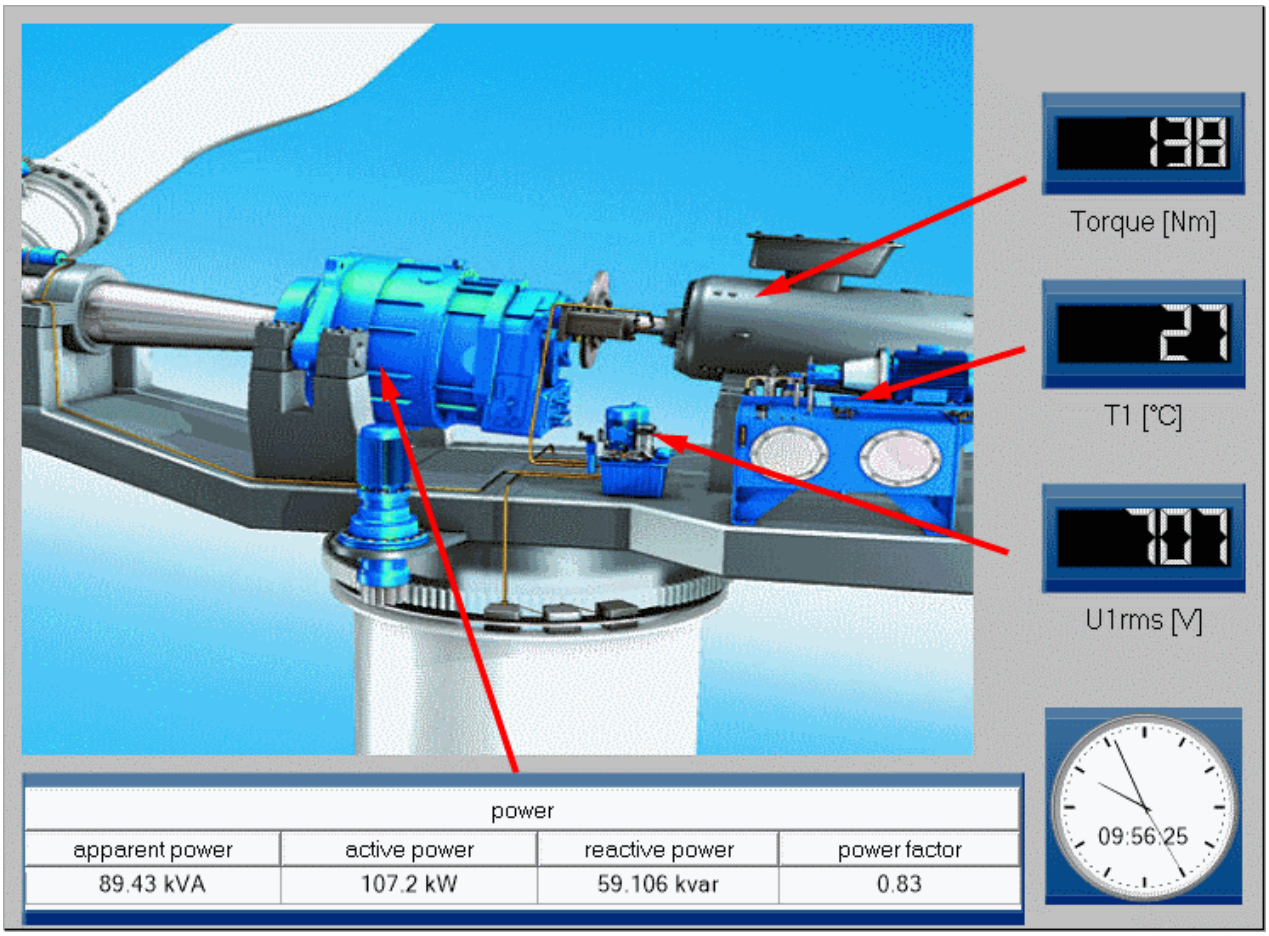
The Generator-push-button displays the dialog "*Generator*" as a popup dialog. If you have not yet created one, do so now. Ensure that the title "Generator" is visible. The push-button's events for the branch *Activated* are set as "Panel" - "Panel *page as dialog*". Click on the arrow next to "Panel *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.

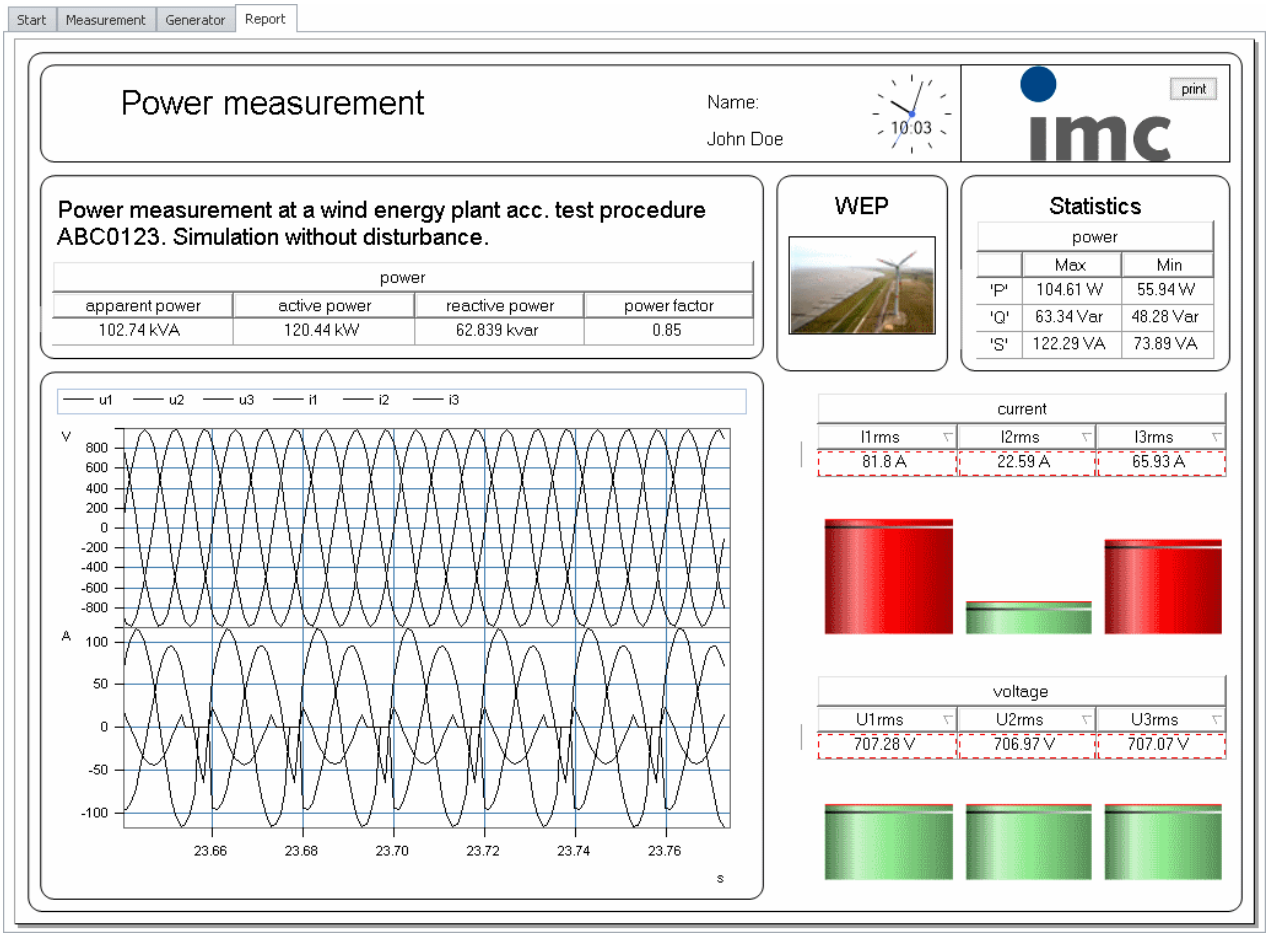
10.7.7.2.3 Panel - Generator

The Generator page is displayed as a popup dialog from the Panel page Measurement.



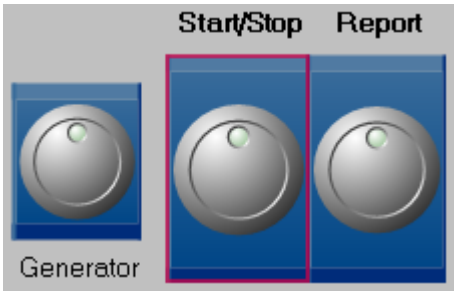
1. This page shows an additional picture "Generator.jpg".
2. From the group Industrial, table for the power signals is inserted. The row count is reduced to 1. By means of Drag&Drop, the cells are filled from the Data Browser with power values. Also, enter headers for the column and table.
3. Add three "*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.

10.7.7.2.4 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.

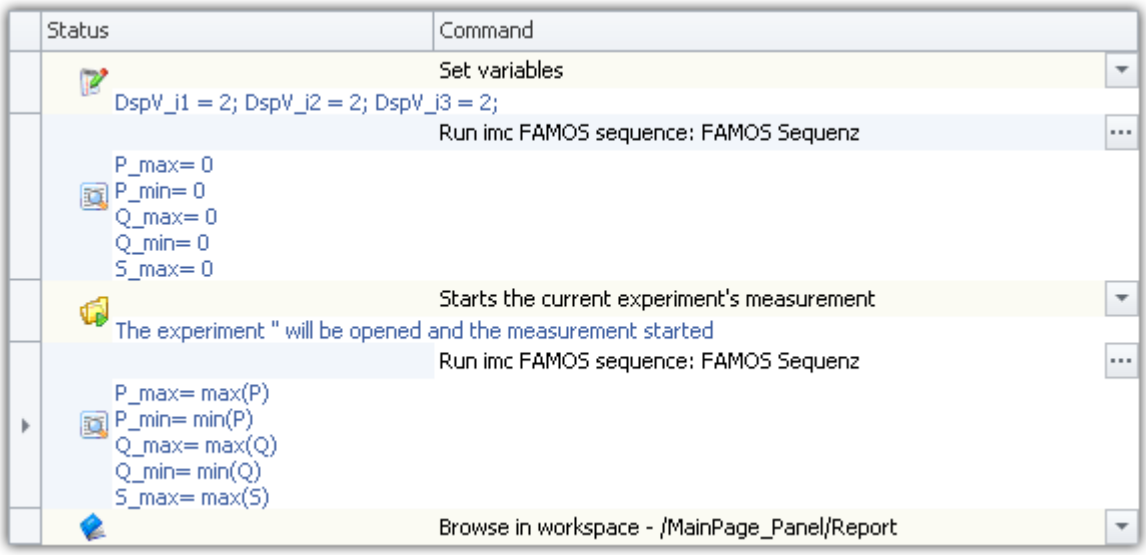
10.7.7.3 Sequencer



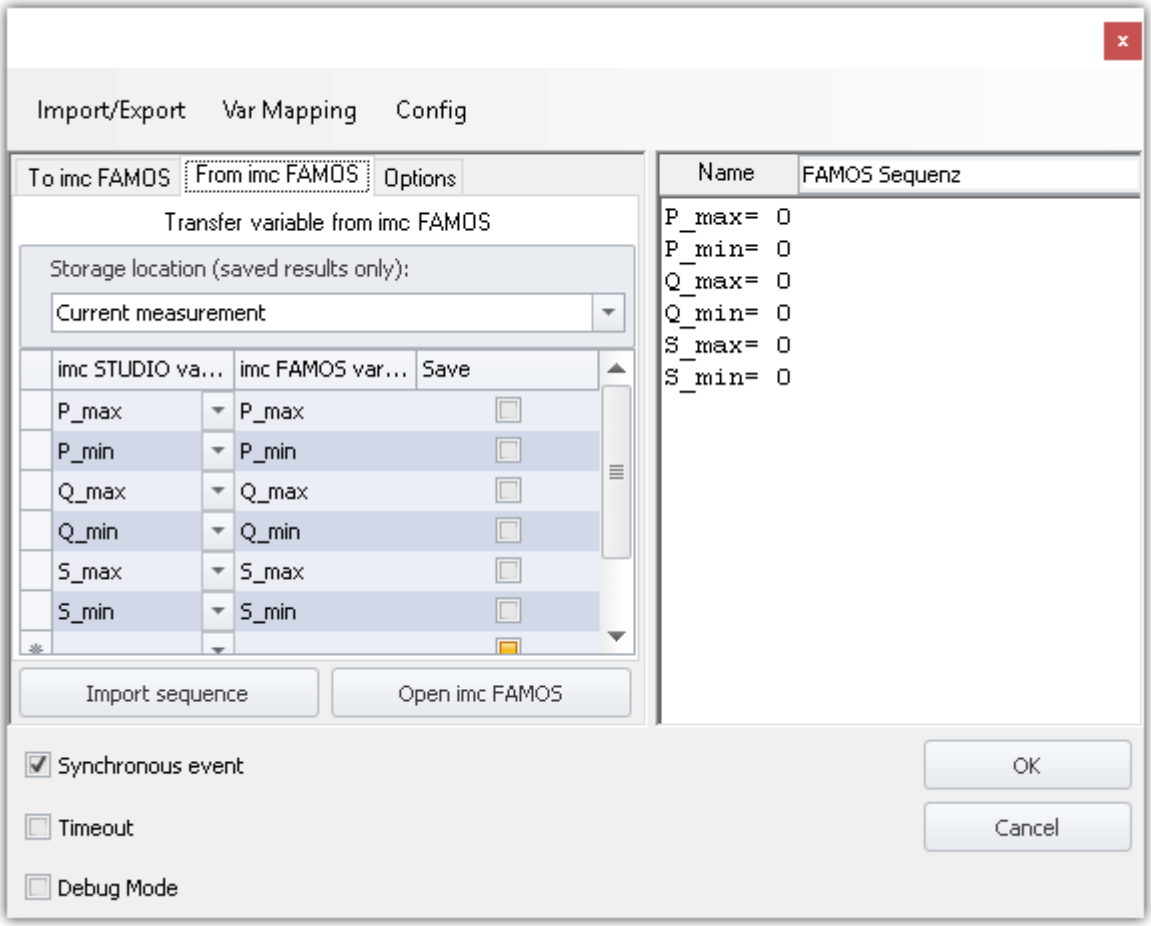
To start the Sequencer push the *Start/Stop* button in the Measurement Panel (see in fig.).

The following steps are now carried out automatically:

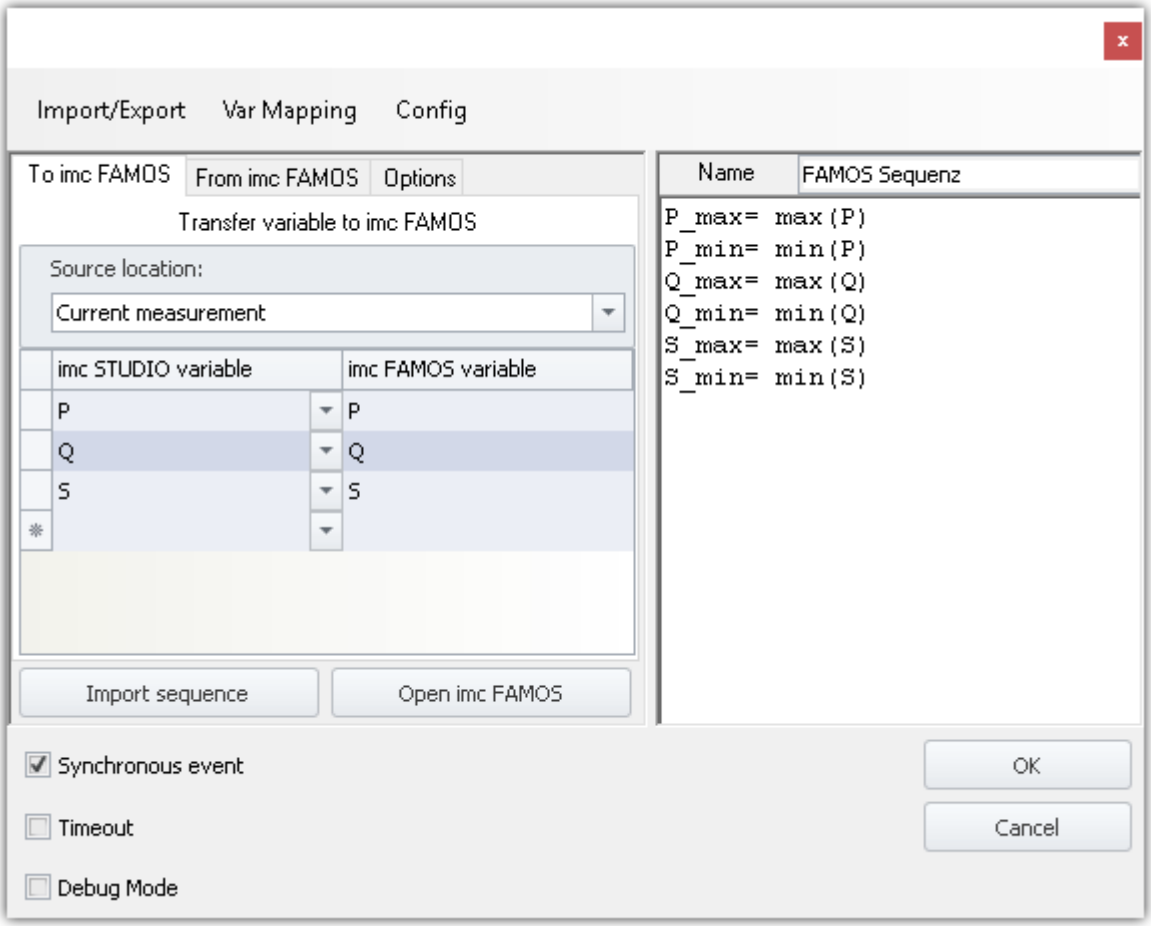
- initializing the variables to be transferred to Famos directly after the measurement
- measurement starts
- measurement stops (*Start/Stop* switch in panel)
- Famos sequence runs. This computes P_max, P_min, Q_max, Q_min ..... etc.
- report page pops-up, showing the calculated results.



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.

11 Monitor

imc STUDIO Monitor makes it possible to connect with one or more measurement devices during a running measurement, particularly for the purpose of observing the current measurement data. Data can be viewed and edited live on multiple workstations.

imc STUDIO Monitor is an independent installation variety of imc STUDIO. This means it is installed and started as a separate application. For Monitor, separate licensing is required accordingly (an edition of imc STUDIO + imc STUDIO Monitor).

The fundamental elements are imc STUDIO's Panel 813, as well as the tool window Devices/Channels 1236.

11.1 Ribbon

Home

	Menu item	Description
	Device Search (imc STUDIO Monitor)	The system searches for all imc devices in the network. The amount of time this takes depends on how many devices are connected and on the network type. Upon conclusion, the devices found are listed under the tab: Devices in the tool window: Devices / Channels. For more information on the correct configuration of all network settings, see the manual: Setup - Connect the device 295.

Other unassigned menu items

	Menu item	Description
	Activate saving (imc STUDIO Monitor)	Activates the saving of all channels selected. (See also the description of the respective symbol in the tool window Devices / Channels 1236 ( ))
	Deactivate saving (imc STUDIO Monitor)	Deactivates the saving of all channels selected. (See also the description of the respective symbol in the tool window Devices / Channels 1236 ( ))

11.2 System Prerequisites and Limitations

Measurement device requirements
The same hardware prerequisite apply as for imc STUDIO Setup.
For the purpose of connecting with imc STUDIO Monitor, the devices must additionally have at least 32 MB of internal device memory available.
Accessing the device: at most, • three imc STUDIO Monitor PCs can be directly connected with one device (SN 12xxx), or • four imc STUDIO Monitor PCs can be directly connected with one device (as of SN 13xxx).

Without imc STUDIO Project Management

In imc STUDIO Monitor , the component "imc STUDIO Project Management" is deactivated. For this reason, there is no database as in imc STUDIO. The dialogs for saving and loading experiments correspond to the standardized "Saved As"-/”Open”-dialogs.

Experiments can be saved at any desired location. The saved measurement data are stored in the associated experiment folder.

Summary of the limitations:

- No use of one's own database
- No compilation in projects having settings in common
- No display of saved measurements in the Data Browser

11.3 Tool Window

11.3.1 Devices / Channels

In this tool window, you can search for and select devices, as well as select channels which you wish to view using Monitor. Also, you can set the circular buffer time for each channel separately and activate/deactivate the data saving.

The window contains a toolbar and three tabs, which contain the respective views for the device/channel selection and additional options.

Toolbar

	Menu item	Description
	Device Search (imc STUDIO Monitor) Network search	The system searches for all imc devices in the network. The amount of time this takes depends on how many devices are connected and on the network type. Upon conclusion, the devices found are listed under the tab: Devices in the tool window: Devices / Channels. For more information on the correct configuration of all network settings, see the manual: Setup - Connect the device 295.
	User-defined Device	Using either the IP-address or the domain name (DNS-name), it is possible to establish a connection to a device without a device search. This is necessary with some structured networks (with routers, Internet, ...), when imc devices can not be integrated by means of a network search.
	Refresh	Removes all channels belonging to de-selected devices from the Channels list.
	Save	Activates/deactivates the saving of all selected channels (corresponds to the menu actions Activate/Deactivate saving (imc STUDIO Monitor) 1235). When data saving is activated, a measurement folder is created having the current time stamp (or according to the data saving options set). All selected channels are saved. If data saving is deactivated again, the measurement folder is closed. Upon each new activation, a new folder is created.

Tab: Devices - Select device

Parameter	Description
Monitor	By clicking on the desired devices *Monitor* box, a connection to the device is established. You can also select multiple devices. Only such devices can be selected which have been initialized for a measurement or on which a measurement is already running. If this condition is not met, the selection is automatically deleted.
Name	Name of the device.
Device status	Indicates the connection status and the device status. E.g. "connected; running"; This means Monitor is connected with the device and the measurement is running.
Serial number	The device's serial number.

Tab: Channels - Selecting channels

Parameter	Description
Monitor	Clicking on the box *Monitor* causes the variable to be selected. Data transfer for the variable begins immediately.
Name	Name of the variable.
Comment	Comment on the variable.
Unit	The variable's unit.
Circular buffer time	All accruing data are held in a circular buffer memory are. This parameters lets you determine how long the maximum time period for this circular buffer memory is. Note: The circular buffer memory only applies to the display in the curve window and does not affect the storage of the measured data!

Tab: Options

Storage options

Parameter	Description
Folder designation	Using this parameter, you set the folder structure for the measured data. • Timestamp - date time (measurement number) The path name is formed from the start time • Continuously numbers
Data folder	Current storage area for measured data.
Saving interval / Amount of saving interval	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. 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.

In order to apply changes to the options, press the button "Apply".

11.4 Information and Tips

Placeholder EXPERIMENT.PATH

The placeholder EXPERIMENT.PATH can now also be used in imc STUDIO Monitor, in order to find the path to the experiment file.

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.*

11.5 Tutorial

Below we present some examples of the plug-in Monitor.

11.5.1 Monitoring - First Steps

imc STUDIO Monitor can connect with imc devices as soon they were initialized or a measurement is running.

Proceed as follows:

- Run a device search to obtain a list of all available devices.
- Select the device to obtain the list of all device variables.
- Select the channels you wish to observe.
- Configure a Panel page, in order to view the measured data (no further explanation presented here).

Device Search

To search for devices available in the network, click on the device search symbol ( )

- in the tool window: Devices / Channels or
- in the ribbon: Home (or Panel-Control) > Device Search (imc STUDIO Monitor).

Note: Connecting with the device

The same conditions for a successful search apply as for imc STUDIO. If a device is not found, there are a variety of possible causes.

For more information on correct configuration of all network settings, see the manual: Setup - Connect the device 295.

Tab: Devices - Device selection

Once the device search has concluded, you can connect with the desired device.

- Go to the main window Panel
- In the tool window Devices / Channels, open the tab Devices

Here you will find the list of all devices found.

- Select the desired device by setting a checkmark in the column Monitor (☑).

Note: Firmware

In order to be able to connect with a device, the firmware version (imc DEVICES) must be installed which matches the device.

The firmware must not only be present on the imc STUDIO computer for configuring the device, but also on the imc STUDIO Monitor computer, in order for it to read from the device.

Transfer of measured data begins with the selection of a device, as long as at least one channel is selected on the tab *Channels*.

If the device is de-selected, transfer of the measured data stops.

Tab: Channels - Channel selection

To select the channels you wish to see, open the tab Channels in the tool window Devices / Channels.

In the default setting, the channels are grouped according to the devices selected.

- Activate the channel desired by selecting it in the column Monitor (☑).

Note

If no channel is selected here, none of this channel's data are transferred.

If a comment was assigned to the channel in imc STUDIO, it is also displayed, in the *Comment* column. As well, the column *Unit* displays the unit of the measured data.

Circular buffer time

All incoming data are kept in a circular buffer memory. The volume of this buffer memory can be set for each channel separately in the column *Circular buffer time*.

Note: Circular buffer time

The circular buffer memory only applies to the display in the curve widow and does not affect storage of the measured data!

Activating data storage

By default, data storage is deactivated. You can activate the data storage for all channels selected.

Note: Saving the experiment is necessary

Since the measured data are saved with the experiment, a defined storage location is needed.
Save the experiment, otherwise the button for activating the data storage is deactivated.

To activate the data storage,

- click on the button: Save 1236 ( ) in the tool window: Devices / Channels or
- the menu action: Activate saving (imc STUDIO Monitor) 1235 ( ).

To deactivate the data storage,

- click again on the button: Save 1236 ( ) in the tool window: Devices / Channels or
- the menu action: Deactivate saving (imc STUDIO Monitor) 1235 ( ).

A variety of options is available for the data storage. The description of all storage options is found in the description of the tool window: Devices / Channels 1237.

12 Data Processing

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 1243	Processing and analysis of measured data during running measurement
Power Quality 1253	Expansion package for power quality analysis as per EN 50160 (IEC 61000-4-30 Class A)
imc STUDIO BusDecoder 1269	Expansion package for decoding Field-bus log channels
Powertrain Monitoring 1269	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.

12.1 Ribbon

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.

12.2 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 1248:

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 Trigger_48 (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.

Prerequisite knowledge for using this documentation:

imc Online FAMOS and imc Inline FAMOS basically differ only in their user interfaces and in the medium which performs the calculations. For this reason, no more on the topic of imc Inline FAMOS's functioning will be discussed.

Learning how to operate imc Inline FAMOS by means of this documentation requires familiarity with imc Online FAMOS. This documentation only describes the user interface, and the differences from imc Online FAMOS.

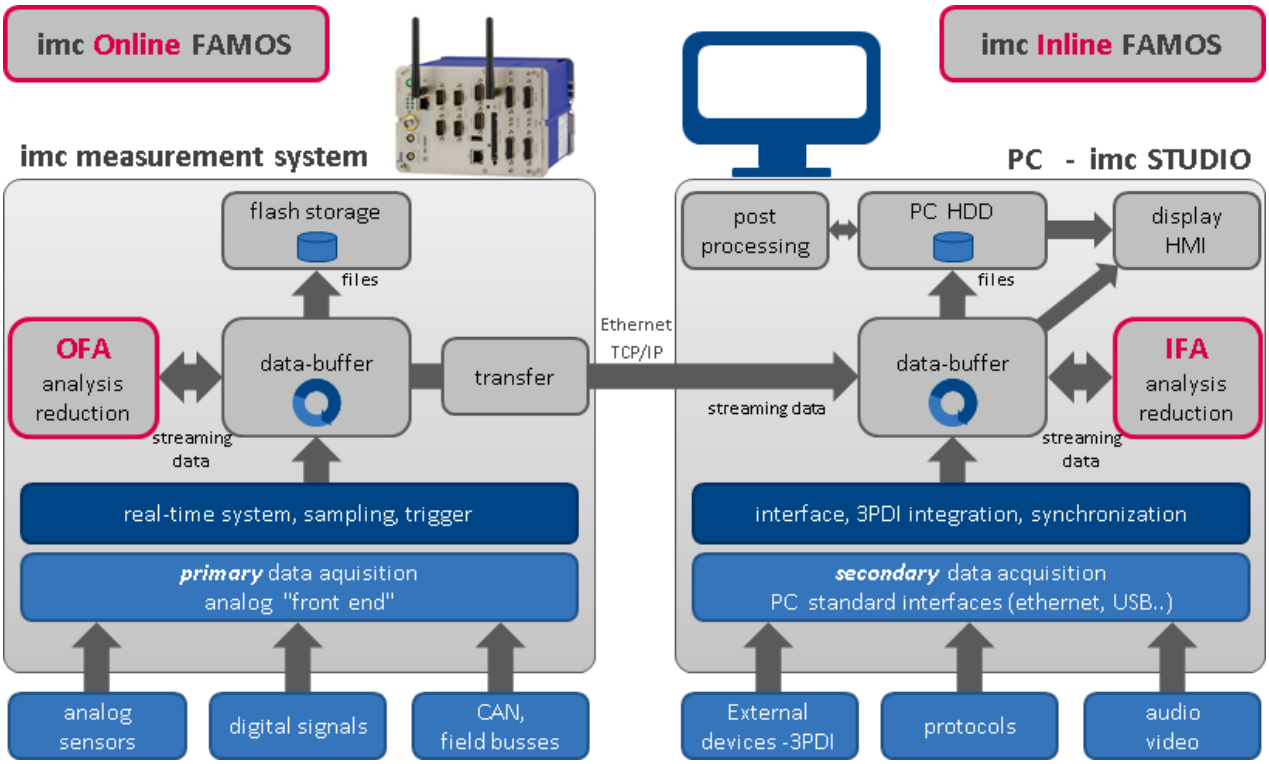
12.2.1 Comparison: imc Online FAMOS / imc Inline FAMOS

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 (Trigger 48)), 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

There are no version grades as with imc Online FAMOS (imc Online FAMOS / imc Online FAMOS Professional).

Data flow schematic

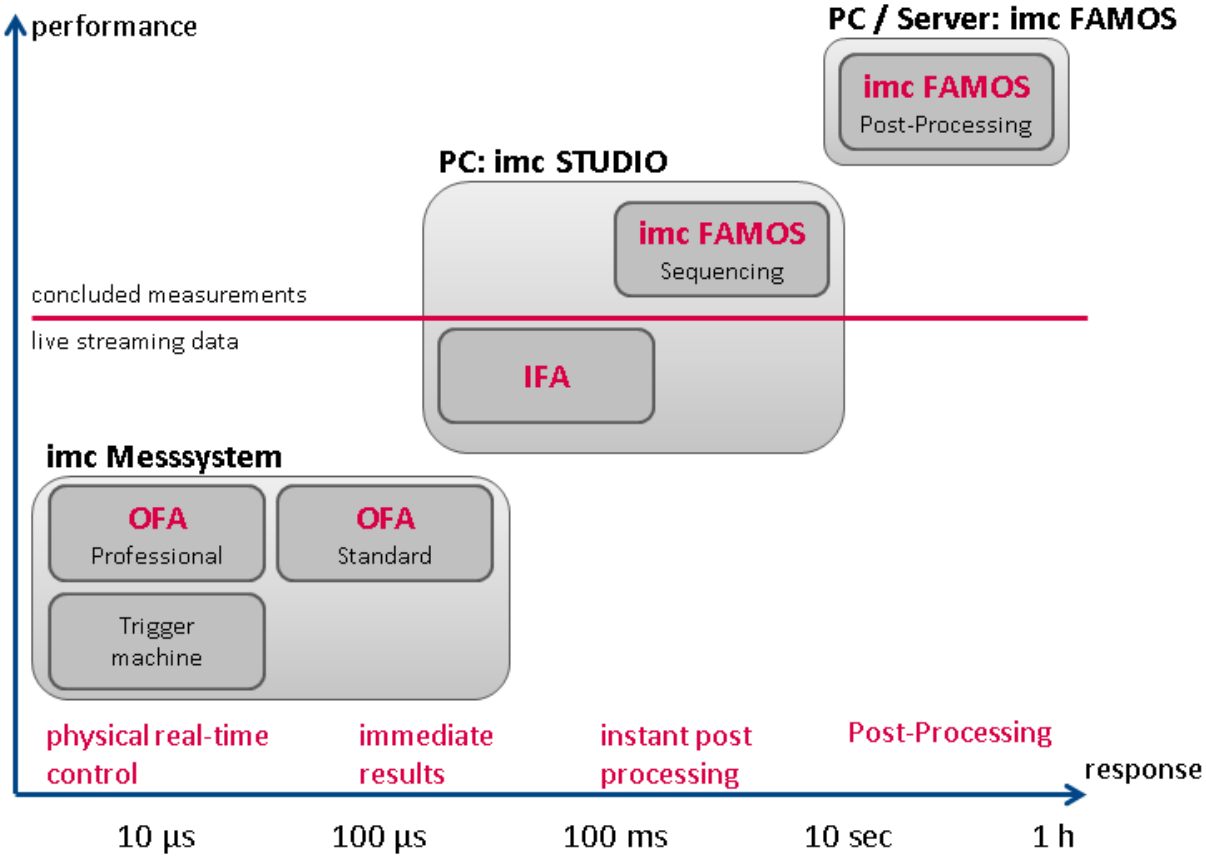


Data flow schematic

When should imc Inline FAMOS, imc Online FAMOS or imc FAMOS 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



12.2.1.1 Example

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
- 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(Trigger_48)
    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.

12.2.2 System Prerequisites 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 (Trigger 48) or a trigger x.

2. Channels belonging to different devices

   Channels which are assigned to the Start/Stop trigger (Trigger 48) 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.

12.2.3 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
- many system functions
- all CAN-functions
- all ECU-functions
- ReadyForPowerOff
- SyncOverload
- OnSyncTask (Synchronous Tasks)
- OnPowerOff
- IntegralP/IntegralP2

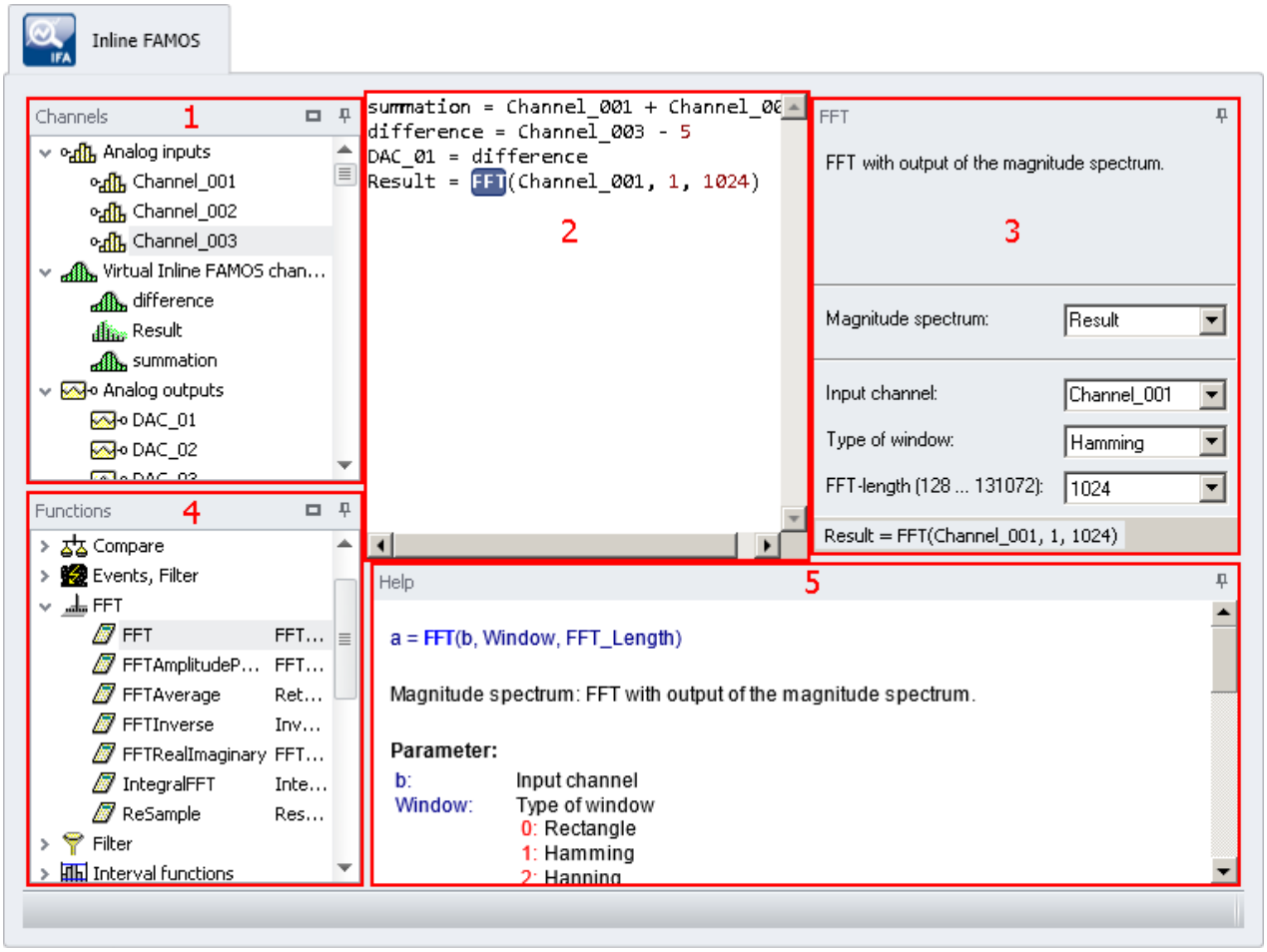
In the Functions Reference, it is noted if a particular function is available only in imc Online FAMOS or only in imc Inline FAMOS.

12.2.4 Operation

12.2.4.1 Starting imc Inline FAMOS

Open the plug-in Data Processing via the Navigation pane. In the plug-in's main window, 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.

12.2.4.2 User Interface



The window is subdivided into five regions:

1. Tool window: Variables List
2. Editor (Text box) for the calculation operations
3. Tool window: Functions-Assistant
4. Tool window: Functions List
5. Tool window: Help

Region 1: Variables List

Here is the list of all available variables supported by 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. A description of the variable types is presented in the chapter: "*Main Window Elements* 748" of the imc Online FAMOS-documentation.

Region 2: 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 Functions-Assistant: using the Functions-Assistant, it is easy to set function parameters and enter the functions into the editor.

Functions are entered into the editor along with parenthesis.

Region 3: Functions-Assistant

The Functions-Assistant provides support in parameterizing the functions.

Region 4: 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 variables.

The mathematical functions are always updated instantaneously. To learn about their workings, see the Help-box or the Functions Reference.

Region 5: 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.

12.2.4.3 Brief Introduction

The Functions-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 Functions-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 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.

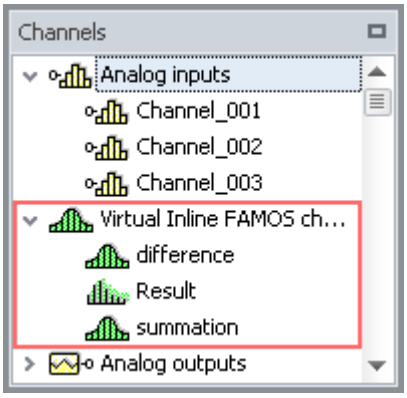
Adding a variable or function

- 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* ( ).

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.

12.2.4.4 Supplemental Files

Supplemental files such as characteristic curves for imc Inline FAMOS-functions are treated the same way as supplemental files for imc Online FAMOS. In order for imc Inline FAMOS to recognize the supplemental files, they must be imported into the experiment.

Reference

On this topic, see the description for: *Setup* > *Ribbon* > *Configuration* > *Supplemental Files* 105.

12.2.5 Context Menu

Context menu in the Editor

Right-clicking the mouse over the Editor causes the following context menu to appear:

	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 744.
	Properties	(F5)	Opens the Properties window. There, the properties of virtual channels can be edited subsequently (see Virtual channels 751).
	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.

Context menu in Variables list

Right-clicking the mouse over 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.

Context menu in the Functions list

Right-clicking the mouse over the Functions 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.

Context menu in the Help window

Right-clicking the mouse over the Help window calls 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.

12.3 Power Quality

Expansion package for power quality analysis conforming to EN 50160 (IEC 61000-4-30 Class A)

Power Quality allows the calculation of power quality indicators according to EN 50160 (IEC 61000-4-30 Class A). The calculations are performed by the PC on the basis of the currently measured data while measurement proceeds. This utilizes the PCs computation resources.

When using the Power Quality for the first time, you should first get to know the user interface 1254 and then work through the example 1257.

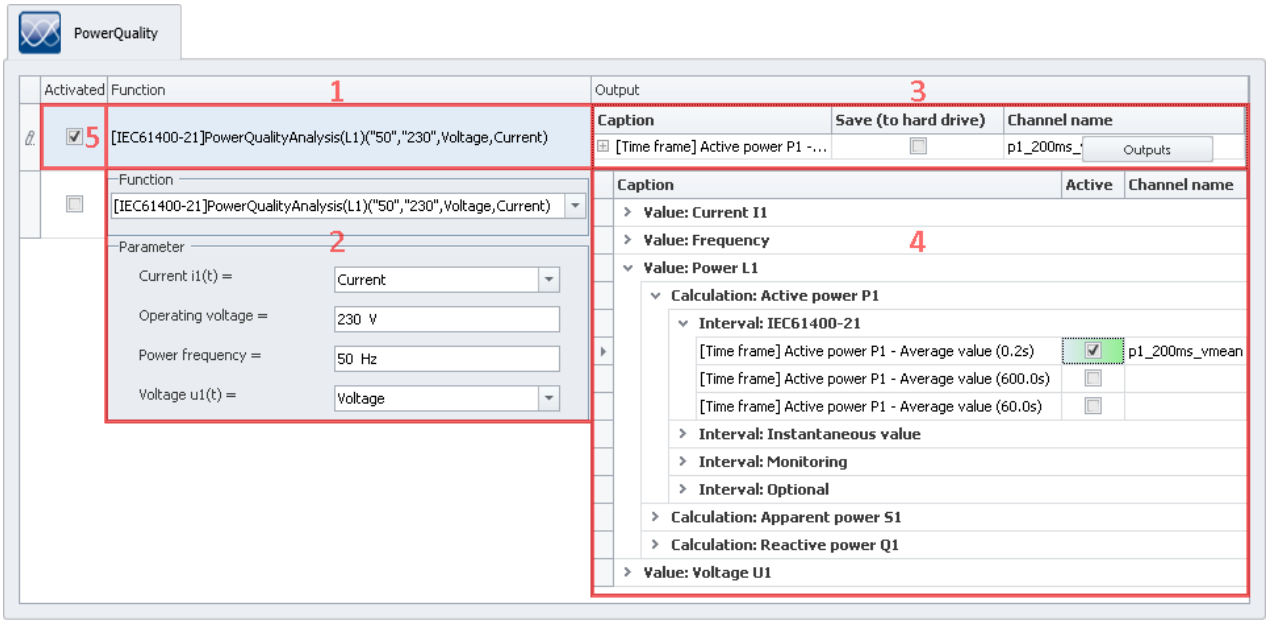
System Requirements and Limitations

To ensure valid results, all the channels subjected to calculations (voltage and amperage) must be sampled at frequencies of at least 10 kHz.

12.3.1 Opening Power Quality

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 Power Quality. If it is not selected already, left-click the mouse on the Task name. The Power Quality Editor will appear in the main window.

12.3.2 User Interface



The window is subdivided into five regions:

1. Function: Lists any functions whose parameters have already been set.
2. Function: Drop-down window for selecting a function and setting its parameters
3. Output: Displays the activated outputs
4. Output: Drop-down window for parameterizing and activating the calculations
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 along with their parameters are displayed here.

- Each function has its own group of input parameters, which must be set after selecting the particular function in Region 2.
- Additionally, each function has various output quantities which must be activated/parameterized in Regions 3 and 4.

Three functions are available:

- PowerQualityAnalysis(L1): single-phase power calculation
- PowerQualityAnalysis(L1,L2,L3): three-phase power calculation (delta configuration)
- PowerQualityAnalysis(L1,L2,L3,N): three-phase power calculation (star configuration)

Setting up a function

To select the function desired, click in the column Function in the first empty row. In consequence, Region 2 (the parameterization window) opens; there you can select and configure your function.

Changing a function

To alter a function which has already been set up, click on it corresponding cell.

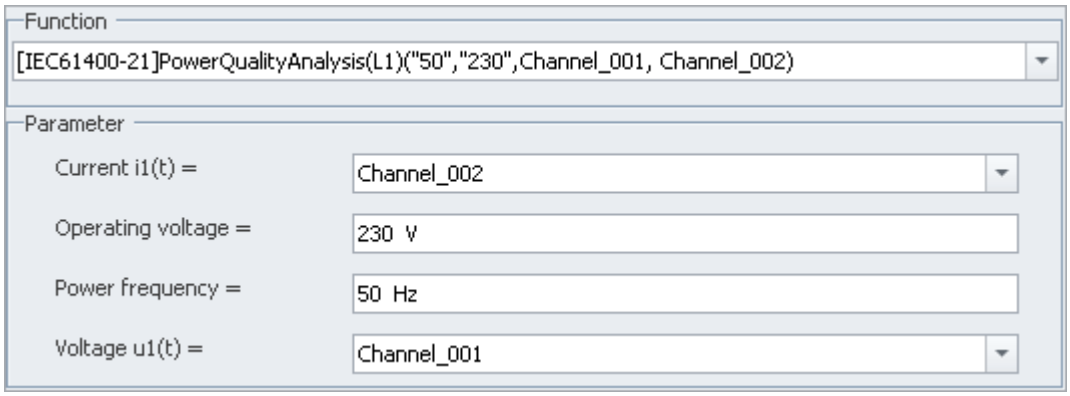
Deleting a function

To delete a function which has already been set up, open the context menu 1265 in the corresponding cell and select *Delete function* 1265.

Region 2: Function parameterization window

Here you select the function and configure the input parameters.

Clicking on the arrow button in the top row opens a list of named functions from which to select. Based on what you select (single- or three-phase), the respective input parameters must be set subsequently.



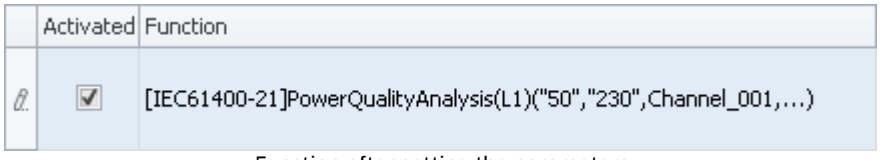
Parameters for the single-phase power calculation

With single-phase systems, the following parameters can be set:

- Power frequency
- Operating voltage
- The voltage u1(t), to be measured
- The current i1(t), also to be measured

The channels specified for u1(t) and i1(t) need not be active nor even exist at the time the settings are made. However, for the sake of simplicity a selection list of active channels is provided.

When you click the mouse anywhere outside of the dialog, the parameter settings are concluded. In consequence the selected function is displayed along with the parameters set.



Function after setting the parameters

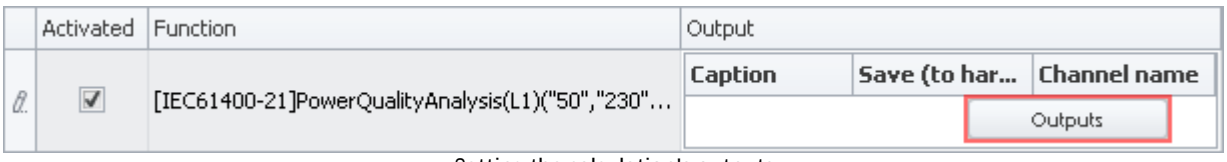
With three-phase power calculations, there would be two additional voltage and two more amperage signals to set.

Region 3: Display of activated outputs

An output in this context is the calculated result quantity returned, e.g. average frequency, maximum active power, etc. Each calculated output quantity generates a channel (virtual channel), which is available in the Data Browser and can be linked with a Widget, such as a curve window. Since the calculations always are based on the instantaneous measurement results, they are continually updated.

Activating and making settings for outputs

To activate and make settings for the currently selected function, click in the column *Output* on the button Outputs. Immediately, the parameterizing window (Region 4) opens; there, select and configure the output quantities.



Setting the calculation's outputs

For each function, the selected output quantities are set separately, for which reason there is an *Outputs* button for each function. For each function, any number of outputs 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 calculations

Here, you can activate and parameterize the output quantities.

The quantities available are listed in a table. One section of this table is shown below:

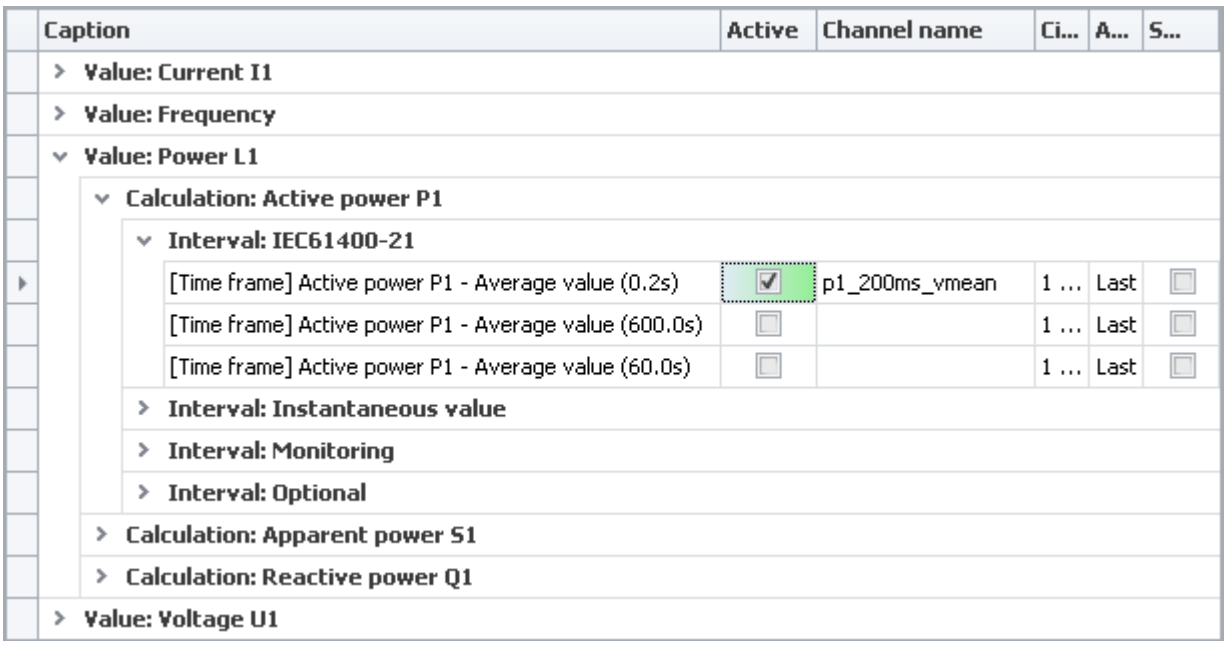


Table of available outputs for a single-phase power calculation

The figure above is based on a single-phase power calculation. If a three-phase power calculation were selected, there would correspondingly be more selection options.

Activating a calculation

Navigate to the desired output quantity and put a checkmark in the column Active.

Checking this box activates the calculation. The channel name is set automatically, but can be edited as desired subsequently. Activated calculations appear as virtual channels in the Channels table (Setup) and in the Data Browser (Panel).

Parameterizing calculations

Each output quantity 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 Power Quality settings, since the same settings, in part, can be made here.

For more information on the individual parameters, see the section: Output Parameters 1262.

Region 5: Activating/deactivating functions

Here, the calculation of a complete function can be activated/deactivated.

Only such calculations are performed which have a checkmark in the column Activated. All newly added functions are activated by default.

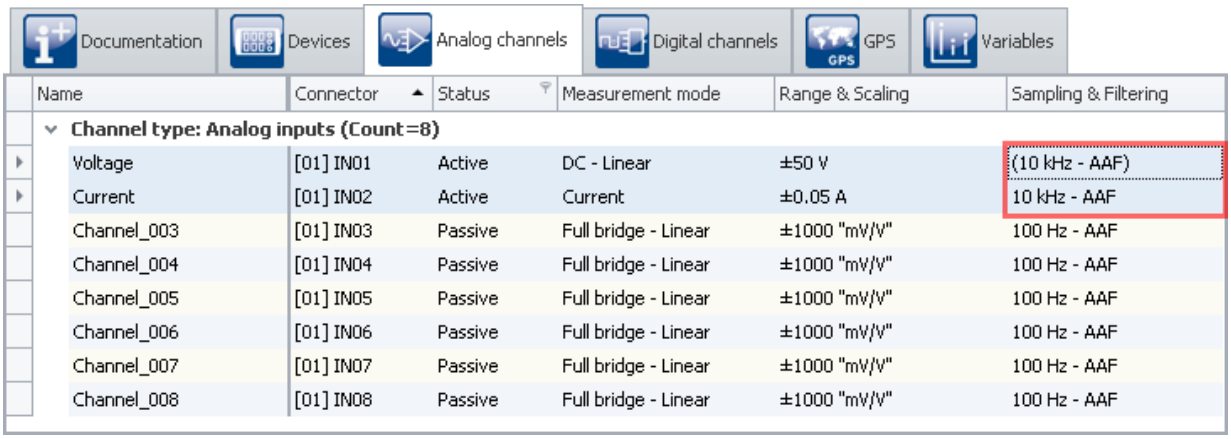
12.3.3 Introductory Example

As our introductory example, Power Quality is used to determine the active power in a 50 Hz power network having 230 V service voltage.

Setup - Setting channels

Make the appropriate settings for the measurement devices in the plug-in Setup's main window. For this example, two measurement channels are required: one measures the amperage and one the voltage.

The sampling rate of both channels must be at least 10 kHz. Meeting this requirement is necessary in order to correctly determine all network analysis results.



Name	Connector	Status	Measurement mode	Range & Scaling	Sampling & Filtering
Channel type: Analog inputs (Count=8)					
Voltage	[01] IN01	Active	DC - Linear	±50 V	(10 kHz - AAF)
Current	[01] IN02	Active	Current	±0.05 A	10 kHz - AAF
Channel_003	[01] IN03	Passive	Full bridge - Linear	±1000 "mV/V"	100 Hz - AAF
Channel_004	[01] IN04	Passive	Full bridge - Linear	±1000 "mV/V"	100 Hz - AAF
Channel_005	[01] IN05	Passive	Full bridge - Linear	±1000 "mV/V"	100 Hz - AAF
Channel_006	[01] IN06	Passive	Full bridge - Linear	±1000 "mV/V"	100 Hz - AAF
Channel_007	[01] IN07	Passive	Full bridge - Linear	±1000 "mV/V"	100 Hz - AAF
Channel_008	[01] IN08	Passive	Full bridge - Linear	±1000 "mV/V"	100 Hz - AAF

Setting the channels

To apply the changes, click on the button Process configuraton 139 (✔). In consequence, the device configuration is available to the other plug-ins. This means that the channels can be subject to operations by the functions package: Power Quality.

Configuring the Power Quality-function

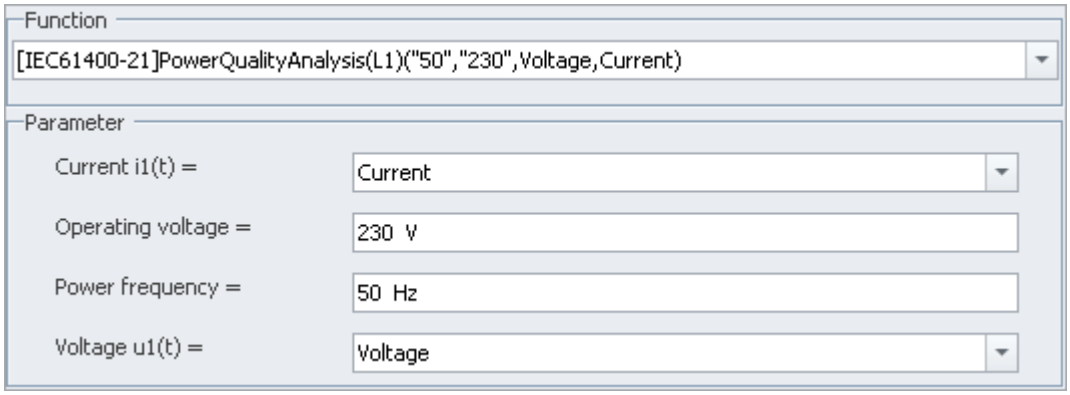
Open the plug-in Data Processing via the Navigation pane. In the plug-in's main window, you will see a tab (Task:) for the functions package Power Quality. If it is not yet selected, left-click the mouse on the task name. The Power Quality Editor will then be displayed in the main window.

Selecting and parameterizing a function

In this example, a single-phase power measurement is to be conducted, in order to calculate the actual power from its result.

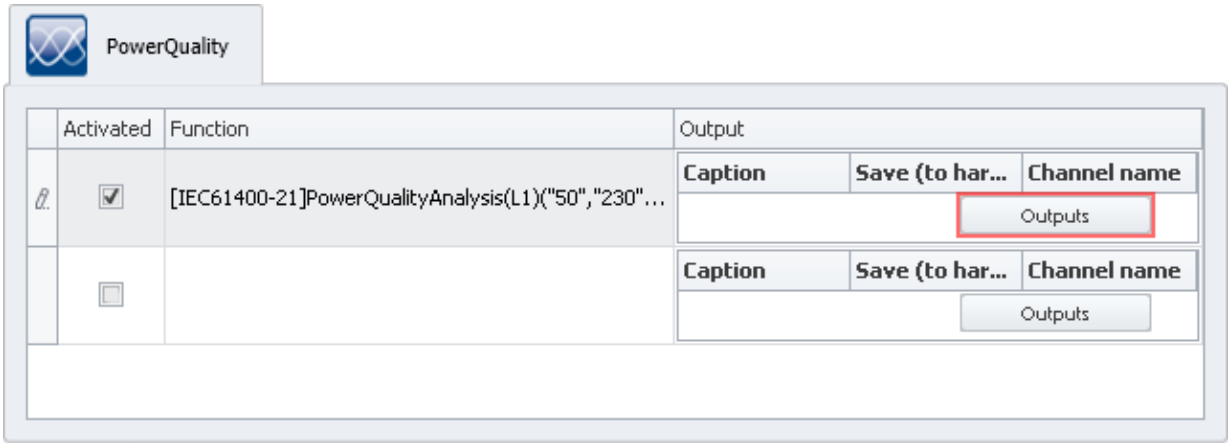
- Click on the top empty line under the column: Function
- In the drop-down list, select the following function: [IEC61400-21]PowerQualityAnalysis(L1)("50","230",,)
- Next, a list of all parameters of the function appears. Parameterize it as follows:

Power frequency	50 Hz (already given)
Operating voltage	230 V (already given)
Voltage u1(t)	Voltage (usually an analog channel which measures the voltage)
Current i1(t)	Current (usually an analog channel which measures the amperage)



Configuring Power Quality-result (output) quantities

To set the actual power as the result quantity, click in the column Output on the button Outputs:

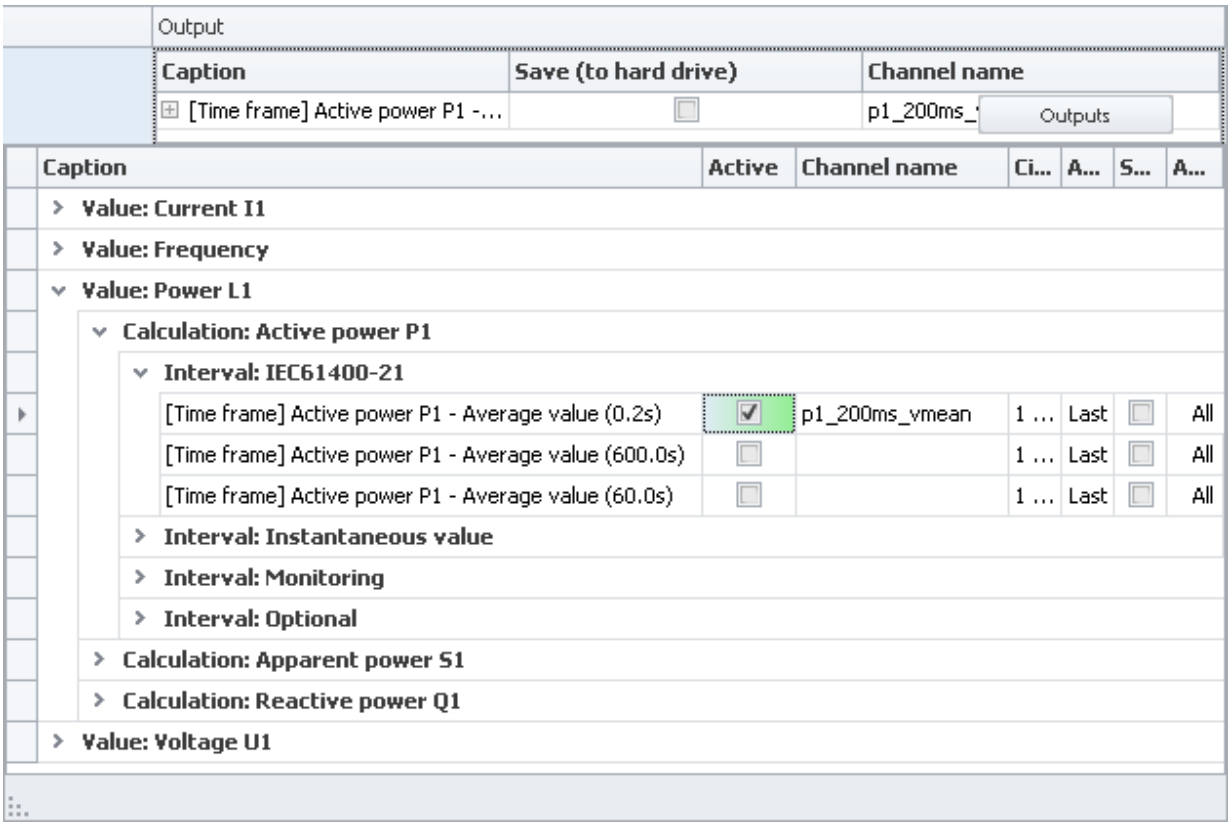


In consequence, a list of various calculations available for selection appears. The desired calculations can be activated/parameterized here.

- The active power is found under *Value: Power L1 > Calculation: Active power P1*.

Here, various intervals are available. In this example, the mean value over 200 ms is to be determined as per the standard IEC 61400-21.

- Open *Interval: IEC61400-21*
- Put a check mark in the column Active for the calculation: *[Time frame] Active power P1 - Average value (0.2s)*



By checking the box, the calculation is activated. The channel name is set to *p1_200ms_vmean*, but can be changed as desired.

Here, make the following settings:

Active	☑
Channel name	p1_200ms_vmean
Circular buffer duration (curve window)	1 min
Available events (curve window)	Last (no other options available)
Save (to hard drive)	☐
Available events (hard drive)	all (not relevant since there is no data saving)
Circular buffer duration (hard drive)	undefined (not relevant since there is no data saving)
Averaging time	0.2 s (permanent setting)

Inspecting the Power Quality-configuration and starting calculation

Check / Syntax-Check

Click on this button to perform a *Syntax-Check*. If any errors occur, you will be notified via the logbook!

Applying the Power Quality-configuration

Even if the Syntax-Check concluded without any errors, the task still will not start. As soon as the configuration is applied, the task starts and the calculations are performed. To do this, click on the button *Apply* ( ).

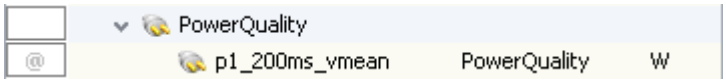
More information on *Check* and *Apply* is presented under: Performing Calculations 1264.

Configuring a results channel in the Setup

In the table of channels in the Setup, the newly defined channel appears as an additional virtual channel. Here, the channel can be parameterized (e.g. the data storage) just like any other virtual channel. The changes made in the Setup directly affect the Power Quality settings, since some of the same settings are made there, too.

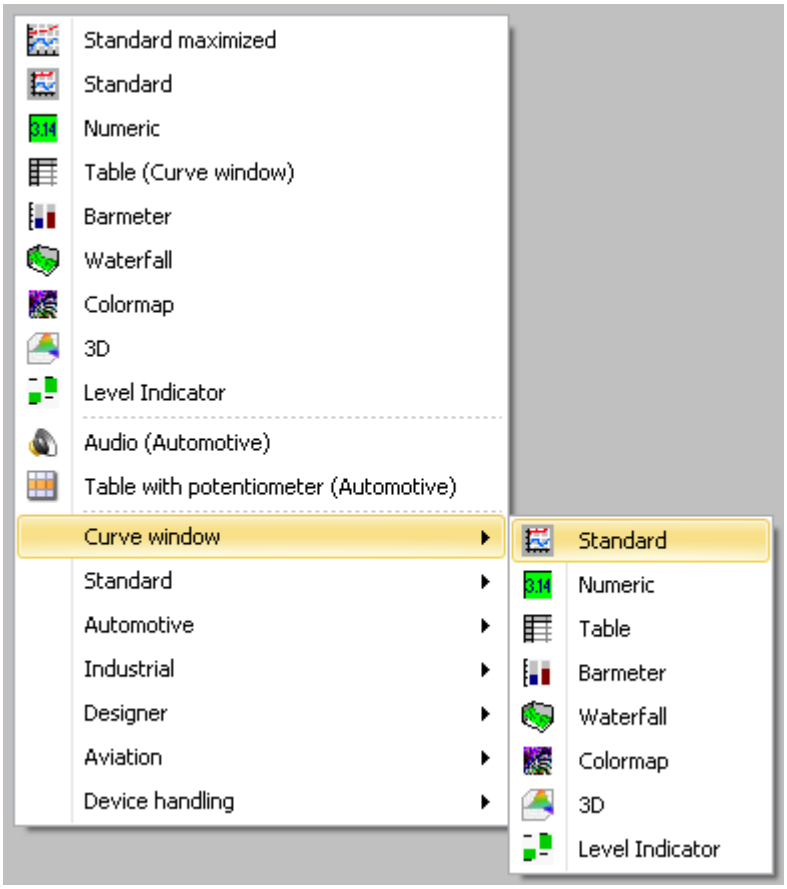
Displaying the results channel in the Panel

Once the configuration has been applied, the results channel appears in the Data Browser with the name *p1_200ms_vmean*. It is found under its own category *Power-Quality-Analysis*:



From there you are able to link the channel to a Panel page with Widgets, e.g. a standard curve window.

To do this, use Drag&Drop to take the Variable *p1_200ms_vmean* to the Panel page. In the menu which then opens, select *Curve window - Standard:*



Now have a curve window on the Panel page, in which *p1_200ms_vmean* is displayed.

Start the measurement now using the Start-Button ( ).

A tip on producing a simulation, without actual signals

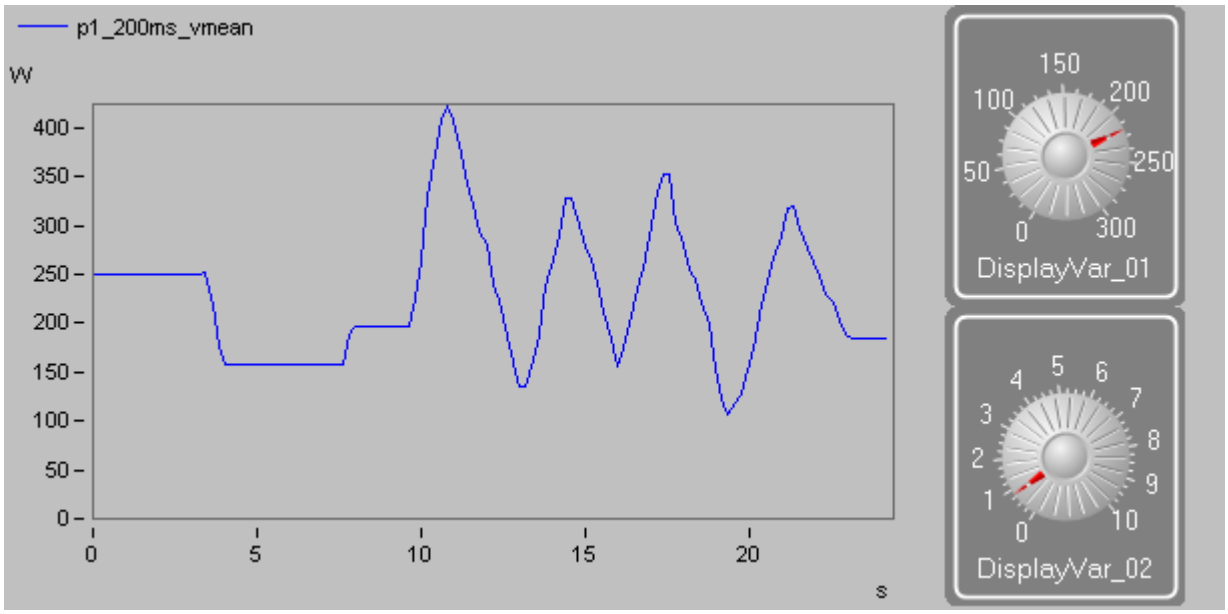
If appropriate voltage and current signals are not available, you can simulate some for the purpose of getting an introduction. To do this, use imc Online FAMOS and generate corresponding virtual channels.

For this simple exercise, generate the virtual channels consisting of a Display-variable's instantaneous values. The value of the Display-variable is edited by means of the Panel for simulation.

imc Online FAMOS-Code:

```
Voltage = Channel_001 * 0 + DisplayVar_01
Current = Channel_001 * 0 + DisplayVar_02
```

In this case, the analog inputs may no longer be called *Current* and *Voltage*, since the virtual channels now have these names for purposes of the calculation.



By means of a potentiometer, the Display-variable's value is adjusted. In the curve window, the calculated values appear.

12.3.4 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:

Parameter [1]		Description
-	Caption	Name of the output quantity (permanently assigned)
-	Active	Activates/deactivates calculation of the output quantity • Deactivated: the output quantity is not calculated • Activated: the output quantity is calculated
-	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).
●	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.
●	Save (to hard drive)	Saves the output quantities to the PC
●	Available events (hard drive)	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 (HD)* is set, then only *Last* is available.
●	Circular buffer duration (hard drive)	For the time period specified, the data are retained on the PC for data storage purposes. The saved measurements only contain data for the duration of the buffer time set.
-	Averaging time	Averaging time in seconds

1 Parameters denoted by ● can also be set in the channel table in the Setup.

If certain parameters are already determined by the nature of the quantity being calculated, such as the averaging duration for *Active power - Instantaneous value* or *Voltage - Flicker - Interval 10 minutes*, these boxes are not editable.

A detailed listing of result quantities is presented under Basics 1265.

Interval Monitoring/Optional

For many result quantities, there is a distinction between: *Interval: Monitoring* and *Interval: Optional*. This distinction simply provides the ability to calculate the same result quantity using two different averaging intervals.

Thus, for instance, it is possible, on the one hand, to calculate the active power with a long averaging interval (e.g. 15 min), and on the other hand to obtain quick preliminary results using a short interval (e.g. 1 s).

Flicker: With the flicker calculation, the algorithm following *Interval: 10 minutes* and *Interval: Optional* is the same; *Interval: Monitoring* offers the ability to determine the arithmetic mean of the instantaneous flicker within a specified time window. *Interval: 10 minutes* calculates the flicker over the 10-minute averaging interval specified in the standard IEC 61000-4-15; *Interval: Optional* calculates according to the same algorithm, but at any arbitrary averaging interval.

10/12 Period

If a result quantity is calculated over 10/12 periods and a specific averaging duration is calculated, the following apply:

- per 10 periods (with 50 Hz rated network frequency; 12 periods at 60 Hz), one value is determined.
- As many values are determined as possible until the averaging interval has elapsed.
- From these values, the minimum, maximum or arithmetic mean is determined depending on the result quantity specified
- The result quantity is only updated after the averaging interval (e.g. in the curve window)

With a network frequency of exactly 50 Hz (non-existent in practice) and an averaging interval of 1 s, this would mean: per 10 periods = 0.2 s one value is determined, i.e. per averaging interval 5 values.

The principle behind this arrangement is described in the standard IEC 61000-4-30.

12.3.5 Performing Calculations

The plug-in Data Processing continually performs calculations, even when no measurement is running. The instantaneous values are used for the measurement.T

In order that the calculation can be performed, the parameters (network voltage, network frequency, voltages and currents) must be complete. Specified channels (which are already present) must be sampled at a minimum of 10 kHz. Also, the network frequency must be 50 Hz or 60 Hz.

Check

When you click on the button: Check ( ), the current configuration is examined to verify that all conditions are met. If any condition is not met, an error message appears in the logbook.

Note

At this point, there is no check of whether the specified channels exist, since they could still be created later (e.g. by imc Online FAMOS).

Apply

After the settings have been checked, clicking on the button: Apply ( )

- stops the running calculations,
- adopts the updated functions with their parameters and desired result quantities,
- and re-starts calculations.

Note

For Apply, the verification process described above is repeated. If any of the calculations can not be performed, for instance because the sampling rate of some channel is less than 10 kHz, none of the calculations are performed! The new specifications (function, parameter, outputs) are not transferred in this case, and the calculation is not re-started.

Activating/deactivating functions

Only such calculations are performed which are selected in the column *Active*. All newly added lines are selected by default as soon as one function is assigned. Deactivated calculations are not performed.

	Activated	Function
	☑	[IEC61400-21]PowerQualityAnalysis(L1)("50","230"...
	☐	[IEC61400-21]PowerQualityAnalysis(L1, L2, L3, N)(...

Calculation activated/deactivated

12.3.6 Context Menu

Context menu in the table

Right-clicking in the table the calls following context menu:

	Menu item	Description
	Delete function	Deletes the function. Is only enabled if the context menu is opened via a row.
	Delete all functions	Deletes all function.

Context menu at the outputs

Right-clicking the mouse over the opened Outputs list opens the following context menu:

	Menu item	Description
	Expand	Expands all or individual groups located under *Caption*.
	Collapse	Collapses all or individual groups located under *Caption*.
	Select all groups	Expands all groups and highlights them; can be used, for example, to set all outputs to active.
	Deselect selection	De-selects all selected outputs/groups

12.3.7 Basics

In this chapter, the result quantities which can be set under *Outputs* 1262 are described.

The result quantities are each discussed separately the following sections:

- Frequency 1266
- Power 1266
- Voltage 1266
- Current 1267

In the tables below, the possible result quantities are briefly discussed. The order of presentation is the same as they appear in the user interface.

Since the possible result quantities fall under the categories *Power*, *Voltage*, *Current* and *Frequency* (on with three-phase power calculations, also *Network frequency*), both for single- and for three-phase power calculations, the breakdown of the next chapters is based only on these categories.

The distinction between *Interval: Monitoring* and *Interval: Optional* is not given consideration since these options do not result in different kinds of result quantities, but are simply based on different respective averaging intervals (see Ausgangsparameter 1262).

Reference: Calculation of power

A good overview of the topic Power measurement is provided in the documentation on imc Online FAMOS 739 in the chapter *Information and Tips*.

12.3.7.1 Frequency

The following result quantity can be selected for the frequency:

Quantity	Interval	Explanation
Frequency	time frame	The frequency is calculated over an interval set as the averaging interval. The available outputs are the minimum, maximum and the arithmetical mean.

12.3.7.2 Power

The following result quantities can be determined for the power:

Quantity	Interval	Explanation	Definition
Reactive power	10/12 periods	Calculates the reactive power over either 10 (50 Hz network frequency) or 12 periods (60 Hz network frequency).The choices of output are minimum, maximum and arithmetical mean.	The *reactive power* denotes the component of the apparent power which is not used by the load.
Apparent power	10/12 periods	Calculates the apparent power over either 10 (50 Hz network frequency) or 12 periods (60 Hz network frequency).The choices of output are minimum, maximum and arithmetical mean.	The *apparent power* is the product of the voltage times the current, disregarding phase offset.
Active power	time frame	Calculates the active power over a specified time window in accordance with the standard IEC 61400-21. The available choices are 0.2 s, 60.0 s and 600.0 s.	The *active power* denotes the component of the apparent power actually used by the load.
Active power	instantaneous value	Calculates the instantaneous value of the active power.	
Active power	10/12 periods	Calculates the active power over either 10 (50 Hz network frequency) or 12 periods (60 Hz network frequency).The choices of output are minimum, maximum and arithmetical mean	
Active power(mean value)	time frame	Computes the arithmetical mean of the actual power over a time window specified as the averaging interval.	

12.3.7.3 Voltage

The following result quantities can be selected for the voltage:

Quantity	Interval	Explanation	Definition
Harmonic distortion	10/12 periods	Calculates the total harmonic distortion of the voltage over either 10 (50 Hz network frequency) or 12 periods (60 Hz network frequency).The choices of output are minimum, maximum and arithmetical mean.	The *Total Harmonic Distortion* is the ratio of the sum of all voltages of all upper harmonics (here, 2nd through 50th) to the fundamental oscillation.

Quantity	Interval	Explanation	Definition
Root mean square	10/12 periods	Calculates the RMS of the voltage over either 10 (50 Hz network frequency) or 12 periods (60 Hz network frequency).The choices of output are minimum, maximum and arithmetical mean.	The *RMS* is the quadratic mean of the input signal.
(Mean) Root mean square	time frame	Calculates the arithmetical mean of the RMS values within a time window specified as the averaging interval.	
Flicker	time frame	Calculates the voltage's flicker according to the standard IEC 6100-4-15 in a fixed 10-minute time window. Also, the arithmetical mean of instantaneous flicker values within a time window specified as the averaging interval can be calculated. Observe notes on *Interval Monitoring/Optional* 1262!	*Flicker* refers to voltage fluctuations which can cause a perceptible flickering of light bulbs. The prevalence, strength and duration of these fluctuations are used as indicators of the network quality.
Harmonics	time frame	Calculates the harmonics of the voltage according to the standard IEC 61400-4-7 (upper harmonics subgroup) over a time window specified as the averaging interval. The choices of output are minimum, maximum and arithmetical mean.	Harmonics are the amplitudes of all integer multiples of the network frequency (fundamental oscillation = 50/60Hz); these are determined up to the 50th harmonic (2.5kHz/3kHz).
Higher frequency contents 2 kHz - 9 kHz	time frame	Calculates the voltage's higher frequency components according to the standard IEC 61400-4-7 over a time window specified as the averaging interval. The choices of output are minimum, maximum and arithmetical mean.	Higher frequency components (2kHz - 9kHz) indicate the amplitudes in the higher frequency range (from 2kHz on in 200Hz steps).
Interharmonics	time frame	Calculates the voltage's interharmonics according to the standard 61400-4-7 (centered interharmonic subgroup) over a time window specified as the averaging interval. The choices of output are minimum, maximum and arithmetical mean.	Interharmonics are sinusoidal oscillations whose frequency is not an integer multiple of the network frequency (determined here up to the 50th interharmonics).

12.3.7.4 Current

The following result quantities can be selected for the current:

Quantity	Interval	Explanation	Definition
Harmonic distortion	10/12 periods	Calculates the total harmonic distortion of the current over either 10 (50 Hz network frequency) or 12 periods (60 Hz network frequency).The choices of output are minimum, maximum and arithmetical mean.	The *Total Harmonic Distortion* is the ratio of the sum of all currents of all upper harmonics (here, 2nd through 50th) to the current of the fundamental oscillation.

Quantity	Interval	Explanation	Definition
Root mean square	10/12 periods	Calculates the RMS of the voltage over either 10 (50 Hz network frequency) or 12 periods (60 Hz network frequency).The choices of output are minimum, maximum and arithmetical mean.	The *RMS* is the quadratic mean of the input signal.
(Mean) Root mean square	time frame	Calculates the arithmetical mean of the RMS values within a time window specified as the averaging interval.	
Flicker	time frame	Calculates the current's flicker according to the standard IEC 6100-4-15 in a fixed 10-minute time window. Also, the arithmetical mean of instantaneous flicker values within a time window specified as the averaging interval can be calculated. Observe notes on *Interval Monitoring/Optional* 1262!	Analog to the definition in the section Voltage 1266.
Harmonics	time frame	Calculates the harmonics of the current according to the standard IEC 61400-4-7 (upper harmonics subgroup) over a time window specified as the averaging interval. The choices of output are minimum, maximum.	Harmonics are the amplitudes of all integer multiples of the network frequency (fundamental oscillation = 50/60Hz); these are determined up to the 50th harmonic (2.5kHz/3kHz).
Higher frequency contents 2 kHz - 9 kHz	time frame	Calculates the current's higher frequency components according to the standard IEC 61400-4-7 over a time window specified as the averaging interval. The choices of output are minimum, maximum and arithmetical mean.	Higher frequency components (2kHz - 9kHz) indicate the amplitudes in the higher frequency range (from 2kHz on in 200Hz steps).
Interharmonics	time frame	Calculates the current's interharmonics according to the standard 61400-4-7 (centered interharmonic subgroup) over a time window specified as the averaging interval. The choices of output are minimum, maximum and arithmetical mean.	Interharmonics are sinusoidal oscillations whose frequency is not an integer multiple of the network frequency (determined here up to the 50th interharmonics)

12.4 Bus Decoder

Expansion package for decoding Field-bus log channels

imc STUDIO BusDecoder is a package of functions for Data Processing.

This plug-in allows either all or individual measurement channels belonging to a log-channel to be decoded/extracted. A log-channel can be a logged Field-bus communication ("Logfile").

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. 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.

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.

12.5 Powertrain Monitoring

General

The component imc STUDIO Powertrain Monitoring has been developed in close cooperation with the company GfM (Gesellschaft für Maschinendiagnose mbH), experts in machine and bearing diagnostics.

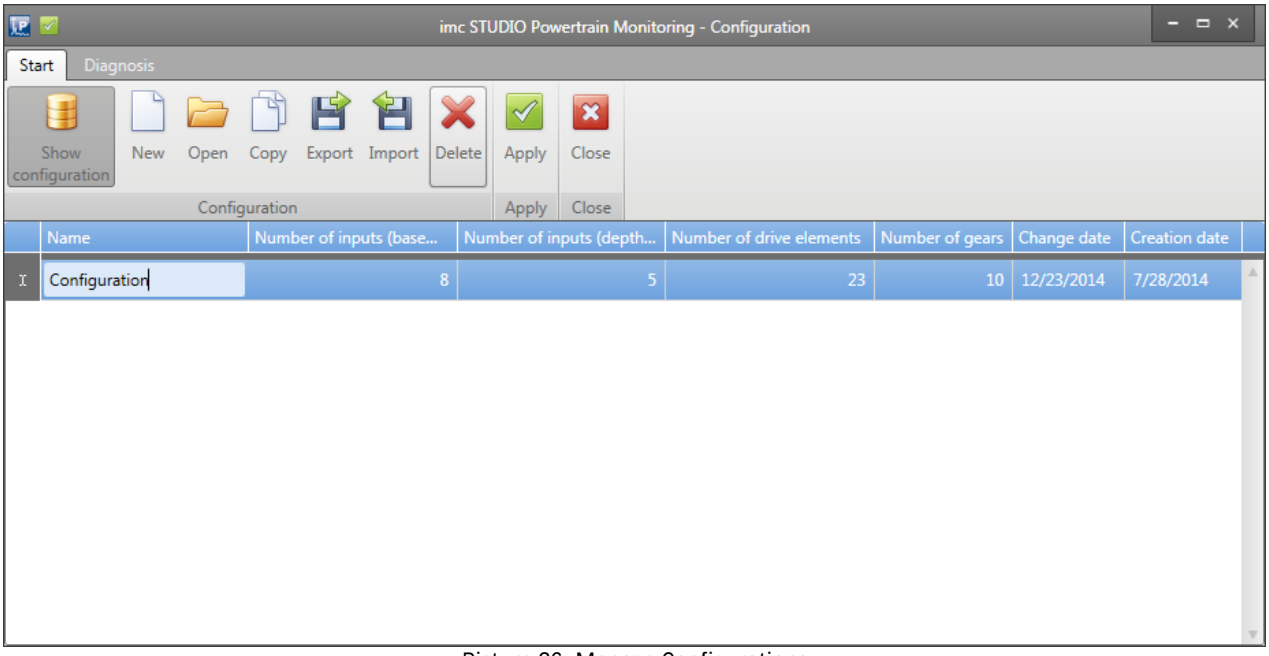
It is dedicated to diagnosis of powertrains. The powertrains can consist of motors, shift gearboxes and engines as well as devices for braking. The diagnosis can be used in field scenarios, test rigs or end of line tests in production.

Powertrain Monitoring offers two different kinds of diagnosis of vibrations: a Base diagnosis 1272 on the basis of characteristic values and a Advanced Diagnosis 1278 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.

For using the component Powertrain Monitoring the product imc STUDIO in the version 5.0 R3 or higher as well as a license of the imc STUDIO Powertrain Monitoring is required. Both can be obtained at imc Meßsysteme GmbH. The license can be activated via the imc LICENSE Manager.

12.5.1 Assistant

12.5.1.1 Configuration

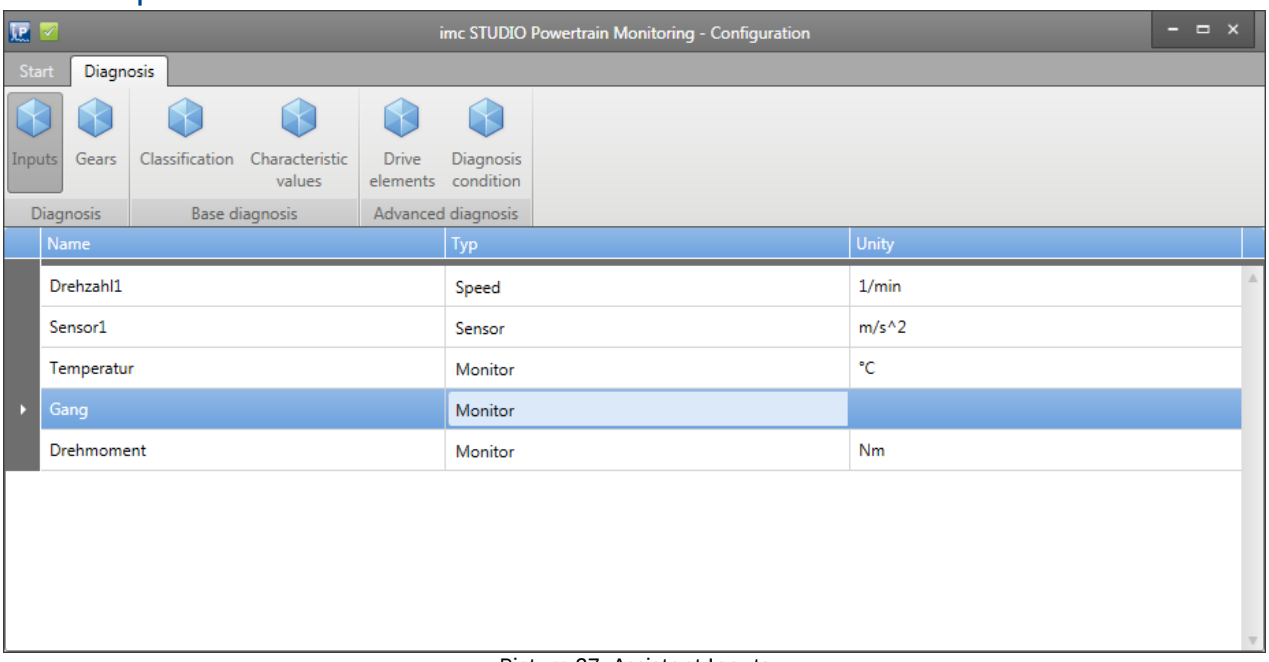


Picture 26: Manage Configurations

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.

12.5.1.2 Inputs



Picture 27: Assistant Inputs

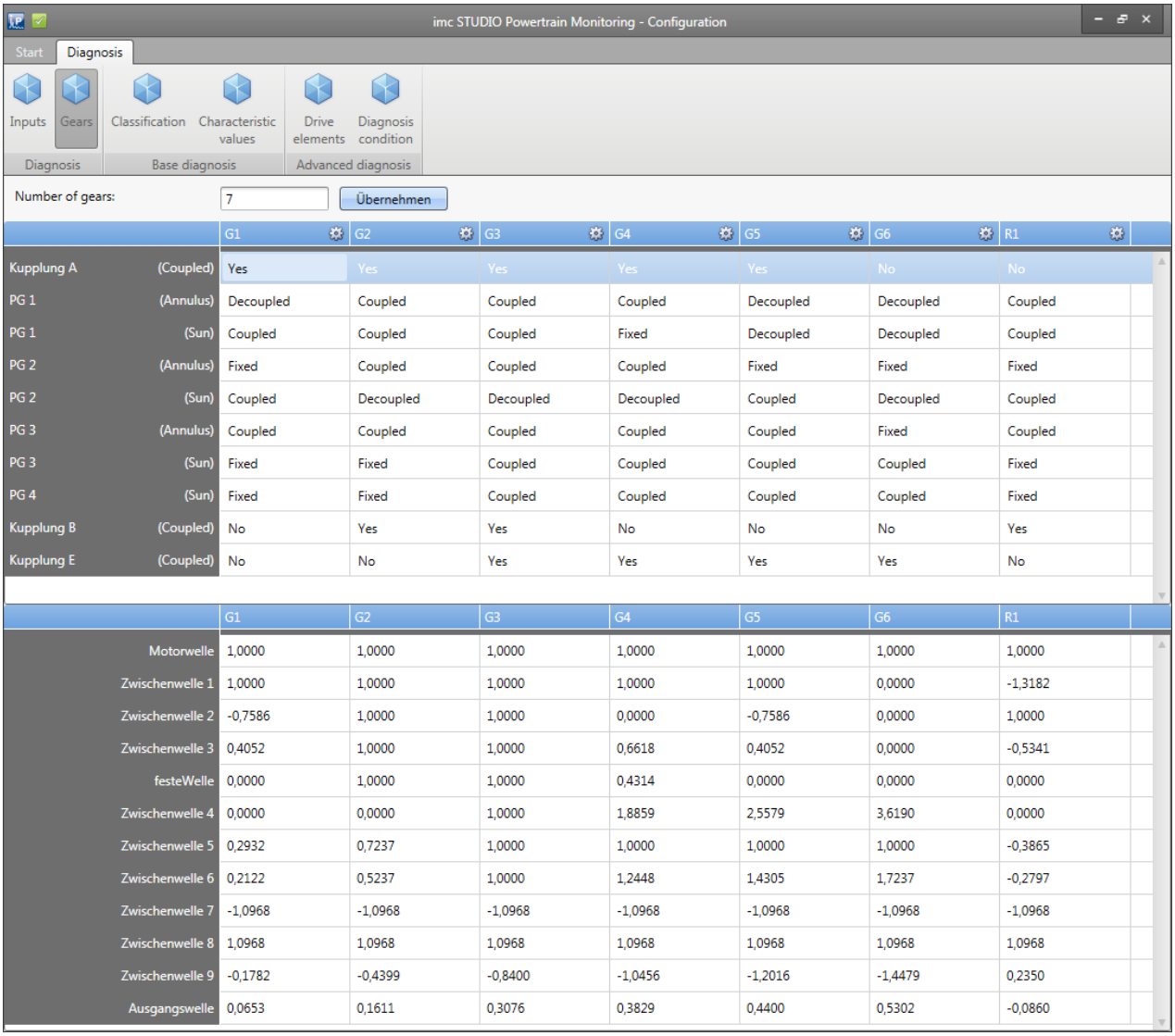
An essential part of both diagnosis procedures are the measurement of structure-borne sound of the test item via accelerometers. These will be added as sensors in the assistant.

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 1278 the measurement time for this sensor must also be set.

12.5.1.3 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

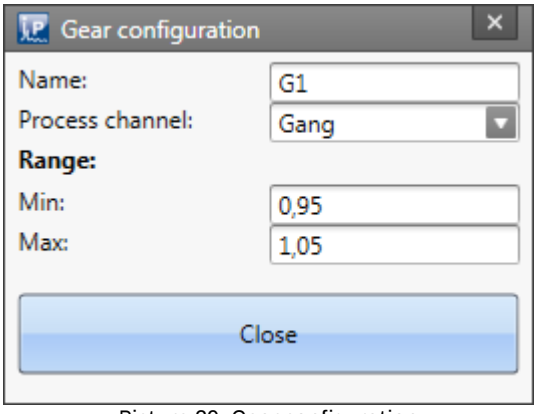
	G1	G2	G3	G4	G5	G6	R1
Motorwelle	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
Zwischenwelle 1	1,0000	1,0000	1,0000	1,0000	1,0000	0,0000	-1,3182
Zwischenwelle 2	-0,7586	1,0000	1,0000	0,0000	-0,7586	0,0000	1,0000
Zwischenwelle 3	0,4052	1,0000	1,0000	0,6618	0,4052	0,0000	-0,5341
festeWelle	0,0000	1,0000	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 28: 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 28).



Picture 29: 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 29).

12.5.2 Editor

To apply the configuration built with the assistant to the current experiment, the Powertrain Monitoring Editor 1276 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:

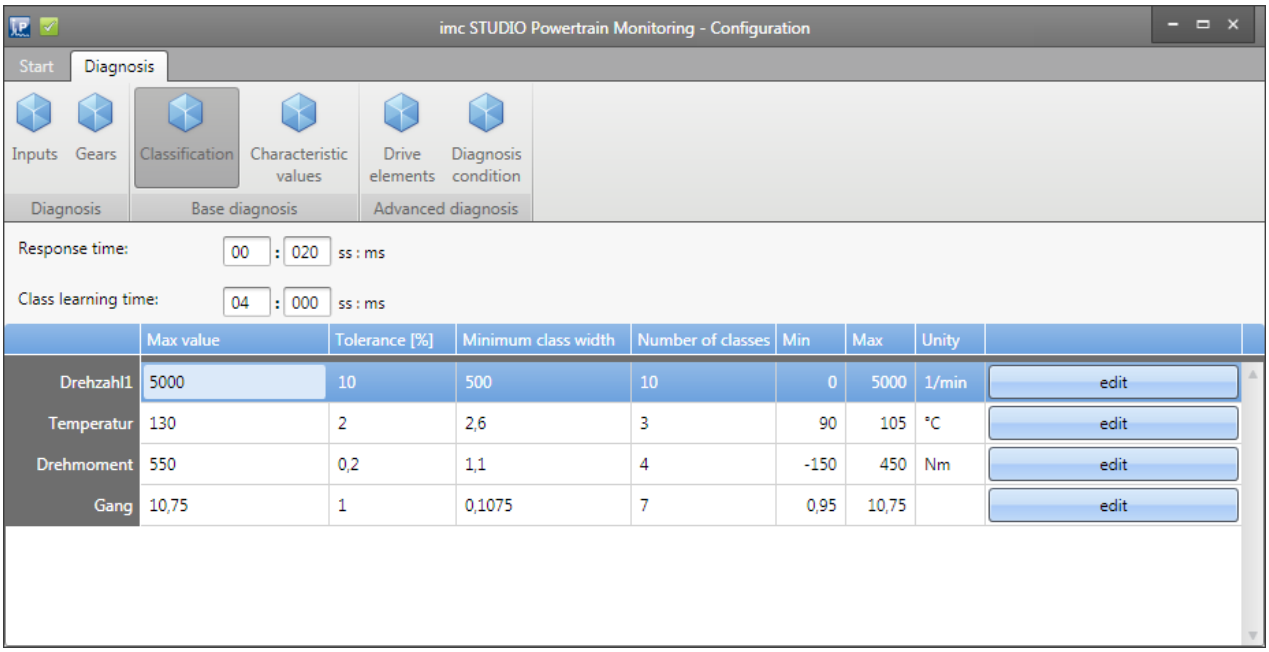
- Accelerometers and speed channels must be measured with 50 kHz.
- Sampling frequencies of process channels must be an integer divider of 50 kHz (e.g. 100 Hz, 1 kHz, ...)
- All channels must be set to undefined measurement time.

12.5.3 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 1273 can be configured with the Powertrain Monitoring Assistant.

12.5.3.1 Assistant

12.5.3.1.1 Classification



Picture 30: 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	☐

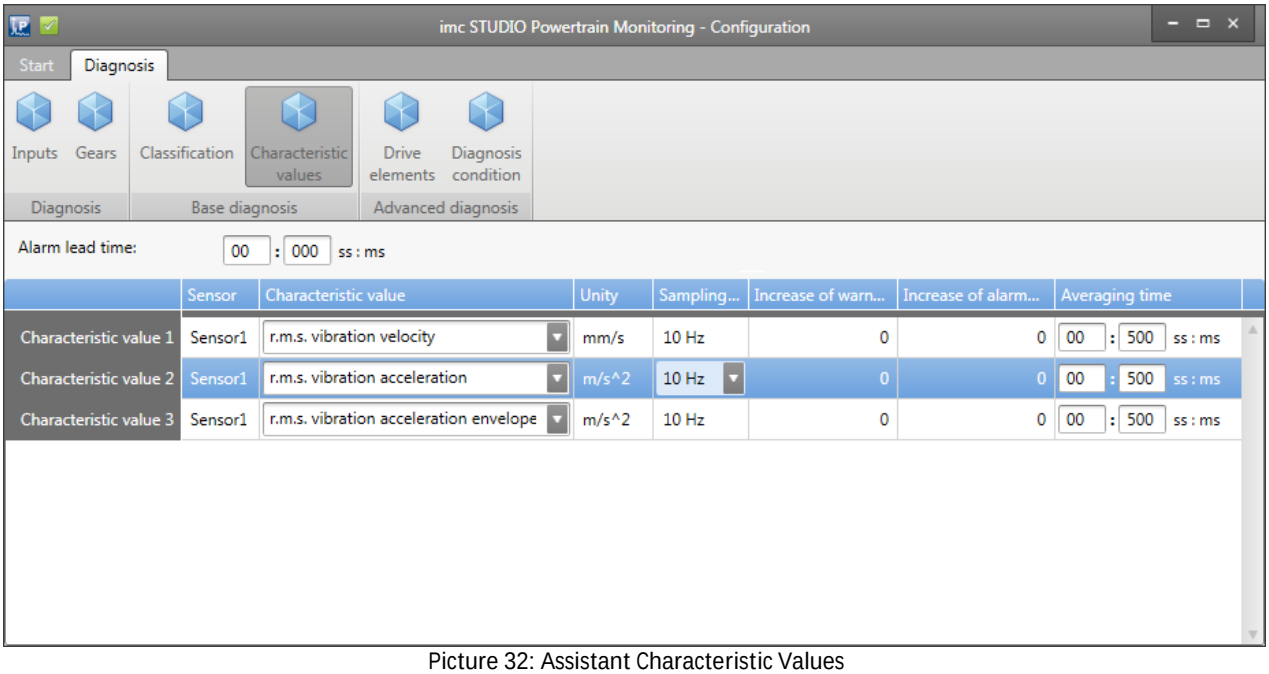
Schließen

Picture 31: Assistant Classification Details

For the classification each process information is used that were defined (see chapter Configuration 1270). 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 31).

An exception is the gear information. This process information can be configured as described in chapter "Gears 1271" and is not modifiable in this section.

12.5.3.1.2 Characteristic Values



Picture 32: Assistant Characteristic Values

The characteristic values can be configured in the corresponding menu point (see Picture 32).

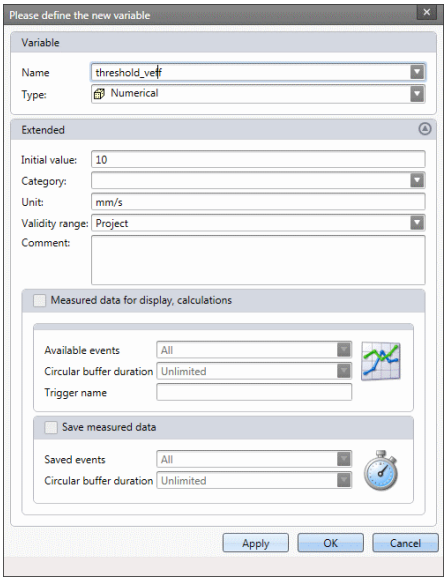
The following characteristic values are available for the base diagnosis for each configured accelerometer:

- 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.

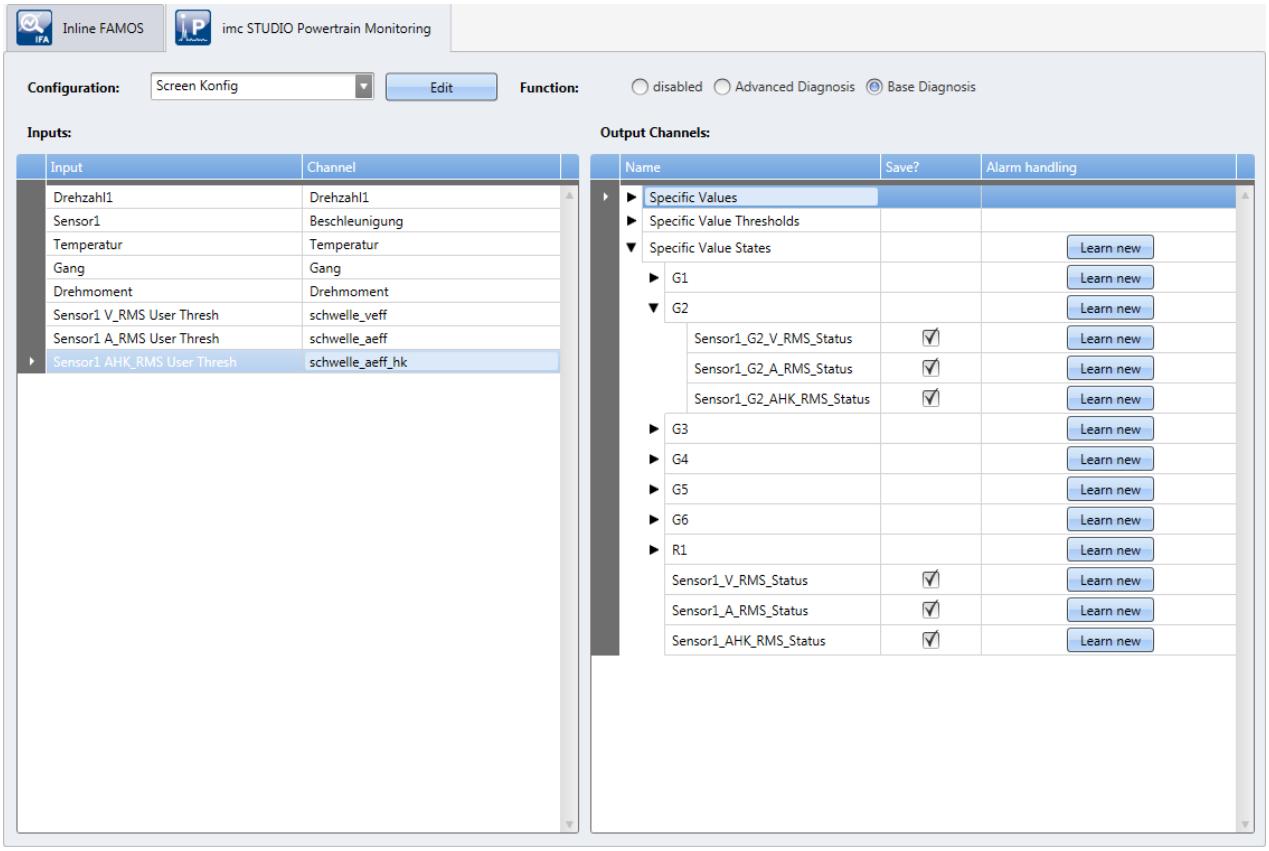
The configured percent increase of warn- and alarm thresholds are multiplied to the maximum characteristic value at the end of the learning phase of the class.

12.5.3.2 Editor



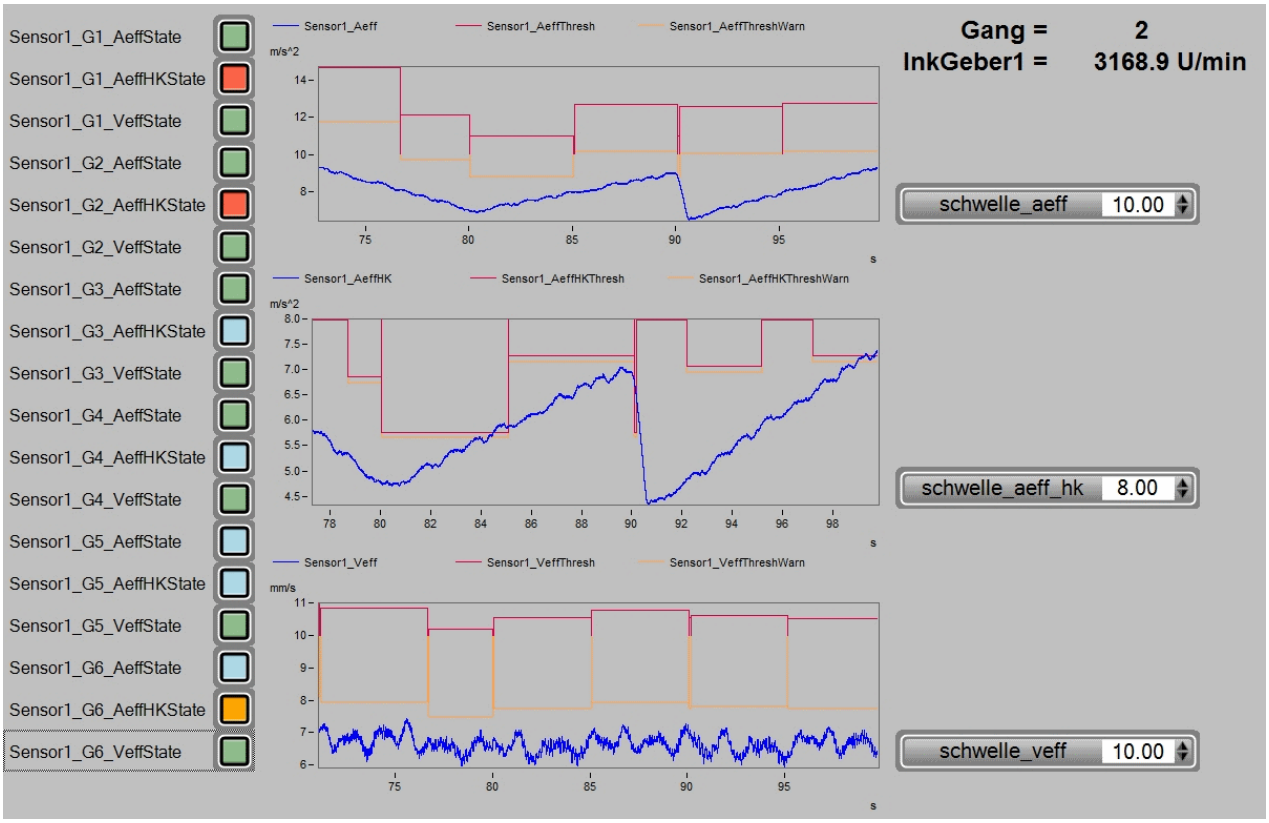
Picture 33: Create User defined variables

The inputs of the configuration must be assigned to the physical measurement channels for the base diagnosis (see also general description of the editor 1272). 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 33).



Picture 34: Base Diagnosis Editor

12.5.3.3 Measurement Data



Picture 35: 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 1276 on the right side at the various state channels.

Picture 35 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.

12.5.3.4 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.

12.5.4 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.

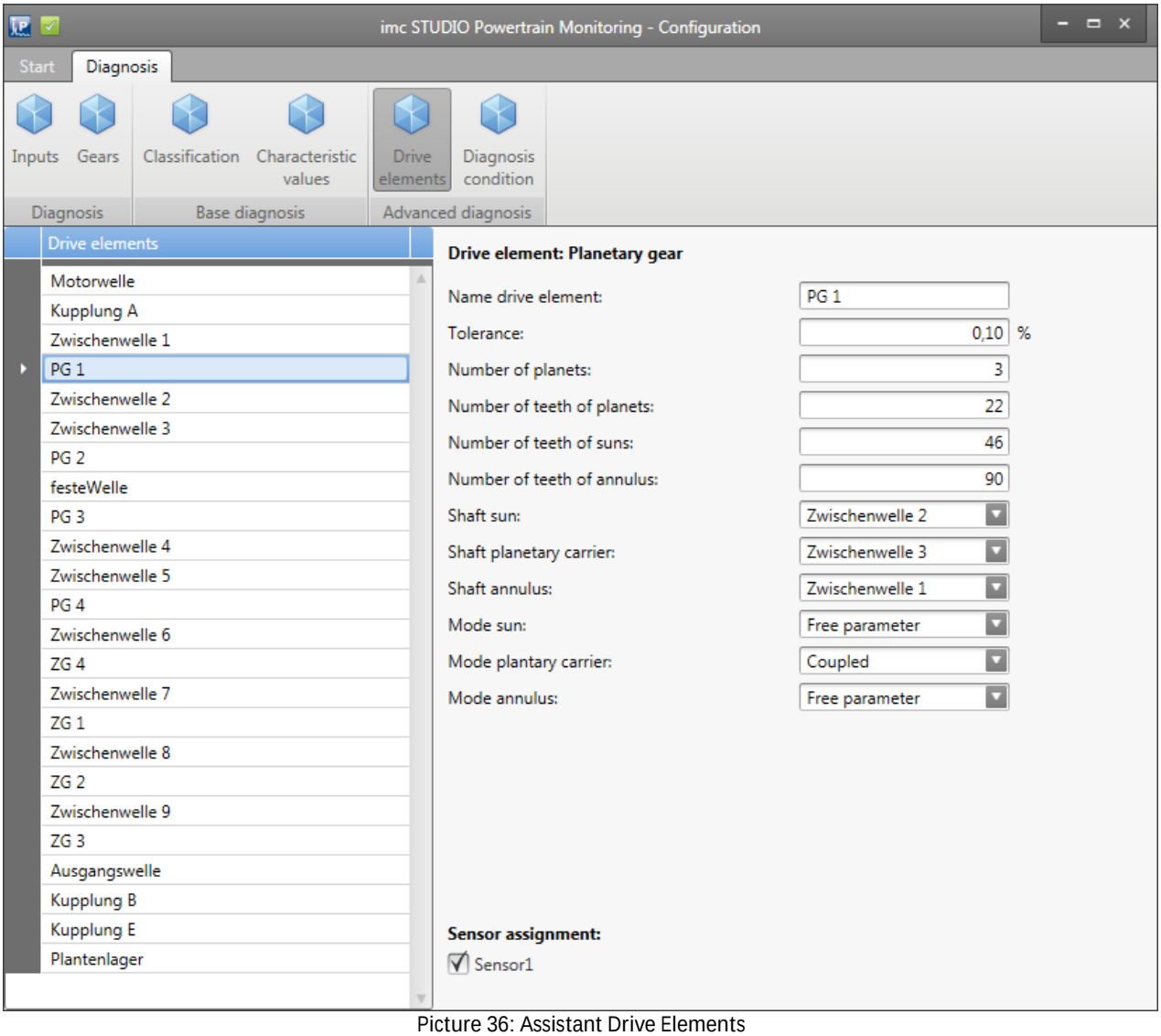
The calculated spectra are then analyzed according to a significance analysis. This is done by first calculating a moving significance threshold that represents the average amplitude over the abscissa. The spectrum is then divided by this threshold and all values below one are set to zero. The remaining 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.

12.5.4.1 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 1283 must be fulfilled over a configured measurement time. The measurement time is configured in the menu point inputs, as described in chapter "Configuration 1270". 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.

12.5.4.1.1 Drive Elements



Picture 36: 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 36). 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.

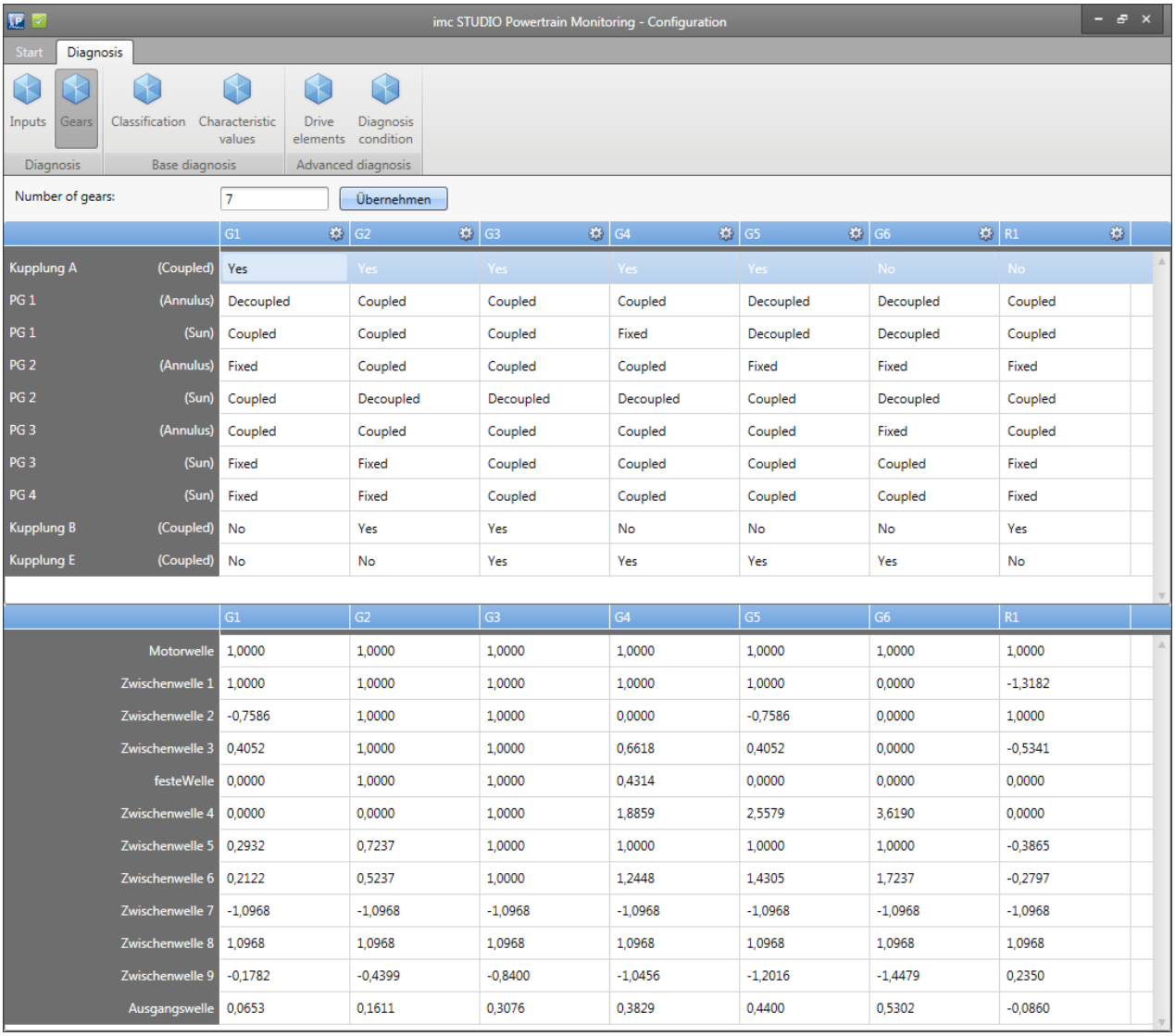
The following drive elements can be configured:

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.

12.5.4.1.2 Gears



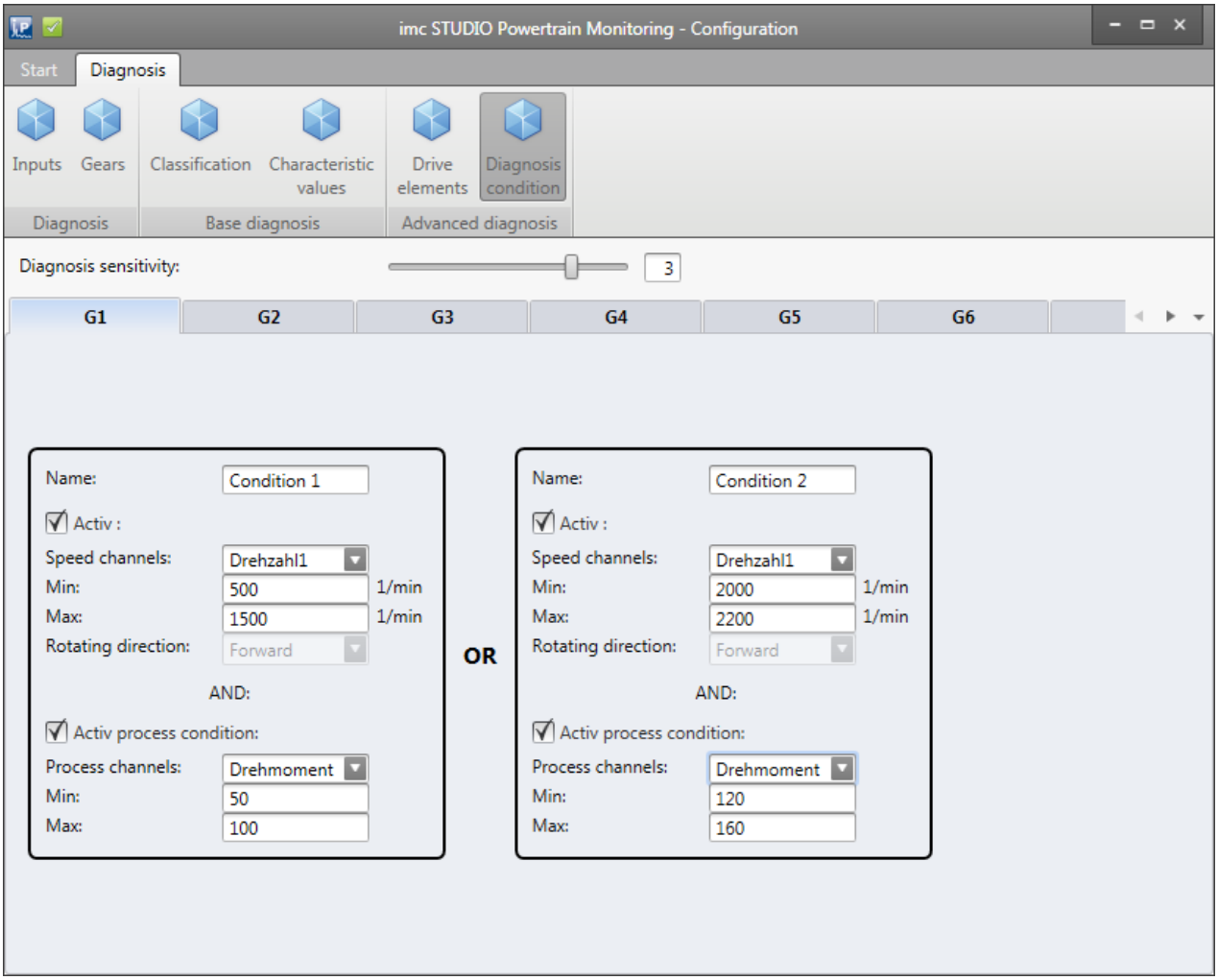
		G1	G2	G3	G4	G5	G6	R1
Kupplung A	(Coupled)	Yes	Yes	Yes	Yes	Yes	No	No
PG 1	(Annulus)	Decoupled	Coupled	Coupled	Coupled	Decoupled	Decoupled	Coupled
PG 1	(Sun)	Coupled	Coupled	Coupled	Fixed	Decoupled	Decoupled	Coupled
PG 2	(Annulus)	Fixed	Coupled	Coupled	Coupled	Fixed	Fixed	Fixed
PG 2	(Sun)	Coupled	Decoupled	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 37: Assistant Gear configuration

In the menu point Gears the free parameter can be set for each configured gear as described in chapter "Gears 1271". 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 37 shows an example configuration of a power train with 10 gears.

12.5.4.1.3 Diagnosis Conditions

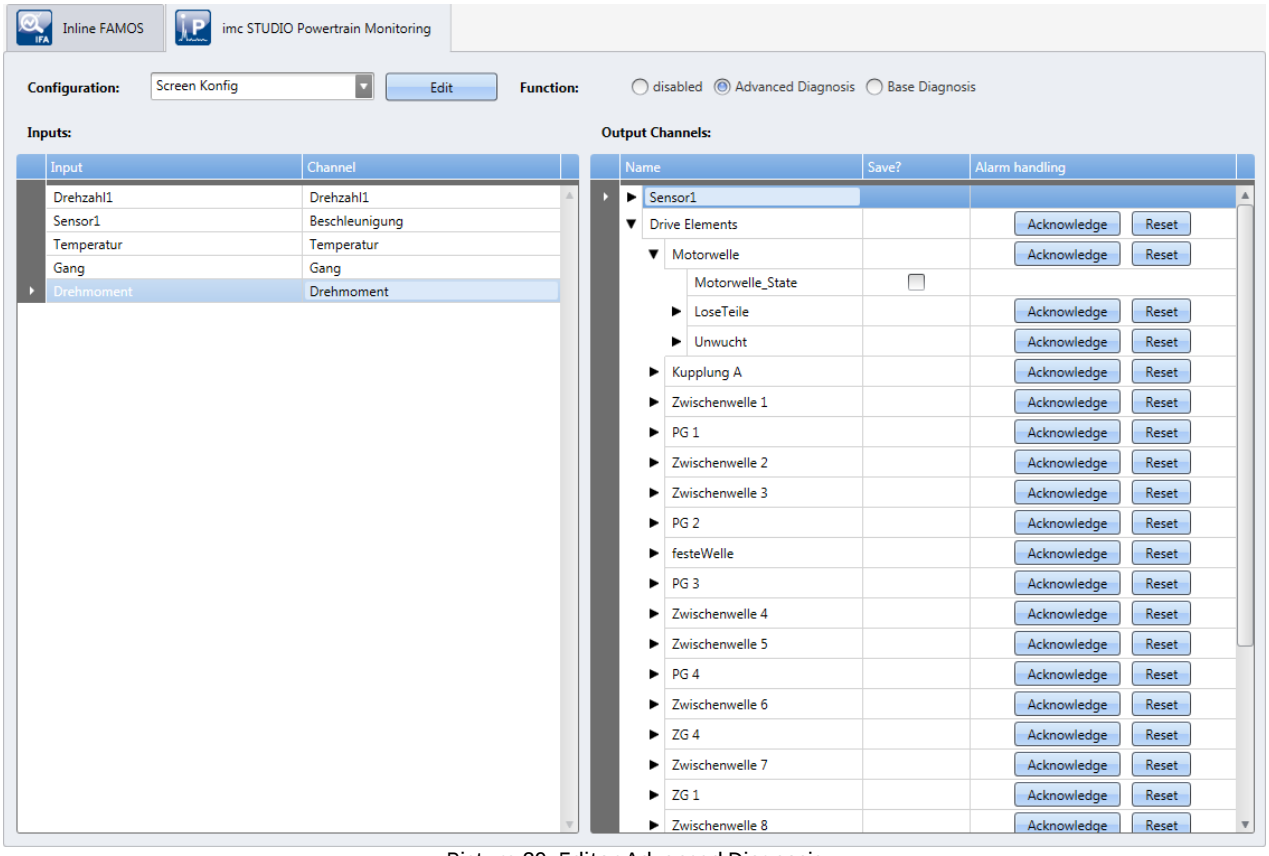


Picture 38: Assistant Diagnosis Conditions

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 38). If one of the conditions are fulfilled over the complete measurement time, the analysis of kinematic patterns will be started.

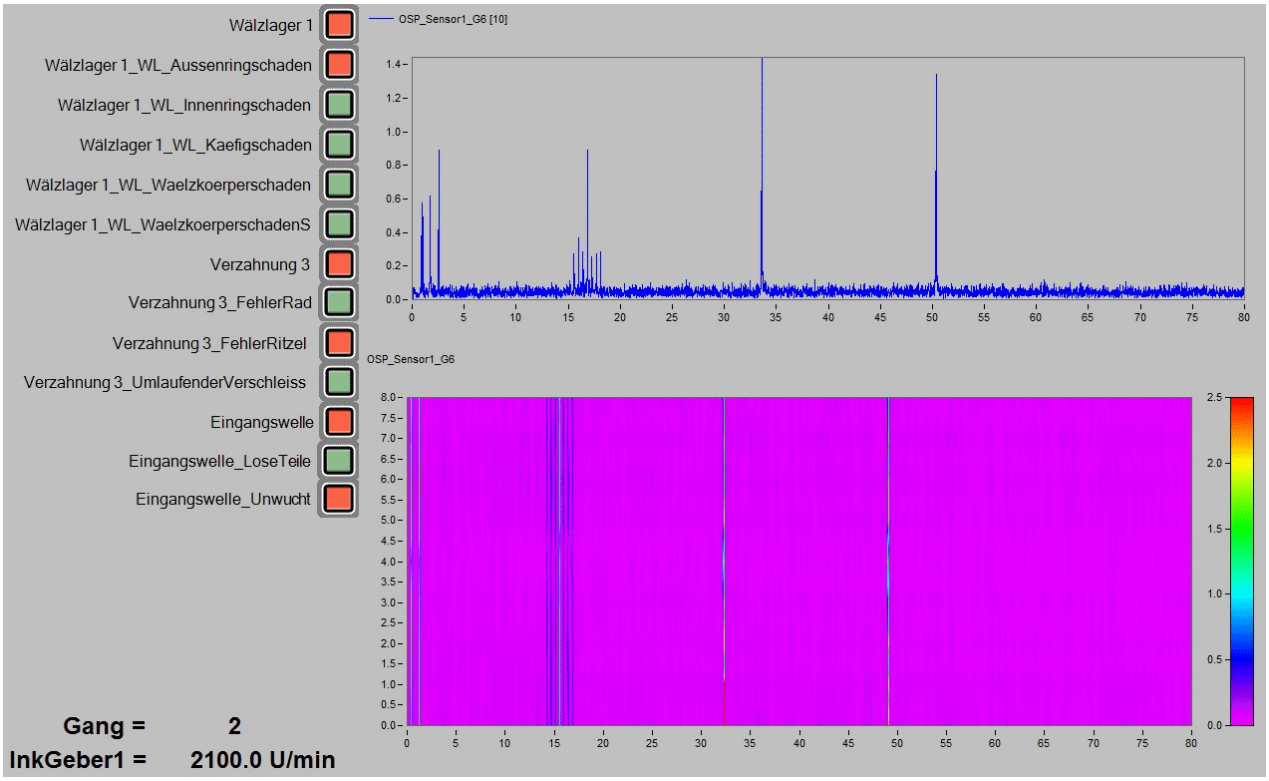
12.5.4.2 Editor



Picture 39: 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 1272).

12.5.4.3 Measurement Data



Picture 40: 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 1279 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 1284). 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 40 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.

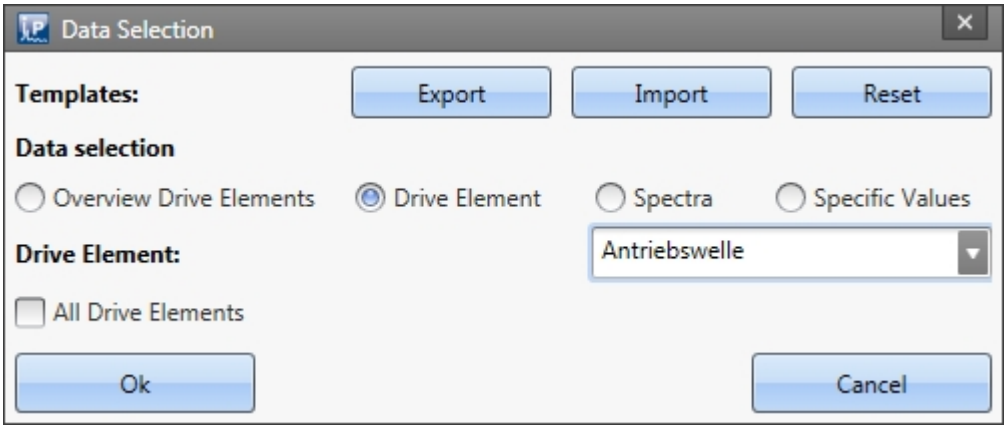
12.5.4.4 Summary of Function

- 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.

12.5.5 Visualization

12.5.5.1 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 41).



Picture 41: 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 1289.

12.5.5.2 Create Panel Pages via Scripting

The described functionality in "Create Panel Pages via Menu Action 1287" 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
}
```

12.5.5.3 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 und 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

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

13 Video

imc STUDIO Video is the imc STUDIO plug-in for capturing and processing video data.

The cameras must be connected to the operating PC on which imc STUDIO is running.You can record and save data from imc measurement devices and Video-data simultaneously.

Notes on the camera drivers

- imc STUDIO uses the camera's drivers. The drivers provide the camera settings, which you can configure on the imc STUDIO user interface.
- The possibilities vary from camera to camera. In consequence, this documentation may describe either more or fewer options than your camera possesses.
- imc STUDIO has no influence on what effect certain driver settings have. E.g., due to particulars of the driver, a positive brightness value may make the picture either brighter or darker.
  To allow you to adjust the settings as desired, a picture preview appears next to the options.
- Please ensure that the most up-to-date driver for the camera used is installed.

Warning about changing the driver

- Remove the camera from the imc STUDIO configuration, if you make a change of the driver manufacturer.
- All experiments with the old camera settings become inoperative.
- This note is also valid for the frame grabber and the compression.

Note on the PC performance

The achievable data rate depends on the performance of the PC used.

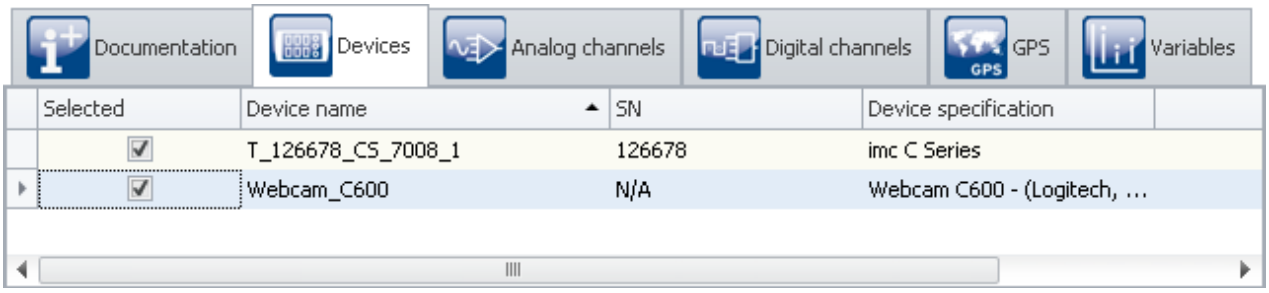
imc STUDIO Product Configuration 36

In order to use cameras, activate the option imc STUDIO Video Adapter in the product configuration.

13.1 Setup Configuration

You can configure and manage cameras in the same way as imc measurement devices. In order to use connected cameras, start a Search for Devices.

Following a device search, the cameras located are displayed in the Devices list.

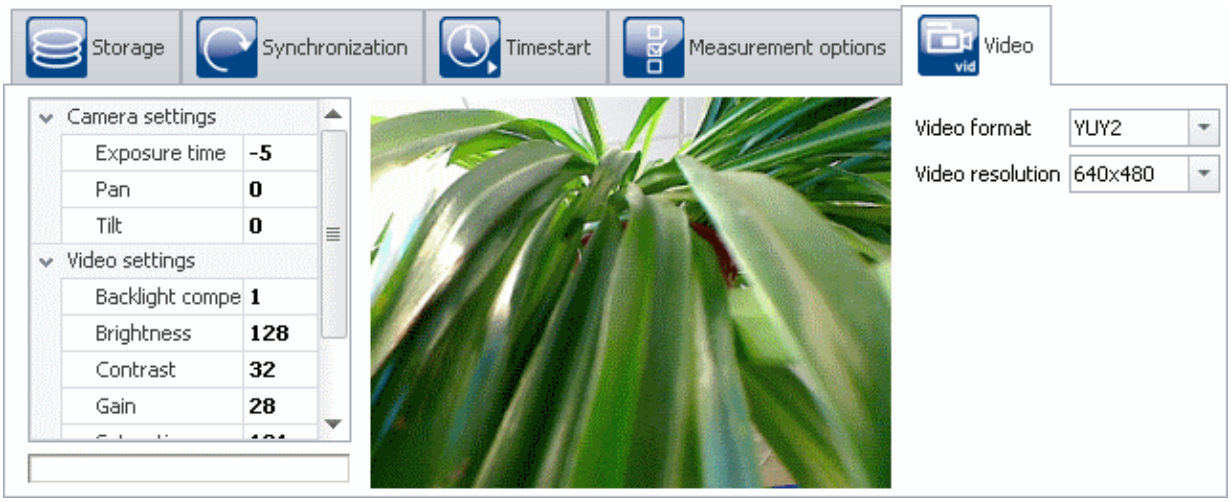


Device list with a camera selected and a device selected

The device search reports whether or not it found any devices. This report only pertains to imc devices. Any cameras will not be taken into account in the concluding report.

Setup: Devices - Settings and Video Preview

Once you have selected the camera from the Devices list, a new dialog appears: Video. Here you can change all settings.



Dialog: "Video"
On the Setup page: "Devices"

The video properties' values can be adjusted using a slider control, or by entering the desired value in the input box. By means of the Video Preview, you can test and correct the image reproduction. You can adjust the video format and the desired resolution. These settings determine the quality and volume of the video data saved.

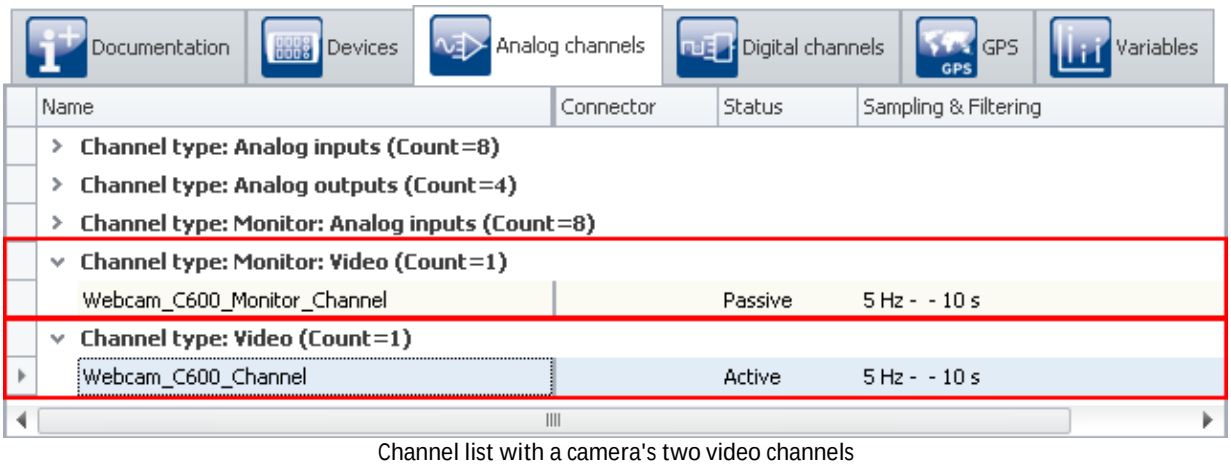
Warning about the use of high resolution and a high sampling rate

When the video resolution is high, you should avoid a high setting for the sampling rate. Otherwise, you may experience problems with the system's capacities.

Setup: Channels - Configuring "Video Channel"

The video channels are configured in the same way as the device channels. To do this, open the Setup page: Analog channels.

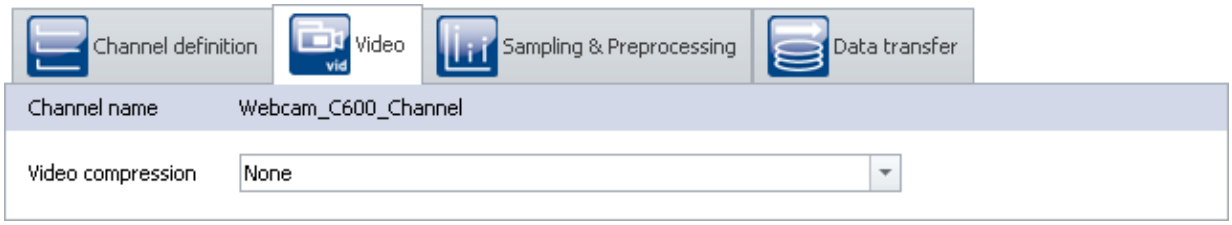
Each camera has a Video- and a Monitor-channel.



Channel list with a camera's two video channels

The main channel serves the purposes of high-speed data capture and storage, e.g. for Snapshots. The monitor channel, having slower sampling rates, serves other purposes such as long-term measurements or observation, without recording the video data.

A new dialog: Video is displayed if you select a video channel. Here, you can set the data compression.



Dialog: "Video"
On the Setup page: "Analog channels"

13.1.1 Saving

Video files are not save automatically!

- Activate data storage for the respective video channel, if you wish to save the video data associated with the measurement.
- Otherwise, the video data are only present until you re-start the measurement or close imc STUDIO, or load a different experiment.

Video channels can be configured just like the measurement channels belonging to imc measurement devices.

In order to activate data storage, put a check in the box at: Save (PC)

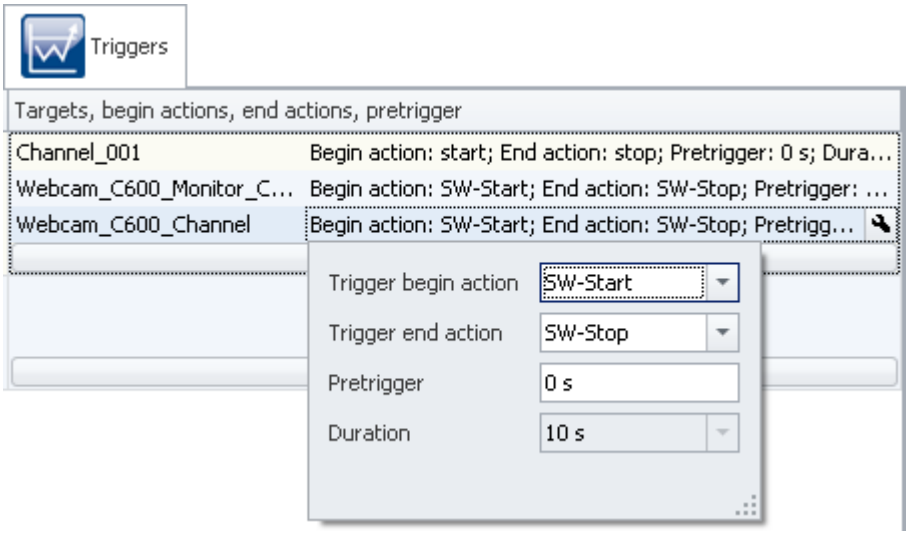
- The video files are saved on the PC's hard drive in the same folder as the stored data for the measurement.
- The size of the video files is limited by the hard drive.

Reference: Introduction to data storage

You can find an introduction to data storage in the manual:
Setup - Dialog: Data transfer 174.

13.1.2 Video measurement with triggering

The video channel can be triggered like measurement channels. To do this, open the Setup page: Triggers. Here, you can set the triggers for the video channels.



Notes on triggering

- Please be aware that you can only use a video channel as the trigger target in conjunction with active measurement channels.
- The main channel and the monitor channel can each be assigned to different triggers belonging to the imc measurement device.

The achievable frame rate decreases in consequence of video-triggering

With triggered data capture, the frame rate achievable decreases by half, since the circular buffer memory used makes greater demands on the system.

- This applies to all video channels which are not assigned to a 1-Trigger.

Save trigger events in individual files

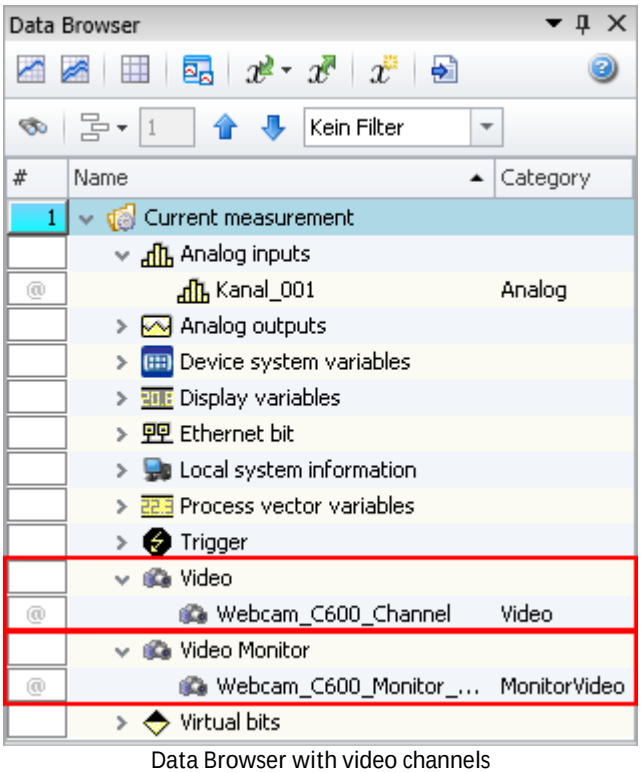
- The video channel's trigger events can also be saved in individual files (Setup page: *Devices > Storage > Save trigger events in individual files (PC)*).

Pretrigger

- With video channels, you can configure a pretrigger just like for channels belonging to a imc measurement device. This means that the pictures preceding the trigger event are recorded along with the data subsequent to the trigger event.
- The pretrigger duration can be set to values from 0 seconds to 10 minutes.

13.2 Displaying Video-Channels on a Panel Page

Click on Process configuration or Download. Then you are able to display the Video-channels on a Panel page.



Data Browser with video channels

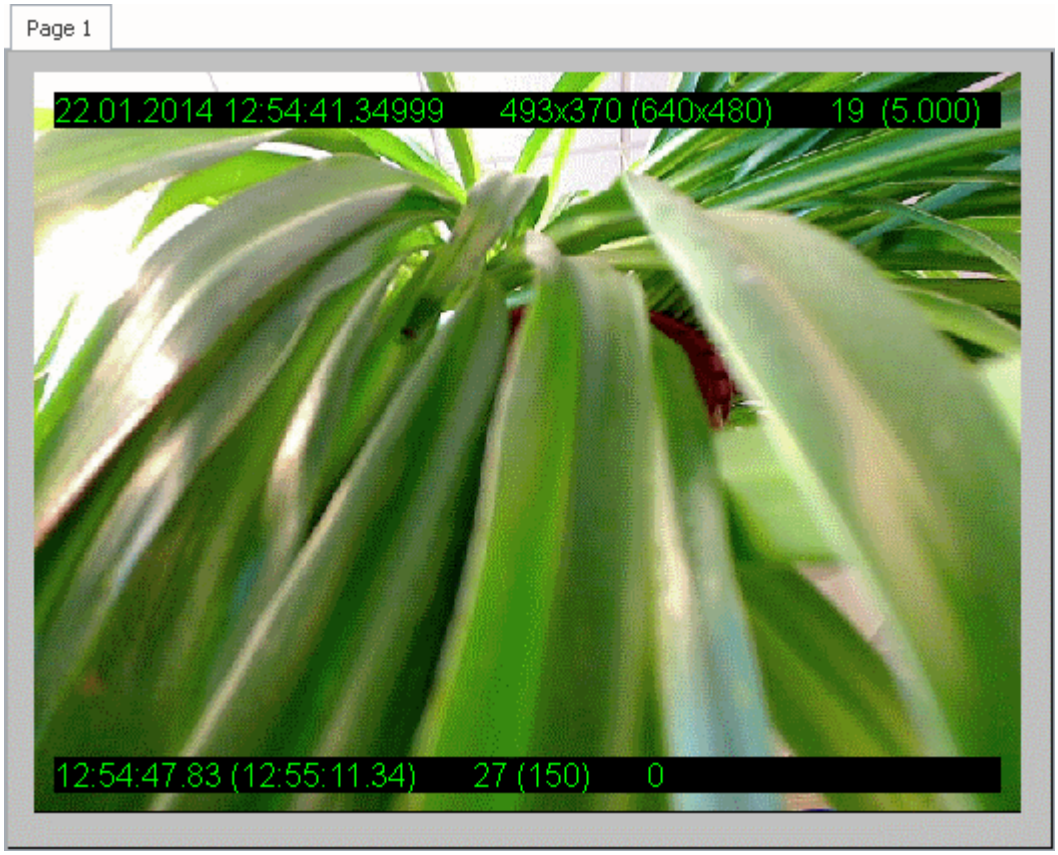
Treat the Video-channels on the Panel just like the other channels.

All active channels and thus also the video channels appear in the Data Browser.

Displaying videos on the Panel page

You can display videos on Panel pages.

Use Drag & Drop to move the video channel to the Panel page. The video-Widget appears along with the image reproduction. To configure the display, open the Properties.



Video-Widget with activated screen display

To each video-Widget, either a main channel or monitor channel, or a saved video can be assigned. You can position multiple videos on a single Panel page.

Notes on display

The Widget shows the current image prior to triggering of the measurement/release of the trigger.

On Screen Display (OSD)

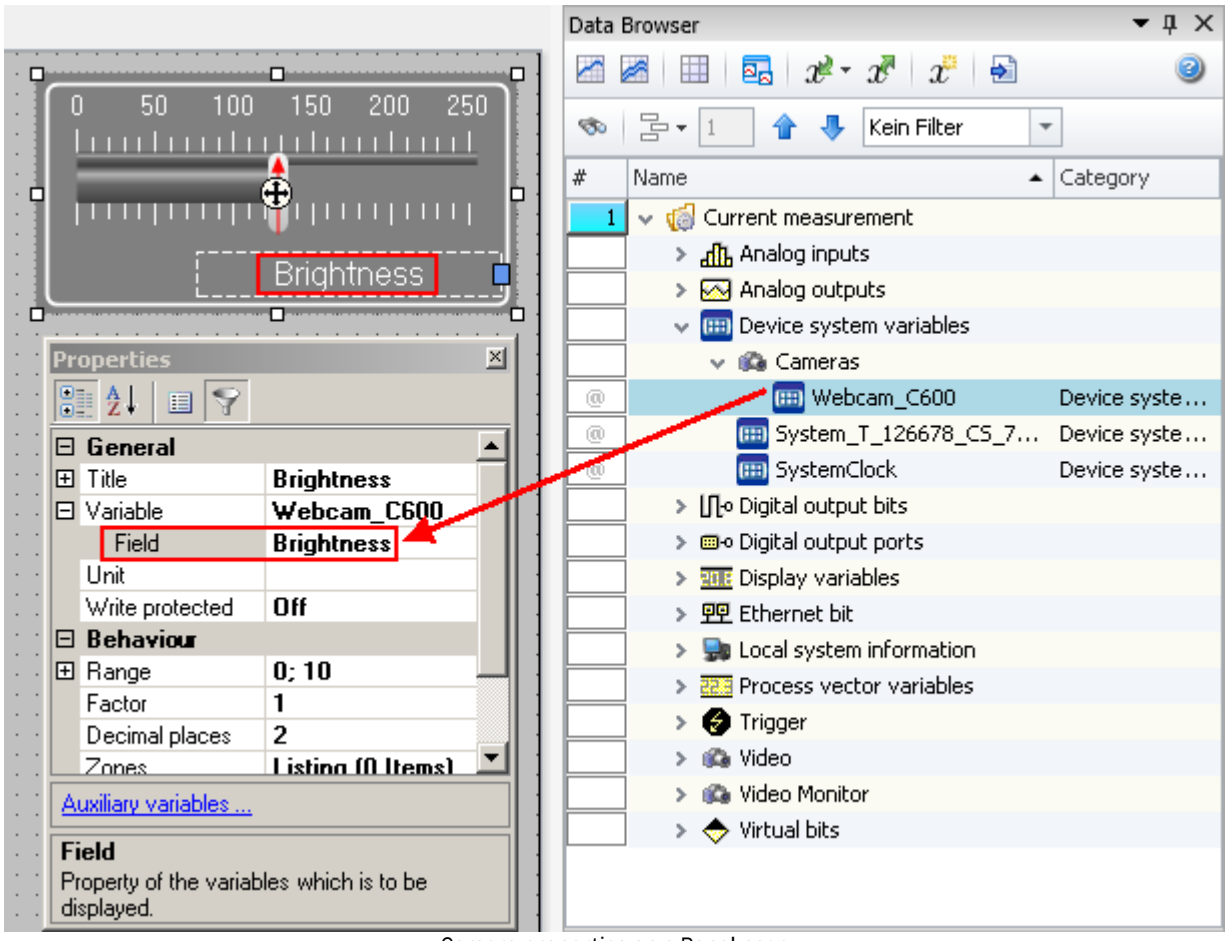
By means of the screen display (OSD), you can observe the sate of the video recording process. For example, with trigger measurements, the display will reflect whether recording is in process or has stopped.

Interpretation of the screen display

1. Date of recording
2. Time (start time)
3. Displayed resolution
4. Video resolution
5. Displayed frame rate
6. Set frame rate
7. Start time of most recent video recording, or time of current playback
8. End time of most recent video recording
9. During recording:
   Count of already recorded images
   During playback:
   Count of images already played back
10. During recording:
    Count of images to be recorded (depends on the measurement duration and the frame rate set)
    During playback:
    Number of recorded images
11. Event number of the current playback (for triggered measurement)
12. Event count

Changing video settings by means of the Panel page

Some camera settings can be changed directly via the Panel. To do this, drag an appropriate Widget (e.g. Automotive: Bar Meter) to the Panel page. Link it to the camera's *Device system variable*.



Camera properties on a Panel page

In order to be able to adjust a different camera property by means of the Widget, select the desired camera property as your choice for: *Field*, which is an extension of the property: *Variable*. It only appears when particular variables are linked with the widget.

The Widget now has direct influence on the future recordings and on the current video output.

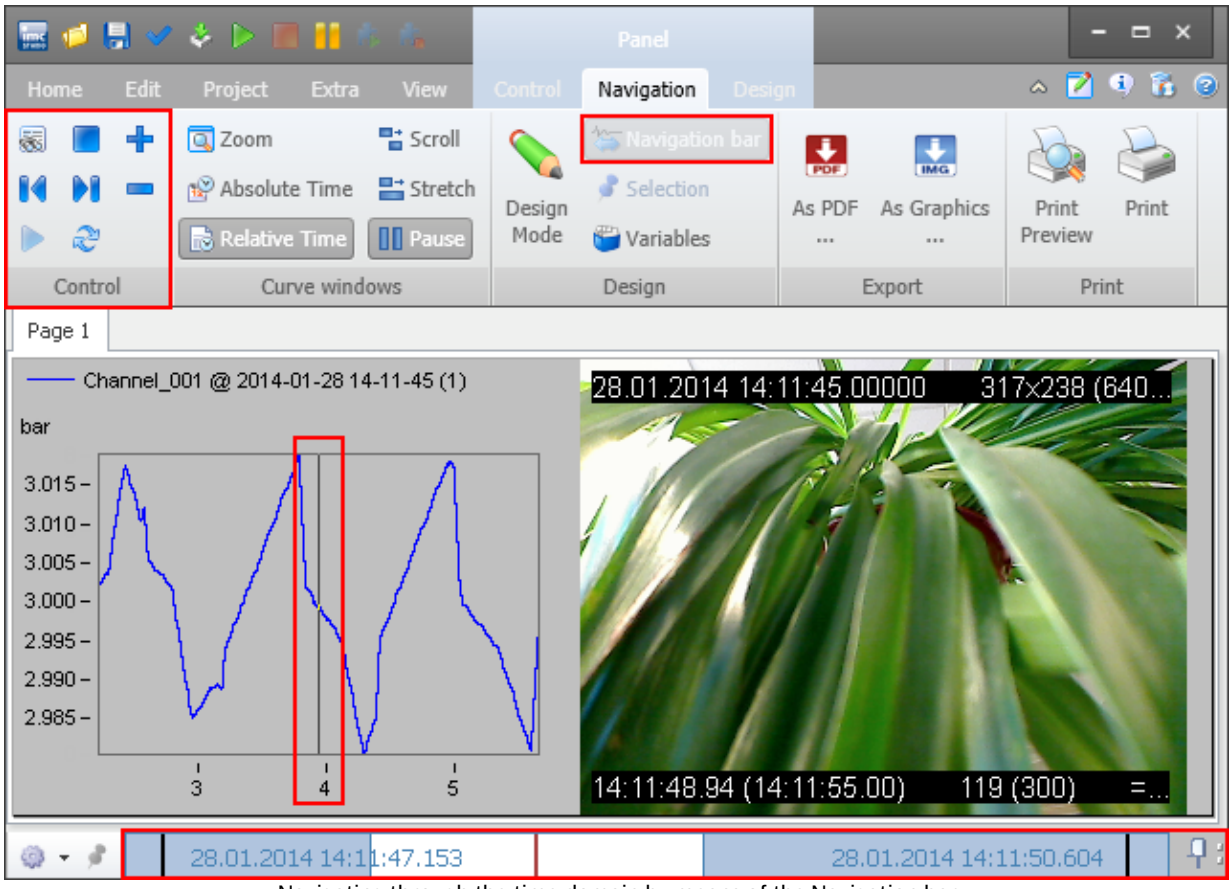
Video measurement - Recording and Playback

On a Panel page, you can view the measurement channels and video data during a running measurement. After conclusion of the measurement, you can play back the measurement channels and the video file jointly.

To do this, connect the corresponding curve window and the video-Widget with the Navigation bar.

By means of the menu item *Panel-Navigation* > *Control*, it is possible to skip to the recording's start time and play it back. You can also navigate to the desired position by means of either the Navigation bar or the curve window.

To learn about the many navigation capabilities, see the documentation on the Navigation bar.



Navigating through the time domain by means of the Navigation bar
Video-Widget and curve window

Playback of saved video files

If data storage is activated both for the measurement channel and the video channel, the measurement results are displayed in the Data Browser as a measurement.

In order to be able to play back various measurements on the same Panel page, do not associate the Widgets with the measurement's *Fixed name*, but rather with its *symbolic measurement number* 1089.

In this way, you can play back the current measurement and the saved measurements in succession on the same Panel page.

13.3 Export/Import of video files

Video channels can be exported/imported just like measurement channels. This can be done either via the Data Browser or by means of commands.

To do this, use the function:

Export Variable	Exports variables' values to a file. In this case, the video recording (values) belonging to the video channel (variable) are saved in a folder (file) chosen by the user.
Load Variable	Creates variables using values from a file. In this case, a video channel (variable) is created. This contains the selected video (values) and can be played back.
Import Variable	Imports the values from a file to existing variables. In this case, data is imported into an existing video channel (variable). This contains the selected video (values) and can be played back.

The available file format is *Windows Video File (AVI)*.

Long waiting times may occur

When importing/exporting large video files, long waiting times may occur.

Additional information on exporting 1645 and importing 1647 is presented in the descriptions in the reference of commands .

13.4 Information and tips

13.4.1 Specifications and Notes

Data capture:

- Two measurement channels per camera are available simultaneously: the main channel and the monitor channel. Among other things, this means that you may apply different settings for the sampling rate and triggering.
- Main channel for high-speed capture and storage, e.g. for triggered snapshots.
- Duplicated monitor channel at low-speed sampling rates, e.g. for untriggered, continuous long-term measurements.

Visualization:

- A video-Widget (Window) is provided to display video data on Panel pages.
- Multiple such video windows can be placed on the pages of the Panel.
- Each video window can be assigned to either a camera's main or monitor channel.
- In the video window, data are displayed even before release of the trigger.

Trigger:

- The main- and monitor channels can each be assigned to different imc measurement device triggers.
- The imc measurement device's triggers are also the triggers for the camera. This means that video channels are triggered at the same time as the associated imc measurement device channels.
- Pre-trigger: As for the imc measurement device channels, it is also possible to configure pretriggers for video channels. This means that the data recorded can also include images relating to situations prior to the trigger events.
- Pre-trigger duration: 0 sec to 10 minutes.

Synchronization:

- Automatic synchronization of the video- and measurement data.
- The achievable accuracy depends on the workload of the entire system.
  Up to dt = [1 frame duration + 20 ms] is achievable.
- The device must be connected with the PC by an Ethernet line of at least a 100 MBit/s , with a maximum of 1 hub or switch in between. The connection must remain intact for the duration of the entire measurement.

Advisory notes:

- For stable and reliable operation we recommend the use of tested and approved camera models and selected software drivers according to the tables: *Supported Cameras* 1698. imc can only grant support for these combinations listed.
- Systems with cameras from different manufacturers are not recommended, since the drivers could also interact uncontrollably.
- For maximum cable length between the control PC and camera, please refer to the respective technical specs for the camera.

Data throughput / frame rate:

- The data transfer rate is specified as the frame rate (typically 60 fps)
- The frame rate is based on pictures of the size 640 * 480 pixels with 1 Byte per Pixel in Bayer encoding, meaning 300 kByte per frame. This results in 17.5 MByte per second being continually written to the data storage medium.
- With the computer equipped accordingly, in Bayer format up to 100 fps can be achieved for 640 * 480 pixels or 200 fps for 320 * 240 pixels.
- The data transfer rate stated is aggregate. When multiple cameras are used, they split the transfer rate. Thus, a camera with 60 fps has about the same transfer volume as two cameras with 30 fps apiece. One camera at full resolution of 640 x 480 generates about the same data volume as four cameras with 320 x 240 resolution.
- With other encoding (such as RGB instead of Bayer), the data volume is increased threefold. This means the achievable frame rate is reduced to one third.
- With triggered data recording (all video channels not assigned to any 1-Trigger), the achievable frame rate is cut in half due to higher demands on the system made by the circular buffer memory used.

Prerequisites for achieving maximum frame rate:

- Windows 7 operating system. The performance of XP and Vista is lower in certain areas.
- Hard drives: Solid State Disk (SSD) or 3.5” SATA hard drives (at least 5400 revolutions per minute) configured as Raid 0 [1]. Please note that 2.5” hard drives are much slower. Particularly in notebooks, slower hard drives are often installed.
- The data carrier may only be filled to a maximum of 70%. Note that writing to a harddisk that is almost full results in significantly reduced writing speed.
- The data carrier may not be fragmented. Note that high writing speed is only achieved if the write head is not forced to excessive displacements.
- Hard drive controller: This must allow data throughput in write mode. Please note that during measurement operation mode, not only video files have to be written, but other data as well!
- Processor: Quadcore with 2.4 GHz (or in case of using Intel I7, two processor cores should be sufficient).
- Interface to camera: 1 GBit Ethernet, Firewire A or B or USB as of Version 2.0.
- No virus scanner for video files.
- No backup tool (or synchronization tool) in use during the measurement.
- No additional programs running on the computer. Also services such as hard drive defragmentation or file indexing may not be running during measurement.

[1] A RAID system consists of multiple hard drives connected together in Stripe-Mode (RAID 0). This increases the capacity as well as the data throughput. It is also possible to connect more than two hard drives, but eventually the hard drive controller imposes limits on the data throughput.

Crucial parameters for optimum frame rate performance:

- The maximum frame rate is determined by the camera’s properties
- The performance capacity of the interface to the camera, e.g. 400 MBit/s for Firewire A
- The hard drive controller's and its PC interface and driver
- Processor and mainboard chip set
- The hard drive's maximum writing speed
- Compression

13.4.2 Video Compression

Dated: 2013

A program for the video compression is available for a download on the following web page:

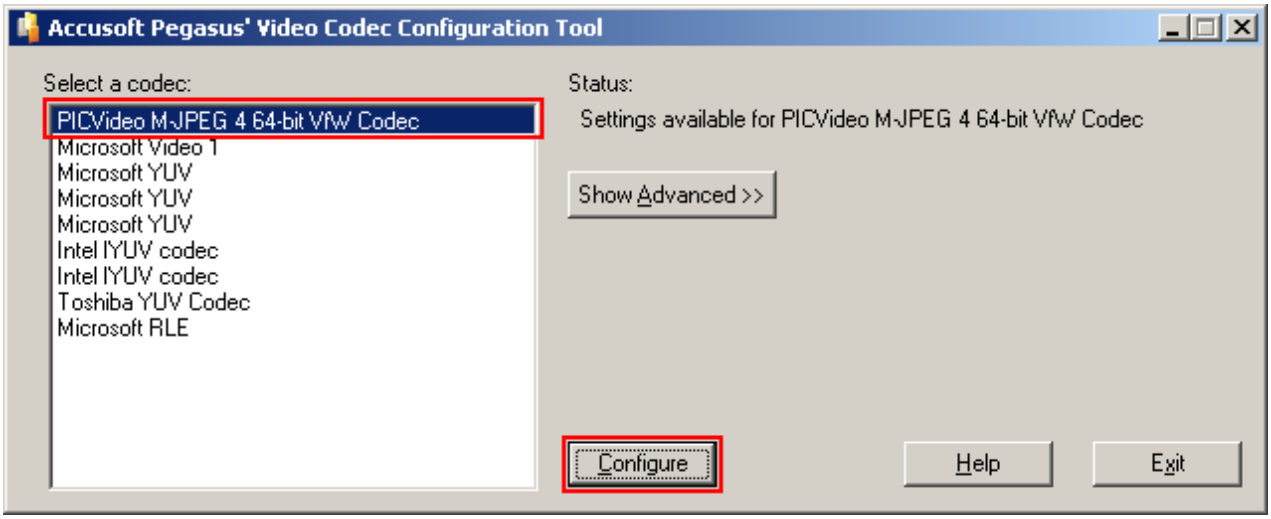
www.accusoft.com/pvmjpegpricing.htm

On this page the following option is tested and recommended by imc: PICVideo Motion JPEG Codec - 32-bit Compress and Decompress.

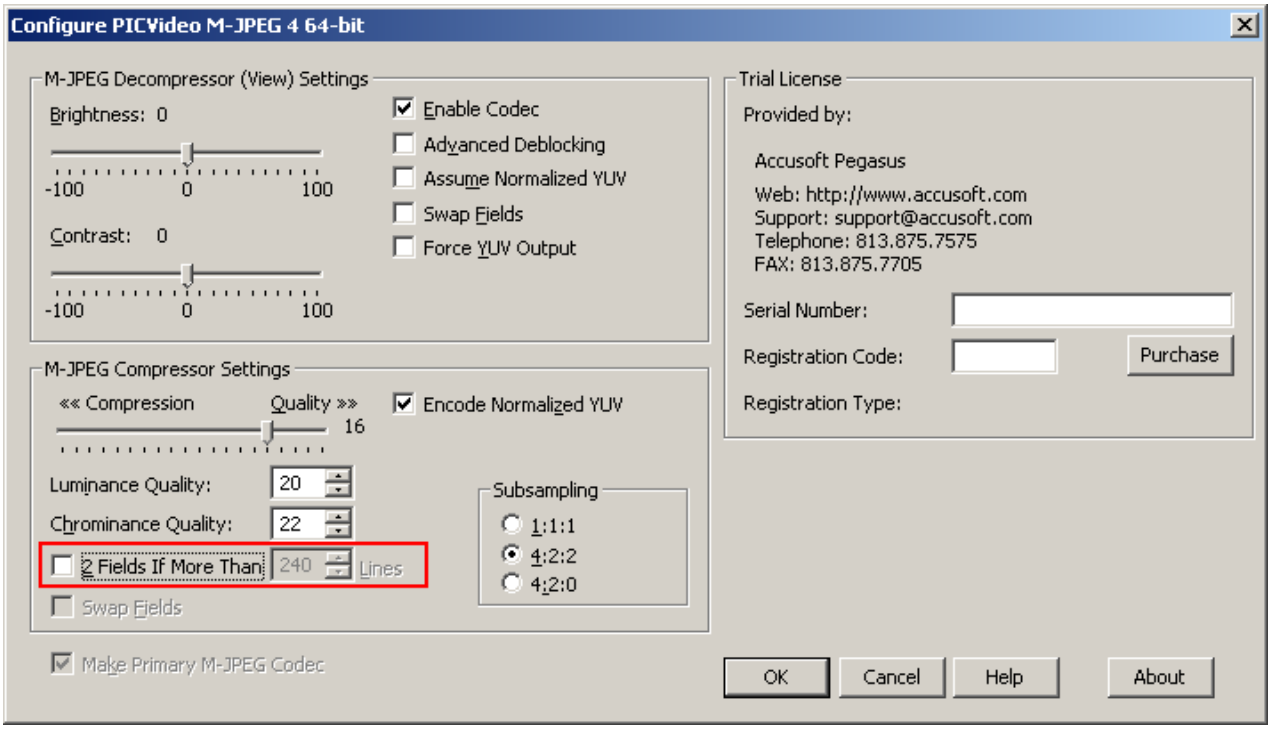
Attention: Version 4 is not supported

Please contact the manufacturer about Version 3 of PICVideo codecs.

After successful installation of the PICVideo, please open the configuration program. Select and configure the Codec: "PicVideo Codes":



Deselect in the configuration dialog the option *2 Fields If More Than ... Lines*:



13.4.3 Camera: GoPro Hero 4

Installation:

To operate the GoPro in imc STUDIO, two additional programpackages are required. You can obtain both packages from the imc Hotline.

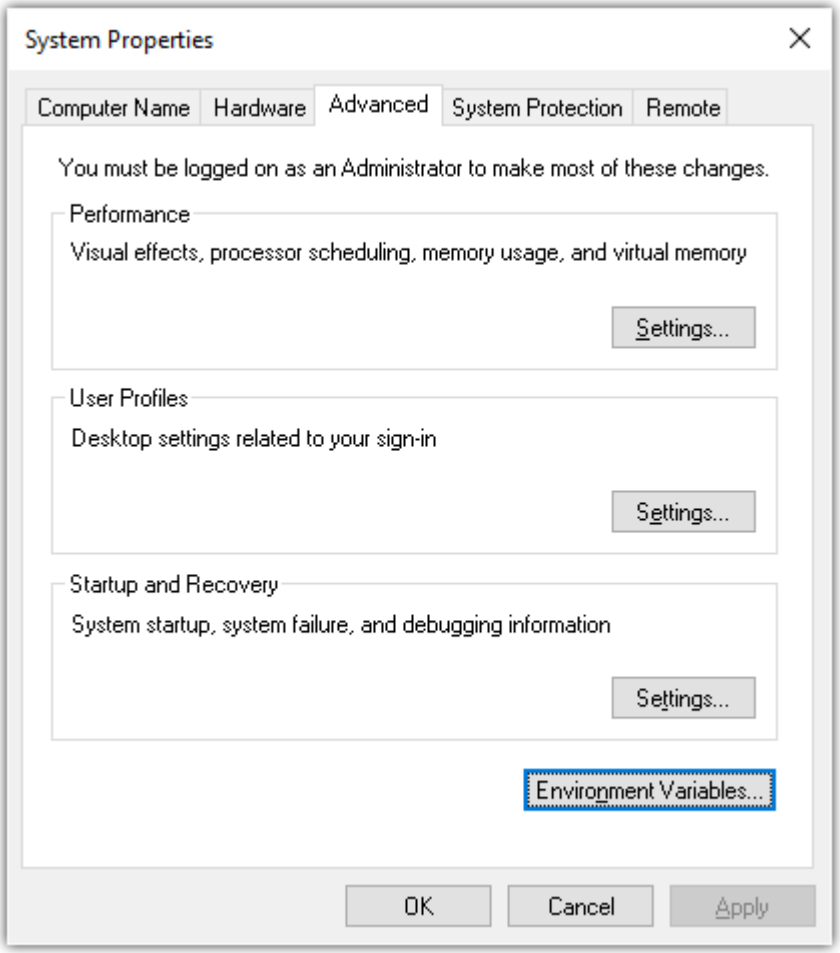
- DatasteadRTSPFilter and
- FFMPEG

RTSPFilter: Run the installer: "DatasteadRTSPFilterInstaller.exe" and follow the instructions.

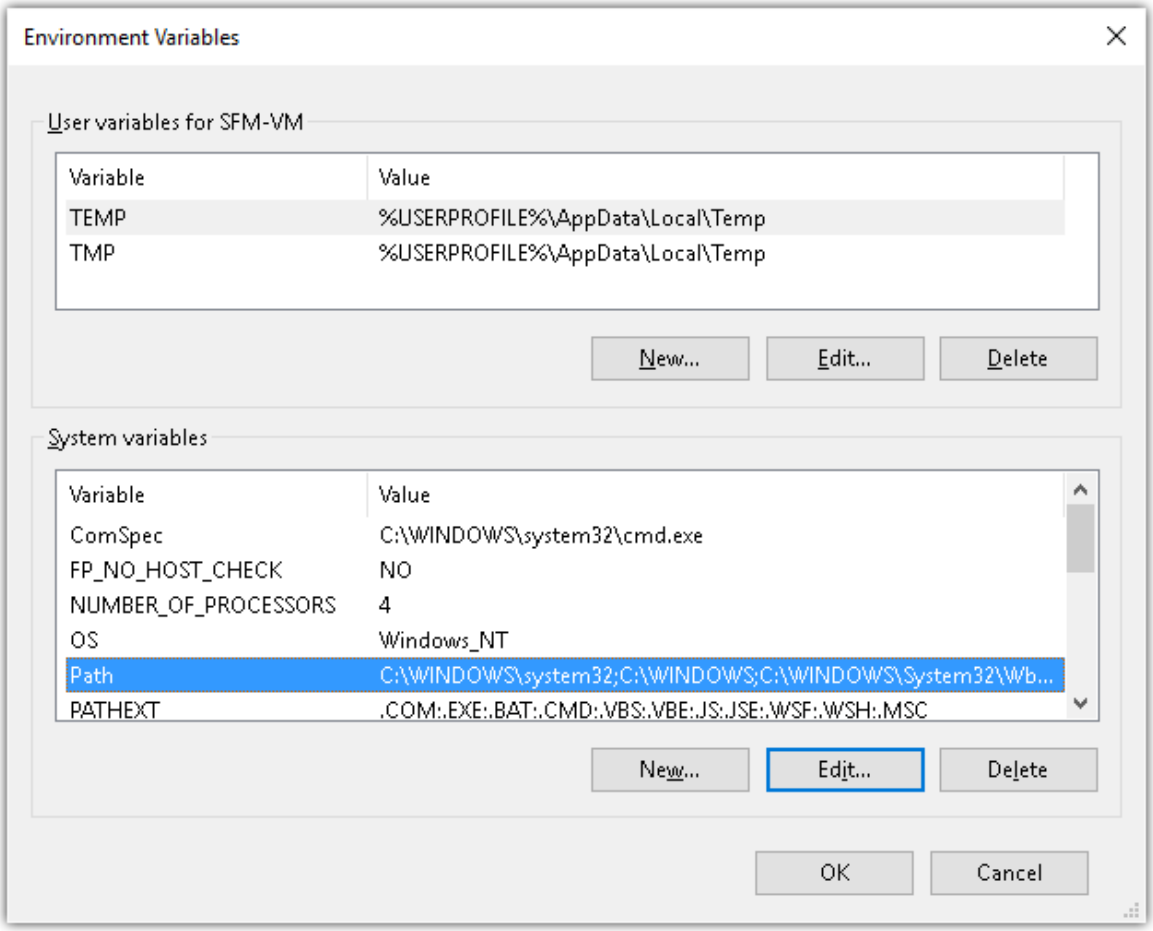
FFMPEG:

- Unpack the archive "ffmpeg-20160404-git-54c9146-win64-static.7z" in the desired installation folder (e.g. "C:\Program Files\ffmpeg-20160404-git-54c9146-win64-static").

  It may not be possible to unpack directly to "C:\Program Files" if Administrator privileges are required. In this case, extract the archive to any other arbitrary location and move it subsequently to the desired location.
- Next, the Windows "Path"-variable is modified accordingly.
- To do this, open the Control Panel and find "Environment variables". Select either "Edit environment variables for your account" or "Edit the system environment variables", depending on you operating system.
- The dialog "System Properties" then appears. There, click on the button: "Environment variables".



- In the new dialog, under "System variables" select the variable "Path". Edit these.



- In the example above, the path "C:\Program Files\ffmpeg-20160404-git-54c9146-win64-static\bin" must be added.

Notes

- In order for the firewall used not interfere with communication between imc STUDIO and the camera, the UDP Port: 8554 must be enabled.
- GoPro itself provides a Wireless Access Point. The IP address of the GoPro is 10.5.5.9 and can not be changed. In order to avoid an IP-address conflict, it makes sense to select a different, non-routed IP-address range, e.g. 172.16.0.0 - 172.16.0.255.

Commissioning:

First, the GoPro must be connected with the WLAN; to do this follow the manufacturer's instructions. Next start imc STUDIO and run a device search.

Special features:

The GoPro saves its recordings on a removable memory card (micro-SD). In the imc STUDIO database, only an info file with the extension "ivi" is saved. In this file, information such as the triggering time, offset, filename (on the GoPro) ... are saved.

Note

The resolution displayed in imc STUDIO and framerate of the GoPro camera is only the same as those of the preview. The recordings are always made according to the values (manually) set on the camera.

Warning: Only use one GoPro camera

In imc STUDIO, only one GoPro camera at a time can be used in an experiment.

13.4.4 Integrated imc STUDIO-PC with video

A video-PC is a PC unit integrated in the imc CRONOS*compact* rack.

Reference to the device documentation

Details and specifications are available in the device manuals.

Note on the Windows version

This description applies to Windows 7. The information pertaining to other Windows versions may vary.

Windows remote connection to the video-PC

Prerequisite: two computers having an Internet connection, in this case the video-PC and a Client computer.

Prerequisites for a remote desktop connection

It is not possible to establish a remote connection with every edition of Windows. The remote computer (video-PC) must run one of the following Windows 7 editions as its operating system:

- Windows 7 Professional
- Windows 7 Ultimate or
- Windows 7 Enterprise.

On the client computer, what Windows 7 edition is running does not matter. It is possible to establish a remote connection to the video-PC with any edition.

The video-PC must be switched on and may not be either in Standby mode or hibernation. If necessary, check and adjust the PC's energy saving setting beforehand.

Check the video-PC's firewall configuration and ensure that the port for the remote desktop (normally port 3389) is open.

Configuring the remote computers (video-PC)

Open the Remote settings:

- Open *Control Panel* > *System and Security* > *System*.
- Select *Remote settings* at left. You may need to enter an Administrator password.
  - Under *Remote Assistance*, activate the option *Allow Remote Assistance connections to this computer*.
  - Under *Remote Desktop* select and configure one of the two alternatives for allowing the connection.
- Confirm by clicking in *OK*.

Establishing a remote connection from the client computer to the video-PC

- Open the Start Menu.
- Type the word *Remote* in the search box.
- Click on the result: *Remote Desktop Connection* to start it. The *Remote Desktop Connection* dialog appears.
- In the input box *Computer*, enter the name or the IP-address of the video-PC. To configure the connection, click on the button *Show Options* to open the corresponding dialog.
- Start the process of establishing the connection by clicking on *Connect*.
  - If necessary, enter a user name and password.
- if it was possible to establish the connection, the video-PC's login dialog appears.
  - Enter the user name and password here, too, if required.

VNC connection to the video-PC

Prerequisite: two computers, in this case the server (video-PC) and a client computer.

Installation of the VNC software (UltraVNC)

- On the client-PC, install only the component selection: *UltraVNC Viewer*.
- On the video-PC, install only the component selection: *UltraVNC Server*.

Server configuration (video-PC)

Specify a password

- Open the context menu belonging to the VNC-Task bar symbol: blue eye.
- Select Admin Properties.
- Under *Authentication* > *VNC Password*, enter a password to apply to the connection.
  ATTENTION: The connection is not encrypted. Do not use any important passwords associated with other programs!

Check the Firewall settings

Check the firewall configuration and ensure that the program winvnc.exe is present in the list of exceptions.

Establishing the VNC connection from the client computer to the video-PC

- On the client computer, start the *VNC Viewer*.
- Enter the video-PC's (the Server's) IP-address.
- Start the setup for the connection and enter the password, if required.

14 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, all fundamental functions of imc STUDIO are available, e.g.

- read and write device and channel settings (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,
- 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

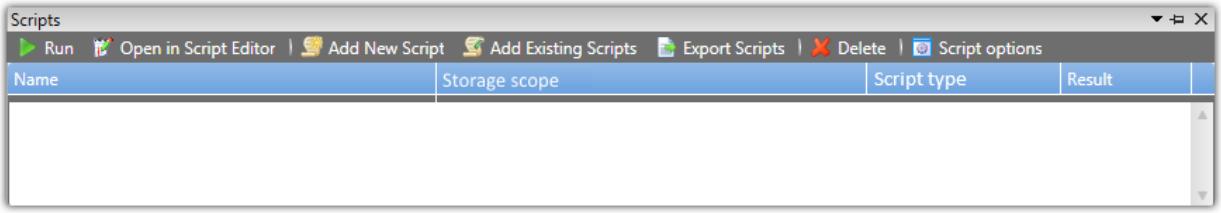
14.1 Menu bar and context menu

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"

Hint

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

Menu action	Description
Run	Runs 1317 the selected script.
Open in Script Editor	Opens 1317 the selected script in the editor.
Add New Script	Opens the dialog for creating 1312 a script.
Add Existing Scripts	Opens the Import-Dialog 1318.
Export Scripts	Opens the Export-Dialog 1318.
Delete	Deletes the selected script.
Script options	Opens 1390 the script options in the imc STUDIO options dialog.

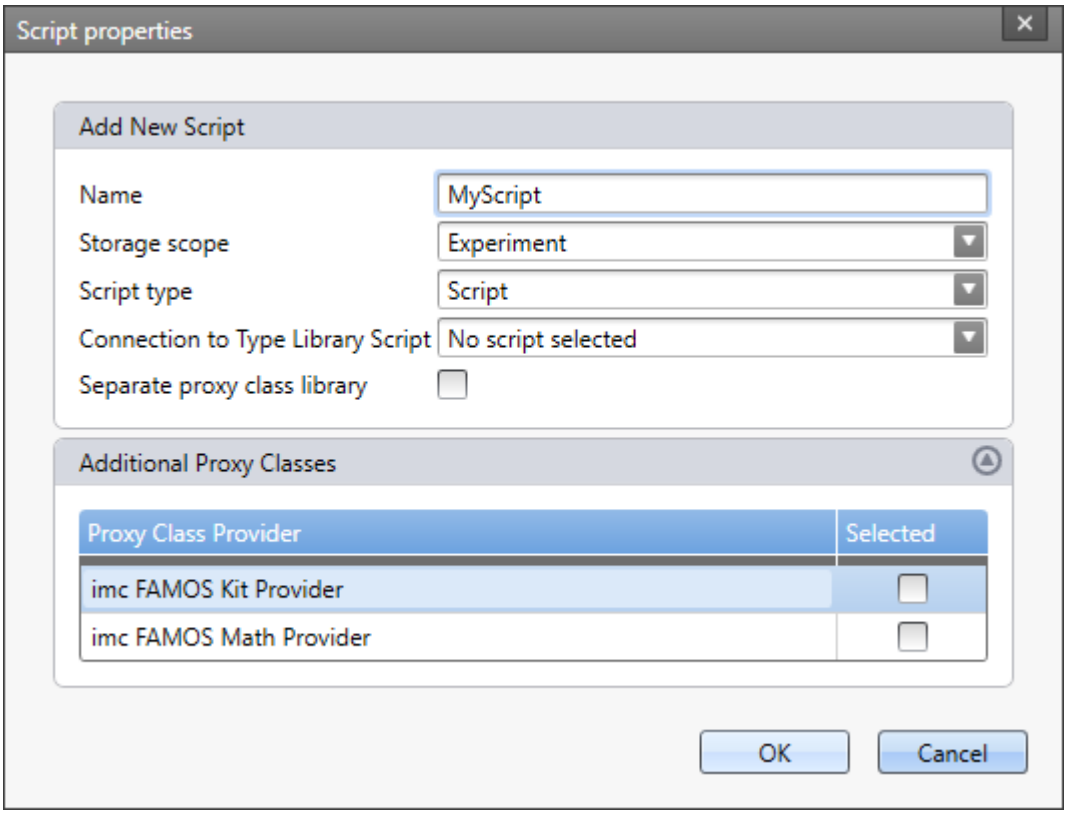
To edit a script, select the menu item *Open in Script Editor* Open in Script Editor.

Context menu

Menu actions	Sub-menu
Add	• Add new Script 1312 • Add existing Scripts 1318
Edit	• Regenerate proxies 1319 • Show build log
Script actions	• Script Properties: New Storage scope 1316 • Regenerate Script Widgets 1358 (Panel script)

14.1.1 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.

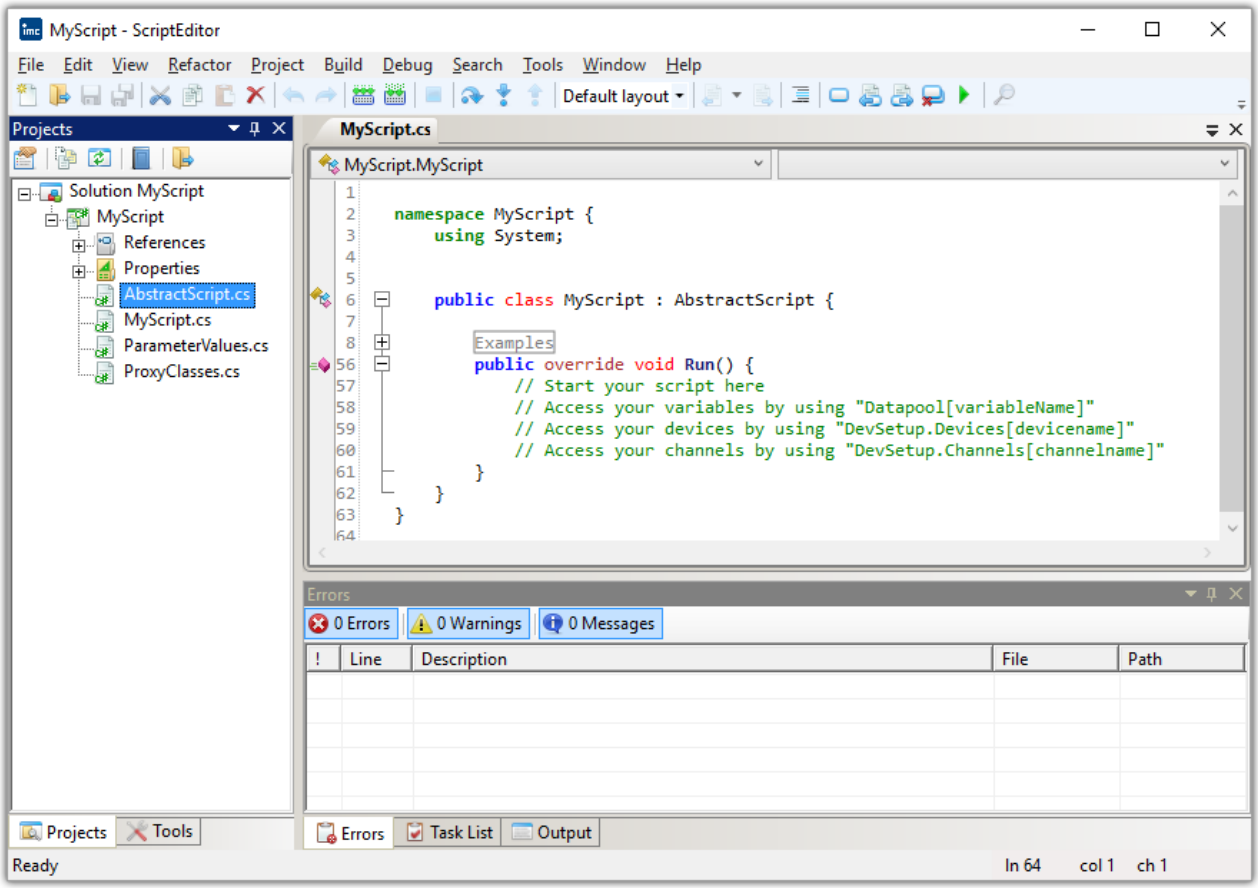


Dialog: Script properties

For the other script types 1314, this dialog and the editor template looks different. See also the corresponding sections: Panel-script 1358, Context-script 1365, Type-Library-script 1369, Event-script 1372.

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*.

14.1.1.1 Script type

The following script types are available. The script type should be selected which accords with the intended application.

Script Type	Description
Script 1356	• A Script is called and run by means of a *command*.
Panel Script 1358	• 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 1365	• 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 1369	• 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 1372	• 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 1407	• 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 1358, Context-script 1365, Type-Library-script 1369, Event-script 1372 or rather Third Party Device Script 1407

14.1.1.2 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* 1316.
Type	The selected *script type* 1314.
Connected Type-Library-Script	Selection from a list of created *Type-Library-Scripts* 1369. 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 1354.
Additional Proxy Classes	Selection of *imc FAMOS functions* Math-Library and Kits. These classes can be regenerated, see Regenerate proxies 1319
Activity Scope	*for Context scripts only:* Scope (Experiment, Project, Application), in which the script is active.

Note: Name assignment

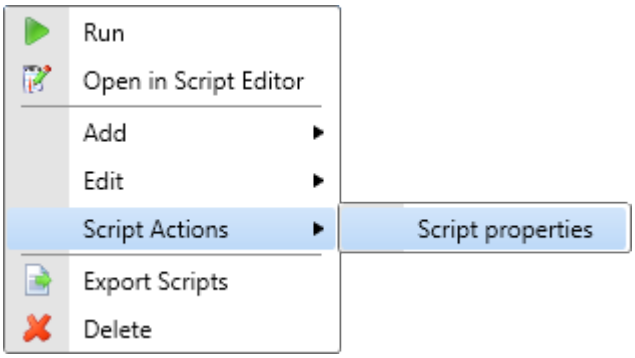
- Note that the same script name may not be used multiple times, even when using different Script Storage Scopes.
- Example:
  - In the Experiment *Exp_001*, a script "*Test*" is created with the Storage Scope *Experiment*.
  - In another experiment, *Exp_002*, a script "*Test*" is created with the Storage 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).

14.1.1.3 Storage scope

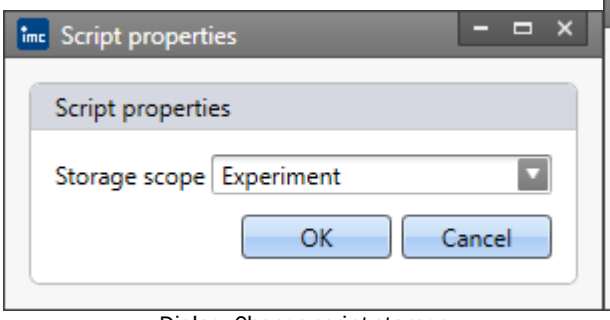
The option Storage Scope (*Save With / Script storage)* specifies the script's location in memory.
In order to use scripts in other experiments belonging to the same project, the script is created with the setting *Save with* > *Project.*

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 *applications-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 script is saved in the applications folder and only be listed in the main experiment.

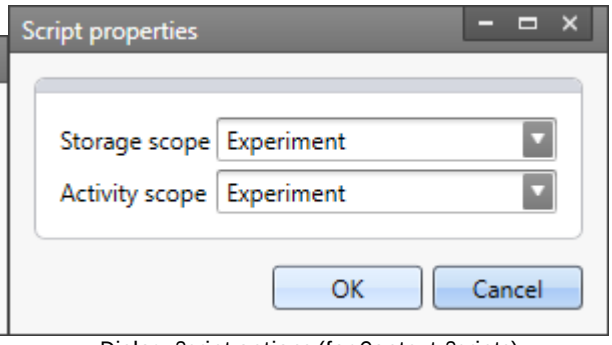
This option can be changed in the context menu *Script actions* > *Script Properties* > New Storage Scope.



Script actions: Script properties



Dialog: Change script storage

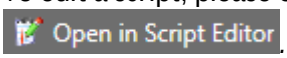


Dialog: Script options (for Context-Scripts)

Important: Save Project and Experiment

Please save your project and experiment after changing the *Storage Scope*, otherwise changes might be lost.

14.1.2 Edit scripts

To edit a script, please open the tool window Scripts and click on *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 transfered, if the corresponding *storage scope* (Experiment, Project) is saved.

14.1.3 Run scripts

To run a script, the following mechanism are available depending on the script type:

Script type	Mechanism

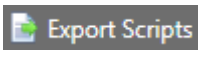
Script 1356	• Command in theSequencer
	• Command on an event.
	• Tool window scripts
	  ▪ Menu
	  ▪ Context menu

Panel script 1358	• Browsing to a page (`Run` method) • Leaving the page (`Stop` method)
Context script 1365	• Loading/opening the corresponding context (Run method) • Leaving/closing the context (`Dispose` method)
Type library script 1369	• Cannot be executed. • Is used in other scripts.
Event script 1372	• Command on an event.
Third Party Device Script 1407	• Cannot be executed • Refresh of the Third Pary Device after closing the editor.

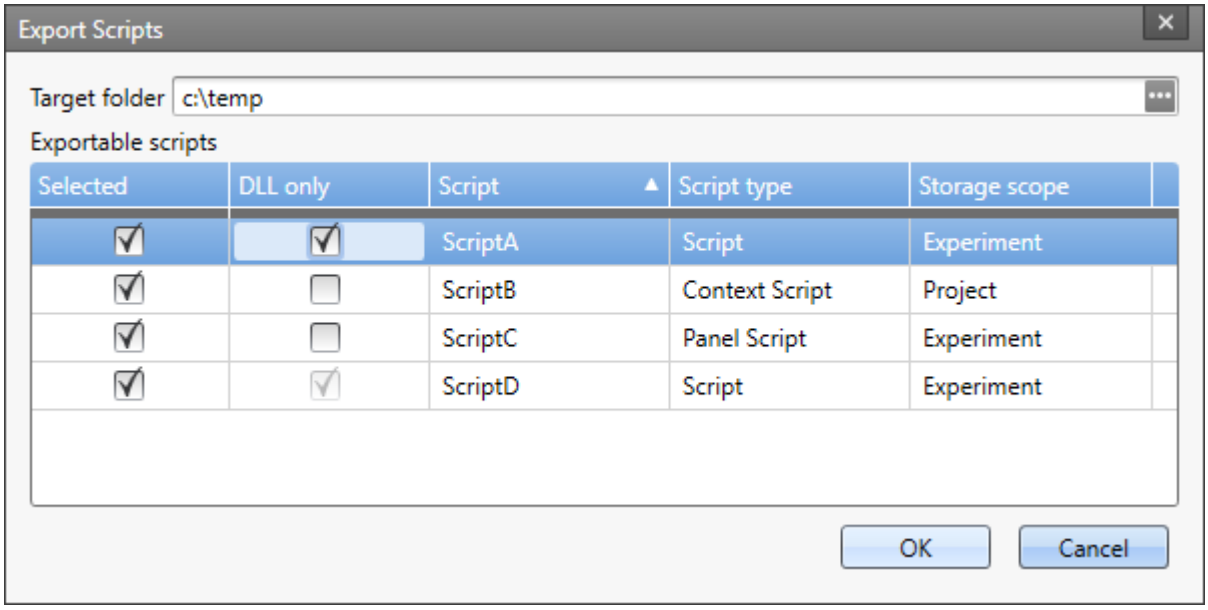
After running the script using Run, you can display the Build-Log by selecting *Edit* > *Show Build log* in the context menu.

14.1.4 Export and Import

Export

To export scripts, click in the Scripts tool window on *Export scripts* 

. Subsequently, a dialog showing all scripts opens.

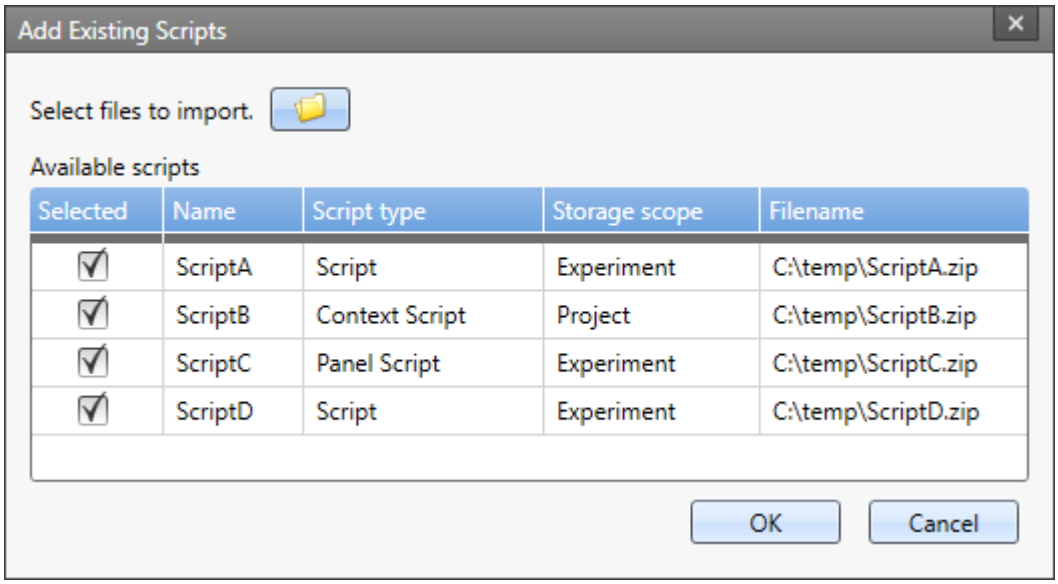


Dialog: Export scripts

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.

Import

Click in the tool window Scripts on *Add existing scripts* Add Existing 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. The 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.

Hint

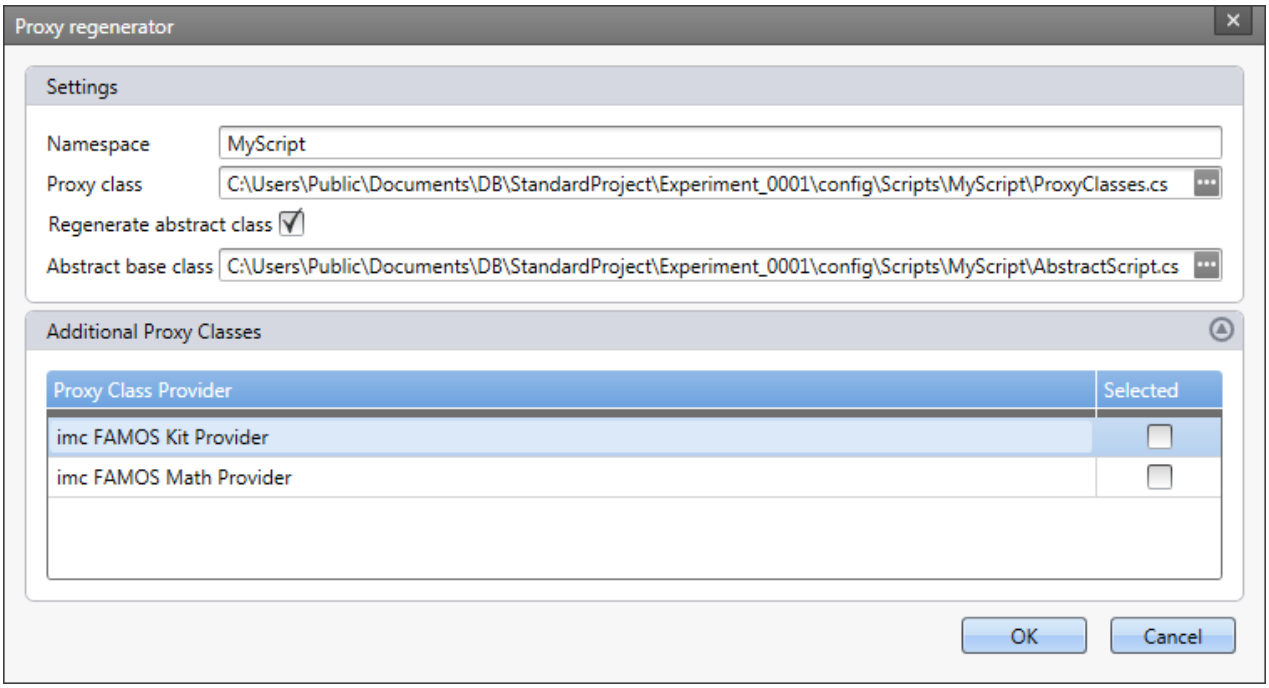
If Resources are used in the C#-Project, please get sure, that they the property "Build action" is set to "Resource" or "Embedded Resource" and/or "Copy to output directory" is set to true.
With these settings the Resources folder is considered when using DLL-Export.

14.1.5 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. Also, if you wish to use imc FAMOS functions and haven't selected the provider when creating the script. Select the script in the tool window Scripts and click in the context menu on *Edit > Regenerate proxies.*

The following dialog opens.

If no new imc FAMOS functions should be added, keep the current settings and click on *OK* button. Otherwise choose Kit and/or Math and then click on *OK*.



Dialog: Proxy regenerator

Attention

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 1403.

14.1.6 Debugging

To debug a script, the following steps must be performed after compiling.

Step 1 : Add breakpoint

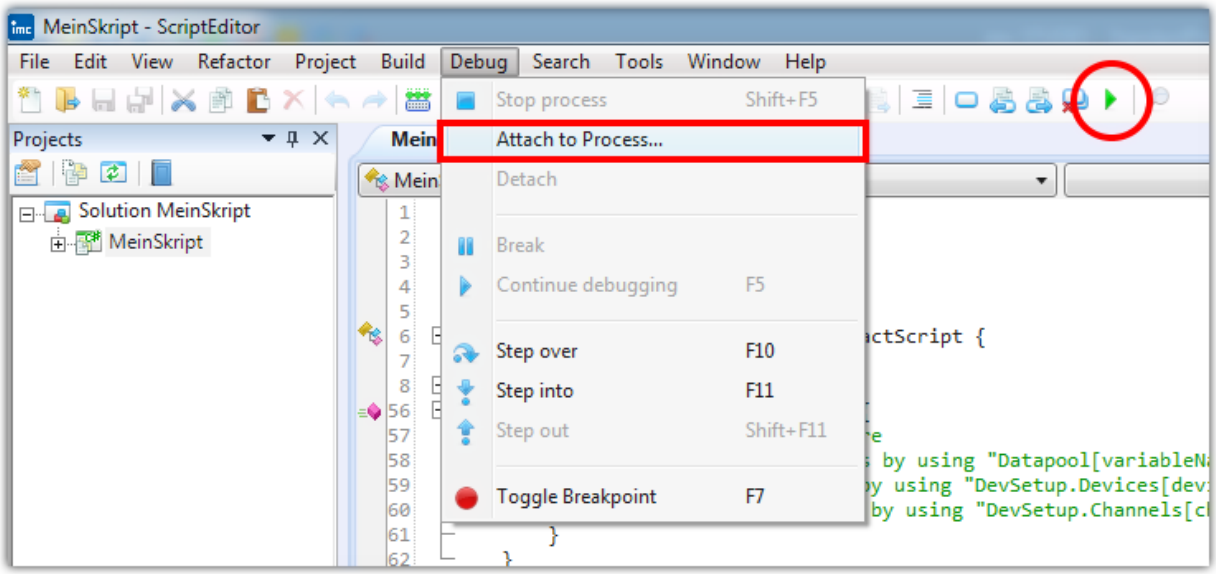
Press <F7> or use the menu item *Debug* > *Toggle Breakpoint* to add a breakpoint. The program counter will stop in this line, when reaching this statement.

Editor: set Breakpoint

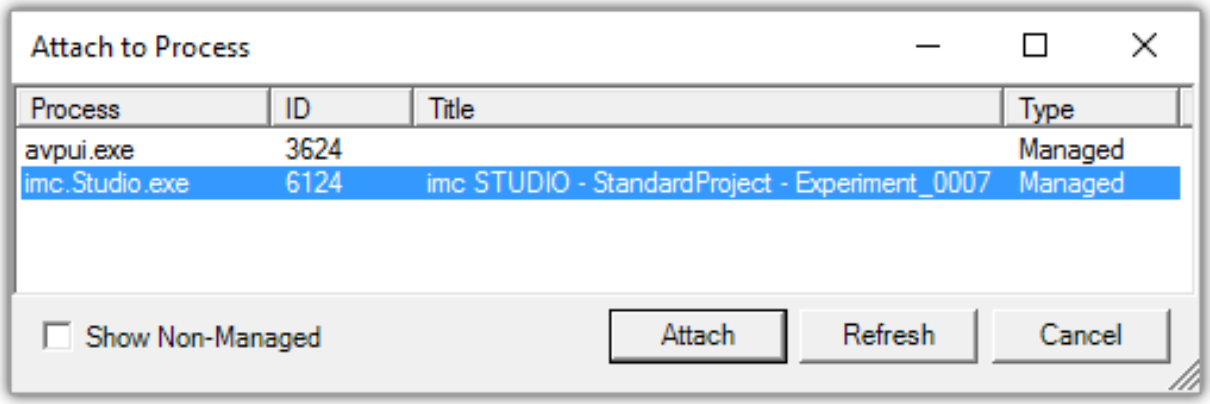
Step 2 : Attach to imc STUDIO

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.

14.2 Script (general)

14.2.1 Device functions

Notes

- For setting the parameters, class *EClassID* is used. Setting values of the parameter are available in the *ParameterValues*-Class. See also Device Parameters 1374.
- 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.V2.DevSetup;  // DeviceAdapters
using imc.Studio.Interfaces.DevSetup; // EDeviceCoreType_V2
```

The parameters are used as follows:

```
// EClassID_Parameter as string or from EClassID-Class
//var myPara = myChannel.Parameters["eMyParameter"];
var myPara = myDevice.Parameters[EClassID.eDeviceName]; // Example
```

```
// set parameter value
myPara.SetValue("SetValue");
// get parameter value
string sParameter = myPara.GetValue().ToString();
// get printout of the parameter value
string sParameterText = myPara.Text;
// select list of parameter values, if available
string [] sParaTextArray = myPara.TextList;
```

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.V2.DevSetup; // add Library
```

```
// IP
DevSetup.SearchDevice("192.168.0.2", DeviceAdapters.imcDevcies2x);
```

```
 // DNS
DevSetup.SearchDevice("imcDev__09123456", DeviceAdapters.imcDevcies2x);
```

```
// The return value of type "EnumItem" is the according device.
EnumItem myDevice = 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 strSN = "123456";
EnumItem myDevice = DevSetup.Devices.GetItemBySerialNumber(strSN);
```

```
// getting the device identifier
string myDevName = myDevice.Parameters[EClassID.eDeviceName].GetValue().ToString();
```

```
// getting the device name
string myDevNickName = myDevice.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(myDeviceNickName,true, EDeviceCoreType_V2.Measurement);
// selecting device via name
DevSetup.SelectDevice(myDeviceName,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
myDevice = DevSetup.Devices[myName];
// via serial number
myDevice = DevSetup.Devices.GetItemBySerialNumber(strSN);
```

Deselect a device

```
using imc.Studio.Interfaces.DevSetup;
DevSetup.DeselectDevice("DeviceNickName", true, EDeviceCoreType_V2.Measurement);
```

Remove device from List

```
DevSetup.RemoveDevice("DeviceNickName", true);
```

Device control actions

To connect to the device or start and stop the device, the following functions have to be executed.

```
var myStatus = myDevice.Parameters[EClassID.eDeviceControlAction];
// for action values see ParameterValues.eDeviceControlAction
// connect to device
myStatus.SetValue(ParameterValues.eDeviceControlAction.Connect);
myStatus.ExecuteAction();
// start measurement
myStatus.SetValue(ParameterValues.eDeviceControlAction.Start);
myStatus.ExecuteAction();
// stop measurement
myStatus.SetValue(ParameterValues.eDeviceControlAction.Stop);
myStatus.ExecuteAction();
// disconnect from device
myStatus.SetValue(ParameterValues.eDeviceControlAction.Disconnect);
myStatus.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 bSelected = myDevice.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)myDevice.Parameters[EClassID.eMeasurementStatus].GetValue();
string stateText = myDevice.Parameters[EClassID.eMeasurementStatus].Text;
```

Manufacturer

```
string manufacturer = myDevice.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

```
IEnumerator<EnumItem> myList = DevSetup.Devices.GetEnumerator();
myList.Reset();
myList.MoveNext(); // get next element in the list
EnumItem myDevice = myList.Current;
if (myDevice != null)
   string myDevName = myDevice.Parameters[EClassID.eDeviceName].GetValue().ToString();
}else{
   // do something, e.g.
    myDevName = String.Empty;
}
```

```
// Alternative
using System.Linq;
// ...
EnumItem myDevice = DevSetup.Devices.First(); // first known device in the list
```

Storage

Device specific options - Device

```
// Storage interval ; 0 = End of measurement
myDevice.Parameters[EClassID.eUMIntervalTime].SetValue(60d); // 60 seconds
// Storage interval count
myDevice.Parameters[EClassID.eUMIntervalCount].SetValue(128); // all = 0
// Storage locaction: 0: Removable medium, 1: Hard disk (HD); 2: Network drive
myDevice.Parameters[EClassID.eUMDrive].SetValue(0);
// Save trigger events in individual files (valid for ALL devices!)
myDevice.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.

```
myDevice.Parameters[EClassID.eUMDrive].SetValue(0); // Removable medium
myDevice.Parameters[EClassID.eDeleteFilesOnSelectedDisk].ExecuteAction();
```

Experiment options (applies to all devices) - PC

Note

These options can be set by setting the parameter of a device, but it will be applied for all devices in the experiment.

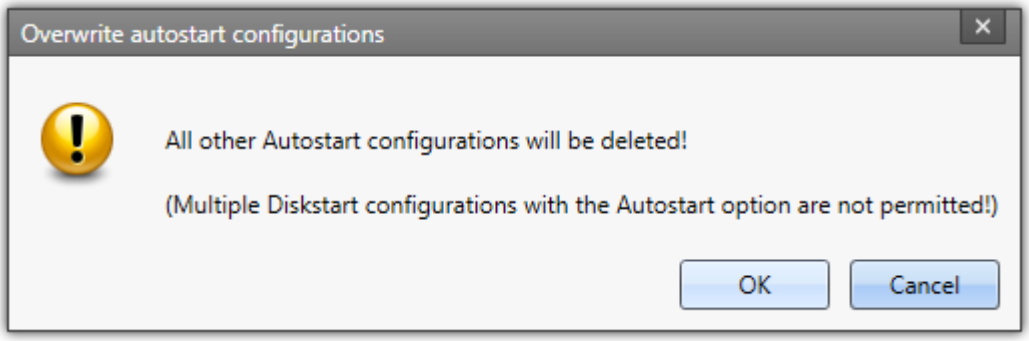
```
// Storage interval ; 0 = End of measurement
myDevice.Parameters[EClassID.ePCIntervalTime].SetValue(60d); // 60 seconds
// Storage interval count
myDevice.Parameters[EClassID.ePCIntervalCount].SetValue(128); // all = 0
// Save trigger events in individual files
myDevice.Parameters[EClassID.eExtraTriggerDirPCHD].SetValue(true);
```

Diskstart

Write Diskstart to destination

```
myDevice.Parameters[EClassID.eDiskStartSelfStartOption].SetValue(true); // Self-start
var myStatus = myDevice.Parameters[EClassID.eDiskStartAction];
// See ParameterValues.eDiskStartAction for setting values, e.g.
// 2503: Device HD; 2505: Device intern;
// saving to removable device
myStatus.SetValue(ParameterValues.eDiskStartAction.Device_PCMCIA);
myStatus.ExecuteAction();
// saving to PC-HD
myStatus.SetValue(ParameterValues.eDiskStartAction.Device_PC_HD);
myStatus.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):

```
using imc.Studio.Interfaces.V2.Gui;
UserRequest.SetDefaultResultValue("Dev2x_OverwriteSelfStart", EMessageBoxResult.OK);
```

Deletes Diskstart in every selected device

```
foreach(var cConfig in DevSetup.Items[EClassID.eConfig]) {
   var cDeleteAction = cConfig.Parameters[EClassID.eDiskStarDeleteConfig];
   if((cDeleteAction != null) && (cDeleteAction.Base != null))  {
       //cDeleteAction.SetValue(11001); // should be set by default
       cDeleteAction.ExecuteAction();
    }
}
```

14.2.2 Channel functions

Notes

- For setting the parameters, class *EClassID* is used. Setting values of the parameter are available in the *ParameterValues*-Class. See also Channel Parameters 1377.
- The method `ExecuteActions()` is only needed for device and channel actions. For parameters the method `SetValue()` is used.

The following line is needed for the following examples. The channel *Channel_001* is used. The channel name is case sensitive.

```
var myChannel = DevSetup.Channels["Channel_001"];
```

Setting parameters of bits (e.g. comment):

```
var myPort = DevSetup.Channels["DIN001"];
var myBit = myPort.ChildItems["DIN001_Bit01"];
myBit.Parameters[EClassID.eChannelComment].SetValue("MyComment");
```

To use the class *EClassID* the following library has to be added:

```
using imc.Studio.Tools.DevSetupGraphGenerator;
```

General

The parameters are used as follows

```
// EClassID_Parameter as string or from EClassID-Class
//var myPara = myChannel.Parameters["eMyParameter"];
var myPara = myChannel.Parameters[EClassID.eSampleTime]; // Example
```

Set parameter value

```
myPara.SetValue("SetValue");
```

Get parameter value

```
string sParameter = myPara.GetValue().ToString();
```

Get printout of the parameter value

```
string sParameterText = myPara.Text;
```

Select list of parameter values, if available

```
string [] sParaTextArray = myPara.TextList;
```

Metadata columns can be edited via the identifier (here: *MyIdentifier*).

```
var myPara = myChannel.Parameters["MyIdentifier"];
// get or set a parameter value
myPara.SetValue("MySettingValue");
string sParameter = myPara.GetValue().ToString();
```

Channels

Channel name

```
var myDevParameter = DevSetup.Devices["DeviceName"].Parameters;
var myChnParameter = DevSetup.Channels["ChannelName"].Parameters;
// Change Channel name
myChnPara[EClassID.eChannelName].SetValue("Channel_001_new");
```

Balance actions

```
var paraBalanceAct = myChannel.Parameters[EClassID.eBalanceAction];
// set offset
paraBalanceAct.SetValue(ParameterValues.eBalanceAction.Offset);
paraBalanceAct.ExecuteAction();
// set bridge
paraBalanceAct.SetValue(ParameterValues.eBalanceAction.Bridge);
paraBalanceAct.ExecuteAction();
// set tare
paraBalanceAct.SetValue(ParameterValues.eBalanceAction.Tare);
paraBalanceAct.ExecuteAction();
// set manufacture's calibration
paraBalanceAct.SetValue(ParameterValues.eBalanceAction.Reset);
paraBalanceAct.ExecuteAction();
```

Balance state

```
string balanceState = myChannel.Parameters[EClassID.eBalanceStatus].Text;
```

Cable compensation

```
var paraCalibAct = myChannel.Parameters[EClassID.eCalibrationAction];
paraCalibAct.SetValue(ParameterValues.eCalibrationAction.CableCompensation); // Cable
myChannel.Parameters[EClassID.eCalibrationAction].ExecuteAction();
double dR_cable = (double)myChannel.Parameters[EClassID.eCableResistor].GetValue();
```

Channel comment

```
myChannel.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 sPlugInName = "[01] IN01";
var query = from chn in DevSetup.Channels
  where (chn.Parameters[EClassID.ePlugInName] != null) &&
(chn.Parameters[EClassID.ePlugInName].Text.CompareTo(PlugInName) == 0)
  select chn;
// read channel name
string sName = string.Empty;
if ( null != query ){
   sName = query.First().Parameters[EClassID.eChannelName].GetValue().ToString();
}
// write channel name
string sName = "MyChannel";
if ( null != query ){
   query.First().Parameters[EClassID.eChannelName].SetValue(sName);
}
```

If there is a problem with the *from-where-select* 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.

```
// setting status of channel "Channel_1" to "active"
myChannel.Parameters[EClassID.eStatus].SetValue(ParameterValues.eStatus.Active);
// setting status to "passive"
myChannel.Parameters[EClassID.eStatus].SetValue(ParameterValues.eStatus.Passive);
// setting status to "write"
myChannel.Parameters[EClassID.eStatus].SetValue(ParameterValues.eStatus.Write);
// setting status to "read/write"
myChannel.Parameters[EClassID.eStatus].SetValue(ParameterValues.eStatus.Read_Write);
```

Connector

```
string myConName = myChannel.Parameters[EclassID.ePlugInName].GetValue().ToString();
```

Coupling

```
myChannel.Parameters[EClassID.eCoupling].SetValue(ParameterValues.eCoupling.AC);
myChannel.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;
myChannel.Parameters[EClassID.eDuration].SetValue(0); // undefined
myChannel.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 = myChannel.Parameters[EClassID.eFilterCharacteristic];
paraFilterChar.SetValue(ParameterValues.eFilterCharacteristic.Butterworth);

var paraFilterType = myChannel.Parameters[EClassID.eFilterType];
paraFilterType.SetValue(ParameterValues.eFilterType.Bandpass);

myChannel.Parameters[EClassID.eFiltCutoff1].SetValue(10); // 10 Hz
myChannel.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
myChannel.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"
// Alternative: get the value
var paraRange = DevSetup.Channels["Channel_001"].Parameters[EClassID.eRange];
object oRange = paraRange.GetValue();
CValueArray cRange = oRange as CValueArray;
string sRange = cRange[1].ToString(); // range is element with index 1
```

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 = myChannel.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 paraProc = myChannel.Parameters[EClassID.eProcessing];
paraProc.SetValue(ParameterValues.eProcessing.ArithMean);
myChannel.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
myChannel.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.

```
myChannel.Parameters[EClassID.eUserUnit].SetValue("Pa"); // Pascal
myChannel.Parameters[EClassID.eUserOffset].SetValue(1); // 1 Pa
myChannel.Parameters[EClassID.eUserScalingFactor].SetValue(5); // = 1/Sensitivity
```

Strain gauge

```
Coupling
myChannel.Parameters[EClassID.eCoupling].SetValue(ParameterValues.Coupling.HalfBridge;
Bridge resistor
myChannel.Parameters[EClassID.eBridgeResistor].SetValue(120);
Bridge mode
myChannel.Parameters[EClassID.eBridgeBridgeMode].SetValue(1);
// 1=>strain; 2=>stress
Bridge factor N
myChannel.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
myChannel.Parameters[EClassID.eBridgeFactor].SetValue(1);
Bridge unit
myChannel.Parameters[EClassID.eBridgeUnit].SetValue(1);
Transverse strain coefficient v
myChannel.Parameters[EClassID.eBridgeEps].SetValue(0.3);
Modolus of elasicity
myChannel.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 = myChannel.Parameters[EClassID.eSensorAction];
paraReadSensor.SetValue(ParameterValues.eSensorAction.ReadSensorFlash);
paraReadSensor.ExecuteAction();
```

```
// write
var paraReadSensor = myChannel.Parameters[EClassID.eSensorAction];
paraReadSensor.SetValue(ParameterValues.eSensorAction.WriteSensorFlash);
paraReadSensor.ExecuteAction();
```

```
// delete
var paraReadSensor = myChannel.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.

```
myChannel.Parameters[EClassID.ePoint1SetValue].SetValue("20");
myChannel.Parameters[EClassID.ePoint1BalanceAction].ExecuteAction();
myChannel.Parameters[EClassID.ePoint2SetValue].SetValue("50");
myChannel.Parameters[EClassID.ePoint2BalanceAction].ExecuteAction();
var paraCalibAct = myChannel.Parameters[EClassID.eCalibrationAction];
paraCalibAct.SetValue(ParameterValues.eCalibrationAction.TwoPoint); // Two Point
myChannel.Parameters[EClassID.eCalibrationAction].ExecuteAction();
```

Unit

Escaping of quotation marks (") by using backslash (\).

```
// setting User Unit to N/m
myChannel.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):

```
myChannel.Parameters[EClassID.eBalanceAtDeviceStart].SetValue(true);
```

Data transfer

PC

Transfer to PC

```
// activate
myChannel.Parameters[EClassID.eTransferToPC].SetValue(true);
// deactivate
myChannel.Parameters[EClassID.eTransferToPC].SetValue(false);
```

Display, calculations of measured data

```
// Available Events (Display) ; 0 = all; 1 = last
myChannel.Parameters[EClassID.eDisplayAvailableEvents].SetValue(0);
```

```
// Circular buffer time (Display)
myChannel.Parameters[EClassID.eDisplayRingTime].SetValue(60d);
```

Save data

```
// aktivieren
myChannel.Parameters[EClassID.ePCTransferToHD].SetValue(true);
// deaktivieren
myChannel.Parameters[EClassID.ePCTransferToHD].SetValue(false);
```

```
// Saved events (PC) ; 0 = all; 1 = last
myChannel.Parameters[EClassID.eAvailableEvents].SetValue(0);
```

```
// Circular buffer time (PC)
myChannel.Parameters[EClassID.ePCRingTime].SetValue(60d); // 60 Sekunden
```

Device

Save data

```
myChannel.Parameters[EClassID.eUMTransferToHD].SetValue(true);
```

RAM

Buffer time

```
myChannel.Parameters[EClassID.eBufferTime].SetValue(60d); // 60 seconds; auto = 0
```

Curve Window

Display Time / Circular Buffer Time (Curve Window)

```
myChannel.Parameters[EClassID.eDisplayRingTime].SetValue(60d);
```

Graph Color

Does not change during measurement.

```
// preset Color
int myCol = Color.Tomato.ToArgb();
// via RGB
int myCol = Color.FromArgb(0,255,255).ToArgb();
// set
myChannel.Parameters[EClassID.eCurveColor].SetValue(myCol);
```

y-Axis (Scale)

```
myChannel.Parameters[EClassID.eCurveYAxisOption].SetValue(2);
// Input-Range (1), user-defined (2), auto (3)
myChannel.Parameters[EClassID.eCurveYAxisMax].SetValue(20d);
myChannel.Parameters[EClassID.eCurveYAxisMin].SetValue(-20d);
```

Example

Saving all active analog channels to hard disk drive

```
using imc.Studio.Tools.DevSetupGraphGenerator;
```

```
// for all channels
foreach (var channel in DevSetup.Channels){

   var paraType = channel.Parameters[EClassID.eChannelType];

   // analog channels
   if ( (paraType != null) && ((int)paraType.GetValue() ==
ParameterValues.eChannelType.AnalogInput) ){
      var paraState = channel.Parameters[EClassID.eStatus];

      // active channels
      if ( (paraState != null) && ((int)paraState.GetValue() ==
ParameterValues.eStatus.Active) ){
         var paraSaveHD = channel.Parameters[EClassID.ePCTransferToHD];

         // set save to HD
         if (paraSaveHD != null){
            paraSaveHD.SetValue(true);
         } // paraSaveHD
      } // paraState
   } // paraType

} // foreach
```

14.2.3 Panel and Widget functions

Panel

Get the active Panel page

```
PanelPage myPage = Panel.GetActivePage();
```

Set Panel page with name "Page 1" as active

```
string strPageName = "Page 1";
Panel.SetActivePage(Panel.GetIndexOf(strPageName)); // index is needed
```

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 sFilename = @"c:\temp\myPage.dbv";
ind pageIndex = 0; // page index, after the page has to be imported
Panel.InsertPageAfter(pageIndex,sFilename, "Page 1", "Page 1");
string sFilename = @"c:\temp\meineSeite.dbv";
ind pageIndex = 0; // page index, before the page has to be imported
Panel.InsertPageBefore(pageIndex,sFilename, "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.

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 cButton = new Button();
cButton.Text = "MyText";
cButton.Bounds = new Rectangle(100, 50, 100, 50);
// cButton.Click += new EventHandler(eventFunction);
Panel["Page 1"].Controls.Add(cButton as Control);
```

Subsequently, the Windows-control can be copied to the script clipboard.

Widgets

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 Widgets.

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!";
```

Setting zones of lists

```
using System.Linq;
// Automotive, Designer, Industrial
Panel["Page 1"]["List1"].Zones.ElementAt(0).Text.SetText("en", "Entry_1");
Panel["Page 1"]["List1"].Zones.ElementAt(0).SetValue = 0.1;
Panel["Page 1"]["List1"].Zones.ElementAt(0).UpperLevel = 0.1; // sets the upper level
Panel["Page 1"]["List1"].Zones.ElementAt(0).LowerLevel = 0.1; // sets the lower level
```

Note

It is not possible to add Zones via Scripting. If a select list is needed, a ComboBox from the `Windows.Forms`-Class can be used. Entries can be added to this control.

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)

Note

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
```

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 myTable = Panel["Page 1"]["Table1"];
var myCell = myTable.Columns.ElementAt(0).Cells.ElementAt(0); // 1st col, 1st element
myCell.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.

14.2.4 Datapool functions

Note

- Be sure, that the variables are available in the Data Browser or are created in the script.
- Check if variable is available in the script.
- For other Script types (Panel, Event, ...) the Datapool is called with `Core.Datapool`.

```
var myVar = Datapool["myVariable"];
if (myVar != null){
   //... do something
}else{
   // ... throw error
}
```

The following libraries are needed:

```
using imc.Common.Interfaces.DataPool;
```

Variables

Create variables

```
// create numeric and text variable
bool bSuccessD = Datapool.CreateVariable<double>("MyNumericVariable", 0d);
bool bSuccessS = Datapool.CreateVariable<string>("MyTextVariable", "MyText");
```

In the Data Browser the variables are created in the category *Scripting* with default scope *temporary*.

Create variables with scope

```
// create variables with scope
Datapool.CreateVariable<double>("MyTemporaryNumber", 0d, EVariableScope.Volatile);
Datapool.CreateVariable<double>("MyExperimentNumber", 0d, EVariableScope.Experiment);
Datapool.CreateVariable<double>("MyProjectNumber", 0d, EVariableScope.Project);
Datapool.CreateVariable<double>("MyGlobalNumber", 0d, EVariableScope.Persistent);
```

Create variables in a group

```
// Sub-group / folder
bool bSuccessD = Datapool.CreateVariable<double>("Group.Name", 0d);
```

Delete variables

```
// delete variable
bool bSuccessDel = Datapool.DeleteVariable("MyNumericVariable");
```

Set content of variables

```
// sets a text to the variable "MyTextVar" in the Data Browser
string strText = "Hello World!";
Datapool["MyTextVar"].SetContent<string>(myText);
```

```
// sets a number to the variable "MyNumVar" in the Data Browser
double dNumber = 42.21;
Datapool["MyNumVar"].SetContent<double>(myNumber);
```

Get content of variables

```
// gets the content of the text variable "VariableName_str" in the Data Browser
string strVariable = Datapool["VariableName_str"].GetContent<string>();
```

```
// gets the content of the numerical variable "VariableName_num" in the Data Browser
double dVariable = Datapool["VariableName_num"].GetContent<double>();
```

Properties of variables

```
// Variables Properties can be read via "VariableProperties"
DataPoolVariable myVar = Datapool["VariableName"];
string myUnit = myVar.VariableProperties.Unit; // gets the unit
myVar.VariableProperties.Scrope = EVariableScope.Experiment; // set scope
```

Changing the category

Show variable in the "*user-defined variables*" category:

```
DataPoolVariable myVar = Datapool["VariablenName"];
myVar.VariableProperties.Set<string>(EVariableProperty.eCategory, "__USER__");
```

Channels

Get Channel

```
// Channel from Current Measurement
string channelName = "Channel_001";
// from specific time stamp
string channelName = "Channel_001@2013-01-01 09-00-00 (1)";
// from measurement number 1
string channelName = "Channel_001@Measurement#1";
// from last measurement
string channelName = "Channel_001@Measurement#LAST"
var dp_chn = Datapool[channelName];
IDmChannel myIChannel = null;
if (dp_chn != null){
   myIChannel = dp_chn.GetContent<IDmChannel>();
}
```

The measurements must be loaded in the Data Browser.

Get channel data

```
double[] data = null;
if (myIChannel != null){
   if (myIChannel.Length > 0){
      data = chn.Data; // access to channel data. if available
      // Do something
   }
}
```

Get channel data (event triggered)

The circular buffer time (Transfer to PC) has to be set to *unlimited*.

```
double[] data1 = myIChannel.Events[0].Data;
double[] data2 = myIChannel.Events[1].Data;
```

Set Channel

```
DmChannel ChannelOut = new DmChannel(); // create new Channel
ChannelOut = DmChannel.CloneOf(myIChannel); // creates a copy of "myIChannel"
ChannelOut.Name = "Channel_output"; // rename
// ... calculations ...
DatapoolVariable dpOut = Datapool[ChannelOut.Name];
// if variable does not exist, create
if (null == dpOut){
   Datapool.CreateVariable<DmChannel>(ChannelOut.Name,ChannelOut);
}else{
   Datapool[ChannelOut.Name].SetContent<DmChannel>(ChannelOut);
}
ChannelOut.Dispose(); // DmChannel has to be disposed
```

Transfer Channel to imc FAMOS

```
myIChannel.TransferToFamos();
```

Save Channel to file

```
myIChannel.Save("C:\\Users\\myName\\Desktop\\myChannel.raw");
```

Load Channel from file

```
myDmChannel.Load("C:\\Users\\myName\\Desktop\\myChannel.raw");
```

Show channel in a separate curve window

```
myChannel.Show();
```

Create XY channel

```
using imc.Common.Components.DataManager;
using imc.Common.Components.DataManager.Interfaces;
double[] xData = {2, 3, 5, 7, 11, 13, 17, 19}; // 8 elements
double[] yData = {4, 9, 25, 49, 121, 169, 289, 361};

DmChannel channel = new DmChannel("xyChannel", IDmChannel.DmChannelType.XY);
channel.Length = xData.Length;
channel.Cmp.X.SetData<double>(xData);
channel.Cmp.Y.SetData<double>(yData);

if (Datapool[channel.Name] == null){
   Datapool.CreateVariable<DmChannel>(channel.Name, channel);
}else{
   Datapool[channel.Name].SetContent<DmChannel>(channel);
}

channel.Dispose();
```

Note

When using DmChannel objects, it is important that the object is *disposed*.

```
DmChannel myChannel = new DmChannel();
// ...
myChannel.Dispose();
```

Report channels

Report Channels

```
IDmChannel myReport = Datapool["ReportChannel"].GetContent<IDmChannel>();
var Chn_Report = Datapool["ReportChannel"];
```

Write to an existing Report Channel

Current Time is used as time stamp.

```
Chn_Report.SetContent<string>("This is a Report-Channel.");
```

Get the last entry

```
string lastEntry = Chn_Report.GetContent<string>();
```

Get Report Channel entries

```
int ind = myReport.Tsa.FindFirst();
ind = myReport.Tsa.FindNext(ind); // do as long as you got yours
byte [] data = myReport.Tsa.GetData(ind);
// convert data to string needed, e.g.
// string t = Encoding.UTF8.GetString(data);
```

Save Report Channel to file

```
myReport.Tsa.SaveAscii("C:\\Users\\myName\\myReportText.txt",(byte)';',0,1,0,1);
```

DataTable

By the means of a DataTable Tables can be set easily.

user-defined variable of type "DataTable"

```
using System.Data;
var dpVar = Datapool["Table"]; // variable name
if (dpVar != null){
   DataTable dt = dpVar.GetContent<DataTable>();
}
```

Add column

```
string columnName = "Name";
if (!dt.Columns.Contains(columnName)){
   dt.Columns.Add(columnName);
}
```

Add row

```
dt.Rows.Add("MyName");
```

For further information, please search the internet for the type System.Data.DataTable.

Example

Copy active Panel page name to an user-defined variable.

Gets the active Panel page and sets the name to the Datapool text variable "myText".

```
PanelPage myPage = Panel.GetActivePage();
string pagename = myPage.Name.ToString();
DatapoolVariable myText = Datapool["myText"];
if (null == myText){
   Datapool.CreateVariable<string>("myText",pagename);
}else{
 myText.SetContent<string>(pagename);
}
```

Create an user-defined channel and set channel data

```
DmChannel myChannel = new DmChannel();
myChannel.Name = "MyChannel";

myChannel.Data = new double[]{1, 1, 2, 3, 5, 8, 13};

var dpVar = Datapool[myChannel.Name];

if (dpVar == null){
   Datapool.CreateVariable<DmChannel>(myChannel.Name, myChannel, EVariableScope.Experiment);
}else{
   dpVar.SetContent<DmChannel>(myChannel);
}
```

14.2.4.1 Complex Variables

Computer-Variable

To access to the Computer variable, the reference *imc.Studio.PlugIns.CoreExtensions* (GAC) must be added to the script project (*context menu of the project > Add Reference*)

The complex variable can be used in Scripting with the following code:

```
using imc.Studio.PlugIns.CoreExtensions.LocalSystemInformation;
var dpComputer = Datapool["Computer"];
var oComputer = dpComputer.GetContent<object>();
CLocalSystemInformation infoComp = oComputer as CLocalSystemInformation;
```

SystemClock

To access to the system clock, the following .NET reference must be added to the script project:

C:\Program Files (x86)\imc\imc STUDIO 5.0 R3\imc.Studio.Components.UnmanagedObjects.dll

The complex variable can be used in Scripting with the following code:

```
using imc.Studio.Components.UnmanagedObjects;
var dpClock = Datapool["SystemClock"];
var oClock = dpClock.GetContent<object>();
imc.Studio.Components.Clock.CSystemClockImpl infoClock = oClock as
imc.Studio.Components.Clock.CSystemClockImpl;
```

Device system variable

To access to the device system variable, the following .NET reference must be added to the script project:

C:\Program Files (x86)\imc\imc STUDIO 5.0 R3\imc.Studio.Components.UnmanagedObjects.dll

The complex variable can be used in Scripting with the following code:

```
using imc.Studio.Components.UnmanagedObjects;
var dpDevice = Datapool["System_T_123456_CS_7008_1"]; // check for name
var oDevice = dpDevice.GetContent<object>();
CDeviceGlobals_V1 infoDevice = oDevice as CDeviceGlobals_V1;
```

14.2.5 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;
```

Controlling the curve window

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 = y.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();
}
```

Delete all coordinate systems

Please add the reference *imc.Comman.Controls.ImcPanel* to the script project.

`if null`-checks are omitted here.

Afterward only one coordinate system is remaining which cannot be deleted.

```
using imc.Common.Controls.ImcCurves.Interfaces;
using imc.Common.Controls.ImcPanel.Widgets;
var imcCurve = w.Base as CimcCurveWidget;
var imcCurveCtrl = imcCurve.CurveCtrl;
for (int i = imcCurveCtrl.CoSystems.Count - 1; i>= 1; i--){
   imcCurveCtrl.CoSystems.Delete(i, CwIncludeConstants.IncludeRelated);
}
```

For further information also see the tutorial Curve Window and channel selection 1398.

Control with imc FAMOS Kits

Libraries

The class *Curve_Window* contains the imc FAMOS functions for the curve window. Using this class, it is possible to control curve windows in imc STUDIO.

```
using imc.Famos.Kits;
```

Load curve window configuration

```
Curve_Window.General.CwLoadCCV("Curve1",strFilename);
```

Add channel to an existing curve window

The corresponding Panel page must be visible and the Design mode must not be activated.

```
Curve_Window.General.CwSelectWindow("Curve1"); // Widget name
DmChannel channel = new DmChannel();
channel.Name = "MyChannel";
Curve_Window.Elements.CwNewChannel("append last axis", channel);
channel.Dispose();
```

Changing the curve window measurement cursor's position

```
// Run method of Panel script
w.CursorMove += CursorMoveEvent;
public void CursorMoveEvent(object sender,EventArgs e){
   Curve_Window.General.CwSelectWindow("Curve1");
   double xleft = Curve_Window.Properties.CwDisplayGet("measure.x.left");
   Datapool["xleft"].SetContent<double>(xleft);
   // ...
}
// Stop method
w.CursorMove -= CursorMoveEvent;
```

Scroll mode

```
Curve_Window.General.CwSelectWindow("Curve1");
Curve_Window.Properties.CwDisplaySet("scroll",0);
```

Several actions

```
Curve_Window.General.CwAction("");  // see editor's tool-tip for more information
```

Set y-Axis color to color of first line

```
Curve_Window.General.CwSelectWindow("Curve1");
Curve_Window.Elements.CwSelectByIndex("y-axis", 1);
Curve_Window.Properties.CwAxisSet("font.color", -3);
// -1 automatic; RGB-Color (as int) for fixed
```

14.2.6 Logbook

Library

```
using imc.Common.Interfaces.Logbook;
// for ElogbookEntryCategory, ILogbookEntryBase, ...
```

Set Logbook entry

Hint

For other Script types (Panel, Event, ...) the Logbook is called with `Core.Logbook`.

```
try{
   // ...
}catch (Exeption myException){
   Logbook.LogEntry("myScript.cs",myException);
}
```

```
// create Exception
Exception myException = new Exception("My Error!");
// write logbook entry
Logbook.LogEntry("myScript.cs", myException);
```

```
string errorSender = "myScript.cs";
string errorText = "My Error";
int errorCode = 0;
ELogbookEntryCategory myCategory = ELogbookEntryCategory.Error;
Logbook.LogEntry(errorSender,errorText,errorCode,myCategory);
```

```
/*
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 dt = new DateTime(2014,1,1);
Logbook.Export(@"c:\temp\logbook.xml", dt);
```

Show Logbook window

```
Core.Windows.ShowWindow("Logbook")
```

Clear view of the logbook / delete logbook entries in the toolwindow

```
Logbook.ClearViews();
```

Example: 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 _start = 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(_start);

         // save entries to text-file
         StreamWriter myFile = new StreamWriter(filename, true);
         for (int k = 0; k < entries.Length; ++k){
            myFile.WriteLine(entries[k].ToString());
         }

         myFile.Close();
         myFile.Dispose();
      } // of void run
   } // of class
} // of namespace
```

Example: List of messages with sender "Sequencer"

In this example all messages of the logbook entries with sender "Sequencer" are listed.

In the Script options 1390 the *advanced code completion* must be enabled.

```
using System.Linq;
using System.Windows.Forms;
using imc.Common.Interfaces.Logbook;
```

```
string str = string.Empty;

foreach (var entry in Logbook){
    API_LogbookEntry_V1 e = entry.GetAPI<API_LogbookEntry_V1>();

    if (string.Compare(e.Sender,"Sequencer") == 0){
        str += e.Message + "\n";
    }
}

MessageBox.Show(str);
```

```
// Alternative
var query = from entry in Logbook
      where entry.GetAPI<API_LogbookEntry_V1>().Sender == "Sequencer"
      select entry.GetAPI<API_LogbookEntry_V1>();

API_LogbookEntry_V1[] entries =  query.ToArray();
```

```
// Alternative 2
var entries = Logbook
            .Select(logEntry => logEntry.GetAPI<API_LogbookEntry_V1>())
            .Where(logEntry => logEntry.Sender == "Sequencer")
            .ToArray();
```

For getting all errors, the code is similar.

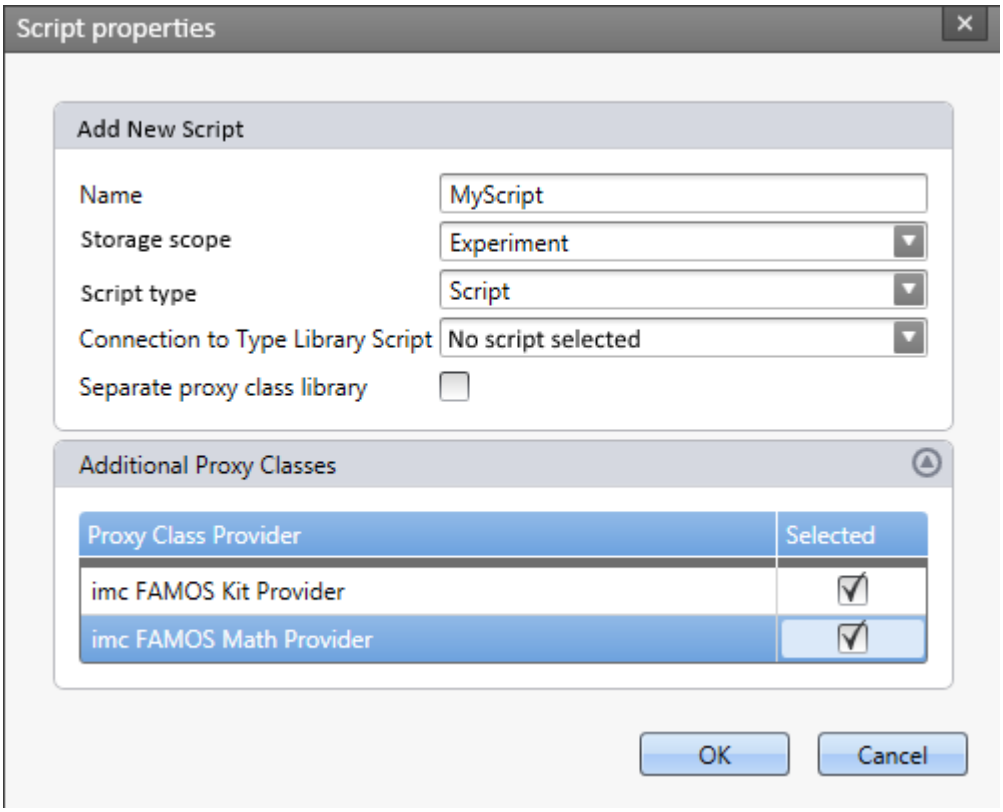
14.2.7 imc FAMOS functions

Important

- For using imc FAMOS functions a imc FAMOS Runtime license must be activated.
- For using imc FAMOS Kits functions the corresponding license must be activated.
- To use imc FAMOS functions, select *imc FAMOS Math Provider* and/or *imc FAMOS Kit Provider* as the Additional Proxy-Classes when creating a script. These can also be updated subsequently using the context menu *Edit > Regenerate proxies*.
- If only a imc FAMOS 64-bit installation is present, only the CurveWindow- and the Time-Stamp-Ascii-Kit are available. To access to the other Kits (SpectralAnalysis, OrderTracking and ClassCounting) there must be a 32-bit installation of imc FAMOS,

Note

- If a script is linked with a Type Library Script and imc FAMOS functions intended to be used, the imc FAMOS Proxy Classes must be activated in the Type Library Script.
- If the imc FAMOS Math Provider is later uploaded into the Type Library Script using *Edit > Regenerate proxies*, then a reference to *imc FAMOS Math* Provider must be added manually in any script which is linked with the Type Library Script.



Sequencer: Script options

If the following line is used, the prefix *imc.Common.Components.Famos.Math / imc.Famos.Kits* is not needed in order to use the imc FAMOS functions:

```
using imc.Common.Components.Famos.Math;
using imc.Famos.Kits;
```

Initialization of channels

The following channels must exist.

```
// get the channel named 'Channel_001'
IDmChannel channelA = Datapool["Channel_001"].GetContent<IDmChannel>();
IDmChannel channelCMP = Datapool["Channel_compare"].GetContent<IDmChannel>();
// init Result channel
DmChannel channelB = null;
```

Calculation of channels

Addition and Multiplication: +, *, Add, Mul

```
// Add 42 to the channel
Famos.Add(channelA, 0,null, 42, out channelB);
// multiply channel by 42
Famos.Mult(channelA, 0, null, 42, out channelB);
```

Integral and Derivation: int, diff

```
// Integration
imc.Common.Components.Famos.Math.FnBasic.Int(channelA, out channelB);
// Derivation
FnBasic.Diff(channalA, out channelB); // if library integrated by means of "using"
```

Filter: FiltLP, smo

```
// Butterworth Filter, 10th Order, Cut-off frequency 50 Hz
FnFilter.LowPass(channelA, 0, 0, 10, 50, out channelB);
// smoothing over 10ms
FnFilter.Smooth(channelA,0.01,out channelB);
```

Edit: cut, cutIndex

```
// Cut data (use time in [s]) ; [0.1s,0.2s]
FnEdit.CopyPartX(channelA, 0.1, 0.2, out channelB);
// Cut data (use index) ; [5,10]
FnEdit.CopyPartI(channelB, 5, 10, out channelB);
```

Approximation: Regr, PTast

```
// linear Regression
FnMisc.FitLin(channelA, out channelB);
// Resample
FnMisc.ResampleRef(channelA, channelCMP, out channelB);
```

Statistics: mean, StDev, SearchLevel, xMax, All0

```
// Mean value
FnStat.Mean(channelA, out channelB);
// Standard deviation
FnStat.StDev(channelA, out channelB);
// find XY-Points in the interval [-5;5] with gain  [-1000,1000]
FnStat.SearchLevel(channelA, 3, -5, 5, 3, -1000, 1000, 0, out channelB);
// find relative maxima over threshold 10
FnStat.XPostofMaxima(channelA, 10, out channelB);
// Zero crossings
FnStat.XPosOfZeros(channelA, out channelB);
```

The return value of a imc FAMOS-function is of the type *bool* and indicates whether the function was run successfully.

Additional functions are available in

- Famos.*,
- FnBasic.*,
- FnCompare.*,
- FnEdit.*,
- FnFilter.*,
- FnMisc.*,
- FnSpec.*,
- FnStat.*,

as well as

- Class_Counting_Kit.* (32 bit),
- Curve_Window.* 1344,
- Order_Tracking_Kit.* (32 bit),
- Spectrum_Analysis_Kit.* (32 bit),
- Timestamp_Ascii_Kit.*.

Return to Datapool

The channel name must be set accordingly, since it has been altered by the imc FAMOS function.

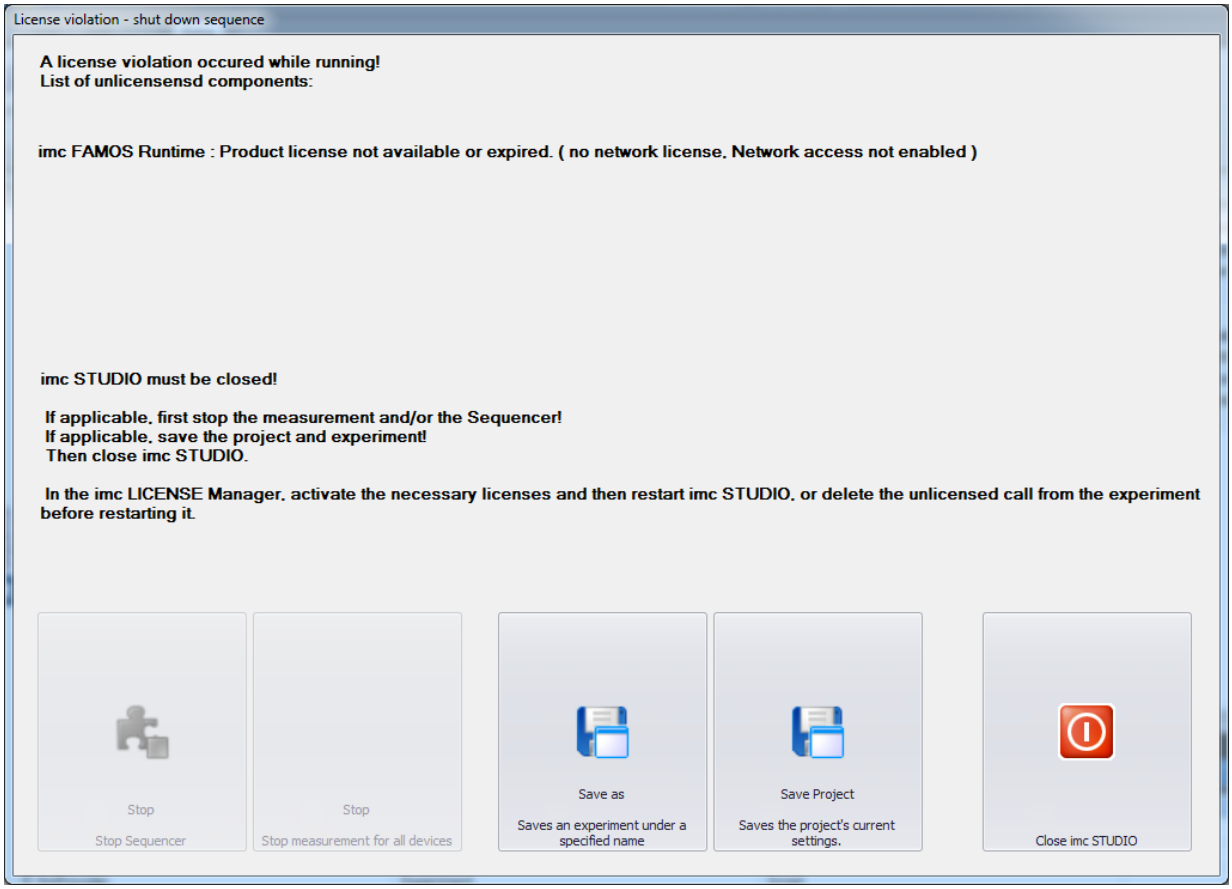
The result is written to the existing user-defined variable "UDvar_Return" of type "Channel".

```
if (null != channelB) {
  channelB.Name = "UDvar_Return"; // Name must be set to Datapool-variable
  Datapool[channelB.Name].SetContent<DmChannel>(channelB);
  channelB.Dispose();
}
```

License violation

If no imc FAMOS Runtime license is activated, the following dialog will appear when executing scripts using imc FAMOS functions.

In this dialog, you have the possibility to stop the Sequencer and the measurement, as well as saving the experiment and the project..



Dialog: License violation

14.2.8 Inline FAMOS functions

Below, the Inline FAMOS functions in Scripting are described. For this purpose, the appropriate license for imc STUDIO Inline FAMOS is required.

Hint

Please use the return value Result 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.

Librariy

```
using imc.Studio.Interfaces.V2.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 taskP;
taskP = 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

```
Result r = InlineFamos.DeleteTask(task.Name); // delete
Result r = task.Delete();
```

Apply Task

```
Result r = task.Apply(); // check result for "IsSuccessful"
```

Task Name

```
string name = task.Name;
Result r = 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 src = task.Setup.SourceCode;
```

```
// Set source code: Path
Result r = null;
using (StreamReader stream = new StreamReader(@"c:\temp\MyTask.ifa.xml")){
   string content = stream.ReadToEnd();
   r = 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");
Result resultSource = task.Setup.SetSourceCode("Channel_out = 2 * Channel_001");
Result resultApply = task.Apply();
```

14.2.9 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
```

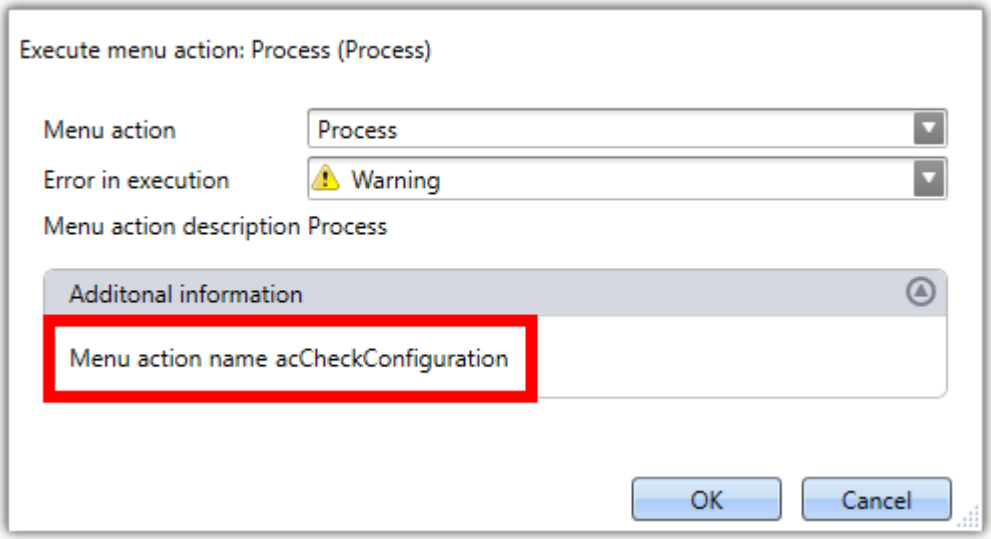
React to a menu action

With a Context script you can react to the manually execution of a menu action.

```
using imc.Studio.Interfaces.V2.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
• Prepare	acCheckConfiguration
• Download	acDeviceDownload
• Suspend data storage	acSuspendDiscRecording
• Resume data storage	acResumeDiscRecording
• Reconfigure	acImcDev30ForceDownload

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

14.2.10 Additional functions

14.2.10.1 Default dialog response

In Scripting the options for the default dialog response can be set as follows:

```
using imc.Studio.Interfaces.V2.Gui;
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"
}
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
// read setting for Overwrite autostart configuration
EMessageBoxResult result = UserRequest.GetDefaultResultValue(OverwriteSelfStart);
// None = Show Dialog
```

Futher information, see chapter Options: Default dialog response 60.

14.2.10.2 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 myText = "My Object";
ScriptClipboard.Add("MyKey",myText);
```

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.V2.DevSetup;
EnumItem dev = DevSetup.Devices.GetItemBySerialNumber("123456");
IEnumItemBase myDeviceInstance = dev.Base.GetAPI<IEnumItemBase>();
ScriptClipboard.Add("MyDeviceInstance",myDeviceInstance);
```

Getting a device object from the Clipboard (Script B)

```
using imc.Studio.Interfaces.V2.DevSetup;
object o = ScriptClipboard["MyDeviceInstance"] ;
IEnumItemBase item = o as IEnumItemBase;
EnumItem myDevice = EnumItem.GetInstance(item);
```

Hint

- 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.
- For details, see Using the script clipboard 1402

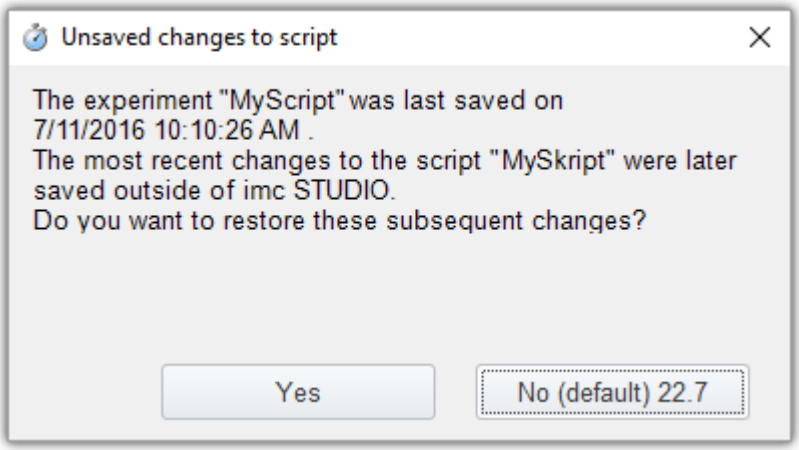
14.2.10.3 Restore script changes

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 1390 a default answer can be set for this dialog.

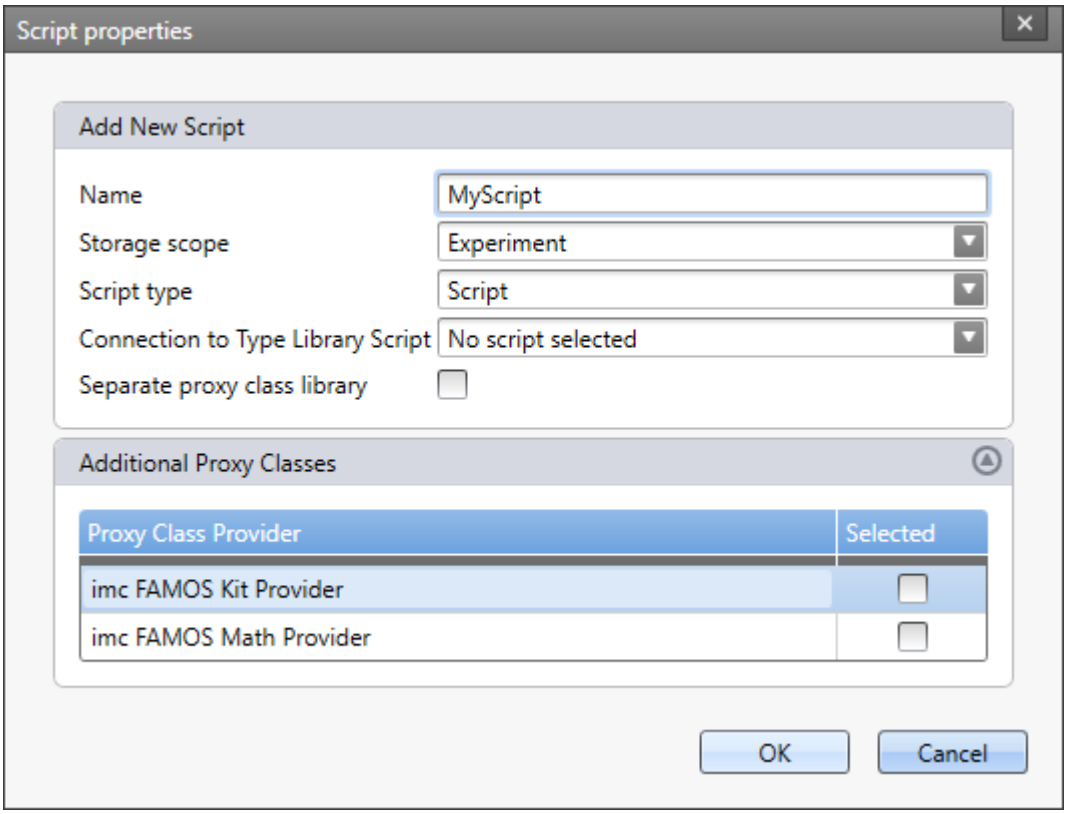
14.3 Script types

14.3.1 Script

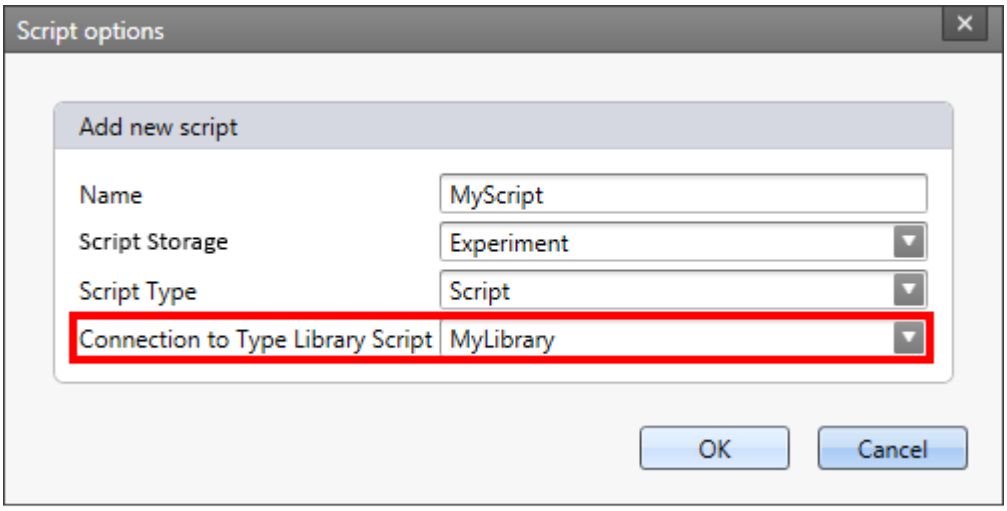
Scripts are run in response to:

- 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*.

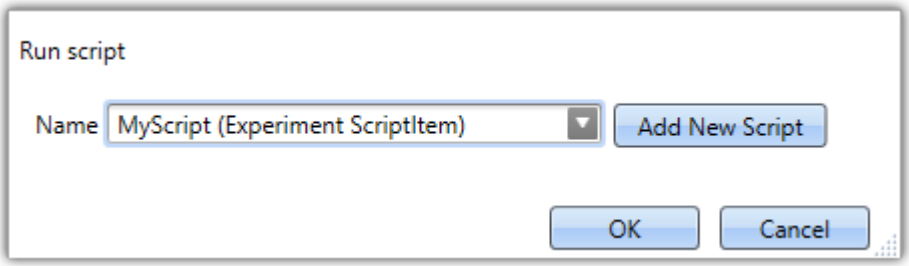
The Storage Scope option can be set to Experiment, Project, Application or Sequencer in which the script is available.



Dialog: create Script without Type Library Script



Dialog: create Script with Type Library 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 "Datapool[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.

Note

Helpful examples are given in the region *Examples*.

Note

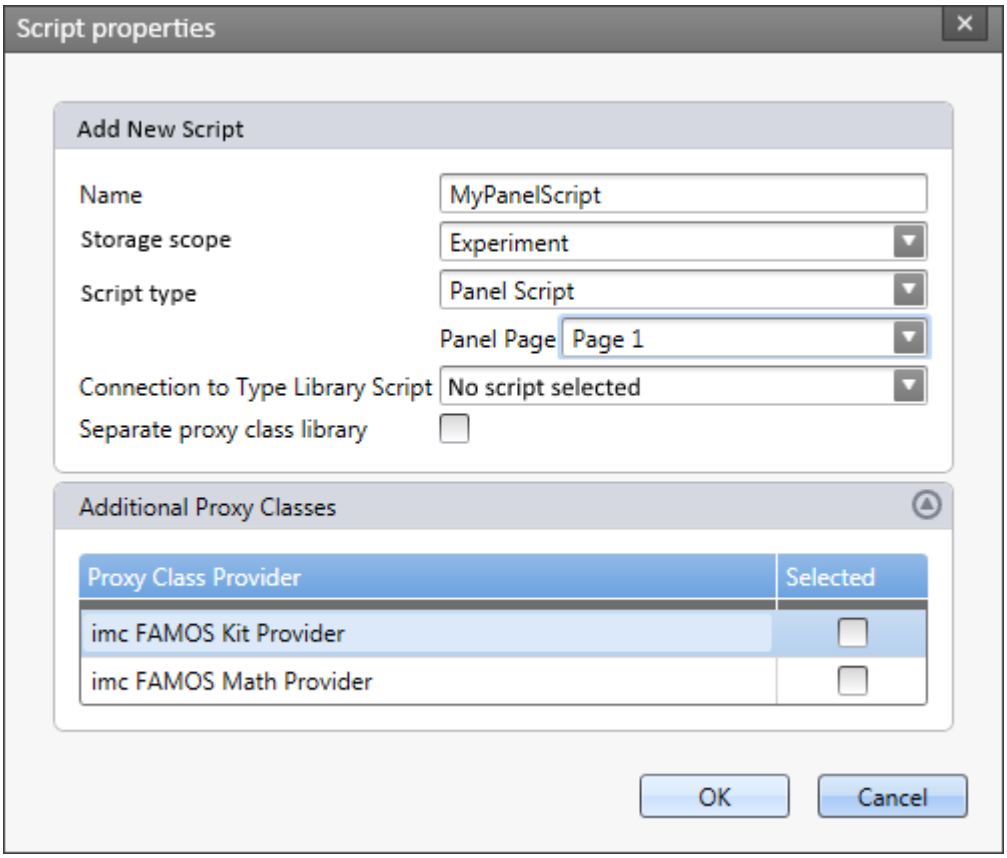
- If a script is linked with a Type Library Script and imc FAMOS functions intended to be used, the imc FAMOS Proxy Classes must be activated in the Type Library Script.
- If the imc FAMOS Math Provider is later uploaded into the Type Library Script using *Edit* > *Regenerate proxies*, then a reference to *imc FAMOS M*ath Provider must be added manually in any script which is linked with the Type Library Script.

14.3.2 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.



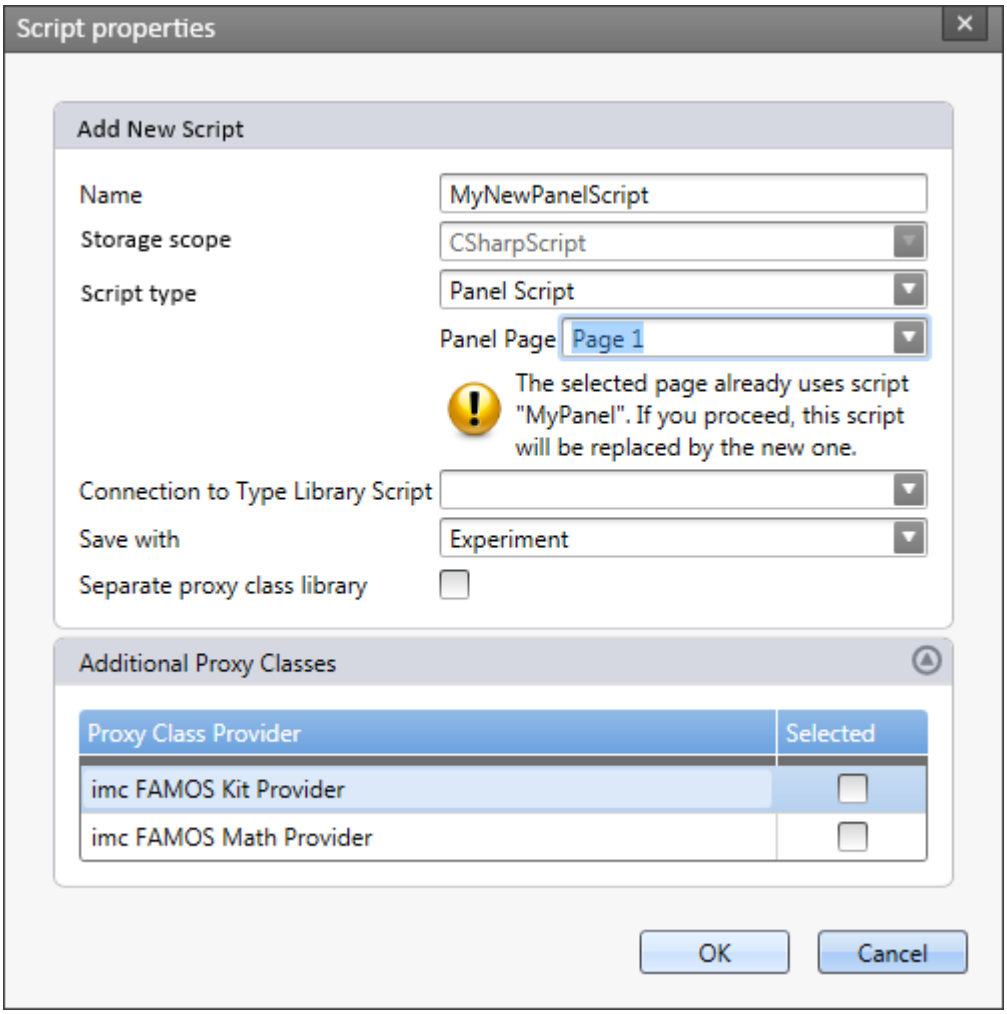
Dialog: Creating a Panel-script

Note

- If a script is linked with a Type Library Script and imc FAMOS functions intended to be used, the imc FAMOS Proxy Classes must be activated in the Type Library Script.
- If the imc FAMOS Math Provider is later uploaded into the Type Library Script using *Edit > Regenerate proxies*, then a reference to *imc FAMOS M*ath Provider must be added manually in any script which is linked with the Type Library Script.

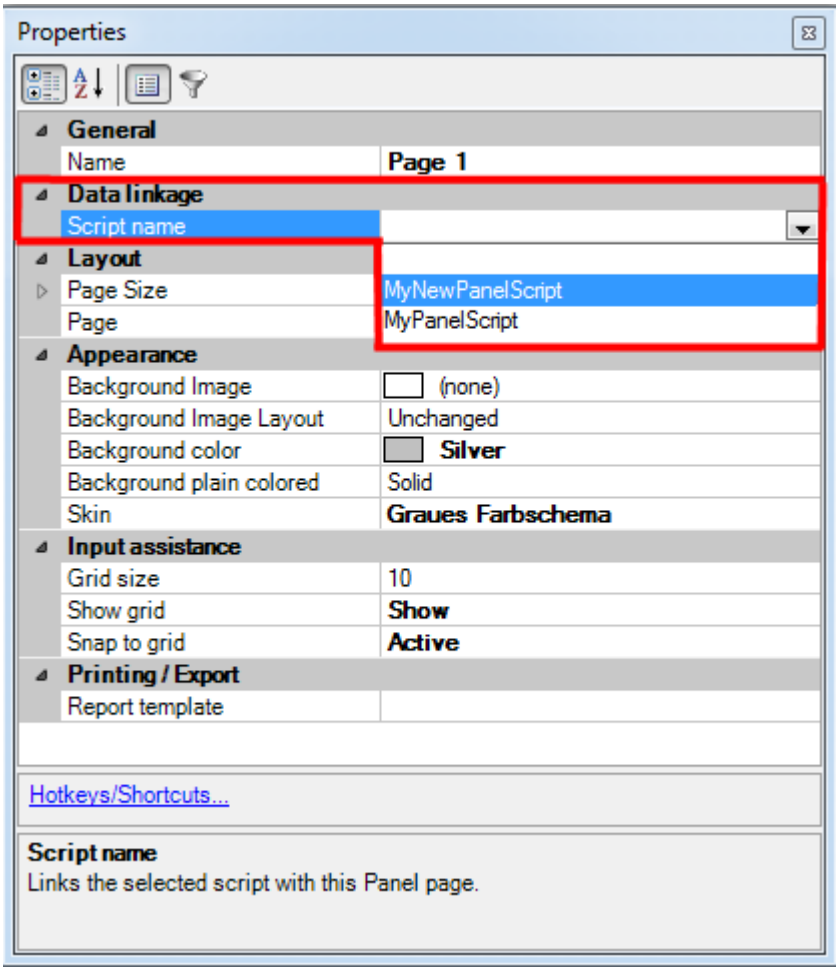
Warning

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.



Dialog: Panel Script - choose already linked page

In the properties of a Panel page the linked script can be changed.



Page properties: Script name

General

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 must be opened
`Stop()`	when changing to an other Panel page	the corresponding Panel must be opened
	when closing the experiment	the corresponding Panel must be opened
	when entering the Design mode	the corresponding Panel 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!";
```

Example:
Switching visibility of a standard meter by a rocker switch

Via Data Browser-Class

```
bool isHooked = false;

public override void Run() {
   if (!isHooked){
      isHooked = true; // State -> hooked
      Rocker_switch1.SwitchOn += new Action<object, EventArgs>(turnon);
      Rocker_switch1.SwitchOff += new Action<object, EventArgs>(turnoff);
   }
}

public void turnon(object myObject,EventArgs args){
   if (isHooked){
      Standard_meter1.Visible = false;
   }
}

public void turnoff(object myObject,EventArgs args){
   if (isHooked){
      Standard_meter1.Visible = true;
   }
}

public override void Stop() {
   if (isHooked){
      isHooked = false; // State -> unhooked
      Rocker_switch1.SwitchOn -= turnon;
      Rocker_switch1.SwitchOff -= turnoff;
   }
}
```

Via Panel-Class

```
bool isHooked = false;

public override void Run() {
   if (!isHooked){
      isHooked = true; // State -> hooked
      Widget w = Panel["Page 1"]["Rocker_switch1"];
      w.SwitchOn += new Action<object, EventArgs>(turnon);
      w.SwitchOff += new Action<object, EventArgs>(turnoff);
   }
}

public void turnon(object mySender, EventArgs ev){
   if (isHooked){
      Panel["Page 1"]["Standard_meter1"].Visible = false;
   }
}

public void turnoff(object mySender, EventArgs ev){
   if (isHooked){
      Panel["Page 1"]["Standard_meter1"].Visible = true;
   }
}

public override void Stop() {
   if (isHooked){
      isHooked = false; // State -> unhooked
      Panel["Page 1"]["Rocker_switch1"].SwitchOn -= turnon;
      Panel["Page 1"]["Rocker_switch1"].SwitchOff -= turnoff;
   }
}
```

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

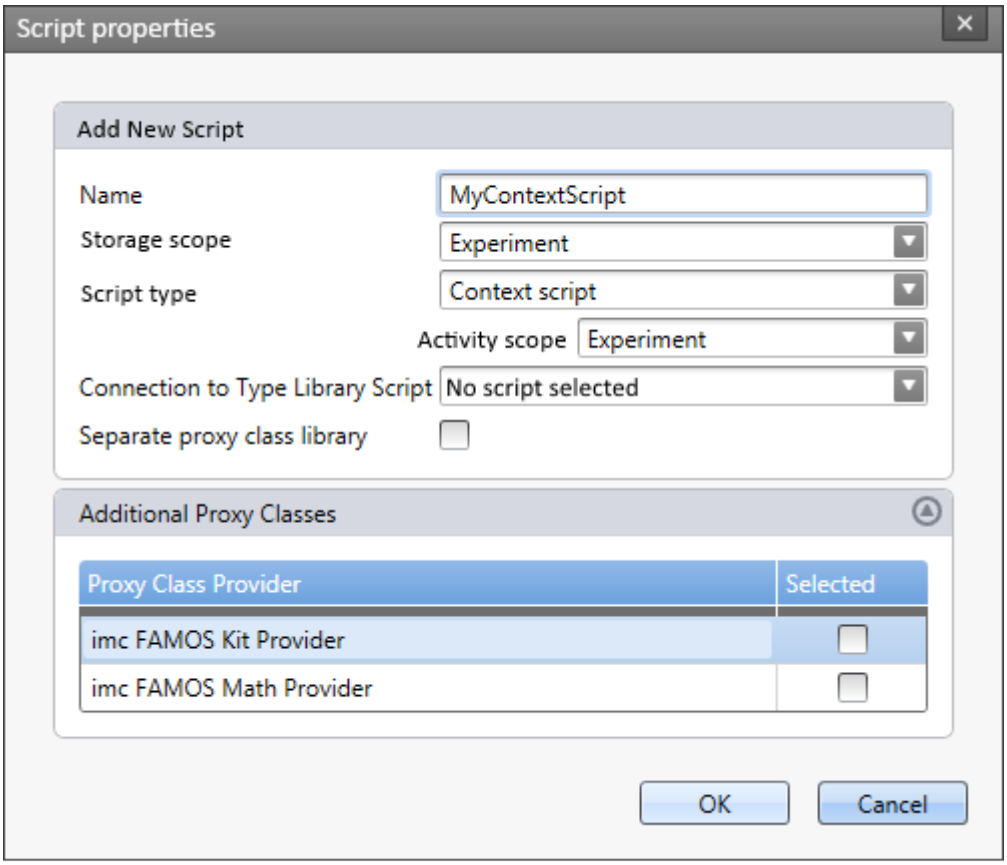
When deleting Widgets, it is necessary to ensure that the script no longer accesses the widget.

Datapool and Logbook

Use class `Core` to access to the datapool or logbook.

```
double myVariable = Core.Datapool["VariableName"].GetContent<double>();
Core.Logbook.LogEntry("MySender", "MyMessage", 0, ELogbookEntryCategory.Warning);
```

14.3.3 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.
      }
   }
}
```

Note

- If a script is linked with a Type Library Script and imc FAMOS functions intended to be used, the imc FAMOS Proxy Classes must be activated in the Type Library Script.
- If the imc FAMOS Math Provider is later uploaded into the Type Library Script using *Edit > Regenerate proxies*, then a reference to *imc FAMOS Math* Provider must be added manually in any script which is linked with the Type Library Script.

React to events

With a Context script you can react to device events. The event names are listed on the Sequencer page.

```
using imc.Studio.Interfaces.V2.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;
}
```

React to a menu action

With a Context script you can react to the manually execution of a menu action.

```
using imc.Studio.Interfaces.V2.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 1351.

Example

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: ParameterChanged
// class: ParameterChanged
private bool isHooked = false;
private bool InValueChanged = false;

public override void Run() {
   if (!isHooked) {
      isHooked = true;
      DevSetup.ValueChanged += new Action<object, DevSetupValueChangedArgs>(ValChanged);
   }     // of if
} // of void Run
```

```
private void ValChanged(object arg1,DevSetupValueChangedArgs arg2){
   // try{
   // if(!InValueChanged){
   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 if InValueChanged
//    } finally {
//       InValueChanged = false;
//    } // of try catch
} // of void
```

```
// Class name "ParameterChanged" -> "ParameterChangedDispose"
protected override void ParameterChangedDispose() {
   // Release all your resources and event hooks here.
   if (isHooked) {
      DevSetup.ValueChanged -= ValChanged;
      isHooked = false;
   } // of if
} // of void Dispose
```

If channel state is set to active, activate saving to PC

```
using imc.Studio.Tools.DevSetupGraphGenerator;
```

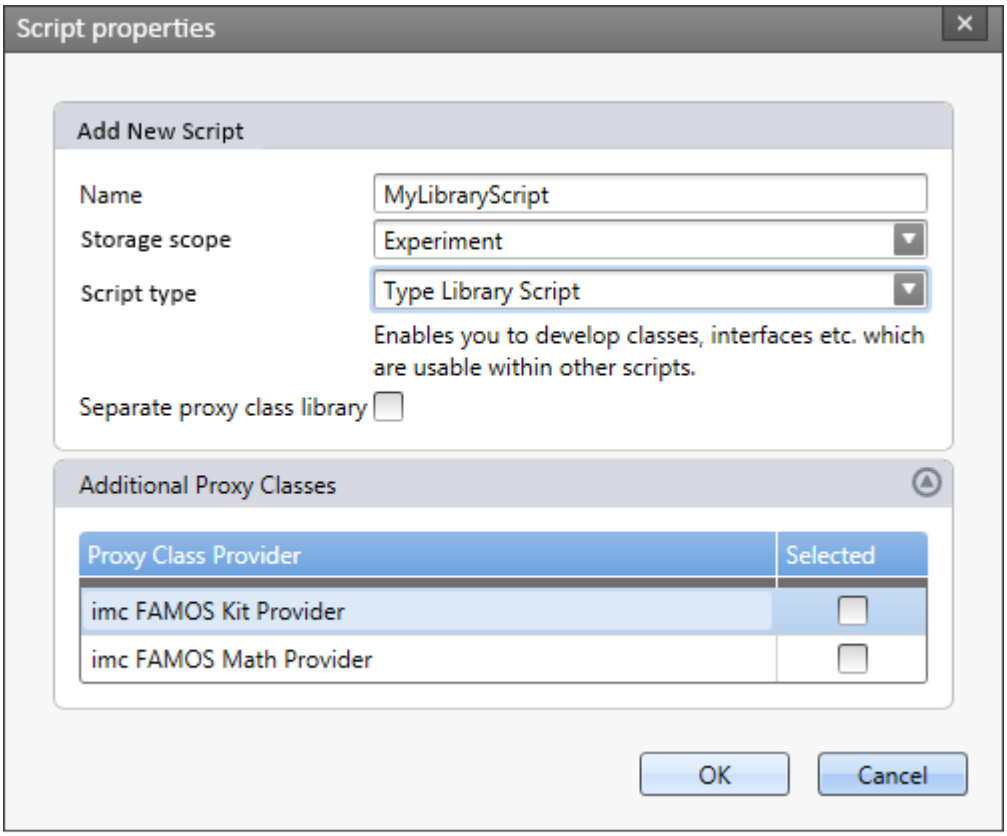
```
// namespace: ParameterChanged
// class: ParameterChanged
bool isHooked = false;

public override void Run() {

   if (!isHooked) {
      isHooked = true;
      DevSetup.ValueChanged += new Action<object,
DevSetupValueChangedArgs>(DevSetup_ValueChanged);
   }
}
```

```
void DevSetup_ValueChanged(object arg1, DevSetupValueChangedArgs arg2)
{
   if (isHooked) {
      foreach (var item in arg2.Changes.Keys){
         // when state is not passive, save channel
         if (item.ClassID == EClassID.eStatus.ToString()) {
            var channel = DevSetup.Channels.GetItemByHWID(item.HWID);
            if (null != channel && null != channel.Base) {
               SaveChannelActive(channel);
            }
         } // if eStatus

         // when selecting device, all non-passive channels are saved
         if (item.ClassID == "Dev_SelForMeasurement"){
            foreach (var channel in DevSetup.Channels) {
               SaveChannelActive(channel);
            }
         } // if Dev_Sel
      } // foreach keys
   } // hooked
}
```

```
protected void SaveChannelActive(EnumItem channel){
   var paraSave = channel.Parameters[EClassID.ePCTransferToHD];
   var paraState = channel.Parameters[EClassID.eStatus];

   if (null != paraSave) {
      if (null != paraState) {
         if ((int)paraState.GetValue() != ParameterValues.eStatus.Passive) {
            paraSave.SetValue(1); // set saving on, if active
         } else {
            paraSave.SetValue(0); // set saving off, if passive
         }
      } // if state
   } // if save
} // SaveChannel
```

```
protected override void SaveActiveChannelDispose() {

   if (isHooked) {
      isHooked = false;
      DevSetup.ValueChanged -= DevSetup_ValueChanged;
   }
}
```

14.3.4 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.

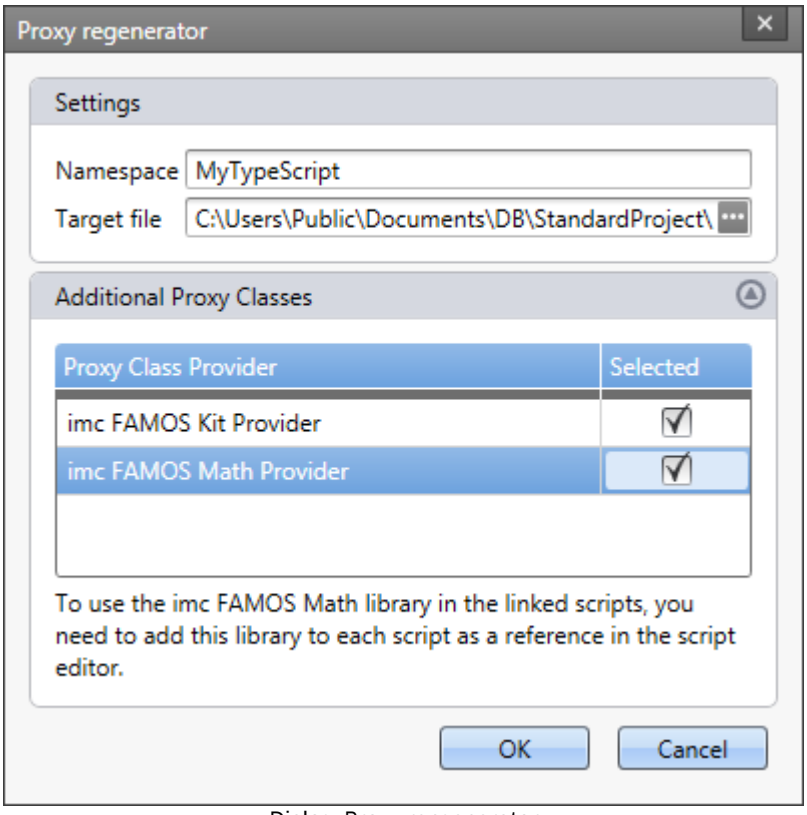
```
using System;
namespace MyLibrary {
   // Insert your TypeLibrary code here
}
```

Important

Please make sure when using Type Library Scripts, 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*.

Note

- If a script is linked with a Type Library Script and imc FAMOS functions intended to be used, the imc FAMOS Proxy Classes must be activated in the Type Library Script.
- If the imc FAMOS Math Provider is later uploaded into the Type Library Script using *Edit > Regenerate proxies*, then a reference to *imc FAMOS M*ath Provider must be added manually in any script which is linked with the Type Library Script.



Dialog: Proxy rengenerator

Hint

It is recommended to set the Assembly Version (see file *AssemblyInfo.cs*):

```
[assembly: AssemblyVersion("1.0.1.0")]
```

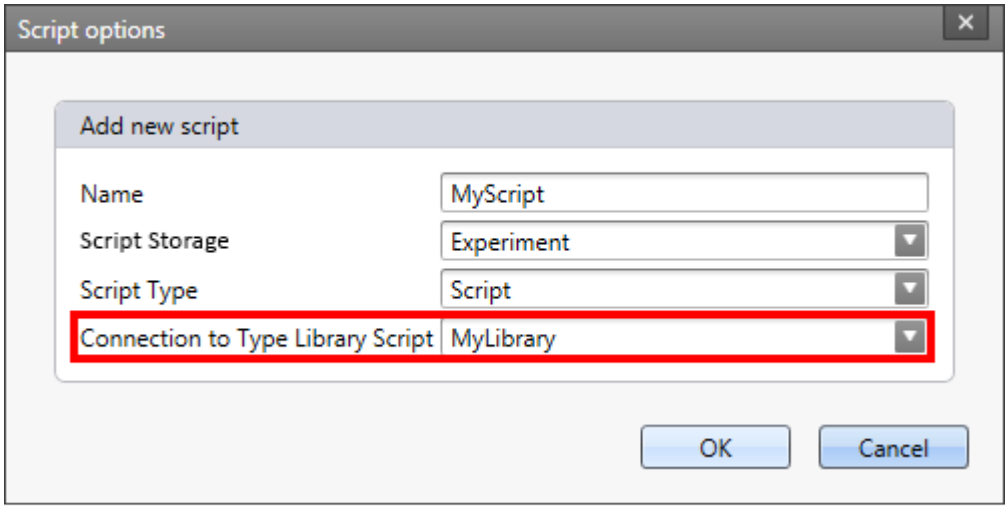
Example

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*.

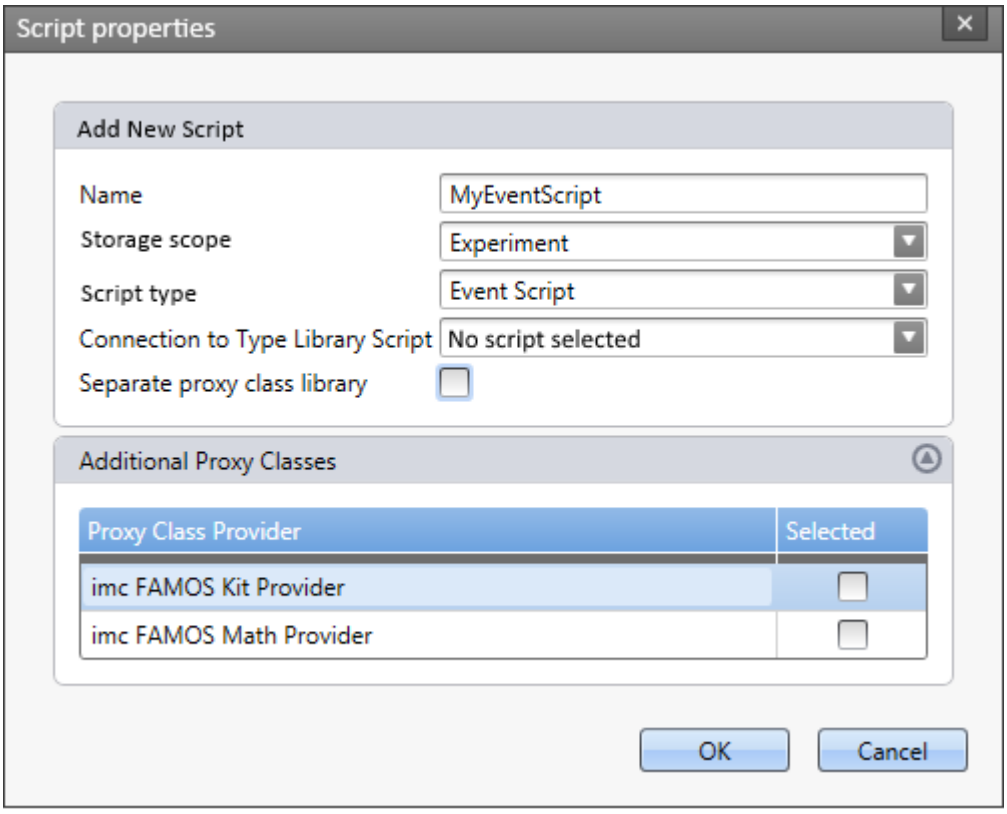


Script with Type Library Script

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;
      }
   }
}
```

14.3.5 Event Script



Dialog: create Event Script

To use an Event Script, go to the Sequencer and the Event tool windows and add the command *Run Script* to an event.

In an Event Script the event arguments / parameters are available.

```
namespace MyEvent {

   public class MyEvent : AbstractScript {

      public override void Run() {
         // Access event arguments by using e.g. this.DevSetupValueChangedArgs
      }
   }
}
```

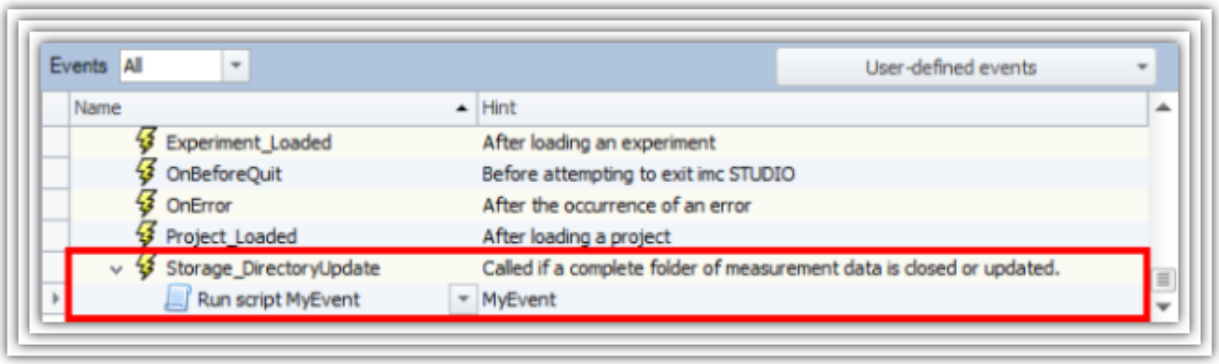
Note

- If a script is linked with a Type Library Script and imc FAMOS functions intended to be used, the imc FAMOS Proxy Classes must be activated in the Type Library Script.
- If the imc FAMOS Math Provider is later uploaded into the Type Library Script using *Edit* > *Regenerate proxies*, then a reference to *imc FAMOS M*ath Provider must be added manually in any script which is linked with the Type Library Script.

Example

In the Sequencer the event *Storage_DirectoryUpdate* is available.

This event is called, when the measurement data in the measurement folder (e.g. interval folder) is completed or updated.



Sequencer: Event Storage_DirectoryUpdate

If an Event Script is run by releasing this event, the data storage parameters are given in `this.QuickStorageInfoEventArgs`.

this.QuickStorageInfoEventArgs	
Directory	Full path to the measurement folder
EventNumber	If using trigger/events, event count and saving in individual files is activated.
MeasureID	Hash code
MeasurementName	Name of the folder
RootPath	Experiment path/folder, where the measurements are saved
StartTime	Start time code
StopTime	Stop time code
Trigger	Trigger name; is empty when only using Trigger_48

```
using System.Windows.Forms;
using System.IO;

namespace MyEvent {

   public class MyEvent : AbstractScript {

      public override void Run() {

       var MyInfo = this.QuickStorageInfoEventArgs;
       if ( MyInfo != null){
         double dStartTime = MyInfo.StartTime;
         double dStopTime = MyInfo.StopTime;
         string strMeasureID = MyInfo.MeasureID;
         string strRootPath = MyInfo.RootPath;
         string strDirectory = MyInfo.Directory;
         string strTrigger     = MyInfo.Trigger;
         int iEventNum         = MyInfo.EventNumber;
         // do something with these parameters

       }else{
         // do something
       } // if
     } // run
   }
}
```

```
// do something with these parameters
string strFullPath = Path.Combine(strRootPath,strDirectory);
double dDuration = dStopTime - dStartTime;
string msg = "Your measurement " + strMeasureID.ToString() + " triggered by  \"" +
      strTrigger + "\" last " + dDuration.ToString("#.###") +
      "s and was saved in " + @strFullPath  + " .\n" +
      "This is the Event #" + iEventNum.ToString();
MessageBox.Show(msg);
```

Additionally, the following settings must be made on the Setup-page:

- Channel_001: Saving (PC) on
- Trigger_01: Event - Virt_Bit01; Action - Channel_001 start/stop
- Devices: Saving Trigger Events to different files (PC)

14.3.6 Third Party Device Script

For more information see chapter Third Party Device Script 1407.

14.4 EClassID - Parameters

14.4.1 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
Download	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

14.4.2 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	
Field bus: Analog inputs	17
Field bus: Digital input / output (ports)	18
Field bus: Digital input / output (bits)	19
Field bus: Sum event	20
Field bus: 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

© 2017 imc Meßsysteme GmbH

Coupling

eCoupling	
AC	1
DC	2
Half-Bridge	3
AC with current supply	4
Full-Bridge	5
Quarter-Bridge	6
Microphone	7
DC charge	8
AC charge	9
TTL	10
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

© 2017 imc Meßsysteme 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

14.4.3 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");
```

Summary	EClassID-Parameter
Description	DescriptionLong
Project	
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 engineer	TestEngineer
Test engineer e-mail	EMailTestEngineer
Test setup	
Test setup description	TestObjectDesc
Test station photo	TestObjPic
Test setup document	TestObjectDoc
Test object	
Test object description	TestSetupDesc
Test object picture	TestStationPhoto
Test object document	TestSetupDoc
Test operator	
User	User
E-mail address user	EMailUser
Annotation (before measurement)	
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
Annotation (after measurement)	
Comment (end)	AnnotationEnd
Environmental conditions (end)	ConditionsEnd

Conditions parameter name (end)	ConditionsParamNameEnd
Conditions parameter value (end)	ConditionsParamValueEnd
Picture (end)	PicEnd
Result	
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	
Test part number	TestPartNo
Test object number	TestObjNo
Short Description	DescriptionShort

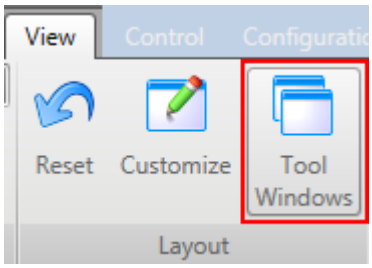
14.4.4 Finding parameters and setting values

Finding parameters and their values

The following describes a way to find the EClassID parameters on the basis of the table description. Another way is described in Finding setting values 1385.

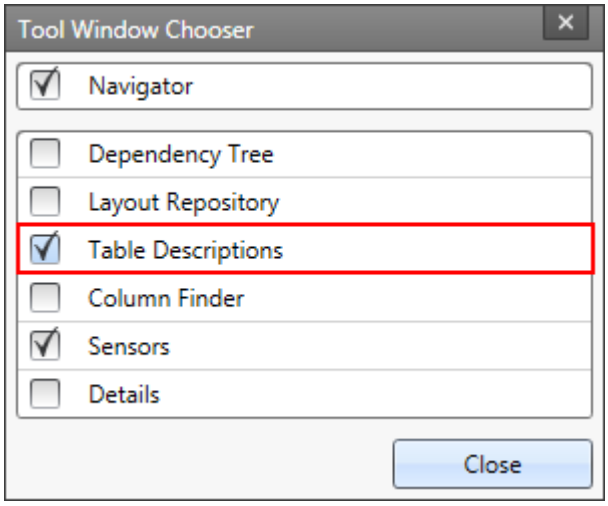
Activating the Table Description tool window

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.

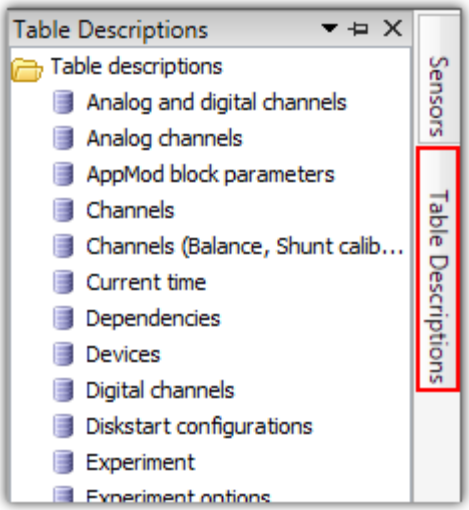


Dialog: Tool Window Chooser

The view can also be saved, so that the tool window is available once again the next time imc STUDIO is started.

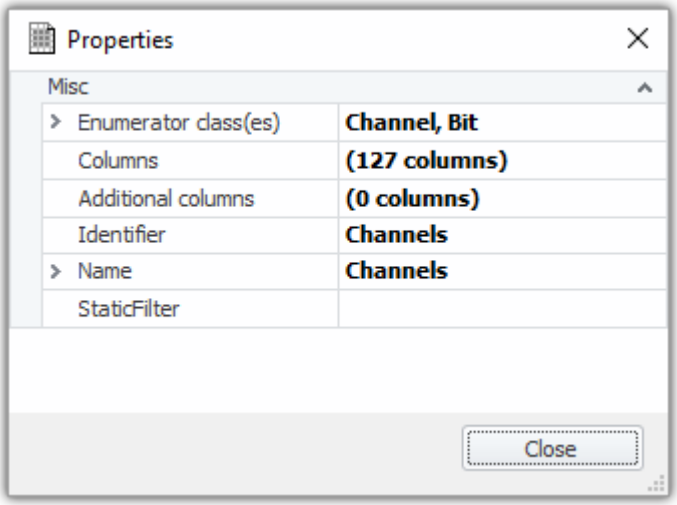
Viewing a table description

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*.



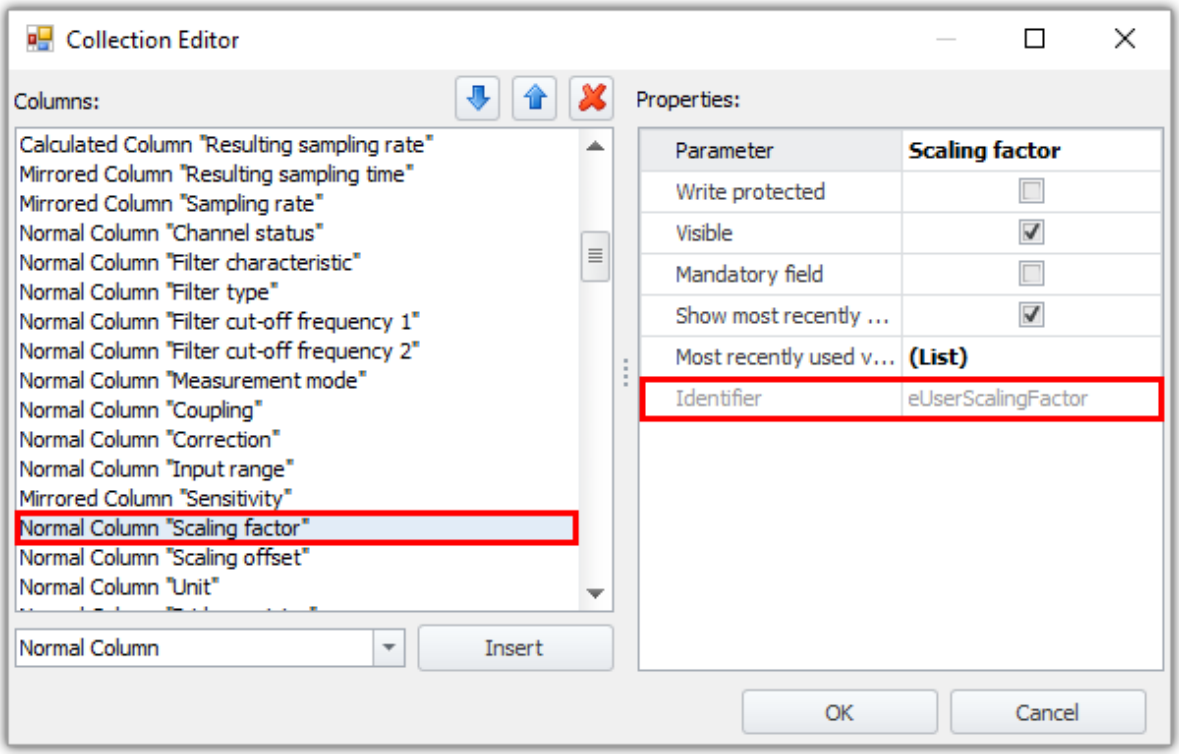
Properties: Analog channels

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*.



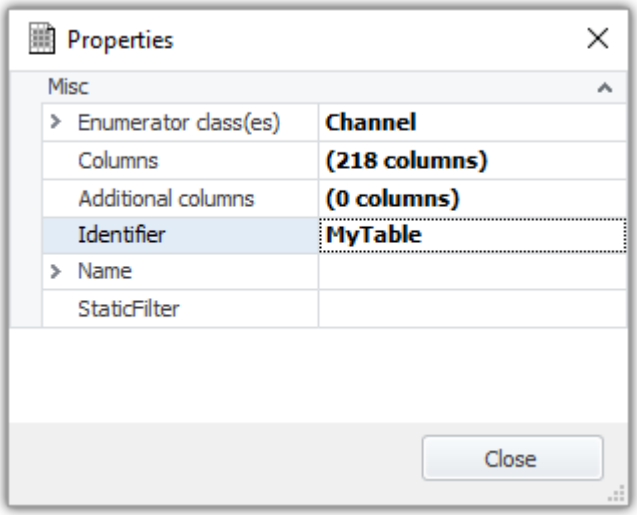
Collection Editor

Finding setting values

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.

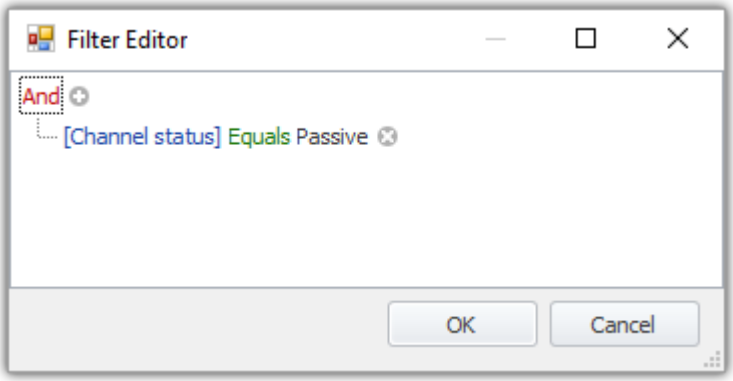


Properties of a user-defined table description

As the Enumerator class, select the desired class containing the parameter, e.g. *Channel*.

Filter Editor

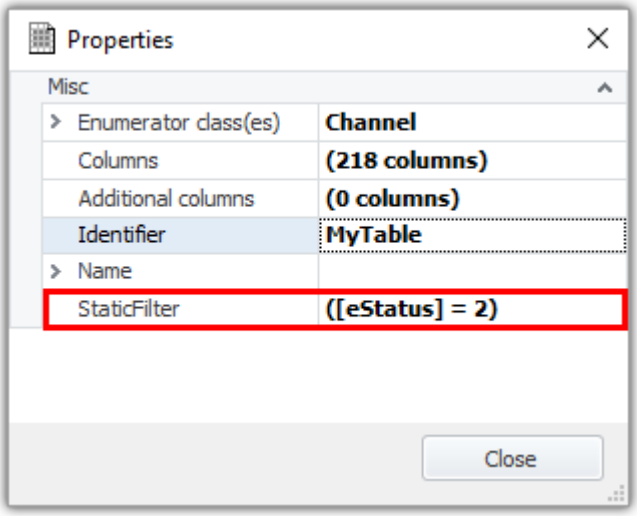
Under *StaticFilter*, select the parameter of which you wish to know what value is set for it, e.g. *Channel status*. For the value, either enter the value desired or select it from the drop-down list. Close the window.



Filter Editor

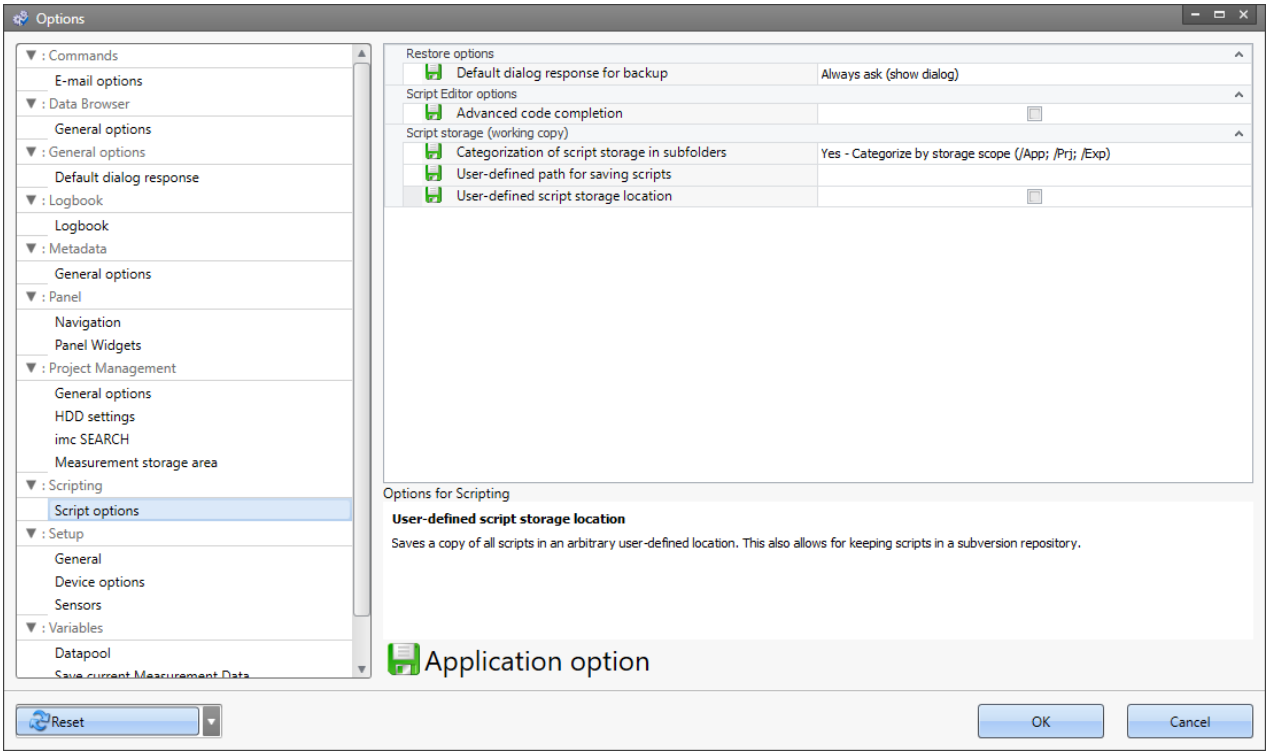
Viewing

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

14.5 Script options



Script options

Script storage (working copy)

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

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

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.

Selection	Description
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 Projekt is used.

For more information, see chapter Restore script changes 1355.

14.6 Tutorial

Here you will find a list of tutorials:

Tutorial	Description
Device and channel settings 1392	A device is added to the device list. After selection of the device, the analog channels are configured.
Channel calculation 1397	Different methods to calculate channels are shown.
Curve Window and channel selection 1398	A channel selection list is created to select, which channel should be shown in the curve window.
RS232-Interface 1401	A example how to control a serial device.
Using the script clipboard 1402	The script clipboard is explained in a little example.

14.6.1 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 dSampleTime = 0.001; // 1 ms = 1000 Hz
double dDuration   = 10.0; //  10 s
double dRange      = 5;  //  +/- 5 V
string strRange    = "\u00b1" + dRange.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 myDevice = DevSetup.Devices.GetItemBySerialNumber(serialNumber);

// if device is not in list, do device search
if ( null == myDevice ) {
   DevSetup.SearchDevices();
   myDevice = DevSetup.Devices.GetItemBySerialNumber(serialNumber);

   // if device is not still in the list -> Warning
   if ( null == myDevice) {
      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 = myDevice.Parameters[EClassID.eDeviceNickname].GetValue().ToString();
string Name = myDevice.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 ( null != paraType ){
      int chnType = (int)paraType.GetValue();
      // for analog channels
      if ( chnType == ParameterValues.eChannelType.AnalogInput) {

         // ... next block

      } // chnType
   } // paraType
} // foreach

// Download
Action.FireAction(this,"acCheckConfiguration");
```

Setting of the parameters is performed as follows:

```
// Parameters
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 ( null != paraStatus ){
   paraStatus.SetValue(ParameterValues.eStatus.Active);
} // paraStatus
// enable saving to PC
if ( null != paraSaveHD ){
   paraSaveHD.SetValue(true);
} // paraSaveHD
// set sample rate
if ( null != paraSampleTime ){
   paraSampleTime.SetValue(dSampleTime);
} // paraSampleTime
// set duration
if ( null != paraDuration ){
   paraDuration.SetValue(dDuration);
} // paraDuration
// set mode to "Voltage"
if ( null != paraMode ){
   paraMode.SetValue(ParameterValues.eChannelMode.DAQ_Voltage);
} // paraMode
// set coupling to DC
if ( null != paraCoupling ){
   paraCoupling.SetValue(ParameterValues.eCoupling.DC);
} // para Coupling
// set Range
if ( null != paraRange ){
   paraRange.SetValue(strRange);
} // 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).

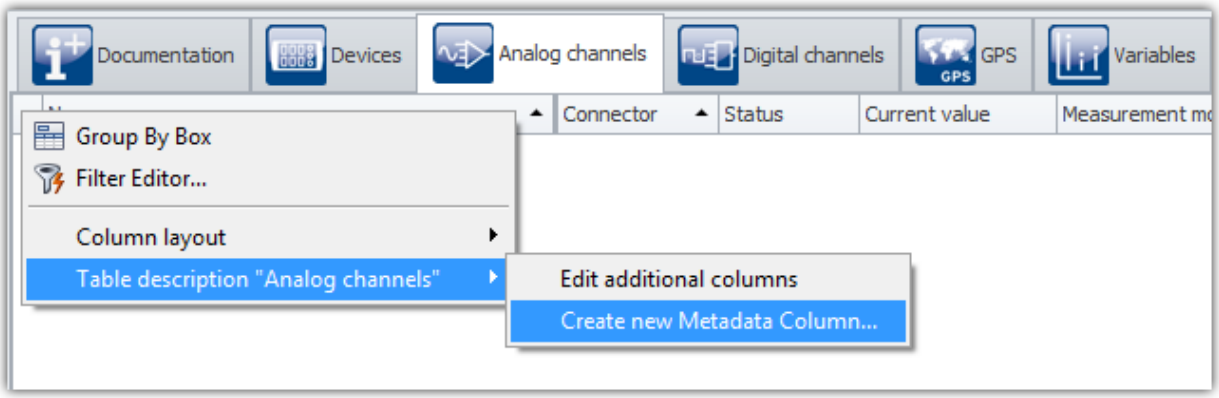
The value of each of these variables is queried in the script and assigned to the corresponding variable.

```
string ERRORSENDER  = "DeviceSettings Script"; // Sender for logbook
string serialNumber = "123456";

double dSampleTime = Datapool["SampleTime"].GetContent<double>(); //
double dDuration   = Datapool["Duration"].GetContent<double>(); //
double dRange      = Datapool["Range"].GetContent<double>();  //
string strRange    = "\u00b1" + dRange.ToString() + " V"; // +/- Symbol and Unit
```

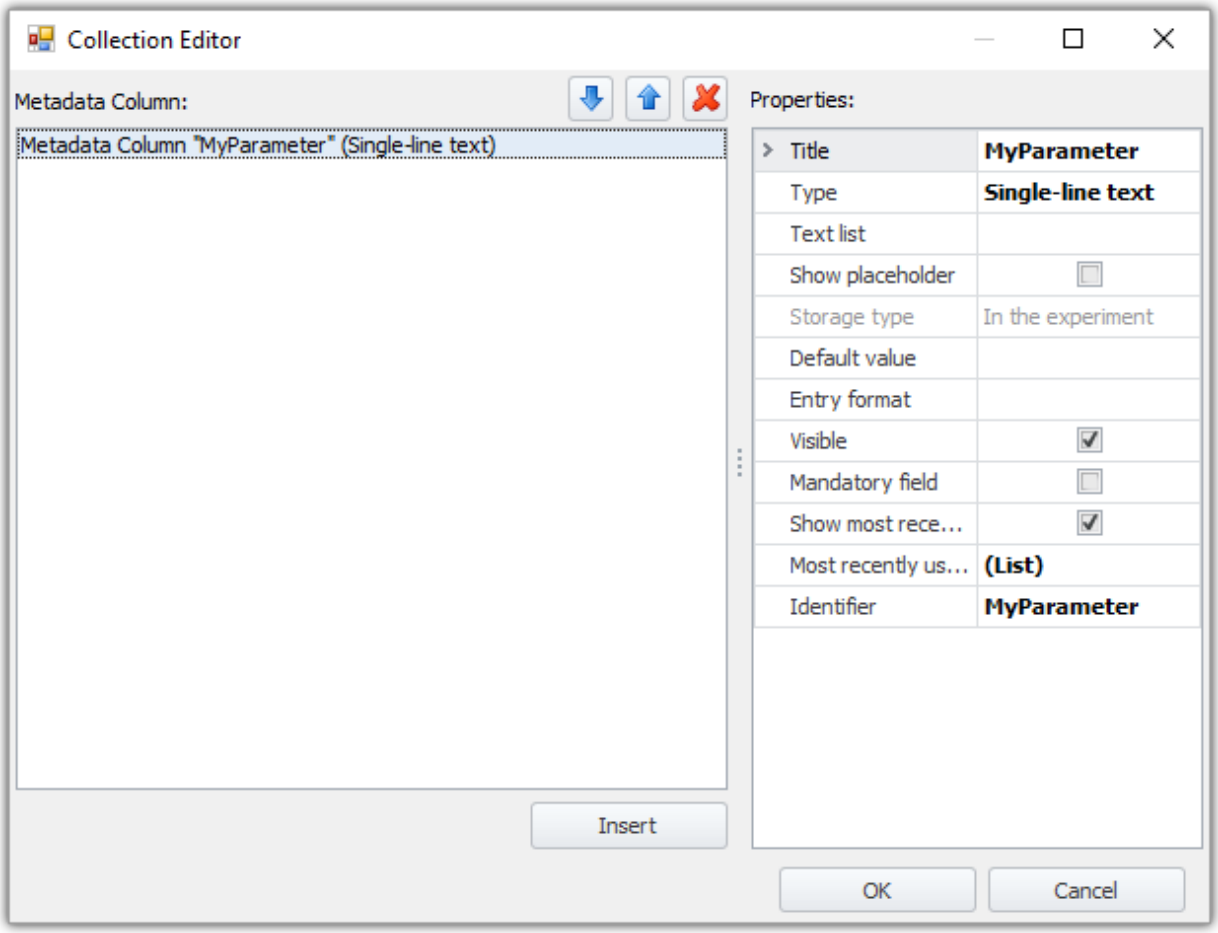
After the script has been compiled, it is possible to set the parameters by means of the widgets while the Design-mode is deactivated. A button which causes the script to run when pressed (command: *Run Script*) is added to the Panel page. When the button is activated, the values are updated to the *Analog channels* page.

Extension: Metadata columns

On the Setup-page *Analog channels*, create your own metadata column 211 *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 paraMy = channel.Parameters["MyParameter"]; // Identifier
// ...
if ( null != paraMy ){
   paraMy.SetValue("#" + count.ToString());
   count++;
}
```

After running the script, the newly added table column of the analog channels fills itself.

14.6.2 Channel calculation

Calculation without imc FAMOS

In this tutorial, analog channels can be subjected to operations using imc FAMOS. 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 is to begin following a measurement, and the current channels are to be added and outputted on an output channel.

The following libraries are needed:

```
using imc.Common.Components.DataManager;
using imc.Common.Components.DataManager.Interfaces;
```

First, the channels are retrieved from the datapool, and the output channel is created.

```
IDmChannel channel1 = Datapool["Channel_001"].GetContent<IDmChannel>();
IDmChannel channel2 = Datapool["Channel_002"].GetContent<IDmChannel>();
DmChannel channelOut = DmChannel.CloneOf(channel1);
channelOut.Name = "Channel_Out";
```

Subsequently, the data can be jointly subjected to calculation operations.

```
double[] data = new double[channel1.Data.Length];

for (int k = 0; k < channel1.Data.Length; k++){
   data[k] = channel1.Data[k] + channel2.Data[k];
}

channelOut.SetData<double>(data);
```

The output channel is then written to the datpool.

```
DataPoolVariable cOut = Datapool["Channel_Out"];
if ( null == cOut ){
   Datapool.CreateVariable<DmChannel>(channelOut.Name, channelOut);
}else{
   cOut.SetContent<DmChannel>(channelOut);
}
```

Subsequently, the measurement is started. At te end of the measurement, the script is run. A user-defined variable *Channel_Out* is created in case it does not exist already.

The two analog channels and the results channel can now be displayed in the curve window.

Extension: Weight and offset

In the example above, the channel values are added together. The weighting depends on the user-defined variables which were previously created.

To do this, create in the Data Browser the user-defined numerical variables *w1* and *w2*, as well as *offset*, and display them as numerical widgets.

To the script above, the following lines are added, which retrieve the values of the weighting and of the offset from the datapool.

```
double w1 = Datapool["w1"].GetContent<double>();
double w2 = Datapool["w2"].GetContent<double>();
double offset = Datapool["offset"].GetContent<double>();
```

The calculation is then performed as follows:

```
double[] data = new double[channel1.Data.Length];

for (int k = 0; k < channel1.Data.Length; k++){
   data[k] = w1 * channel1.Data[k] + w2 * channel2.Data[k] + offset;
}

channelOut.SetData<double>(data);
```

Change the values for the weighting and for the offset and then run the script. The results channel should reflect the weighting and offset accordingly.

Extension: imc FAMOS functions

Hint

When creating the script, the imc FAMOS Providers *Kit* and *Math* must be selected.

Next, add the following library to the beginning of the script:

```
using imc.Common.Components.Famos.Math;
```

The calculation is now performed by means of an imc FAMOS-command, which applies math operations to the two channels while taking the weighting into account.

What is important is for the channel name to be reset, since it is overwritten within the function.

```
bool bSuccess = Famos.Add(channel1, w1, channel2, w2, out channelOut);
channelOut.Name = "Channel_Out";
```

It is also possible to use additional imc FAMOS functions.

14.6.3 Curve Window and channel selection

In this tutorial, a widget is to be used to select an analog channel, which is then displayed in the curve window.

To do this, switch all analog channels to active and perform a *Prepare*. Add a standard curve window to the Panel page and assign it the name "MyCurve".

Create a new Panel Script with the name *ChannelChooser*.

The following libraries are needed:

```
using System.Windows.Forms; // ComboBox
using imc.Common.Interfaces.Panel.Scripting; // eAppendConstant
using imc.Common.Interfaces.Logbook; // Logbook Category
using imc.Studio.Tools.DevSetupGraphGenerator; // EClassID
```

The corresponding widget is created as an attribute in the class.

```
public class ChannelChooser : AbstractScript {
   ComboBox controlName    = new ComboBox();
   string indexVariable = "ChannelIndex";

   public override void Run() {
      // ...
   }

]
```

Widget Zones cannot be added via Scripting yet. Thus, a ComboBox of the Windows.Forms class is used instead.

The combo-box is now to be filled with the list of analog channels and displayed on the Panel page. This is done in the PanelScriptInitialize() method.

```
public override bool PanelScriptInitialize()
{
   // ComboBox erstellen
   ComboBox cb = new ComboBox();
   cb.Name = controlName;
   cb.Location = new System.Drawing.Point(100, 200);
   cb.Items.Clear();

   // Add channels to list
   foreach (var chn in DevSetup.Channels){

      var paraState = chn.Parameters[EClassID.eStatus];
      var paraType = chn.Parameters[EClassID.eChannelType];
      var paraName = chn.Parameters[EClassID.eChannelName];

      if (paraState != null && paraType != null && paraName != null){

         if ( ((int)paraState.GetValue() == ParameterValues.eStatus.Active) &&
           ( (int)paraType.GetValue() == ParameterValues.eChannelType.AnalogInput) ){
           cb.Items.Add(paraName.GetValue());
         }
      }
   } // foreach


   if (Core.Datapool[indexVariable] != null) {
      cb.SelectedIndex = Core.Datapool[indexVariable].GetContent<int>();
   }

   // Hook an event listener
   cb.SelectedIndexChanged += MyCombo_SelectedIndexChanged;
   PanelPage.Controls.Add(cb);
   return true;//Return true if everything is initialized, otherwise return false
}
```

If the selection changes, the signal is to be changed in the curve window.

```
private void MyCombo_SelectedIndexChanged(object sender, System.EventArgs e)
{
   ComboBox cb = sender as ComboBox;
   if (cb != null)
   {
      if (cb.Name == controlName)
      {
         if (!Core.Datapool.CreateVariable<int>(indexVariable, cb.SelectedIndex))
         {
            Core.Datapool[indexVariable].SetContent<int>(cb.SelectedIndex);
         }

         // delete every signal in curve window
         foreach (var s in MyCurve.Signals){
               s.Delete();
         }

         MyCurve.AddSignal(cb.SelectedItem.ToString(),     eAppendConstants.AppendNewLine);

      }
   }
}
```

The event has to be unhooked and the old ComboBox has to be deleted:

```
public override bool PanelScriptDispose()
{
   try
   {
      if (PanelPage != null)
      {
         foreach (System.Windows.Forms.Control el in PanelPage.Controls)
         {
            ComboBox cb = el as ComboBox;

            // Unhook events from the ComboBox named "MyCombo"
            if (el.Name == controlName && cb != null)
            {
               cb.SelectedIndexChanged -= MyCombo_SelectedIndexChanged;
            }
         }
         PanelPage.Controls.Clear();
      }
      return true;// Return true if everything is disposed, otherwise return false
   }
   catch (Exception e)
   {
      // Write to the logbook if anything went wrong
      this.Core.Logbook.LogEntry(this.ToString(), e, ELogbookEntryCategory.Warning);
   }
   return false;
}
```

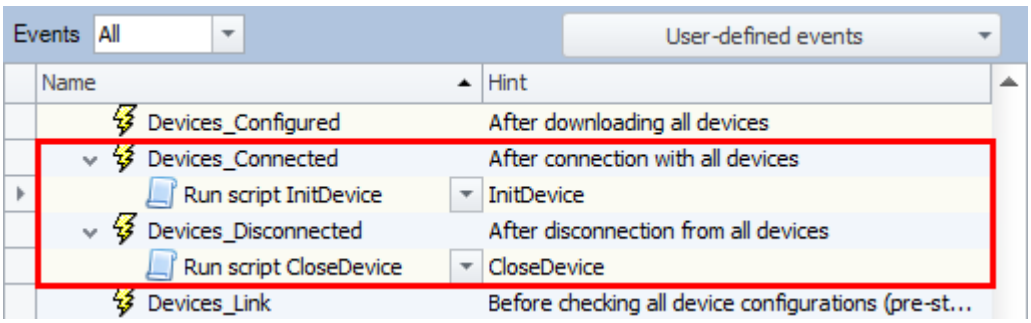
The methods Run() and Stop() are empty.

With this combination, it is now possible to select a curve window's channel directly by means of a combo-box.

14.6.4 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 cCOMPort = new SerialPort("COM1", 9600, Parity.None, 8, StopBits.One );
   // Open the port
   cCOMPort.Open();
   // Store the COM port in the clipboard
   ScriptClipboard.Add("COM1", cCOMPort);
}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 cCOMPort = ScriptClipboard["COM1"] as SerialPort;
   // Close the COM port
   cCOMPort.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 cCOMPort = ScriptClipboard["COM1"] as SerialPort;
   // Read a message from the device
   string strData = cCOMPort.ReadLine();
}catch(Exception x){
   // error handling
}
```

Write data

WriteData.cs

```
try{
   // Get the COM port from clipboard
   SerialPort cCOMPort = ScriptClipboard["COM1"] as SerialPort;
   // Send a message to the device
   cCOMPort.WriteLine("*IDN?");
}catch(Exception x){
   // error handling
}
```

14.6.5 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;
}
```

14.7 Important Information

Close Script Editor before "Experiment > Save as"

Hint

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 FAMOS functions and Type Library Script

Note

- If a script is linked with a Type Library Script and imc FAMOS functions intended to be used, the imc FAMOS Proxy Classes must be activated in the Type Library Script.
- If the imc FAMOS Math Provider is later uploaded into the Type Library Script using *Edit* > *Regenerate proxies*, then a reference to *imc FAMOS M*ath Provider must be added manually in any script which is linked with the Type Library Script.

Use of an 4.0R1 experiment including scripts

Import of an imc STUDIO 4.0R1 Experiment with scripts

Due to changes on the interface and provision of new functions, the following steps are needed for using script from an imc STUDIO 4.0R1 experiment.
Before committing any changes, it is recommended to export the scripts as backup.

Error: The script cannot be run.	Solution
Compiler error CS0535 `..\AbstractScript.cs(6,24) : Error CS0535: 'ScriptName.AbstractScript' does not implement interface member 'System.IDisposable.Dispose()'`	Please insert the method `Dispose()` to AbstractScript.cs: `public virtual void Dispose() {}`
Compiler error CS1729 `..\AbstractScript.cs(163,23) : Error CS1729: 'ScriptName.DataPool' does not contain a constructor that takes 1 arguments`	Please replace every instantiation with `new` by the method `GetInstance(), e.g.` `MyClass myObject = new MyClass(*)` by `MyClass myObject = MyClass.GetInstance(*)`

Compiler error CS0234
`..\ProxyClasses.cs(6754,53) : Error CS0234: The type or namespace name 'EVariableProperty' does not exist in the namespace 'imc.Common.Interfaces.DataPool' (are you missing an assembly reference?)`
	Affected methods:
	`public bool Set<T>(*)`
	`public T Get<T>(*)`
	Please add the reference *imc.Common.Interfaces.UnManaged* to your script project (*Project > Add Reference)* and compile again.

Compiler error CS0029 `Error CS0029: Cannot implicitly convert type 'System.EventHandler' to 'System.Action<object,System.EventArgs>'`	Please replace `EventHandler(*)` by `Action<object,EventArgs>(*)`.

Parallel installation of multiple imc STUDIO versions

Necessary adaptations to make scripts executable in the older version

When a newer imc STUDIO version is installed in parallel, it can happen that the scripts in the older version can no longer be run.

To make the scripts in the older version executable, take the following steps:

- Open the experiment affected
- Use the Explorer to open the script project file (*.csscriptproj) in an editor program
- Edit the command lines according to the table below and save the file
- Save the experiment and restart the script. It should now run correctly.

Table: Altering the *.csscriptproj - file of a script belonging to imc STUDIO 4.0R1

Original Text	Changes

`<Reference Include="imc.Studio.Interfaces.V2">`
 	
`<Private>False</Private>`
`<SpecificVersion>False</SpecificVersion>`
`</Reference>`
	`<Reference Include="imc.Studio.Interfaces.V2, Version=4.0.1.0, Culture=neutral, PublicKeyToken=68a4b1d388e6c0b6">`
	`<Private>False</Private>`
	`<SpecificVersion>True</SpecificVersion>`
	`</Reference>`

`<Reference Include="imc.Common.Interfaces>`
 	
`<Private>False</Private>`
`<SpecificVersion>False</SpecificVersion>`
`</Reference>`
	`<Reference Include="imc.Common.Interfaces, Version=4.0.1.0, Culture=neutral, PublicKeyToken=9350ef35614b8d14">`
	`<Private>False</Private>`
	`<SpecificVersion>True</SpecificVersion>`
	`</Reference>`

`<Reference Include="imc.Studio.Interfaces">`
 	
`<Private>False</Private>`
`<SpecificVersion>False</SpecificVersion>`
`</Reference>`
	`<Reference Include="imc.Studio.Interfaces, Version=4.0.1.0, Culture=neutral, PublicKeyToken=68a4b1d388e6c0b6">`
	`<Private>False</Private>`
	`<SpecificVersion>True</SpecificVersion>`
	`</Reference>`

The scripts are then only executable for this particular version. Adaptation of the reference version for subsequent versions (5.0R1 -> 5.0.1.xxxx) must be performed correspondingly.

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

Hint

The event handling methods are run asynchronously.

15 Third Party Device Interface

Documentation on the imc STUDIO plug-in imc STUDIO Third Party Device Interface.

15.1 Getting started

15.1.1 Introduction

Hint

- 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* and *ChannelLoader* 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:

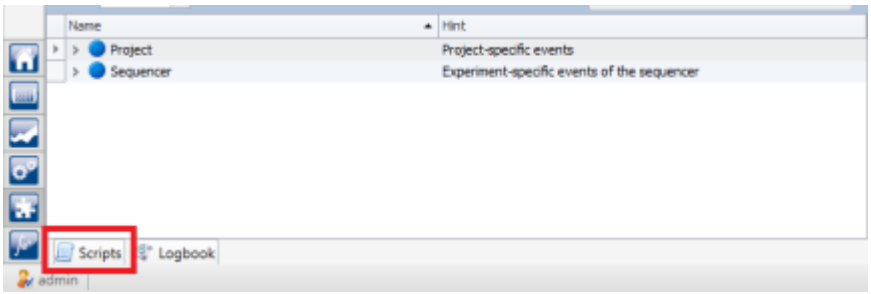
- Experience in C# programming is necessary (C++ for an *unmanaged* data transfer)
- The interface of the Third Party Device has to be known.

15.1.2 Add a new script

Note

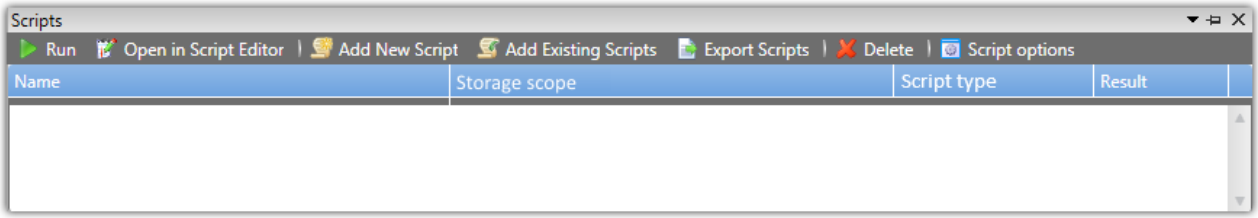
For more information about Scripting, see here 1312.

Scripts are created on the Sequencer page by clicking on the tab *Scripts* at the bottom of the page.



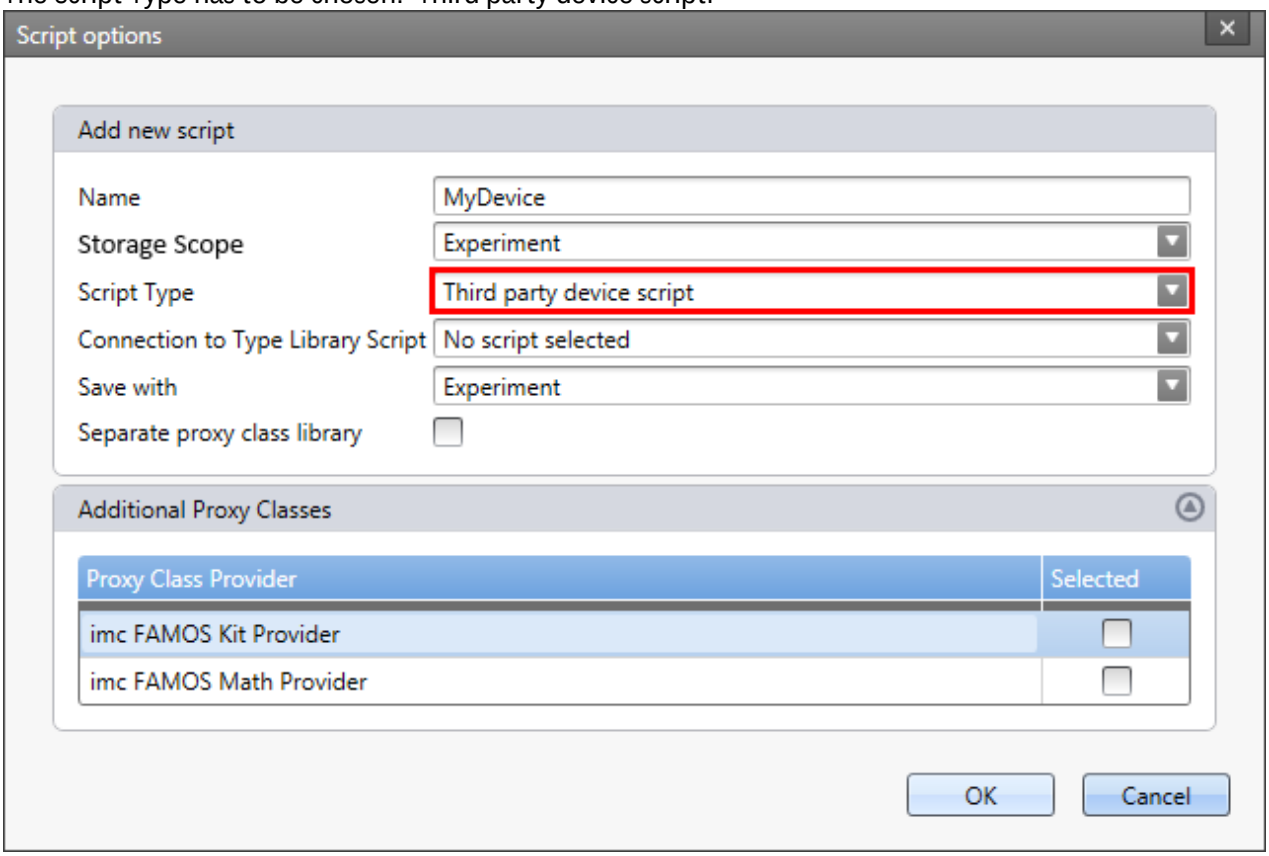
Sequencer: Tool window "Scripts"

The tool window *Scripts* will open. Click on *Add new script* Add New Script to add a new Script



Sequenzer: Werkzeugfenster "Skripte"

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.

Hint

To use another template, please use the 3rd Party Device Management 1414.

15.2 Serial interface

15.2.1 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 cCOMPort = new System.IO.Ports.SerialPort("COM1", 9600,
Parity.None, 8, StopBits.One );
   // Open the port
   cCOMPort.Open();
   // Store the COM port in the clipboard
   ScriptClipboard.Add("COM1", cCOMPort);
}catch(Exception x){
   // error handling
}
```

Close COM Port / disconnect from device

```
try{
   // Get the COM port from clipboard
   System.IO.Ports.SerialPort cCOMPort = ScriptClipboard["COM1"] as System.IO.Ports.SerialPort;
   // Close the COM port
   cCOMPort.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 cCOMPort = ScriptClipboard["COM1"] as SerialPort;
   // Send a message to the device
   cCOMPort.WriteLine("*IDN?");
}catch(Exception x){
   // error handling
}
```

Read data

```
try{
   // Get the COM port from clipboard
   SerialPort cCOMPort = ScriptClipboard["COM1"] as SerialPort;
   // Read a message from the device
   string strData = cCOMPort.ReadLine();
}catch(Exception x){
   // error handling
}
```

15.3 Third Party Device script

15.3.1 Add a device

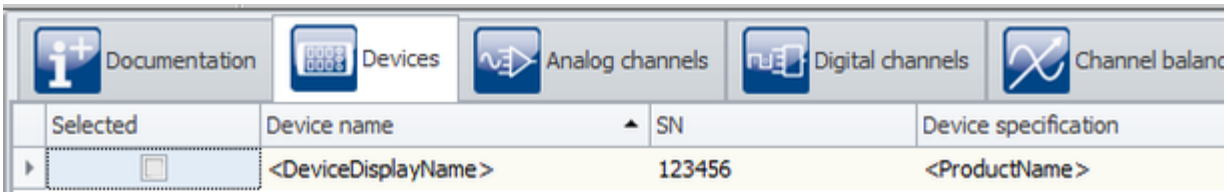
First of all the device (or the devices) has to be configured. In the method `GetAvailableDevices()` the properties of the devices is programmed.

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 1410 to the device.

15.3.2 Add a channel

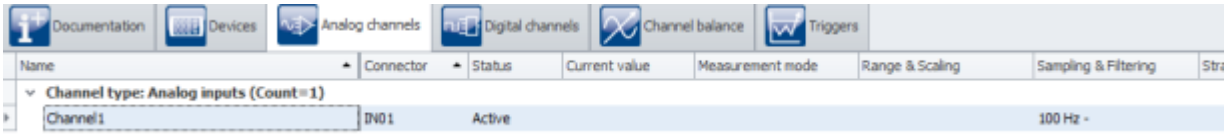
The channel configuration of the parameters is set in the following procedure. This method is run when the device is selected. In this example the configuration is made for only one channel. To configure more channels 1435, use a loop or create a method. Here, the channel type is set to analog input, 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 1437 to the device, the channel type has to be set to "DisplayVar".

15.3.2.1 FFT channel

FFT channel

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)"-

Time-stamped channel

For a time-stamped channel (TSA), the parameter *eXFormat* is set to "Timestamp".

```
var channel = device.Children.GetOrAdd(ESetupNodeType.Channel, "<DeviceName>.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 (protocol channel) set DataType to ASCII
channel.Parameters.UpdateOrAdd(EClassID.eDataType, value: ParameterValues.eDataType.ASCII);
```

The segment is located in the collapsed region "Example for time stamped channel parameters".

15.3.2.2 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;
   }
}
```

15.3.3 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("Editiere Konfiguration");
      // do something
      break;
   case 2:
      MessageBox.Show("Importiere Konfiguration");
      // do something
      break;
   case 3:
      MessageBox.Show("Exportiere Konfiguration");
      // 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.

15.3.4 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 1429), or whether it causes a function call (*callback*) when it has data (see SimplePushDevice 1432).

The SimplePollDevice and the SimplePushDevice are good examples of such Third Party Device.

For further information, please read DeviceDriver and PushDeviceDriver 1424.

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);
```

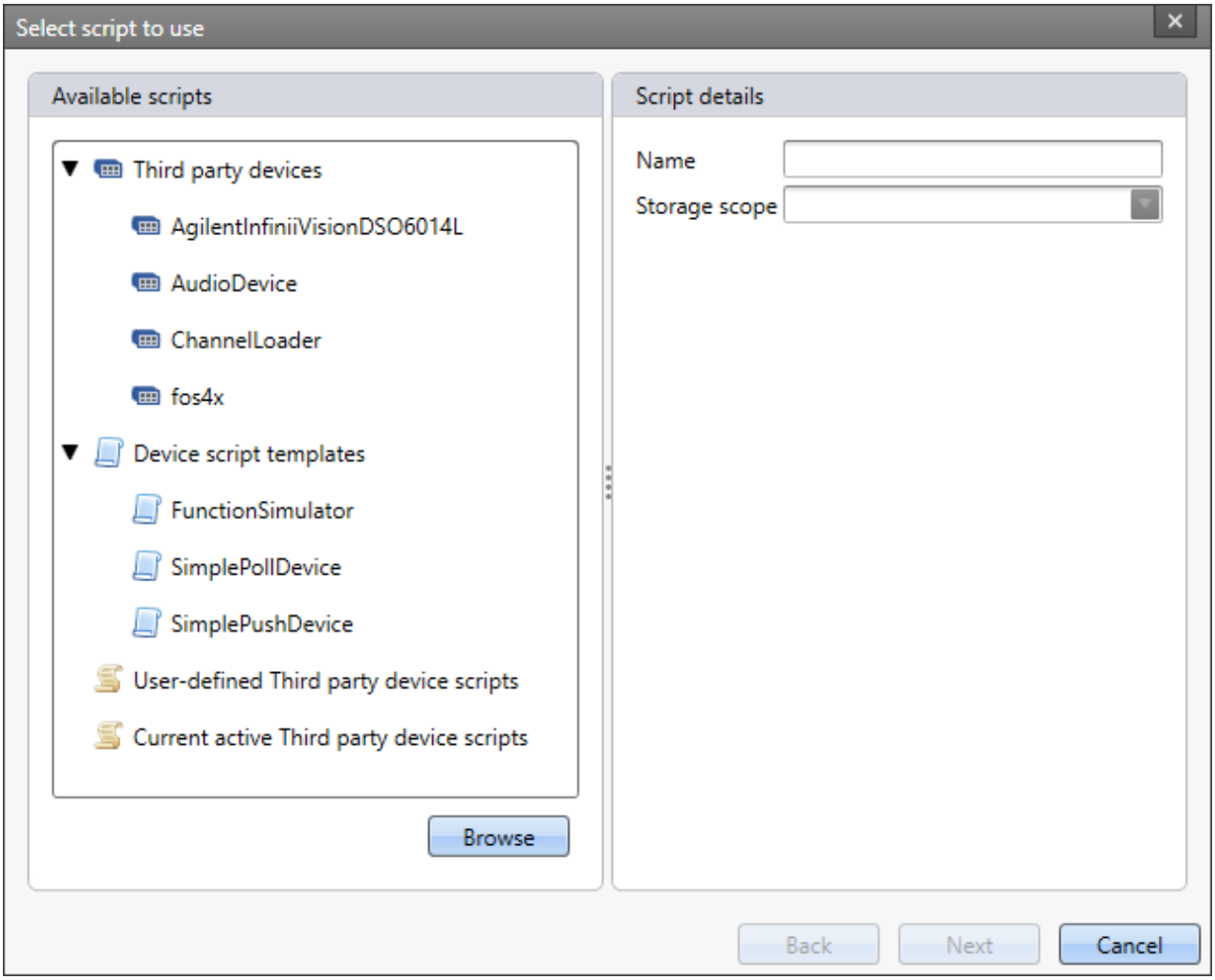
15.4 3rd Party Device Management

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 Agilent Scope and the fos4x are also available. The last of the two device require 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 necessary.
- The AudioDevice and the ChannelLoader can be used without any 3PDI license.

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 1415.

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 1416.

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, ...).

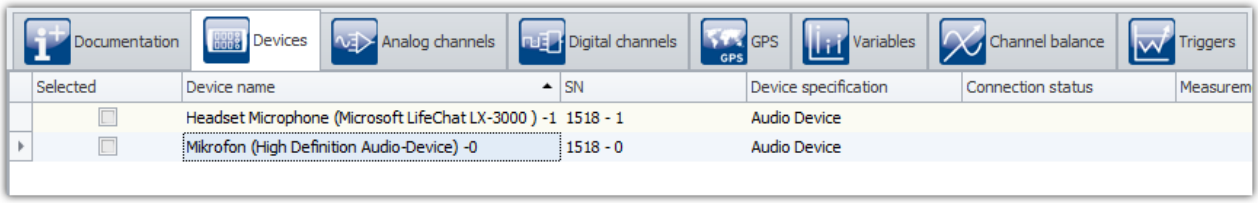
15.4.1 Device Scripts

15.4.1.1 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

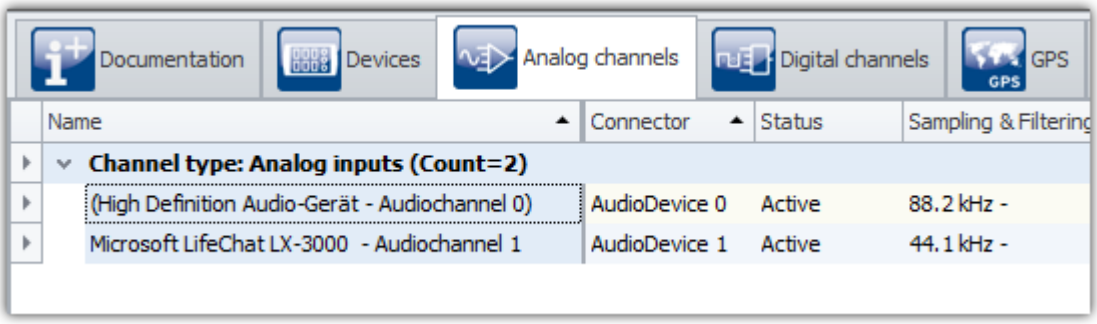
Afterward all audio devices are listed in the device list.



Device list: AudioDevice

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

Notes

- 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.

Remarks to certain devices

Notes

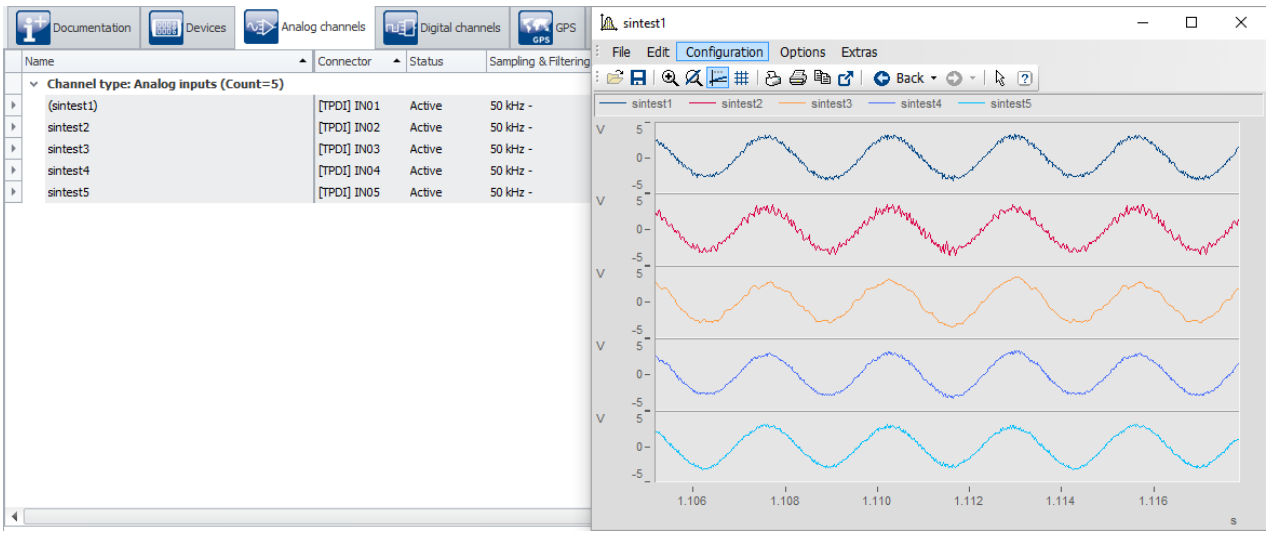
- With the webcam Logitech HD Pro C920, it was noticed that it was only delivering one audio channel, although according to the Windows settings there should be two channels.
  This may also apply to other audio devices.
- With the Headset Microsoft LifeChat LX-3000, it was noticed that after disconnecting and reconnecting the USB cable, a new measurement is only possible after the *second* start.

15.4.1.2 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 performing a *Download* 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

15.4.1.3 Agilent Scope

15.4.1.3.1 First Steps

It is possible to use the device Agilent InfiniiiVision DSCO6014L in imc STUDIO as a 3rd-party device. This device can be configured and controlled via the imc STUDIO interface. Capture of measurement data as individual snapshots is implemented as an event-based data set. Furthermore, synchronized measurement along with imc measurement devices (e.g. imc CRONOS*flex*) is possible. In this documentation, the device is referred to as Agilent Scope.

Preparations

Before the Agilent Scope can be used, certain preparations must be made.

As of 2014, the electronic measurement device branch of Agilent Technologies has been spun off to the company Keysight Technologies. For this reason, this documentation will refer to the company name "Keysight".

Activating the Keysight software

Installing the IO Library Suite

If you haven't already, install the *IO Library Suite*. This software is provided on the imc STUDIO data carrier under the folder:

..\Products\Agilent InfiniiVision DSO6000L\

Version on the DVD: 16.3.17914

Otherwise, it is also possible to download a newer version from Keysight Technologies.

After installation, use the program *Keysight Connection Expert* to add Agilent Scope to the Instruments list by means of a device-scan. For more information, you can also refer to Keysight's help info.

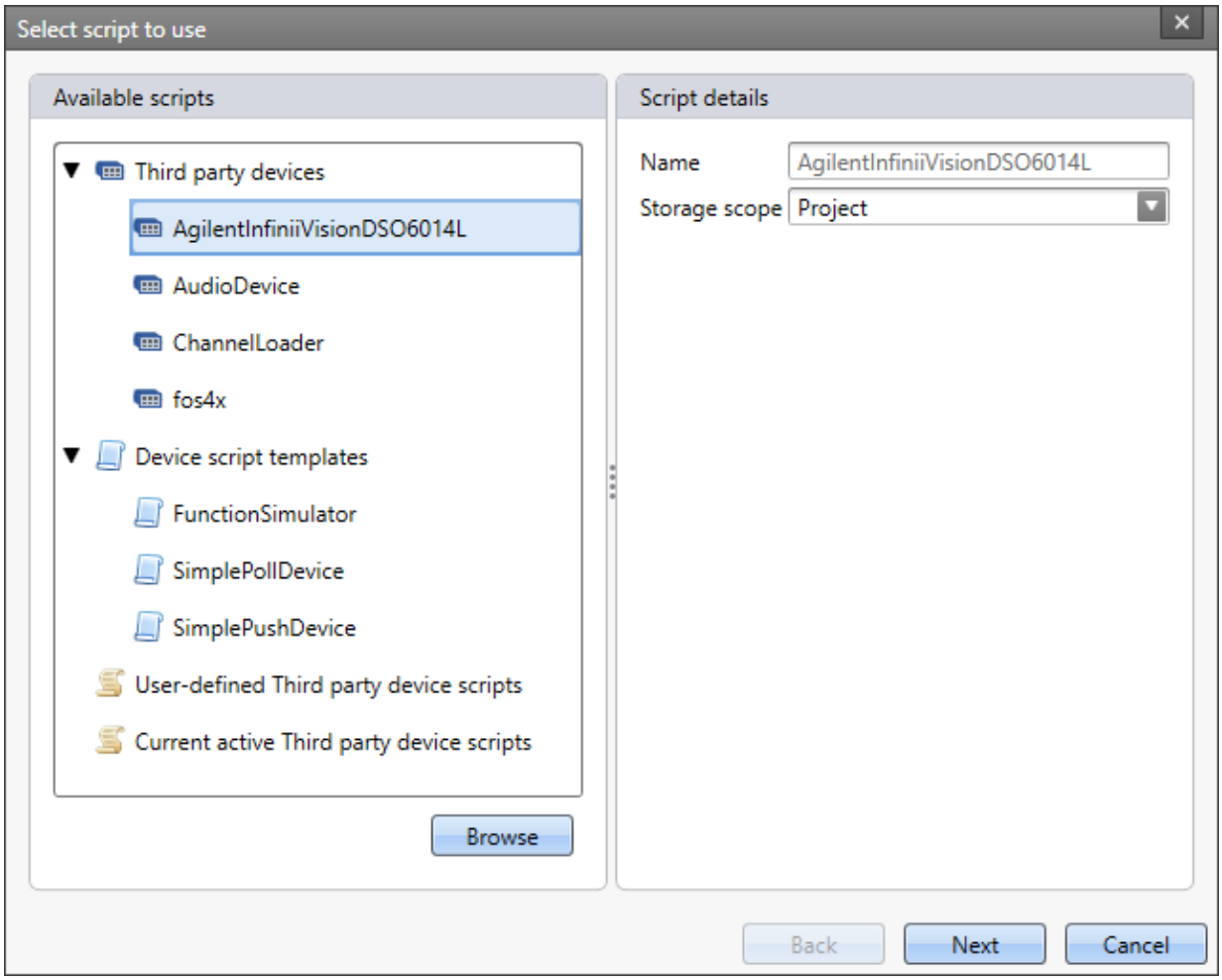
License activation

To use the Agilent Scope, the license imc STUDIO 3PDI-Digital Scope is required.

Adding a device

Integrating the Agilent Scope

Go to the Setup–page. In the menu ribbon, click on *Setup Control > 3rd Party Device Management* and select *AgilentIntiniiVisionDSO6014L*.



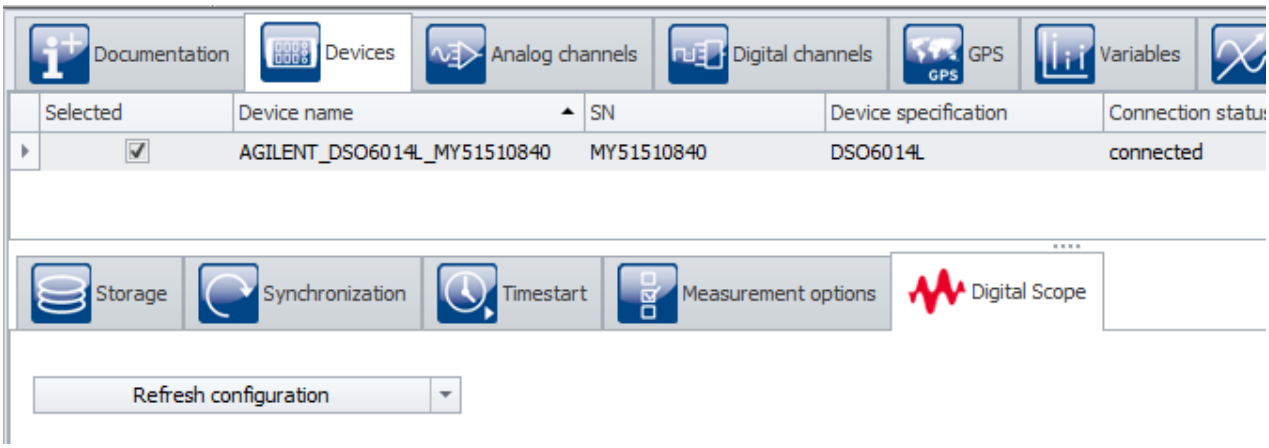
Integrating Agilent Scope

15.4.1.3.2 Settings

15.4.1.3.2.1 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 the device is selected, a new dialog Digital Scope is shown.

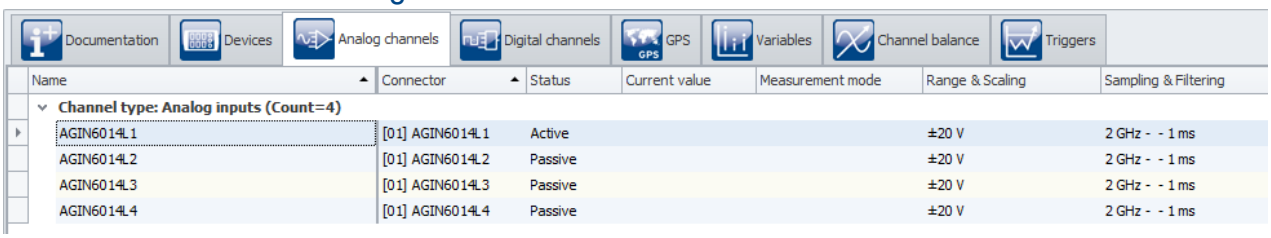


Action	Description
Reset	The device is reset and the input channels' default settings are loaded in imc STUDIO.
Edit configuration	Editing of the device-specific settings in the Browser via the web interface tool. Click on the opened web page on *Browser Web Control* at left next on the blue highlighted link *Remote Front Panel*. A Java-applet opens with settings options for the Agilent Scope.
Refresh configuration	The current settings which were set by the web-interface, for example, are read by the Agilent Scope and the parameter values are updated in imc STUDIO accordingly.

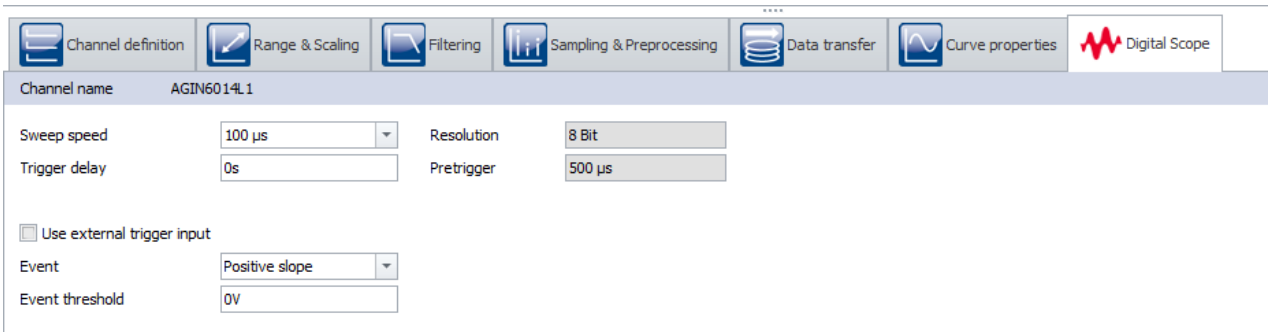
Problems with JAVA

- If no window opens, check whether JAVA is installed.
- Furthermore, the Agilent Scope's IP should be shown in the Java Control Panel in the list of excepted web sites under *Security*.

15.4.1.3.2.2 Channel settings



Analoge Kanäle



Special parameters of analog channels

Dependencies of channel parameters

Sample time and Sweep speed

The sampling rate depends on the sweep speed setting. This setting also determines the measurement duration of individual snapshots.

Sampling rate & processing function

Depending on the kind of processing function selected, the values for the sampling rate settings adjust accordingly. The device offers four kinds of processing functions:

- Normal
- Averaging
- High resolution
- Peak detect (not yet implemented)

Sampling rate & status of input channels

The device has a total of four channels and two channel pairs:

- Channel 1 & Channel 2
- Channel 3 & Channel 4

Reduction of the sampling rate

If both channels (Channel 1 and Channel 2, or Channel 3 and Channel 4) are activated, the sampling rate decreases by one half.

Sampling rate and processing function

The change of the sampling rate in relation to the sweep speed is presented in the table below.

Channel pairs not active

Sampling rate as a function of the processing function - Channel pairs are not active				
Sweep speed [ms]	Normal	Averaging	High Resolution	Peak detect*
0.05	-	2 GHz	-	-
0.1	2 GHz	-	2 GHz	2 GHz
0.2	2 GHz	500 MHz	500 MHz	2 GHz
0.5	500 MHz	200 MHz	200 MHz	200 MHz
1	400 MHz	100 MHz	100 MHz	100 MHz
2	200 MHz	50 MHz	50 MHz	50 MHz
5	80 MHz	20 MHz	20 MHz	20 MHz
10	40 MHz	10 MHz	10 MHz	10 MHz
20	20 MHz	5 MHz	5 MHz	5 MHz
50	8 MHz	2 MHz	2 MHz	2 MHz
100	4 MHz	1 MHz	1 MHz	1 MHz
200	2 MHz	500 kHz	500 kHz	500 kHz
500	800 kHz	200 kHz	200 kHz	200 kHz
1000	400 kHz	100 kHz	100 kHz	100 kHz
2000	200 kHz	50 kHz	50 kHz	50 kHz
5000	80 kHz	20 kHz	20 kHz	20 kHz
10000	40 kHz	10 kHz	10 kHz	10 kHz
20000	20 kHz	5 kHz	5 kHz	5 kHz
50000	8 kHz	2 kHz	2 kHz	2 kHz

* not implemented at this time

Channel pairs active

Sampling rate as a function of the processing function - Channel pairs are active				
Sweep speed [ms]	Normal	Averaging	High Resolution	Peak detect*
0.05	-	500 MHz	-	-
0.1	2 GHz	-	500	2 GHz
0.2	1 GHz	250 MHz	250 MHz	250 MHz
0.5	400 MHz	100 MHz	100 MHz	100 MHz
1	200 MHz	50 MHz	50 MHz	50 MHz
2	100 MHz	25 MHz	25 MHz	25 MHz
5	40 MHz	10 MHz	10 MHz	10 MHz
10	20 MHz	5 MHz	5 MHz	5 MHz
20	10 MHz	2.5 MHz	2,5 MHz	2,5 MHz
50	4 MHz	1 MHz	1 MHz	1 MHz
100	2 MHz	500 kHz	500 kHz	500 kHz
200	1 MHz	250 kHz	250 kHz	250 kHz
500	400 kHz	100 kHz	100 kHz	100 kHz
1000	200 kHz	50 kHz	50 kHz	50 kHz
2000	100 kHz	25 kHz	25 kHz	25 kHz
5000	40 kHz	10 kHz	10 kHz	10 kHz
10000	20 kHz	5 kHz	5 kHz	5 kHz
20000	10 kHz	2,5 kHz	2,5 kHz	2,5 kHz
50000	4 kHz	1 kHz	1 kHz	1 kHz

* not implemented at this time

Resolution and processing function

Depending on the selections for the processing functions and processing values, the resolution in bits (vertical bit resolution) changes accordingly.

The setting for the amount of processing values is only available for the processing function "Averaging".

Processing function & processing values & resolution in bits

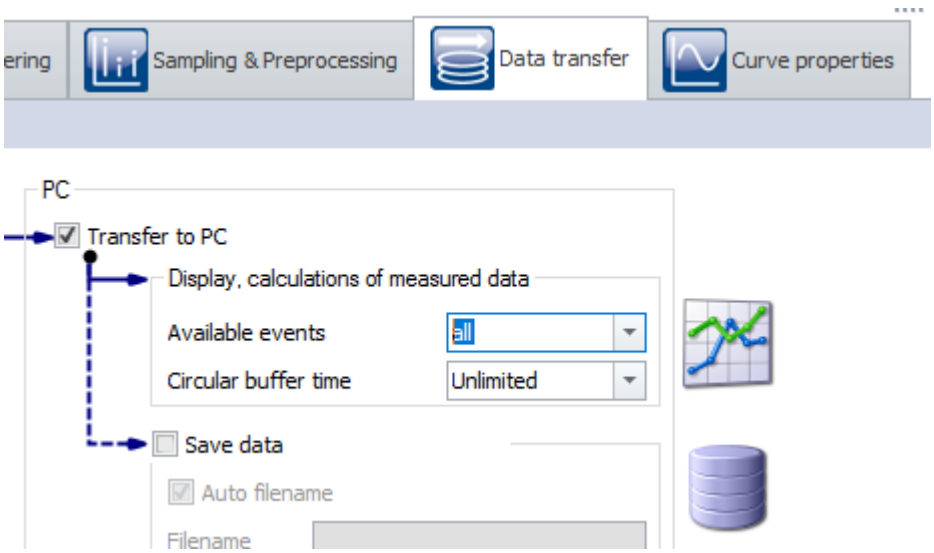
Processing function	Resolution in bits
Normal	8
High Resolution	12
Peak detect*	8
Averaging	*see table below*

* not implemented at this time

Averaging	
Processing values	Resolution in bits
2	8
4, 8	9
16, 32	10
64, 128	11
256, 512, 1024, 2048, 4096	12
8192, 16384, 32768, 65536	12

Settings for the display

In order to display all of the measured data ("shots") in the curve window, the channel's circular buffer memory must be set to *unlimited* and the Available events must be set to *all*.



Data transfer settings

15.4.2 Device script templates

15.4.2.1 DeviceDriver and PushDeviceDriver

Hint

- 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);
      }
   }
}
```

15.4.2.2 FunctionSimulator

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 1413.

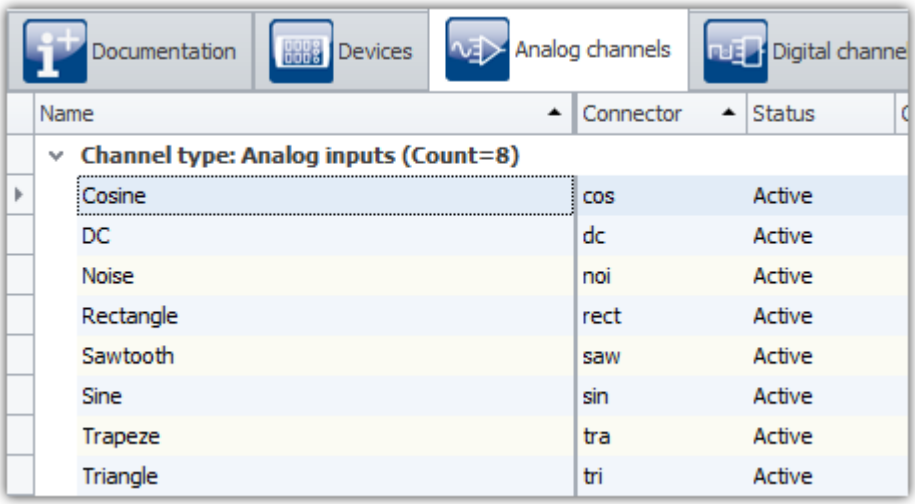
To make it easier to add analog channels, the function `AddFunctionChannelToDevices()` is provided here.

```
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()`.

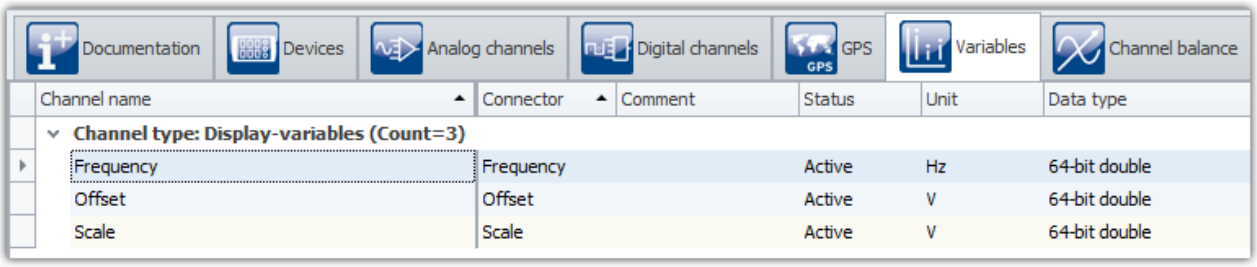
```
private void AddDisplayVariable(SetupNode device, string variableName, string
variableNameShort, string unit) {
   var channel = device.Children.GetOrAdd(ESetupNodeType.Channel, variableName);
   channel.Parameters.UpdateOrAdd(EClassID.eChannelName, value: variableName);
   channel.Parameters.UpdateOrAdd(EClassID.ePlugInName, 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)

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 1424 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 downloading, 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.

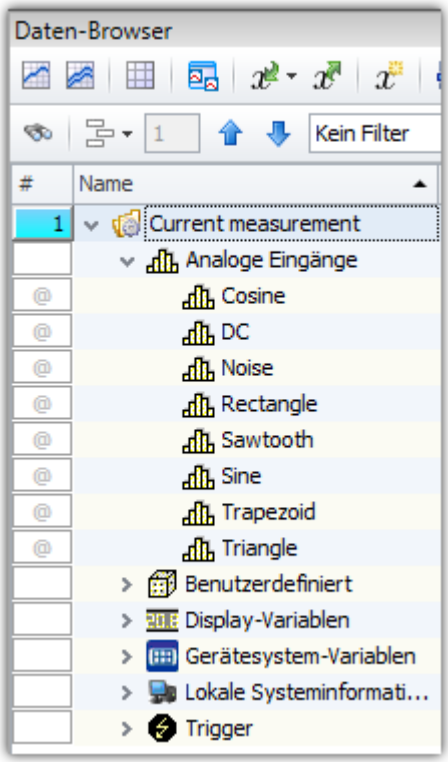
```
public DataTransferDriver() {
   m_processDataTimer = new Timer(OnProcessData, null, Timeout.Infinite, Timeout.Infinite);
}
private void OnProcessData(object state) {
// ...
   // processing Display-variables
   if (dataTransferNode.Name == "Frequency") {
      if (dataReader.ReadData(out frequency) == 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 1424.

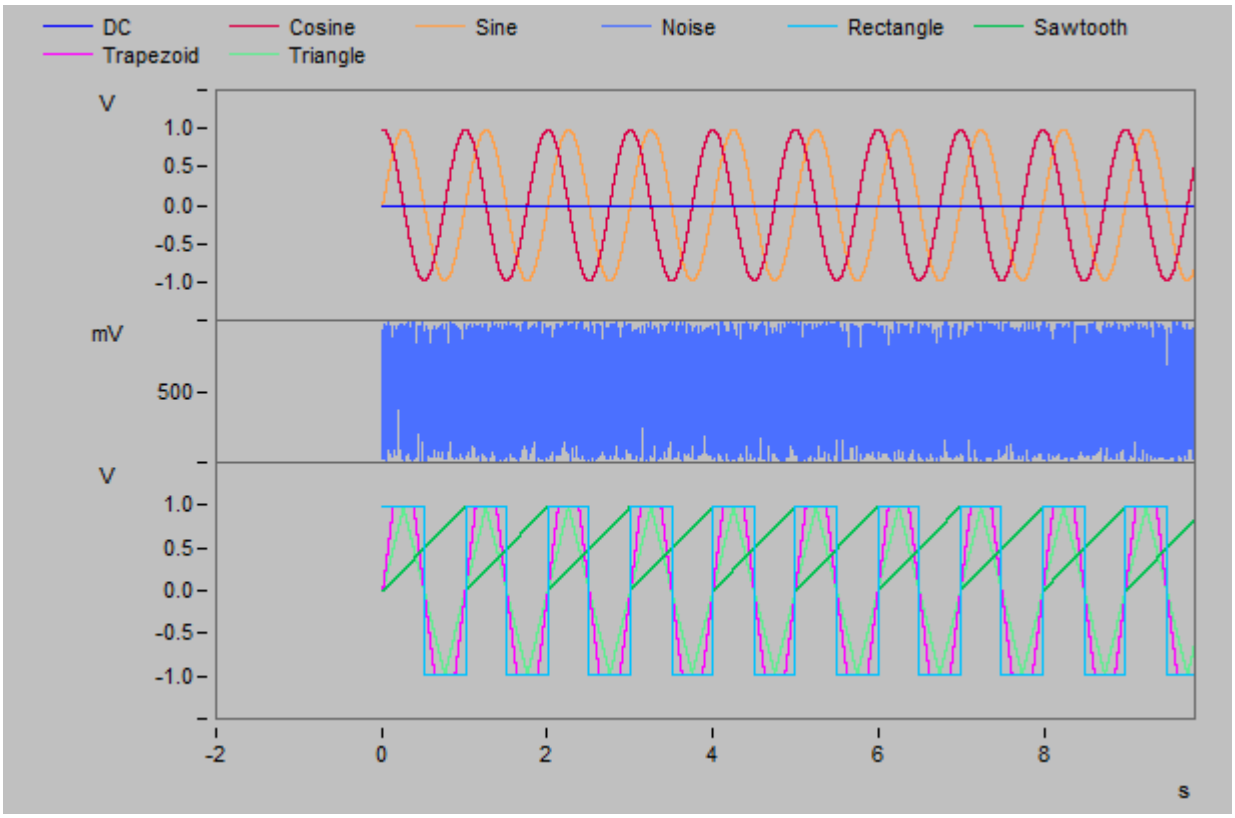
Functionality in imc STUDIO

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.

15.4.2.3 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 1424 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 downloading, 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.

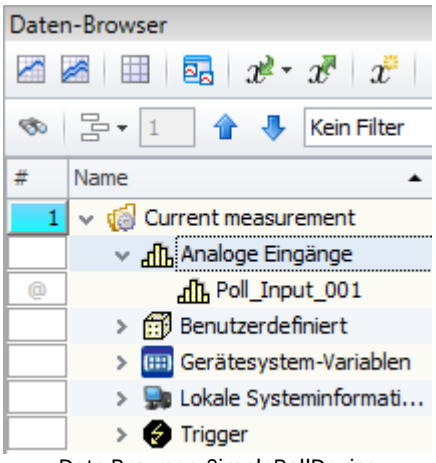
```
public DataTransferDriver() {
   m_processDataTimer = new Timer(OnProcessData, null, Timeout.Infinite, Timeout.Infinite);
}
private void OnProcessData(object state) {
// ...
   double[] data = m_deviceDriver.ReadData(dataTransferNode.Name); // retrieving data from the
device
   EResult result = dataWriter.WriteData(data);
//...
}
```

DeviceDriver.cs

For details, see the chapter DeviceDriver and PushDeviceDriver 1424.

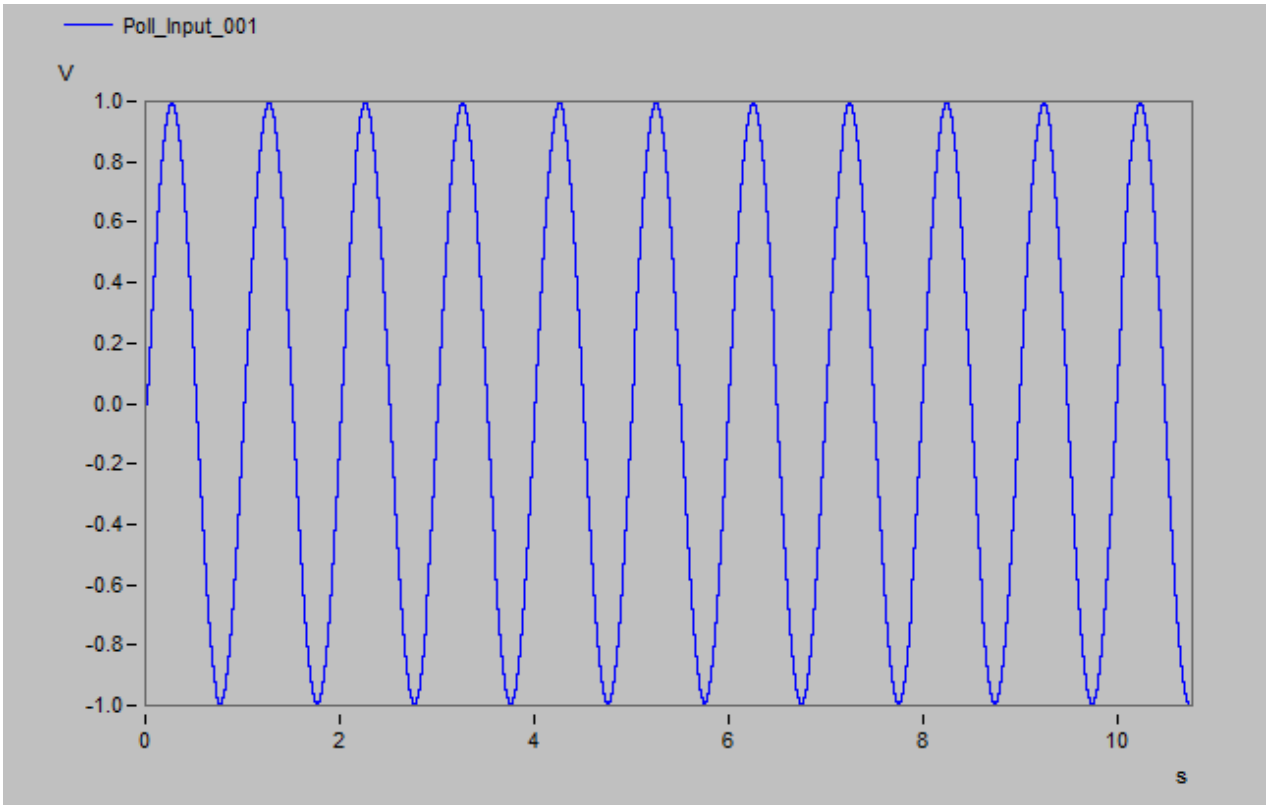
Functionality in imc STUDIO

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

15.4.2.4 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 1424, 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 downloading, 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.

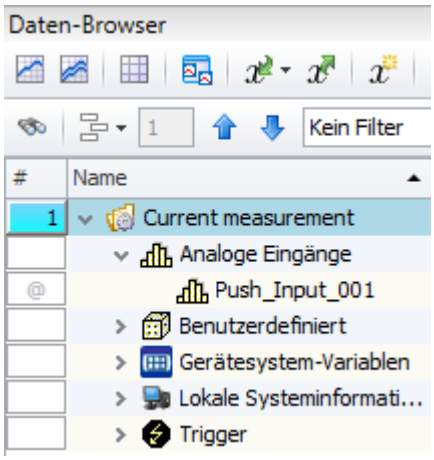
```
private void OnProcessData(string channelName, DataEventArgs e) {
// ...
   switch (e.DataEvent) {
      case DataEvent.Begin:
         dataWriter.Open();
         result = dataWriter.BeginWrite(e.Time.Value);
         // Error-Handling
         break;
      case DataEvent.Data:
         result = 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 1424.

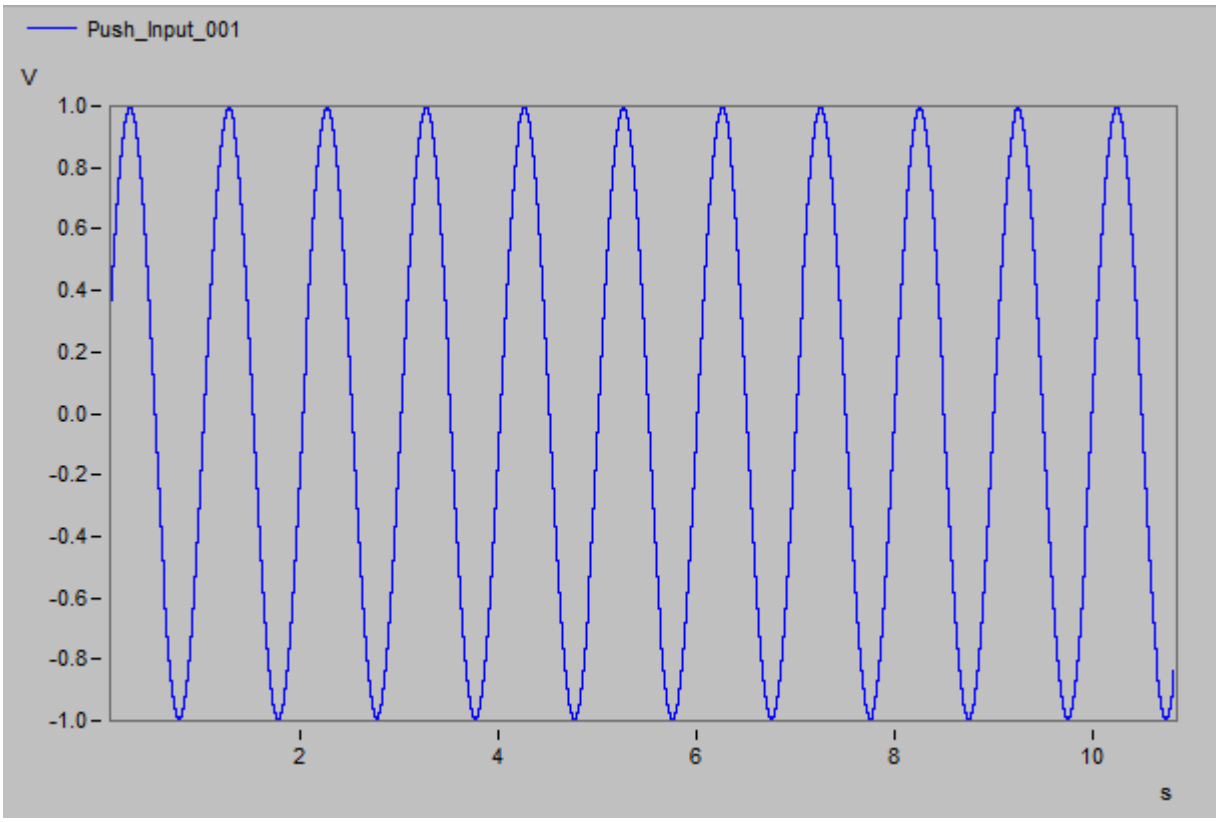
Functionality in imc STUDIO

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

15.5 Tutorial

Im Folgenden finden Sie einige Tutorials:

Tutorial	Beschreibung
Add a channel 1435	A new channel is added to the SimplePollDevice.
Add a Display variable 1437	A Display variable for setting the signal frequency is added to the SimplePollDevice.

15.5.1 Extension of the SimplePollDevice

15.5.1.1 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

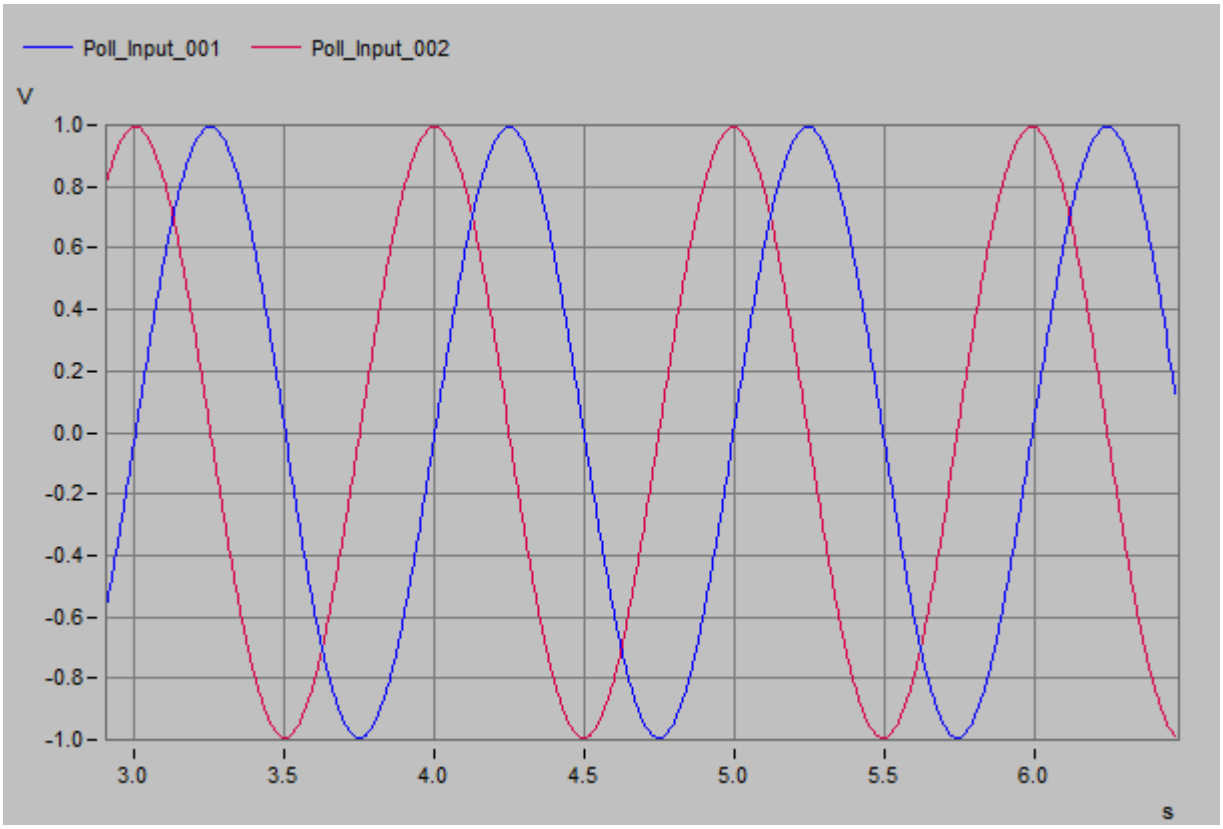
In the class DataTransferDriver, the SimplePollDevice is configured. In the method `Prepare()`, the channels are configured.

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

15.5.1.2 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()`.

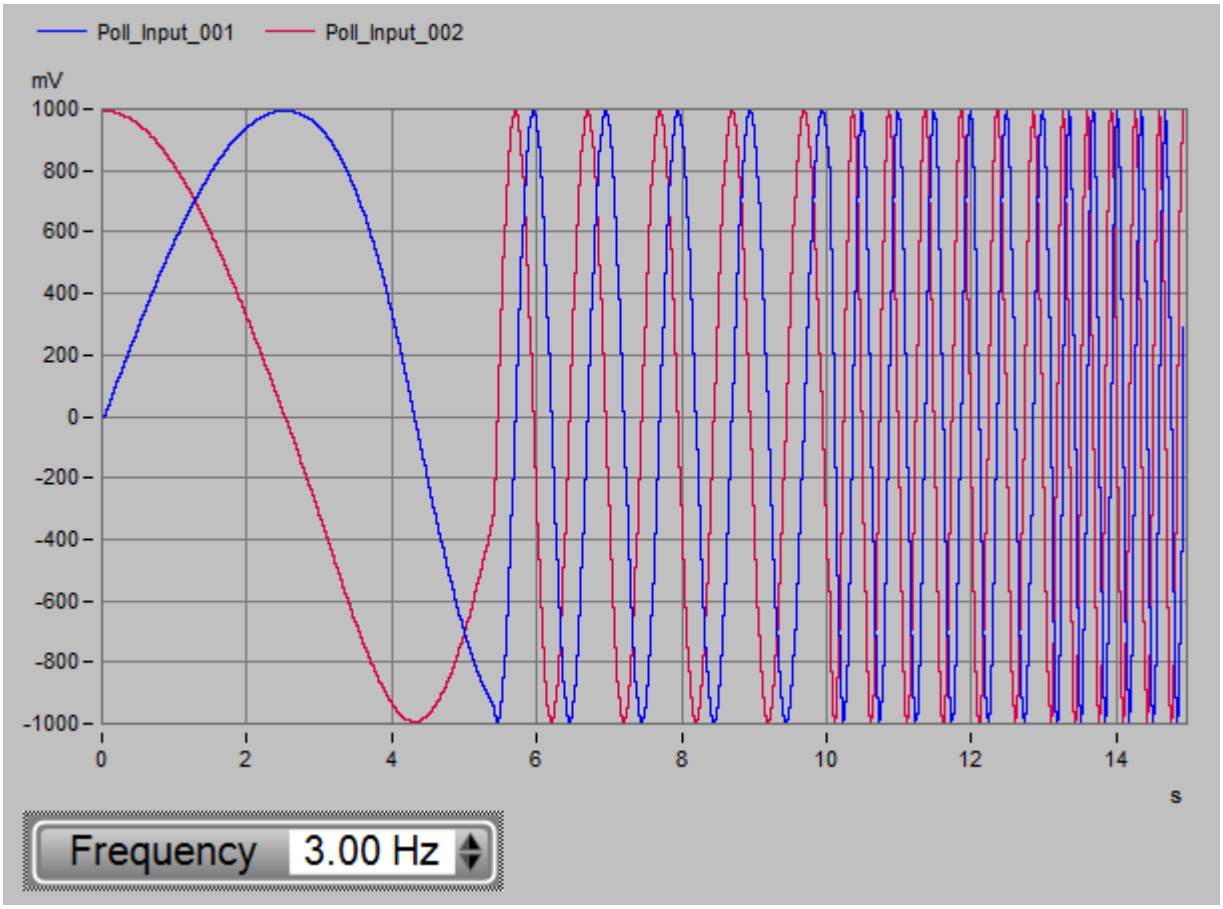
```
if (dataTransferNodes != null) {
   // defining parameters (inserting into existing framework)
```

```
   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 1425.

16 Shared Components

Documentation for components used by multiple imc STUDIO plug-ins.

Component	Description
imc Online/Inline FAMOS Function Reference 1440	Descriptions of all imc Online FAMOS and imc Inline FAMOS functions
Command Reference 1592	Descriptions of all commands.
Command Sequence 1650	Description of the creation and operation of a sequence of commands.
Placeholder 1655	Description of all placeholders.
User-defined variables 1671	Description of the creation of User-defined variables.

Note

These documents can contain references to products not included in your installation. The links to them will not work in the help.

16.1 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.

16.1.1 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 1490.

(^) 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 1490.

(=) 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

a: 1st operand

b: 2nd operand

IsEqual: Result

IsEqual = 1, if both operands are equal, else 0.

Example

```
OnlTriggerMeasure( Trigger_48 )
    Value = CurrentValue( Channel_01, 0, 0.0 )
    If Value = 1
        VirtChannel = Channel_01 + 10
    Else
        VirtChannel = Channel_01 + 5
    End
End
```

(<>) Inequality operator

Inequality-operator. Compares the two operands for inequality.

Prerequisite:
• Only usable with control commands!

IsUnequal = a <> b

a: 1st operand

b: 2nd operand

IsUnequal: Result

IsUnequal = 1, if the two operands are unequal, else 0.

(<) 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

a: 1st operand

b: 2nd operand

IsLess: Result

IsLess = 1, if the first operand is less than the second, else 0.

Example

```
OnlTriggerMeasure( Trigger_48 )
    Value = CurrentValue( Channel_01, 0, 0.0 )
    If Value < 1
        VirtChannel = Channel_01 + 10
    Else
        VirtChannel = Channel_01 + 5
    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

a: 1st operand

b: 2nd operand

IsLessEqual: Result

IsLessEqual = 1, if the first operand is either less than or equal to the second, else 0.

(>) 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

a: 1st operand

b: 2nd operand

IsGreater: Result

IsGreater = 1, if the first operand is greater than the second, else 0.

(>=) 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

a: 1st operand

b: 2nd operand

IsGreaterEqual: Result

IsGreaterEqual = 1, if the first operand is either greater than or equal to the second, else 0.

( ) 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
```

16.1.2 Functions (alphabetic)

16.1.2.1 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 b: Input channel (Signal to be evaluated) FrequencyRating: The signal's frequency rating 1: A-rating 2: B-rating 3: C-rating	TimeConstant: Time constant for averaging In imc Online FAMOS: ≥ 0.0: Time constant is defined by the user in s e.g. 0.125 for FAST-weighting, 1.0 for SLOW-weighting 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 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)

b: Input channel

General information about the Accu* functions

Prerequisite:
• Only available in: imc Inline FAMOS!

AccuLength

Length of entire measurement

Result = AccuLength(Signal)

Result: Results channel

Signal: Input signal

Remarks:

The function determines the length 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 length of the input signal 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)

b: Input channel; in radians

AND

Logical And. Logical conjunction of b and c.

Prerequisite:
• Only usable with control commands!

a = b AND c

b: 1st operand

c: 2nd operand

a = 1, if the first operand is unequal to 0 and the second operand as well, else 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

```
OnlTriggerMeasure( Trigger_48 )
    Value1 = CurrentValue( Channel_01, 0, 0.0 )
    Value2 = CurrentValue( Channel_02, 0, 0.0 )
    If Value1 > 0 AND Value2 > 0
        VirtChannel_01 = Channel_01 + 10
    Else
        VirtChannel_01 = Channel_01 + 5
    End
End
```

Asin

Inverse sine of input channel

a = Asin(b)

b: Input channel; in radians

Atan2

Inverse tangent of b/c

a = Atan2(b, c)

b: Numerator

c: Denominator

AudioBoardThirds

Third-octave-vector: Creates a third-octave vector.

Prerequisite:
• Only available in: imc Online FAMOS!

a = AudioBoardThirds(b)

a: Vector with 33 elements

b: Input channel whose values are to be transformed into a vector

The input channel data are given in the following format:

..... Framebit1 Framebit2 HW1 LW1 HW2 LW2 ... HW33 LW33 .....

Two frame bits are always prefixed to the input's raw data potion. Next come 2 * 33 values as the raw data. A return value is determined from the sequence of HIWORD (HW) and LOWORD (LW) in the raw data.

Each set of 68 input values is used to generate a result vector with 33 elements.

It is not possible to combine data generated by the function with other data formats by, for instance, using the Addition function.

16.1.2.2 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)

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)

16.1.2.3 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 field bus.

Reference

See the example in the chapter: *CAN-Bus Interface* 382 > ... > *Outputting a waveform to a DAC module* 432.

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

The characteristic curve is limited to 50000 values.

ClHistogram

Classification histogram: amplitude time-of-stay procedure

a = ClHistogram(b, MinValue, MaxValue, Total, ResUnit, EndClass)

a: Results b: Input channel 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 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	ResUnits: 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
Hysteresis: Hysteresis for peak identification; ≥ 0.0	1: open
Axes: Axes configuration, x-axis is	Option: Algorithm versions
0: target class or amplitude	0: main algorithm
1: starting class or mean value	
ResType: Result matrix is of type	
0: target/starting class	
1: amplitude/mean	

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( Trigger_48 )
    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	Hyst: Hysteresis; in physical units, ≥ 0
b: Input channel	ResUnit: Unit of the Result
MinValue: Lower limit of range	0: classes
MaxValue: Upper limit of range	1: like input channel
Total: Number of classes; ≥ 2	

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	ResUnit: Result units
b: Input channel	0: classes
c: speed in rpm	1: like input channel
MinValue: Lower limit of range	EndClass: End classes are
MaxValue: Upper limit of range	0: closed
Total: Number of classes; ≥ 1	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	Axes: Axis order, speed is
b: first input channel	0: Z axis
c: Speed in rpm (second input channel)	1: X axis
MinValue1: Lower limit of range	ResUnits: Result units
MaxValue1: Upper limit of range	0: classes
Total1: Number of classes; ≥ 1	1: like input channel
MinValue2: Lower limit of range	EndClasses: End classes are
MaxValue2: Upper limit of range	0: closed
Total2: Number of classes; ≥ 1	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	Total2: Number of classes; ≥ 1
b: first input channel	ResUnit: Result units
c: second input channel	0: classes
MinValue1: Lower limit of range 1	1: like input channel
MaxValue1: Upper limit of range 1	EndClass: End classes are
Total1: Number of classes; ≥ 1	0: closed
MinValue2: Lower limit of range 2	1: open
MaxValue2: Upper limit of range 2	

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 b: Input channel	Delay: delay time "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)

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

b: input channel

Value: single value, which is supplied to the virtual 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_01, 2.5 )
Virt2 = CreateVChannel( Channel_01, Virt_Bit01 )
Virt3 = CreateVChannel( Channel_01, 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

b: input channel

Value: single value, which is supplied to the virtual 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_01, 2 )
VirtInt2 = CreateVChannelInt( Channel_01, Virt_Bit01 )
VirtInt3 = CreateVChannelInt( Channel_01, pv.X123 )
```

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 Channel: Channel from which the value is to be taken	Option: Which value? 0: Last value 1: Minimum 2: Maximum 3: Mean value	Init: Initialization 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( Trigger_48 )
    Value = CurrentValue( Channel_01, 2, 0.0 )
    If Value > 100
        VrtChannel_01 = Channel_01*0 + 100
    Else
        VrtChannel_01 = Channel_01*0
    End
End
```

16.1.2.4 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

```
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( Trigger_48 )
    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.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( Trigger_48 )
    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( Trigger_48 )
    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( Trigger_48 )
    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

b: input channel

Delay: delay time

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

b: Input channel

"Filename": Coefficient vector file

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:

$0\ a_2\ a_1\ b_2\ b_1\ b_0\quad a_2\ a_1\ b_2\ b_1\ b_0\quad \ldots\quad a_2\ a_1\ b_2\ b_1\ b_0$

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).

Zero will be returned in case of failure or absent storage medium.

b: Reserved parameter (set to 0)

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_01 * 0 + GetDiskFreeTime ( 0 )
; returns time duration continuously
```

DiskRunDir

Gets the number of the current directory on the internal storage medium

Prerequisite:
• Only available in: imc Online FAMOS!

a = DiskRunDir(b)

a: Current directory number

b: Reserved parameter (set to 0)

Zero 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

16.1.2.5 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( Trigger_48 )
    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 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( Trigger_48 )
    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 b: Input channel	tau: Time constant for exponential weighting 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_01, 0.125, 1000 )
```

Computes RMS of exponentially weighted window and reduces by 1000.

16.1.2.6 F

FFT

Magnitude spectrum: FFT with output of the magnitude spectrum

a = FFT(b, Window, FFT_length)

a: Result; Magnitude 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 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).

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

Returns averages FFTs

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 50: 50% 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 need not be an exponent of 2; internal interpolation is possible. 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 effective values.

Example

```
Res = FFTAverage( Signal_Vib, 1000, 3, 0, 1, 100 )
```

Calculation of averaged FFTs with a Blackman window and window widths of 1024. Every FFT is factored in and the arithmetic mean of 100 spectra is taken.

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 *Miscellaneous* 792 > *Tuning, Tips and Tricks* > *imc Online FAMOS - Digital Filters* 797.

FiltBP

Processes channel data with a band pass filter

Result = FiltBP(Data, Characteristic, Para, Order, CutoffFreqLow, CutoffFreqUp)

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..40 otherwise: 1..100 CutoffFreqLow: Lower cutoff frequency (Hz) CutoffFreqUp: Upper cutoff frequency (Hz)

FiltBS

Processes channel data with a band stop filter

Result = FiltBS(Data, Characteristic, Para, Order, CutoffFreqLow, CutoffFreqUp)

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..40 otherwise: 1..100 CutoffFreqLow: Lower cutoff frequency (Hz) CutoffFreqUp: Upper cutoff frequency (Hz)

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)

16.1.2.7 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 Ne: input rpm, engine Na: output rpm, live axle	MinNe: minimum input rpm MinNa: minimum output rpm 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 Seconds: current seconds, 0 ... 59 Minute: current minutes, 0 ... 59 Hour: current hours, 0 ... 23	Day: current day, 1 ... 31 Month: current month, 1 ... 12 Year: current year, e.g. 2009

The decimal fraction of seconds is precise to within one 64th of a second.

The function parameters may be local single values, display variables or process vector elements.

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( Trigger_48 )
    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, Field-bus inputs, digital Field-bus ports and digital ports.

If no measurement duration is defined, the value 0 is returned.

Example

```
; measurement duration of physical channel Channel_01 in s
_MeasDuration_01 = GetDuration( Channel_01 )
```

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( Trigger_48 )
    Histo = ClHistogram( Channel_01, -10.0, 10.0, 32, 1, 1 )
    ; 1st histogram value, histogram index: 0
    Histo_0 = Channel_01*0 + GetHistoValue( Histo, 0 )
    ; Last histogram value, histogram index: 31
    Histo_1 = Channel_01*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( Trigger_48 )
    Histo = ClRainFlow( Channel_01, -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_01*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_01*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 Field-bus system, Slot 1 2: Error in Field-bus system, Slot 2 3: Error in Field-bus system, Slot 3 4: Error in Field-bus system, Slot 4 5: Error in Field-bus system, Slot 5 6: Error in Field-bus system, Slot 6 7: Error in Field-bus system, Slot 7 8: Error in Field-bus 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 Field-bus system 1: Errors in the Field-bus system, Slot 1 2: Errors in the Field-bus system, Slot 2 3: Errors in the Field-bus system, Slot 3 4: Errors in the Field-bus system, Slot 4 5: Errors in the Field-bus system, Slot 5 6: Errors in the Field-bus system, Slot 6 7: Errors in the Field-bus system, Slot 7 8: Errors in the Field-bus 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.

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.

Beispiel

```
; 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( Trigger_48 )
    StartTimerPeriodic( 1, 0.1, 0 )
End

OnTriggerMeasure( Trigger_48 )
    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( Trigger_48 )
    sc = GetSampleCount( Channel_01 )
    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 encoder, or field bus inputs, as well as digital and digital field bus ports.

Example

```
_Samplinginterval_01 = GetSamplingTime( Channel_01 )
```

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_01, 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.

16.1.2.8 H

HighLowRatio

The pulse duty ratio of the signal in a window, with reduction

a = HighLowRatio(b, RF)

a: Result

b: Input channel

RF: Reduction factor in samples; ≥ 1

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 1551).

Hyst

Hysteresis: Applies a hysteresis filter to the input channel

a = Hyst(b, Width)

a: Result

b: Input channel

Width: Hysteresis width

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.

16.1.2.9 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

Channel: Input channel

Reset: If this value is not 0, the integral is reset to 0

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_01, 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

Channel: Input channel

Reset: If this value is not 0, the integral is reset to 0

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_01, 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:

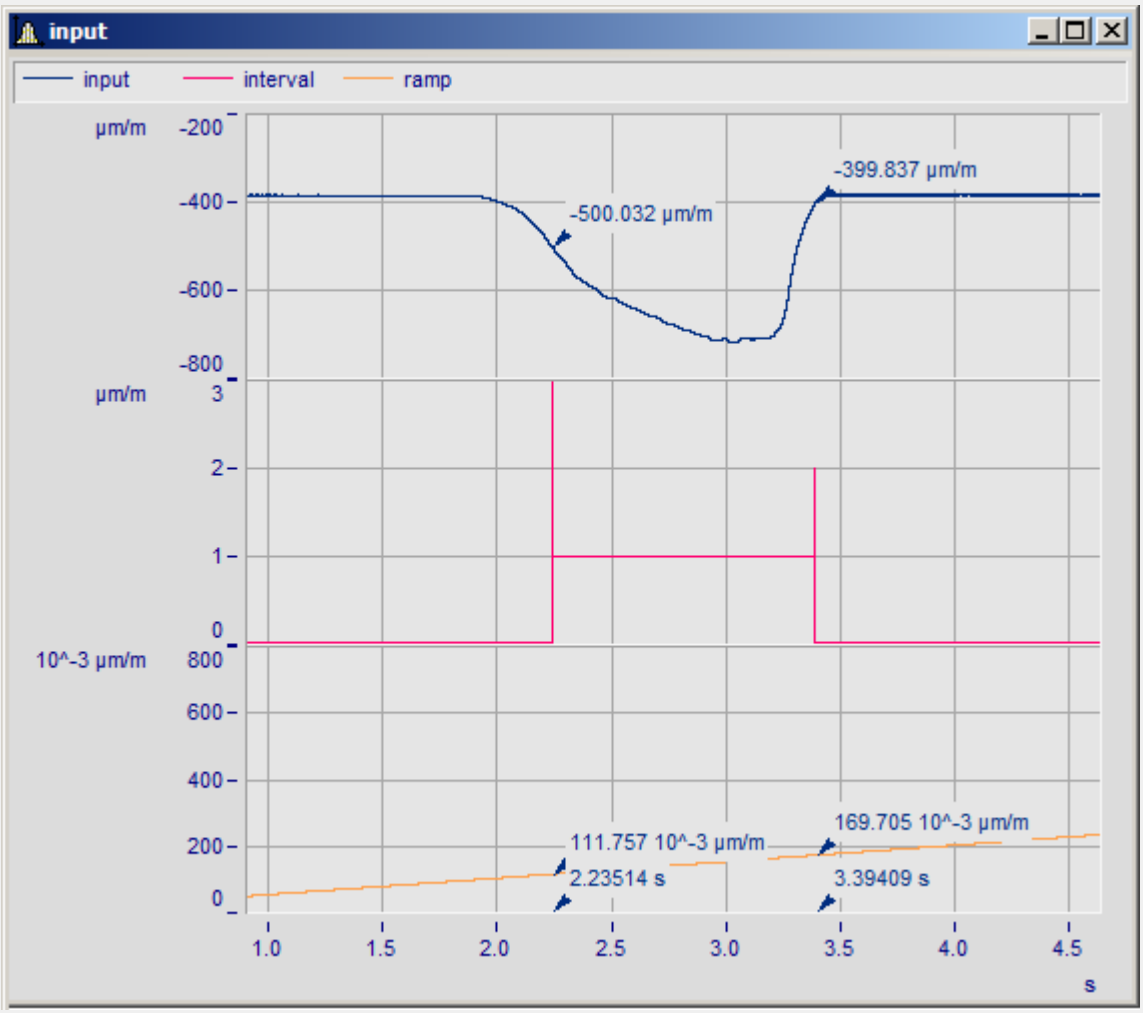
Example

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.

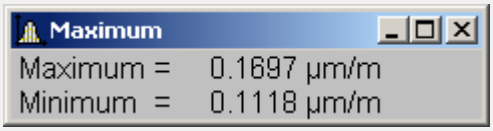
The program for this appears as follows:

```
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.



IntervalFrom1Level

Generates a channel specifying whether the signal's values are within a specifiedinterval.

Result = IntervalFrom1Level(Signal, Level, Levelcode)

Result: results channel Signal: input channel	Level: interval limit 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_01, 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 Signal: input channel	Level1: first interval limit 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 counter 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 Impulse time: Incremental encoder channel set to measurement mode impulse time.	Period duration: Length of period Multiplier: Reserved, always = 0

The input signal comes from an incremental counter 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_01, ivl, 360.0, 60, "Deg", 1 )
```

IntervalMax

Determines the input signal's maximum within the specified interval.

Result = IntervalMax(Signal, Interval)

Result: result channel

Signal: input signal

Intervall: interval channel

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.

Example

```
; Determine min and max of a periodic signal between 20 and 40 degrees.
; the angle is obtained from an incremental counter 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

Signal: input signal

Interval: interval channel

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.

Example

```
; Determine the mean of a squared sine signal
; Theoretical value = 0.5, independent of the interval limit specified by
; IntervalFrom1Level
saw1 = SawTooth(Channel_01, 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

Signal: input signal

Interval: interval channel

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.

Example

```
; Determine min and max of a periodic signal between 20 and 40 degrees.
; the angle is obtained from an incremental counter 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 Intervals: Input channel containing the original interval codes. Sampling interval stated in s.	MinFrequency: Minimal frequency of the pulses contained, stated in Hz. 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_01, 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 counter'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_01, 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.

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_01, 0, 0.01, 628 )
signal1  = Sin( saw1 )
interval = IntervalFrom1Level( saw1, 1.0, 0 )
iRMS1    = IntervalRMS( signal1, interval )
```

IsSyncronized

Determines whether radio clock is synchronized.

Prerequisite:
• Only available in: imc Online FAMOS!

a = IsSyncronized(b)

a: Result	b: clock, only for establishing the data rate.
1: Radio clock synchronizes 0: Radio clock does not synchronize -1: No radio clock present in this device	

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 can be used in the synchronous tasks 1575.

Example

```
; The LED flashes while the signal is larger than 9.
res = Greater( Channel_01, 9 )
LED_01 = JKFlipFlop( res, res )
```

16.1.2.10 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 of b and c.

a = LogAnd(b, c)

a = 1, if b is not equal to zero and c is not equal to zero, else 0

LogNot

Gives the inverse argument b

a = LogNot(b)

LogOr

Logical Or.

a = LogOr(b, c)

a = 1, if b is not equal zero or if c is not equal zero, else 0.

LogXor

Logical And of b and c.

a = LogXor(b, c)

a = 1, if b is inverse to c; else 0.

Lower

Returns the lesser value of the arguments

a = Lower(b, c)

16.1.2.11 M

Max

Maximum of a window, with reduction

a = Max(b, WS, RF)

a: result b: input channel	WS: window size in samples; 1 * RF, 2 * RF, ... , 10 * RF 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) of the reduction factor.

Example

```
high = Max( Channel_01, 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 b: input channel	WS: Window size; in samples; 1 * RF, 2 * RF, ... , 10 * 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 can include 1 to 10 times as many values as the reduction factor.

Example

```
average = Mean( Channel_01, 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 b: input channel	WS: Window size in samples; 1 * RF, 2 * RF, ... , 10 * 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 can include 1 to 10 times as many values as the reduction factor.

Example

```
Low = Min( Channel_01, 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

b: input channel

Duration: Pulse duration in samples; ≥ 1

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 can be used in the synchronous tasks 1575.

MonoflopRT

Retriggerable monoflop

a = MonoflopRT(b, Duration)

a: result

b: input channel

Duration: Pulse duration in samples; ≥ 1

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 can be used in the synchronous tasks 1575.

16.1.2.12 N

NorthCorrection

Correction of north transition in a window with WF samples.

a = NorthCorrection(b, WF)

a: result

b: argument

WF: window size in samples; ≥ 1

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 1572 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 1572 is in a combination as illustrated in the example below. Instead of averaging, the standard deviation, i.e. the function StDev 1550, can be used.

Example

```
NC = NorthCorrection( Channel, 10 )
NC_Mean = Mean( NC, 10, 10 )
Result = WindRoseCorr( NC_Mean )
```

NOT

Negation of a.

Prerequisite:
• Only usable with control commands!

LogicalNot = NOT a

LogicalNot: Result

a: Operand

LogicalNot = 1, if the operand is equal to 0, else 0.

Example

```
OnlTriggerMeasure( Trigger_48 )
    Value = CurrentValue( Channel_01, 0, 0.0 )
    If NOT Value > 0
        VirtChannel_01 = Channel_01 + 5
    Else
        VirtChannel_01 = Channel_01 + 10
    End
End
```

NumberOfPulses

Returns the number of pulses in reduction window

a = NumberOfPulses(b, RF)

a: Result

b: Input channel

RF: Reduction factor in samples; ≥ 1

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 1551).

Example

```
_tmp = Stri( Channel_01, 0, 0 )
pulses = NumberOfPulses( _tmp, 100 )
Returns for every 100th input sample the number of pulses in the last 100 input samples.
```

16.1.2.13 O

OnECUCmdReturn_ECU_001

See ECU-functions 1474.

OR

Logical Or. Logical disjunction of a and b.

Prerequisite:
• Only usable with control commands!

LogicalOr = a OR b

LogicalOr: Result

a: 1st operand

b: 2nd operand

LogicalOr = 1, if the first operand is unequal to 0 or the second operand is unequal to 0, else LogicalOr = 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

```
OnlTriggerMeasure( Trigger_48 )
    Value1 = CurrentValue( Channel_01, 0, 0.0 )
    Value2 = CurrentValue( Channel_02, 0, 0.0 )
    If Value1 > 0 OR Value2 > 0
        VirtChannel_01 = Channel_01 + 10
    Else
        VirtChannel_01 = Channel_01 + 5
    End
End
```

General information about the Otr functions

Prerequisite:
• Only available in: imc Online FAMOS!

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 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_01, 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 encoder 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 encoder 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.

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.

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.

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 2

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 VibrationSignal: Signal with sinusoidal oscillation PeriodLength: Number of samples per period; ≥ 2.0 PeriodAmount: The number of periods over which an average is taken	Option: Determines what is calculated 0: Magnitude (effective value) 1: Phase (in degrees)

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 encoder 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 encoder 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.

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 encoder 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

InputSignal: Signal from Incremental Encoder

DxResult: Delta-X for the result channel, > 0

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 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 encoder 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 encoder 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.

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.

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.

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.

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.

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 encoder 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 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, corrected by -0.1°, is passed to the function OtrResampleAngle, together with the periodic measurement signal on Channel_01. 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_01, 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_01, 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 encoder 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 encoder 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.

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.

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 encoder 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 Vibration: Time history of vibration signal RPM: Time history of RPMs. Scaled in revs/min	Order3dB: Order at which the low-pas filter dampens the signal by 3dB FilterOrder: The low-pass filter order

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%.

16.1.2.14 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* 782.

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: *Application Notes* > *Power Measurement* 782.

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: *Application Notes* > *Power Measurement* 782.

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 1551).

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

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 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 1551).

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 b: Input channel 1	c: Input channel 2 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 1551).

16.1.2.15 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_01
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

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.

There must be at least one source channel for Trigger 48; acquisition continues only as long as the channels for Trigger 48 are are taking measurements.

Example

```
RecordEvent( Greater( Channel_01, 9 ), "High water mark passed" )
```

If the level on Channel_01 exceeds 9, this is recorded as an event.

RecordText

Return text: The specified text is output.

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). Otherwise, the function RecordText has the same constraints as the function RecordEvent: there must be at least one source channel for Trigger 48, acquisition continues only as long as the channels for Trigger 48 are are taking measurements.

Red

Reduction: reduces the number of values in the input channel

a = Red(b, RF)

a: Result

b: Input channel

RF: Reduction factor in samples; ≥ 1

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_01, 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.

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

Signal: Input channel

Count: Number of values to replace

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.

Beispiel

```
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

Signal: Input channel

Count: Number of values to replace

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.

Beispiel

```
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

b: Vectorized input channel

VectorLen: Length of result vectors

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 process vector element can be restored in the device automatically if the device is prepared for a measurement.

For instance, a process vector element'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: *Process Vector* > *Saving/Restoring Process Vector Variables* 763.

Example

```
OnInitAll
    Int Restore pv.x = 0  ; for Integer process vector elements
    Restore pv.y = 0.0    ; for Float process vector elements
End
```

With the syntax `Restore pv.x = 0`, the process vector 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; 1 * RF, 2 * RF, ... , 10 * 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 can be an integer multiple between 1 and 10 of the reduction factor

Example

```
eff = RMS( Channel_01, 50, 10 )
```

Returns for every tenth input value the RMS-value of the last 50 input values.

Note

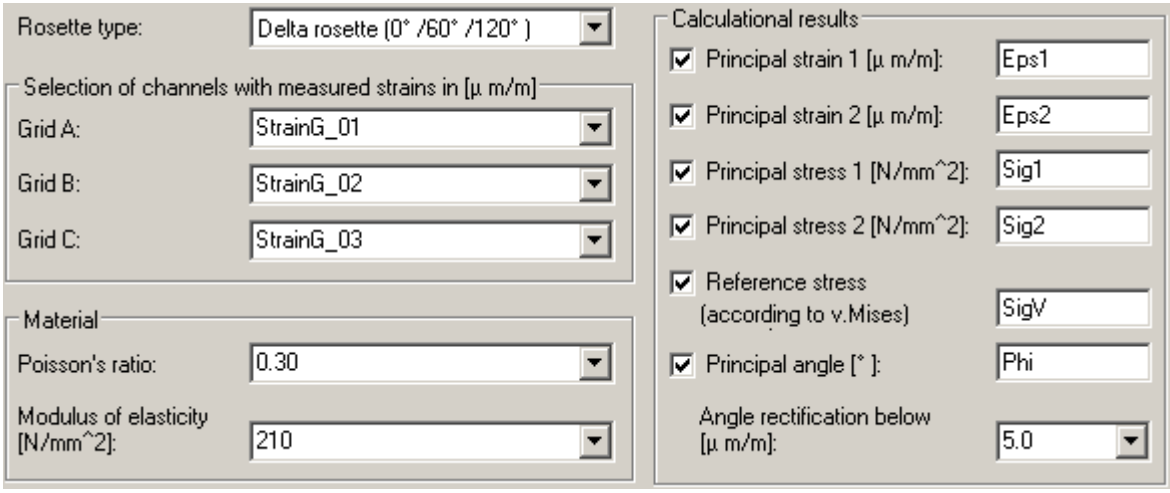
If the window size is not 1*RF, then there aren't enough values available initially. In that case, the available values are taken as the calculation basis.

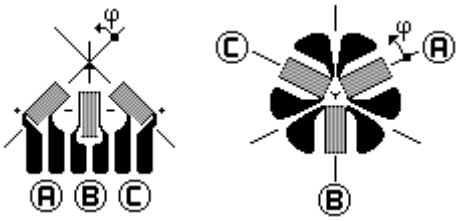
Rosette1

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_v [N/mm^2] Phi: Result Grid A angle $\emptyset_{P,Q}$ [°] Gitter_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 ν EModul: Modulus of elasticity in GPa ν 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°:

$$\varepsilon_{1,2} = \frac{\varepsilon_a + \varepsilon_b + \varepsilon_c}{3} \pm \sqrt{\left(\frac{2\varepsilon_a - \varepsilon_b - \varepsilon_c}{3}\right)^2 + \frac{1}{3}(\varepsilon_b - \varepsilon_c)^2}$$

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°:

$$\emptyset_{P,Q} = \frac{1}{2}\arctan\left(\frac{2\varepsilon_\mathrm{b} - \varepsilon_\mathrm{a} - \varepsilon_\mathrm{c}}{\varepsilon_\mathrm{a} - \varepsilon_\mathrm{c}}\right)$$

Angle grid at 60°:

$$\emptyset_{P,Q} = \frac{1}{2}\arctan\left(\frac{\sqrt{3}(\varepsilon_\mathrm{b} - \varepsilon_\mathrm{c})}{2\,\varepsilon_\mathrm{a} - \varepsilon_\mathrm{b} - \varepsilon_\mathrm{c}}\right)$$

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 1540

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)

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 can be used in the synchronous tasks 1575.

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_01, 6 ), Greater( Channel_01, 0.5 ))
_S=  LogOr( Greater( Channel_01, 8), Less( Channel_01, 0))
LED_01 = RSFlipFlop( _R, _S)
```

RunAutoBalance

Function for calling automatic adjustment of the selected channels

RunAutoBalance()

RunAutoShuntCalibration

Function for calling the automatic shunt calibration of the selected channels

RunAutoShuntCalibration()

16.1.2.16 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 b: input channel	c: control 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 b: Input channel y0: Start value	dy: Increment P: Cycle count in samples

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* 409). 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_01, Channel_02 )
```

or

```
SendMessage_Message _001( 0, 0.1, Channel_01, Channel_02 )
```

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( Trigger_48, 1000 )
    int VChanInt = SingleValueChannel( Trigger_48, 1000 )
End

OnTriggerMeasure( Trigger_48 )
    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( Trigger_48, 1000 )
End

OnTriggerMeasure( Trigger_48 )
    If Channel_01 > 100
        MinMax = 100
    End
    If Channel_01 < 5
        MinMax = 5
    End
End
```

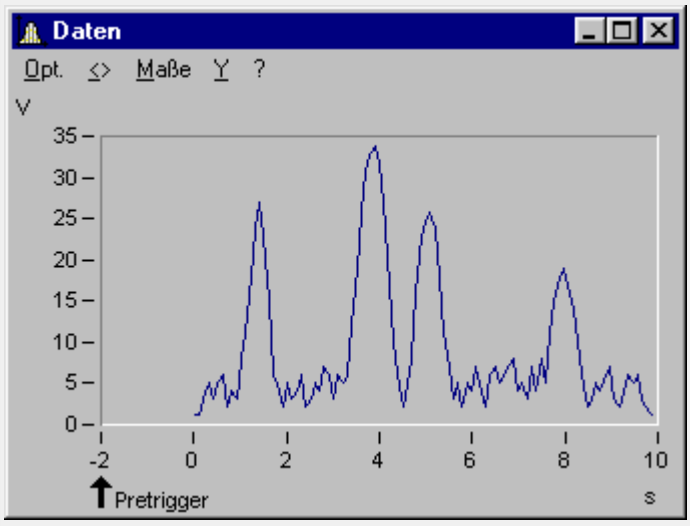
The virtual channel created is filled according to assignments of single values. Whenever an assignment is carried out, this single value is added to the virtual channel as a sample value. Thus it can happen that the virtual channel contains no samples. On the other hand, a large number of samples can 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.

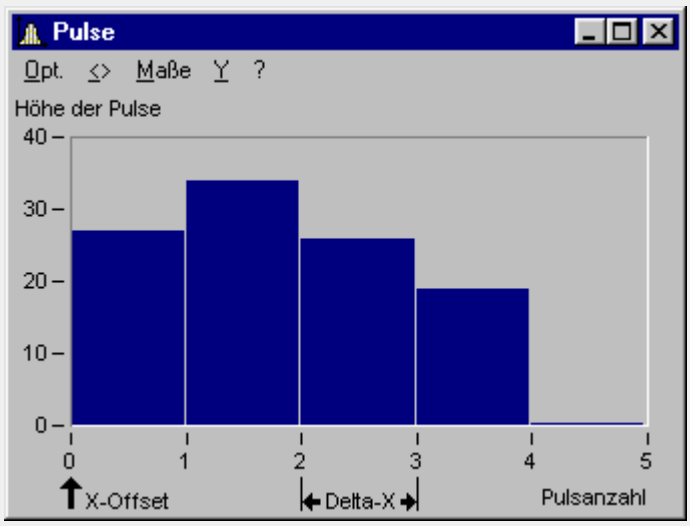
By default, the clock rate is set to 1s and no units are displayed in the curve window.

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

Signal: Input channel

Count: Number of values to skip

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` und `ReplaceFirstValuesN`, the values during the transients' phase are skipped instead of replaced.

Beispiel

```
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

b: Input channel

Max: Maximum slope

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 Signal: signal to be evaluated FrequencyRating: signal's frequency rating 0: no rating 1: A-rating 2: B-rating 3: C-rating	TimeConstant: Time constant for averaging In imc Online FAMOS: ≥ 0.0: Time constant is defined by the user in s e.g. 0.125 for FAST-weighting, 1.0 for SLOW-weighting 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 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.

Requirement:
• 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 b: Input channel	WS: Window size in samples; 1 * RF, 2 * RF, ... , 10 * 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*RF.

Example

```
erg = StDev( Channel_01, 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 b: Input channel	S_upper: Upper threshold 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_01, 4, 1 )*5
```

Sum

Sum of input channel with reduction.

a = Sum(b, RF)

a: Result

b: Input channel

RF: Reduction factor in samples; ≥ 1

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 )
```

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 InputChannel: Channels whose values are to be summated.	Reduction: Reduction factor 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_01, 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, process vector elements 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
```

16.1.2.17 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 ( Trigger_48 )
    ...
    If IsText
        Text3 = Text1 + TextFormatH( Number, 8 )
        RecordText( Text3 )
        ; The text "Number = 0x000003E8" is returned
        IsText = 0
    End
    RecordEvent( Channel_01, "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

Number: Real number to be converted to text

DecPlace: Number of digits after the decimal

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 ( Trigger_48 )
    ...
    If IsText
        Text3 = Text1 + TextFormatE( Number, 6 )
        RecordText( Text3 )
        ; The text "Number = 1.000000E03" is returned
        IsText = 0
    End
    RecordEvent( Channel_01, "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

Number: Real number to be converted to text

DecPlace: Number of digits after the decimal

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 ( Trigger_48 )
    ...
    If IsText
        Text3 = Text1 + TextFormatF( Number, 6 )
        RecordText( Text3 )
        ; The text "Number = 1.000000" is returned
        IsText = 0
    End
    RecordEvent( Channel_01, "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

Number: Number to be converted to text

Places: Number of decimal places

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 ( Trigger_48 )
    ...
    If IsText
        Text3 = Text1 + TextFormatH( Number, 8 )
        RecordText( Text3 )
        ; The text "Number = 0x000003E8" is returned
        IsText = 0
    End
    RecordEvent( Channel_01, "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 ( Trigger_48 )
    ...
    If IsText
        Text3 = Text1 + TextFormatI( Number )
        RecordText( Text3 )
        ; The text "Number = 1000" is returned
        IsText = 0
    End
    RecordEvent( Channel_01, "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 b: Input (channel or single values)	min: Minimum max: Maximum

TransRec

Transitional Recording: Data reduction for analog data

a = TransRec(b, Precision)

a: result

b: input channel

Precision: Precision in units of the argument

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.

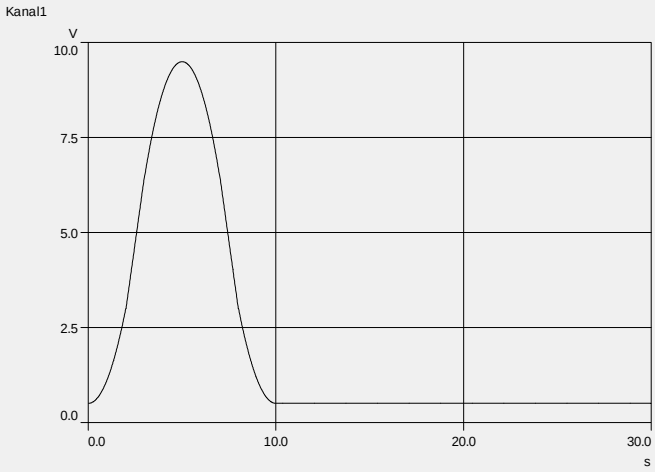
This means that rapid signal movements are sampled at a high sampling rate. However slow-paced signal movements are sampled at as few points as will still allow the signal to be reconstructed.

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.

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.

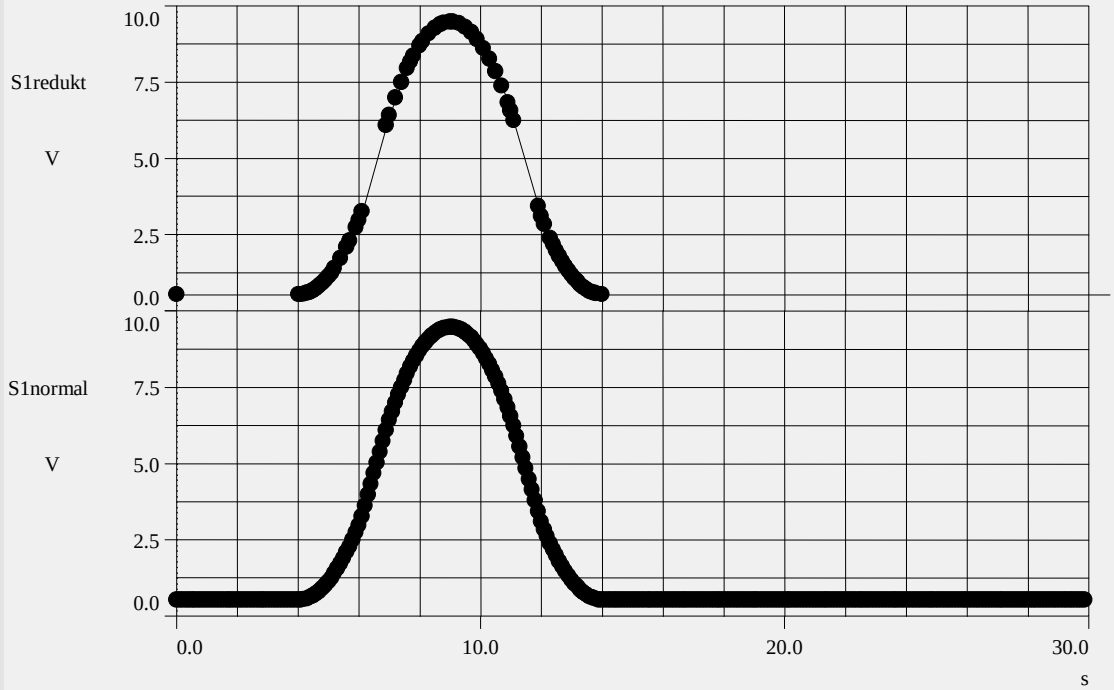


In this example, the signal is recorded once with and once without TransRec and saved.

As tolerance threshold set for the data reduction is 10mV.

S1reduct is measured with data reduction

S1normal is measured without data reduction



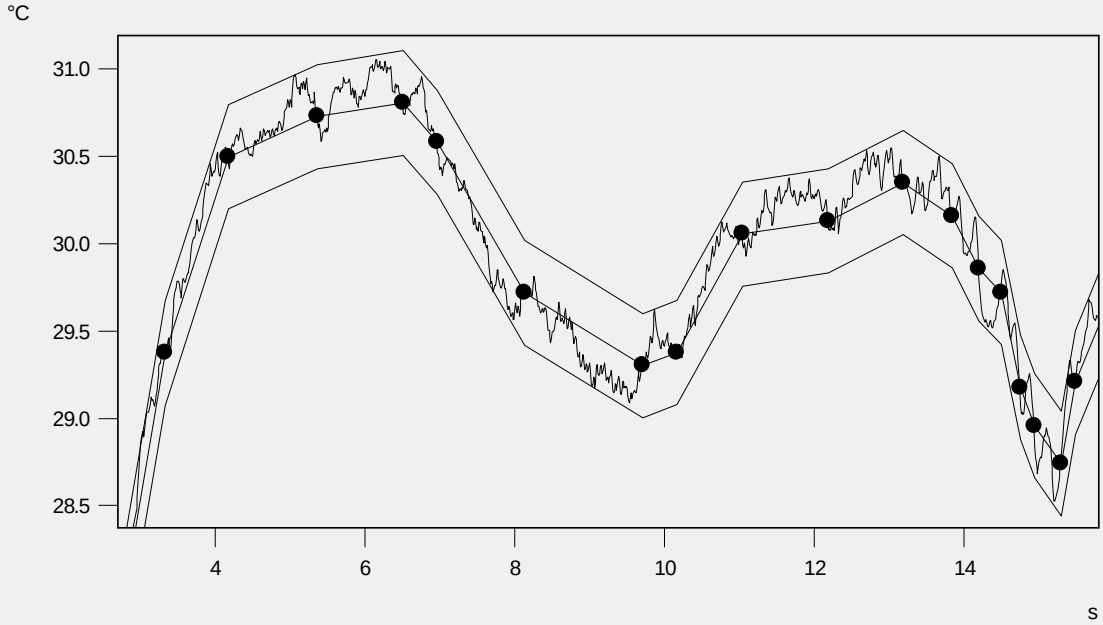
The curves are displayed on top of each other. The filled circles represent sample values, the lines between them are interpolations

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_01 - Channel_02
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.

16.1.2.18 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_01 ), -100 )
```

The input channel is converted into decibel and restricted to a lower border of -100 dB.

16.1.2.19 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, generating virtual channels from vectors

```
OnInitAll
    int v1 = VectorChannel( Trigger_48, 1000, 3 ) ; vector with integers
    ; v1[1] = 0, v1[2] = 0, v1[3] = 0  happens automatically
    v1[3] = 2

    v2 = VectorChannel( Trigger_48, 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.

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

VectorChannelSet

Writing a vector: A vector is written to a segmented virtual channel.

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 *Setup-Configuration* > *Supplemental files*.

Note

The vector data is limited to 50000 values.

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 b: Input channel	VLength: Number of samples per vector; > 1 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 b: input channel	VLength: Length of a vector; > 1 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_01, 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

Trigger: Trigger to which the vector is to be assigned

Element count: Number of the vector's elements

A static vector is declared using the function VectorStatic. All elements are initialized at zero. 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( Trigger_48, 3 ) ; vector with integers
    ; vs1[1] = 0, vs1[2] = 0, vs1[3] = 0  performed automatically

    vs2 = VectorStatic( Trigger_48, 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_01
    vs2[2] = pv.Channel_02
    vs2[3] = pv.Channel_03
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
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]
```

Notes on assignment

The first element is addressed with the value "1". The second with "2", etc.

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.

With TimeConstant = 0, there is no subsequent time weighting. The reduction factor in this case must 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. Permitted signal sampling intervals: 0.2ms...2ms corresponding to 500 Hz...5kHz
11	Wd, x and y directions and for horizontal recumbent direction Permitted signal sampling intervals: 0.2ms...2ms corresponding to 500 Hz...5kHz
12	Wf, Motion sickness Permitted signal sampling intervals: 5ms...500ms corresponding to 2Hz...200Hz
13	Wc, Seat-back measurement Permitted signal sampling intervals: 0.2ms...2ms corresponding to 500 Hz...5kHz
14	We, Measurement of rotational vibration Permitted signal sampling intervals: 0.2ms...2ms corresponding to 500 Hz...5kHz
15	Wj, Vibration under the head of a recumbent person Permitted signal sampling intervals: 0.2ms...2ms corresponding to 500 Hz...5kHz

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. Permitted signal sampling intervals: 0.2ms...2ms corresponding to 500 Hz...5kHz
17	Hz, Whole-body vibrations, standing, seated position, measurement direction: z Permitted signal sampling intervals: 0.2ms...2ms corresponding to 500 Hz...5kHz
18	HxI, Whole-body vibrations, recumbent position, measurement direction: x Permitted signal sampling intervals: 0.2ms...2ms corresponding to 500 Hz...5kHz
19	Hb, Whole-body vibrations, body position not specified Permitted signal sampling intervals: 0.2ms...2ms corresponding to 500 Hz...5kHz
20	Hh, Hand transmitted vibration, for all measurement directions Permitted signal sampling intervals: 0.02ms...0.2ms corresponding to 5kHz...50kHz

per ISO 7505, 1st edition, 1986-05-01:	
20	hand transmitted vibration, weighting filter. Permitted signal sampling intervals: 0.02ms...0.2ms corresponding to 5kHz...50kHz

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_01, 512, 0 )
_tmpV2 = VectorizeAndSkip( Channel_02, 512, 0 )
a = VIsAnyGreater(_tmpV1, _tmpV2 )
```

a: 1 when a sample of Channel_01 larger than Channel_02

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

This function compares the elements in a series of vectors and finds the mean value of the elements.

I.e. the mean of all of the 1st elements is found followed by the mean of all of the 2nd elements etc.

The resulting mean values are assigned to a vector of the same dimension as the original vectors.

Over time, these too are grouped into a series of mean value vectors.

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

This function compares the elements in a series of vectors and finds the minimum elements.

I.e. the minimum of all of the 1st elements is found, followed by the minimum of all of the 2nd elements etc.

The resulting minimum values are assigned to a vector of the same dimension as the original vectors.

Over time, these too are grouped into a series of minimum vectors.

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 b: Input vector series	XValue: X-value 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 b: Input vector series	YValue: Y-value to find ErrorValue: Error value to return if y-value not found

16.1.2.20 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 1505 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 1505 and WindRoseCorr is in a combination as illustrated in the example below. Instead of averaging, the standard deviation, i.e. the function StDev 1550, can be used.

Example

```
NC = NorthCorrection( Channel, 10 )
NC_Mean = Mean( NC, 10, 10 )
Result = WindRoseCorr( NC_Mean )
```

16.1.3 Functions with Control Commands

Prerequisite for all functions:
• Only usable with control commands!

Function Groups	Operators and Symbols	Additional functions
Measurement Process States 1573	(=) Equal Operator (Comparison) 1442	AND 1449
Loops, Conditions 1579	(<>) Inequality operator 1442	CloseSaveInterval 1455
Timer Functions 1582	(<) Less than operator 1443	CurrentValue 1463
CanMsg 1584	(<=) Less than or equal operator 1443	DelayBuffer* 1464
Controller Functions 1588	(>) Greater than operator 1443	ECU* 1473
	(>=) Greater than or equal operator 1444	GetDateTime 1482
		NOT 1506
		OnECUCmdReturn_ECU_* 1474
		OR 1507
		ReadyForPowerOff 1535
		SyncOverload 1553
		VectorChannel 1564
		VectorChannelSet 1564
		VectorStatic 1566

16.1.3.1 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.

From version imc STUDIO 3.1R1, loops and conditions are supported too.

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
```

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 1553.

Reference

For more information on controller see in the documentation on imc Online FAMOS:
imc Online FAMOS Professional > *PID-Controller* 785.

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( Trigger_48 )
    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( Trigger_48 )
    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( Trigger_48 )
    VirtChannel_01 = Channel_01 + 5
    If VirtBit_01 > 0
        VirtChannel_01 = Channel_02 + 10
    Else
        VirtChannel_01 = Channel_02 + 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 1582).

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( Trigger_48 )
    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_01
End

OnPowerOff( 0.01 ) ; Cycle time: 0.01s
    DisplayVar_01 = 0
    DOut01_Bit01  = 0
    DOut01_Bit02  = 0
    ReadyForPowerOff()
End
```

16.1.3.2 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( Trigger_48 )
    If VirtBit_01 > 0
        VirtChannel_01 = Channel_01 + 1
    End
    If VirtBit_05 = 0
        VirtChannel_02 = Channel_02 + 5
    Else
        VirtChannel_02 = Channel_02 + 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.

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".

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( Trigger_48 )
    Value = CurrentValue( Channel_01, 0, 0.0 )
    Switch Value
        Case 1
            VirtChannel_01 = Channel_01 + 1
            VirtChannel_02 = Channel_02 + 1
        End
        Case 2
            VirtChannel_01 = Channel_01 + 10
            VirtChannel_02 = Channel_02 + 2
        End
        Default
            VirtChannel_01 = Channel_01 + 100
            VirtChannel_02 = Channel_02 + 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( Trigger_48 )
    Histo = Histogramm( Channel_01,  -10, 10, 16, 1, 1 )
    HistoSum = 0
    For i = 0 Till MaxLoops Step 1
        HistoSum = HistoSum + GetHistoValue( Histo, i )
    End
    VrtSum = Channel_01*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( Trigger_48 )
    Histo = ClHistogram( Channel_01,-10,10,16,1,1)
    HistoSum = 0
    i = 0
    While i <= 15
        HistoSum = HistoSum + GetHistoValue( Histo, i )
        i = i + 1
    End
    VrtSum = Channel_01*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( Trigger_48 )
    Value = CurrentValue( Channel_01, 0, 0.0 )
    Switch Value
        Case 1
            VirtChannel_01 = Channel_01 + 1
            VirtChannel_02 = Channel_02 + 1
        End
        Case 2
            VirtChannel_01 = Channel_01 + 10
            VirtChannel_02 = Channel_02 + 2
        End
        Default
            VirtChannel_01 = Channel_01 + 100
            VirtChannel_02 = Channel_02 + 3
        End
    End
End
```

16.1.3.3 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( Trigger_48 )
    ; 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( Trigger_48 )
    ; 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( Trigger_48 )
    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
```

16.1.3.4 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( Trigger_48 )
    VID    = Channel_01*0 + CanID
    VLen   = Channel_01*0 + CanLen
    VByte0 = Channel_01*0 + CanByte0
    VByte1 = Channel_01*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( Trigger_48 )
    VID    = Channel_01*0 + CanID
    VLen   = Channel_01*0 + CanLen
    VByte0 = Channel_01*0 + CanByte0
    VByte1 = Channel_01*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 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( Trigger_48 )
    VID    = Channel_01*0 + CanID
    VLen   = Channel_01*0 + CanLen
    VByte0 = Channel_01*0 + CanByte0
    VByte1 = Channel_01*0 + CanByte1
End
```

16.1.3.5 Controller Functions

Prerequisite:
• Only available in: imc Online FAMOS Professional!

Reference

For more information on controller see in the documentation on imc Online FAMOS: *imc Online FAMOS Professional* > *PID-Controller* 785.

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 P_component: P-component	I_component: I-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 Hysteresis: The distance between two reversal points	Output_Inversion: Should the output be inverted? 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.

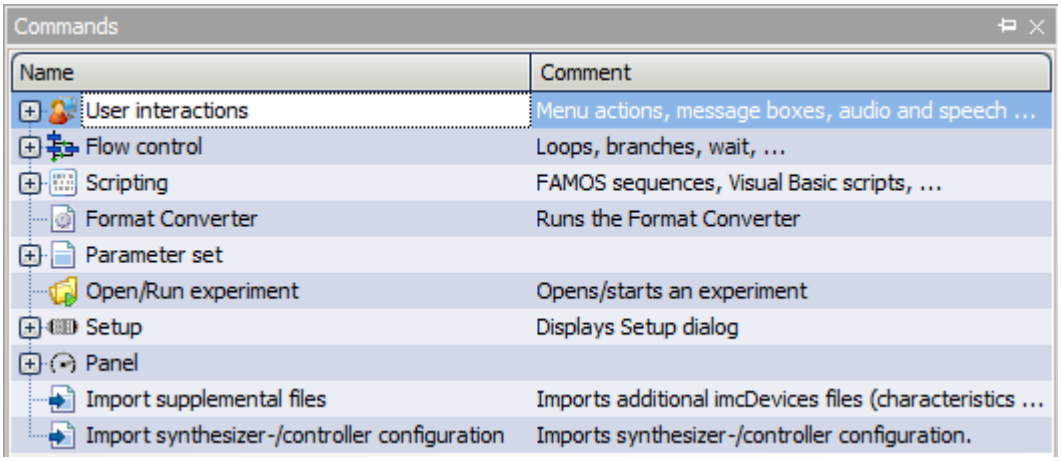
When the controller's parameters are changed, these changes only take effect upon the next call of the function .Calc(), because that is the first (and only) time a new controller output is computed. The function .Calc() is only permitted in synchronous tasks.

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
```

16.2 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.



Tool window Commands

Commands are available in various areas of imc STUDIO:

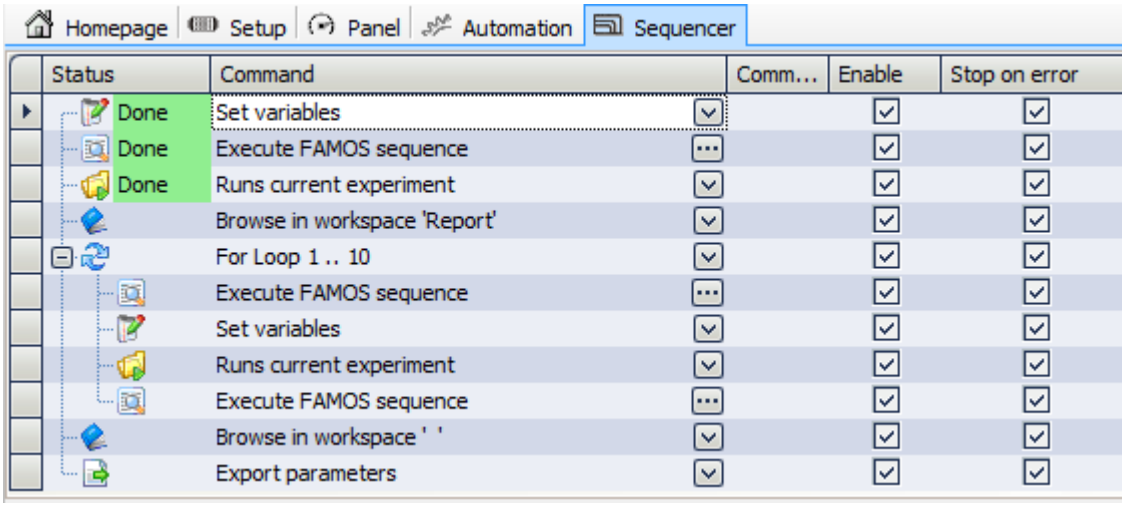
- imc STUDIO Sequencer
- imc STUDIO Panel
- imc STUDIO Automation

Note

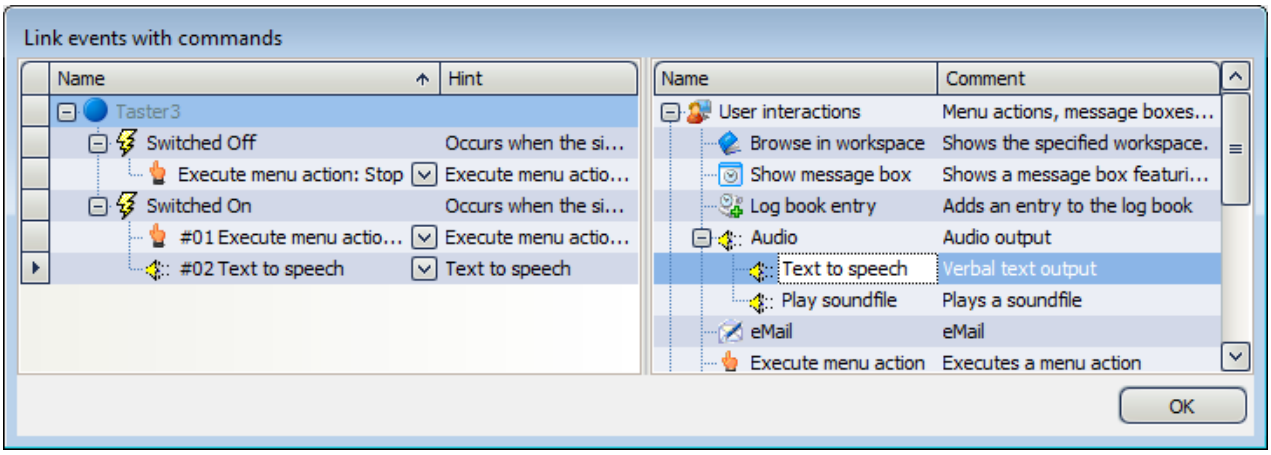
Not all commands described here are available in all areas mentioned, since not all of them are relevant to every area. The commands respectively available are displayed in the associated command list.

In automations, commands are called either from the Sequencer or the Automation.

Manually, commands are linked with Widgets on Panel pages.



Commands in the Sequence Table of the plug-in Sequencer

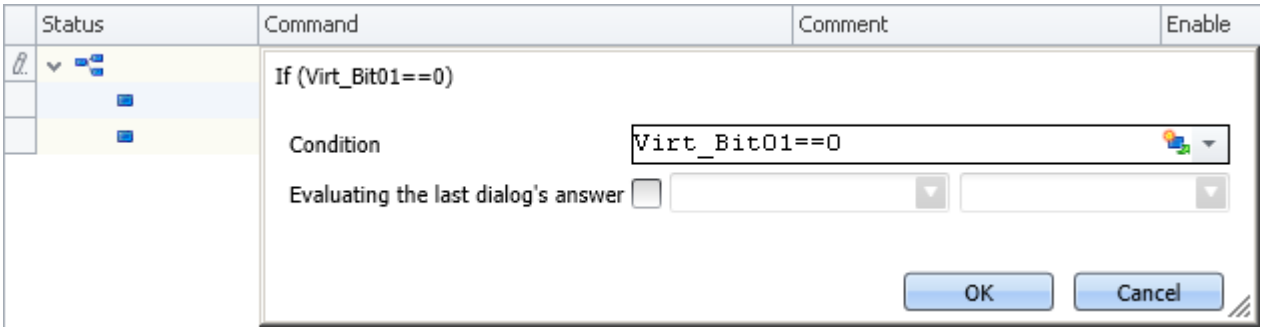


Commands in the Sequence Table of the "Event Dialog" belonging to the plug-in Panel
Linked with events belonging to a Panel control

16.2.1 Flow control

16.2.1.1 If

If enables the routine to branch off in response to *If* and *Else* instructions.

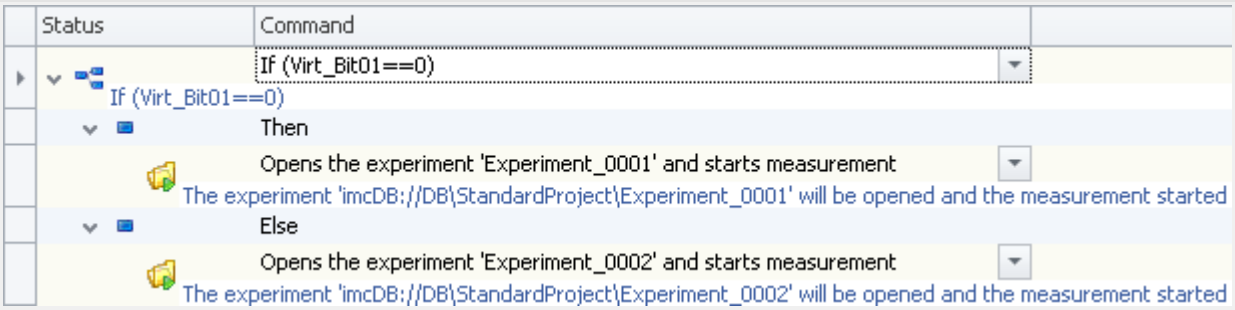


Note that a comparison is performed using the operator "==".

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 1598.

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 on loading an experiment from the Sequencer

Note that the command *Open Experiment/Start measurement* 1615 does not overwrite the Sequencer. The Sequencer is ignored when the command *Open Experiment/Start measurement* 1615 is selected.

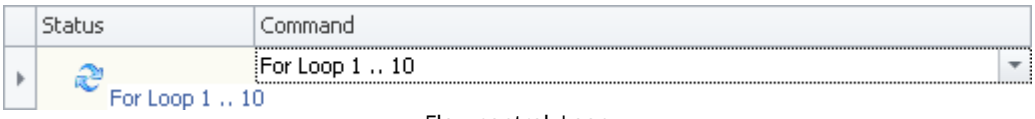
16.2.1.2 Loop

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.

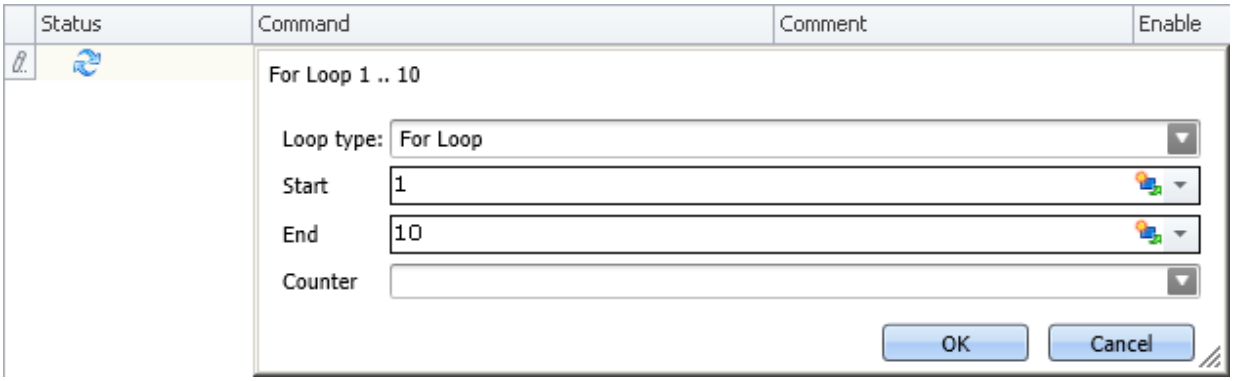


Flow control: Loop

Right-click next to the *For Loop* entry to open its Properties dialog:

- Select the *Loop type: For Loop* or *While Loop*.

For Loop

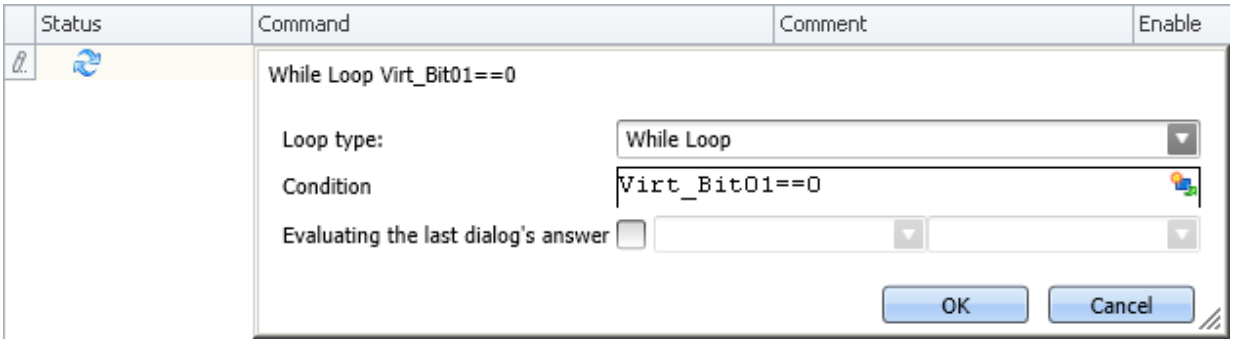


Flow control: For Loop

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 1216 and While Loop 1218 is found in the examples for the plug-in Sequencer.

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 1598.

16.2.1.3 Stop Sequencer

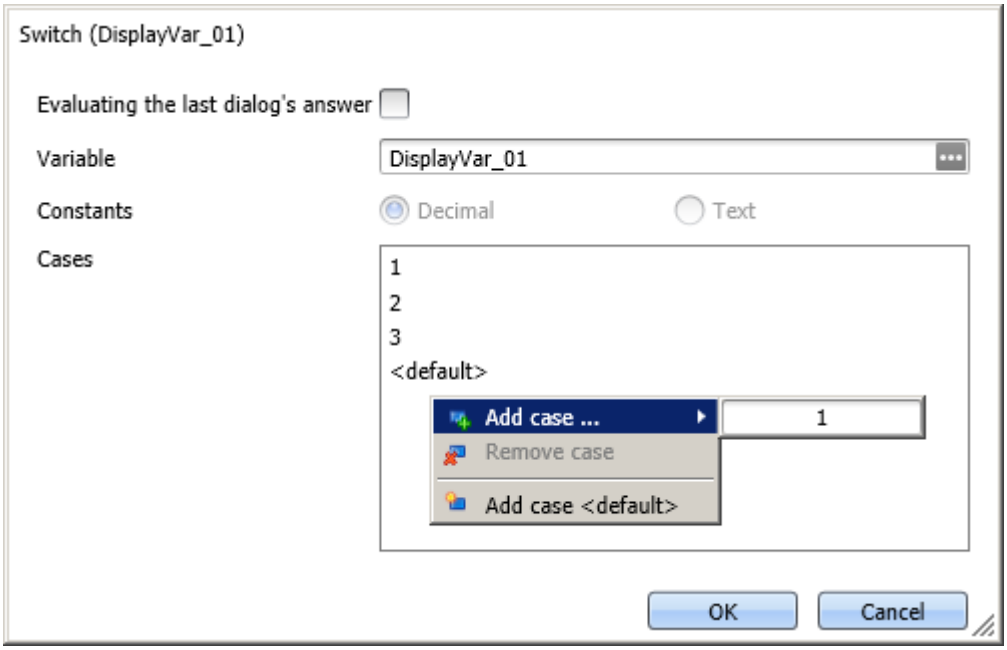
When this command is run, the Sequencer is stopped. The command only takes effect on the Sequencer running. Other command-lists, such as ones which are started by means of a Panel-button, are not stopped in that way.

16.2.1.4 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.

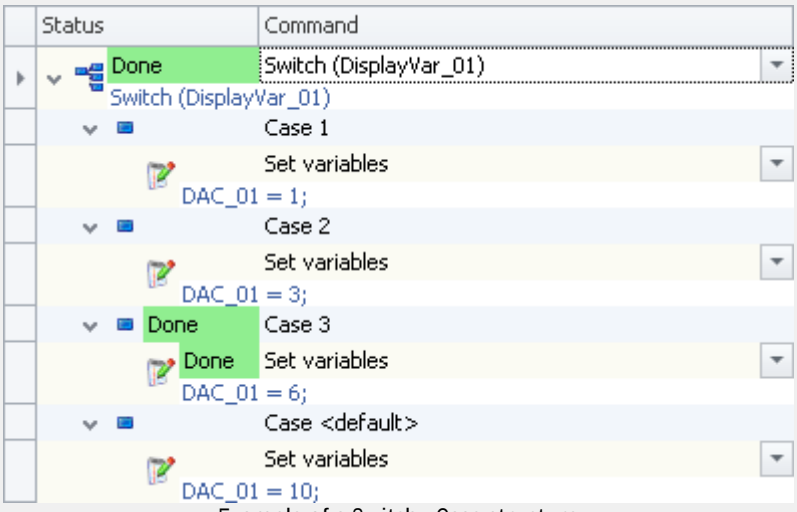
Additional *cases* are added using the right mouse button.

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 1598.

Example

In the example, the Displayvariable DisplayVar_01 is linked to a pop-down list in the Panel.

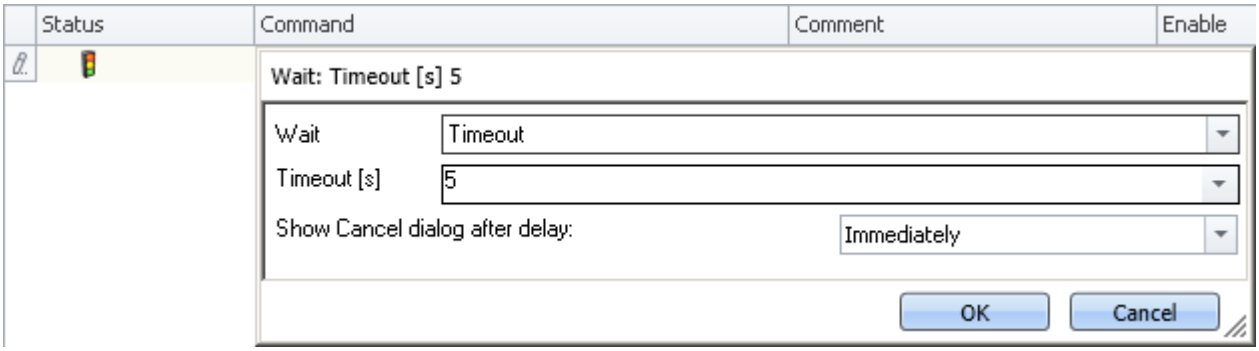


Example of a Switch - Case structure

The selected line determines the value of the Display variable. The analog channel DAC_01 is set in response to the value.

16.2.1.5 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 "==".

16.2.1.6 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 1594
- Loop as While Loop 1594
- Switch 1596

For the evaluation, always the response to the last dialog is used. This includes *Show message box* 1602 and *Show Panel page as dialog* 1620.

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".

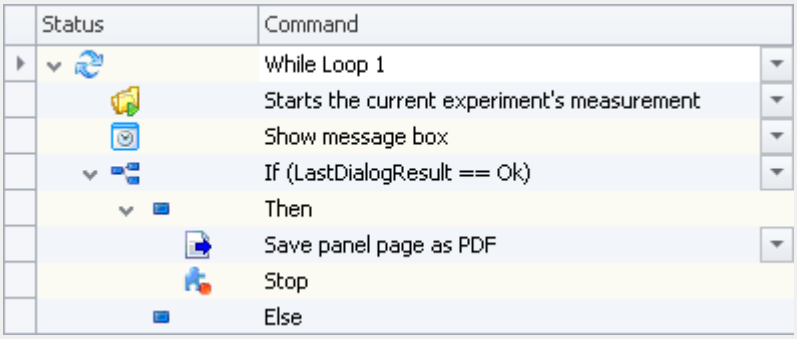
Is no response available yet?

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 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

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.

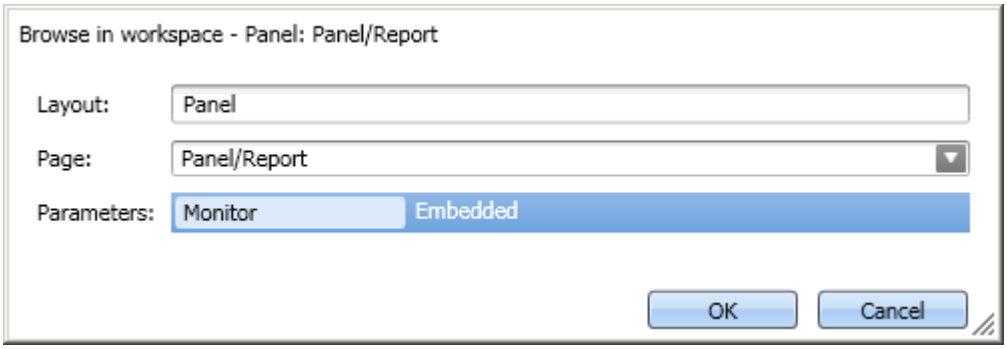
16.2.2 User interacion

16.2.2.1 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 view* 814). 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* 829).

Reference: Tutorial

An example of the command: *Browse in workspace* is presented in the user's manual:
Sequencer > Tutorial 1202.

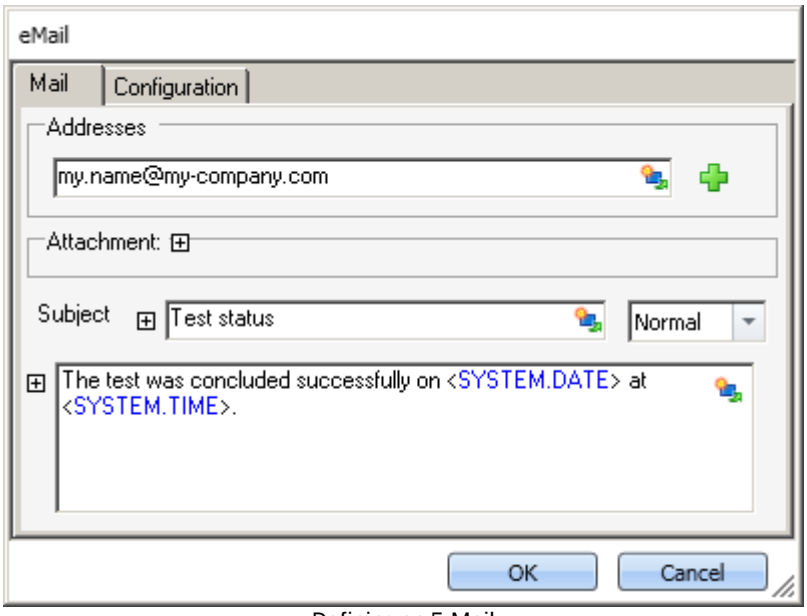
16.2.2.2 E-mail

The command E-mail enables automated sending of an email 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

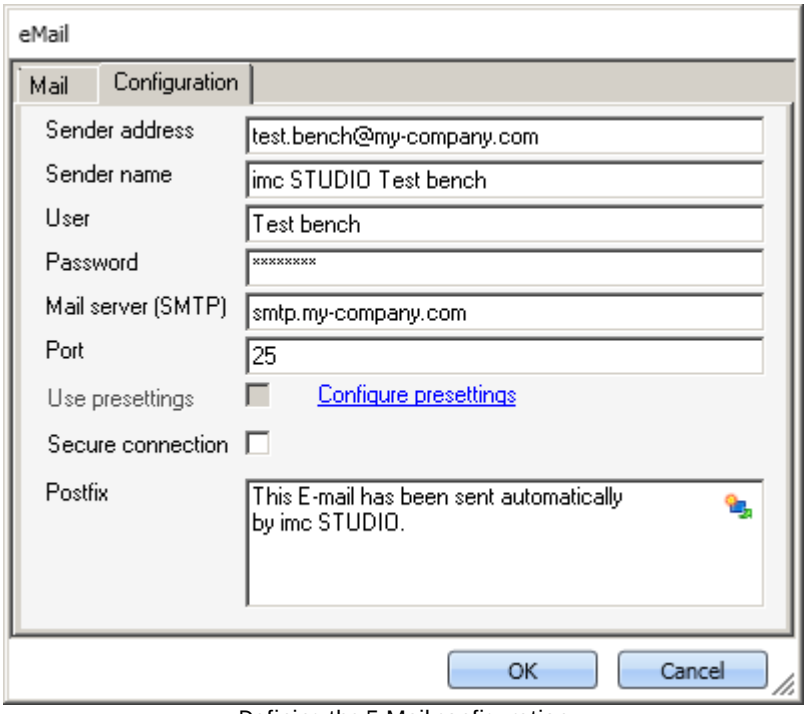
By right-clicking, or by clicking on the placeholder symbol ( ), you can apply a variety of placeholders.

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 email address).
Sender name	The name of the sender of the email (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 email.

FAQ

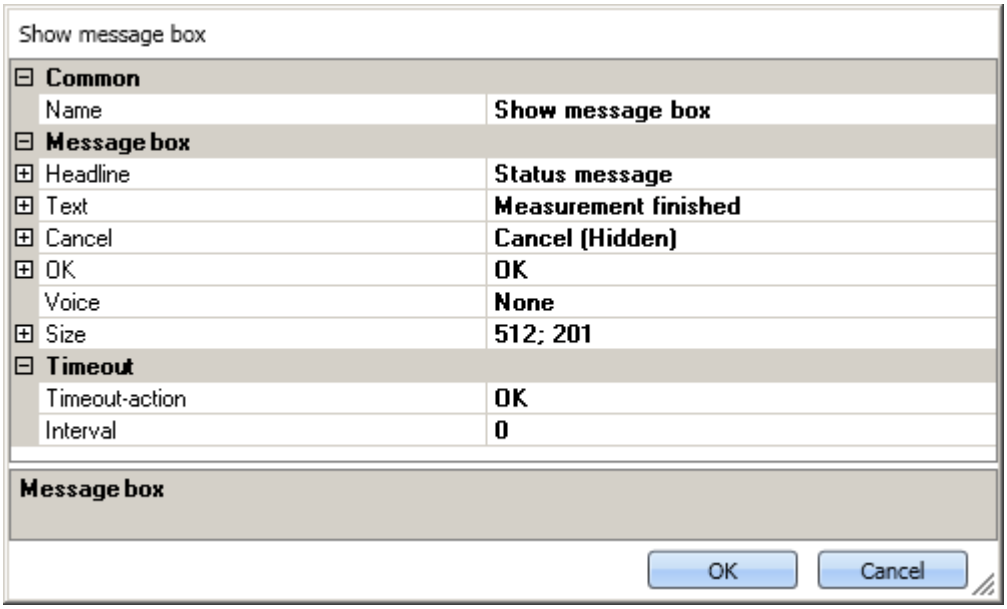
Why do E-mails not get sent, even though the settings are correct?

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.

16.2.2.3 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 1598.

Reference: Tutorial

An example of the command: *Show message box* is presented in the user's manual:
Sequencer > Tutorial 1202.

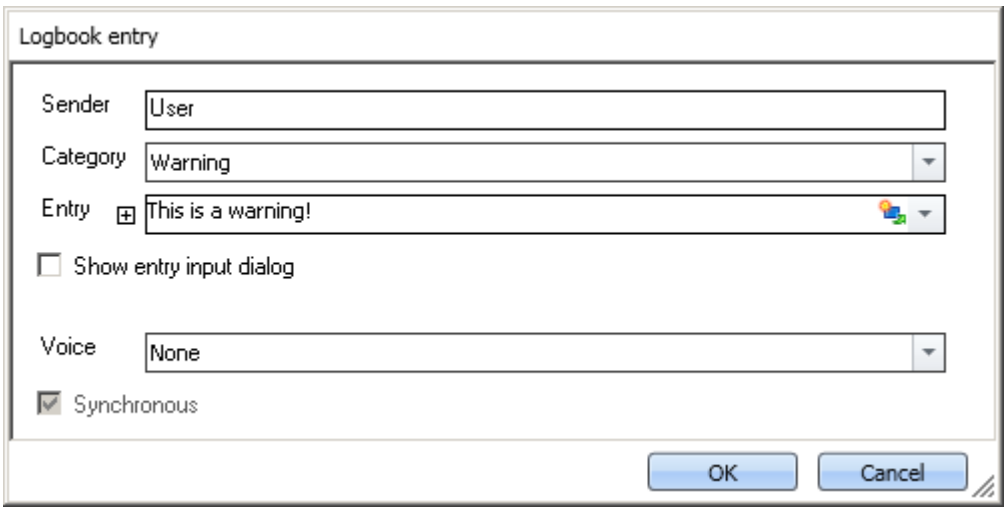
16.2.2.4 Open Logbook-Viewer

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 69.

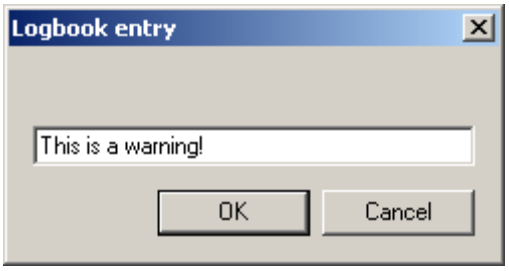
The Logbook-Viewer can be started with defined filter settings. For example, messages of the category: *Information* 70 and all Duplicates 71 can be hidden.

16.2.2.5 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"
Voice	The message text can be spoken by the computer voice.

Parameter	Description
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

16.2.2.6 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. The action can be directly linked with the synonymous Widget: "Execute menu action". 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"

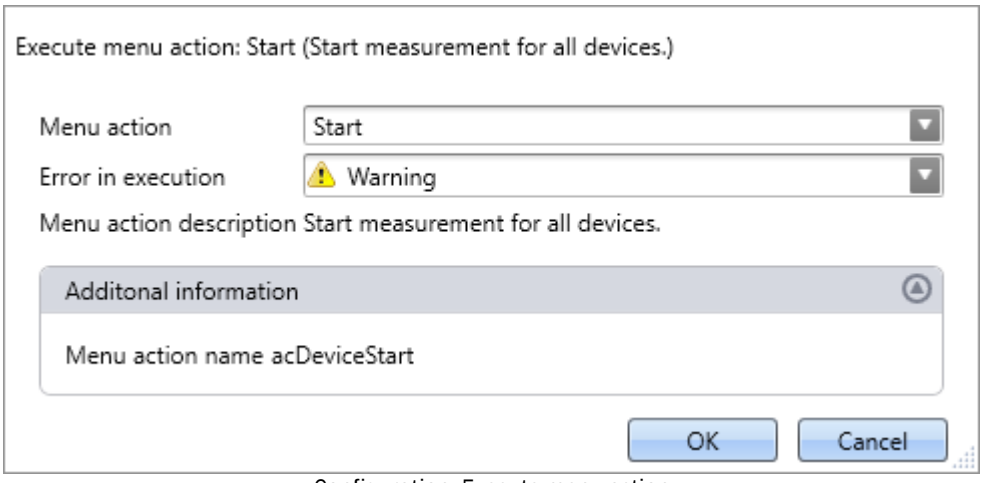
The sequence does not wait for the function behind the action to be concluded. For the sequence, the action is "done" when the action is acknowledged.

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.

16.2.2.7 Set measurement number

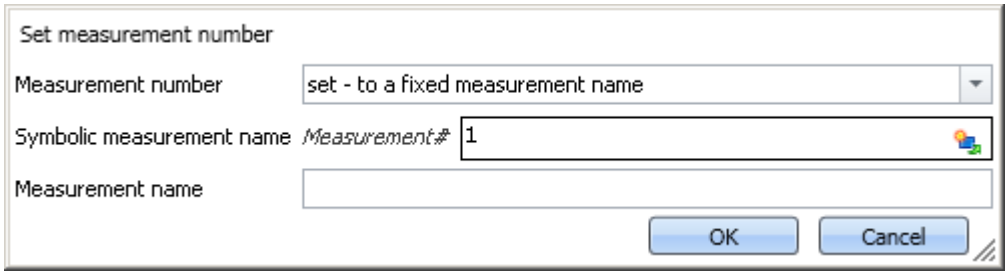
The command *Set measurement number* assigns a measurement number to a measurement.

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.

set - to a fixed measurement name

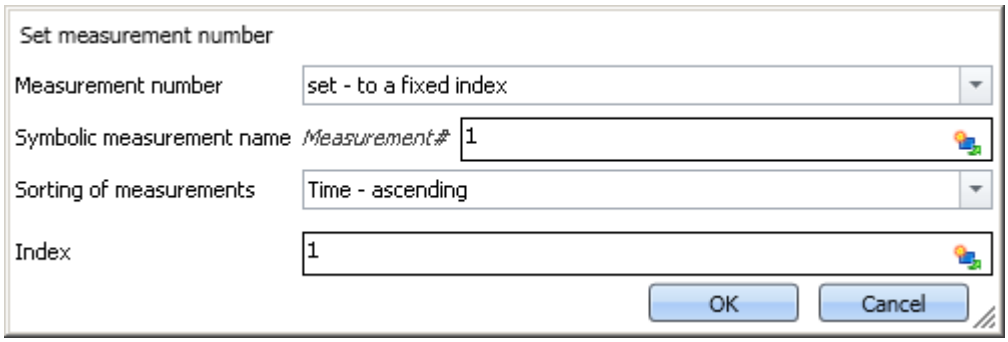


Command "Set measurement number to a fixed measurement name"

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

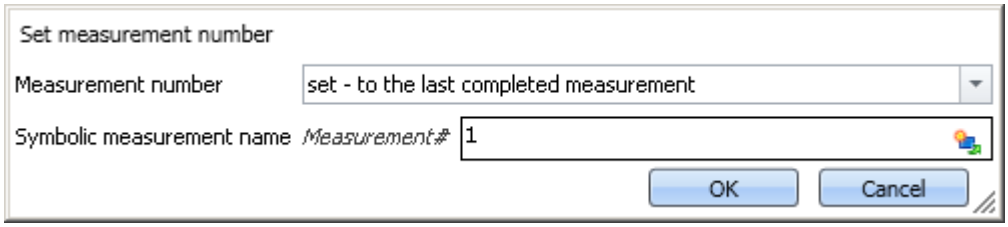


Command "Set measurement number to a fixed index"

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

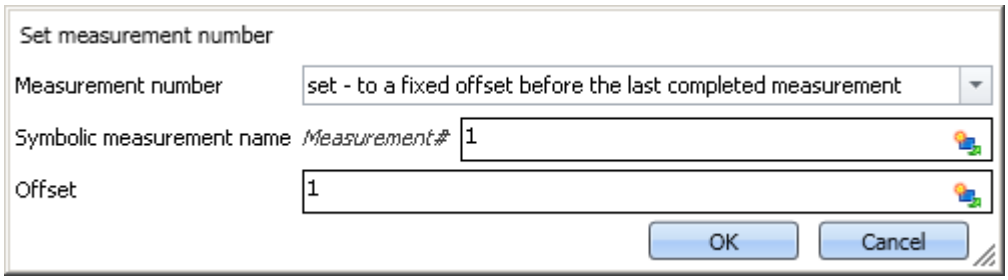


Command "Set measurement number to the last completed measurement"

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

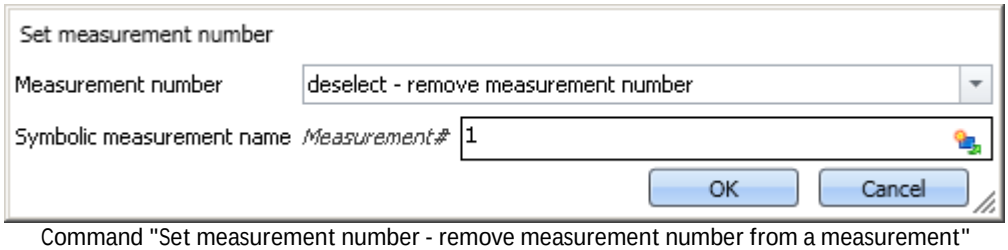


Command "Set measurement number to a fixed offset before the last measurement"

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



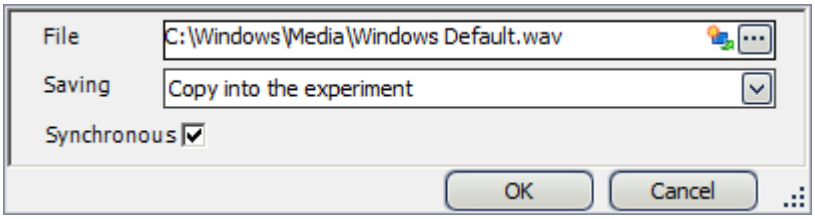
Command "Set measurement number - remove measurement number from a measurement"

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.

16.2.3 Read/Write data

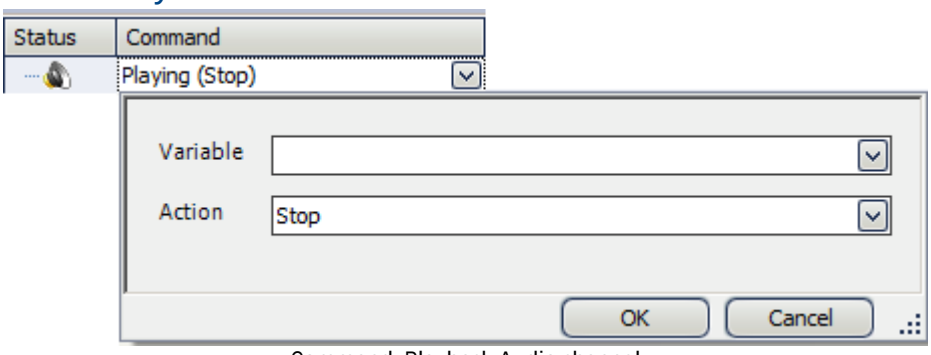
16.2.3.1 Play soundfile



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.

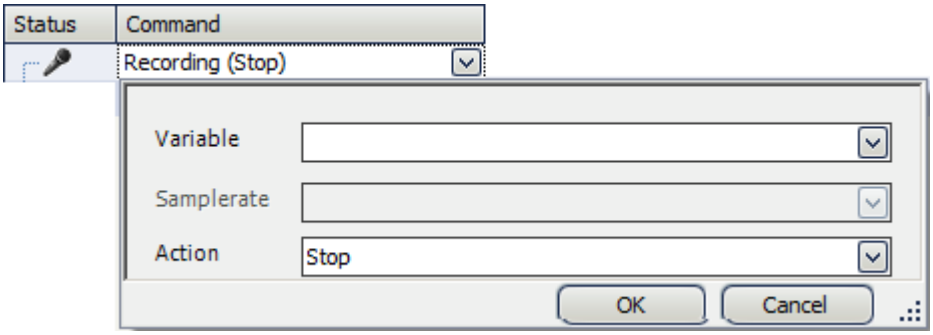
16.2.3.2 Playback audio channel



Command: Playback Audio channel

Parameter	Description
Variable	Report channel (audio), in which the recording to be played back was saved; on this topic, see also the section Report Channel 1675.
Action	• Start: Running this command starts the playback of what is saved under the variable. • Stop: Any playback of the variable in progress is stopped when this command is run.

16.2.3.3 Record audio channel



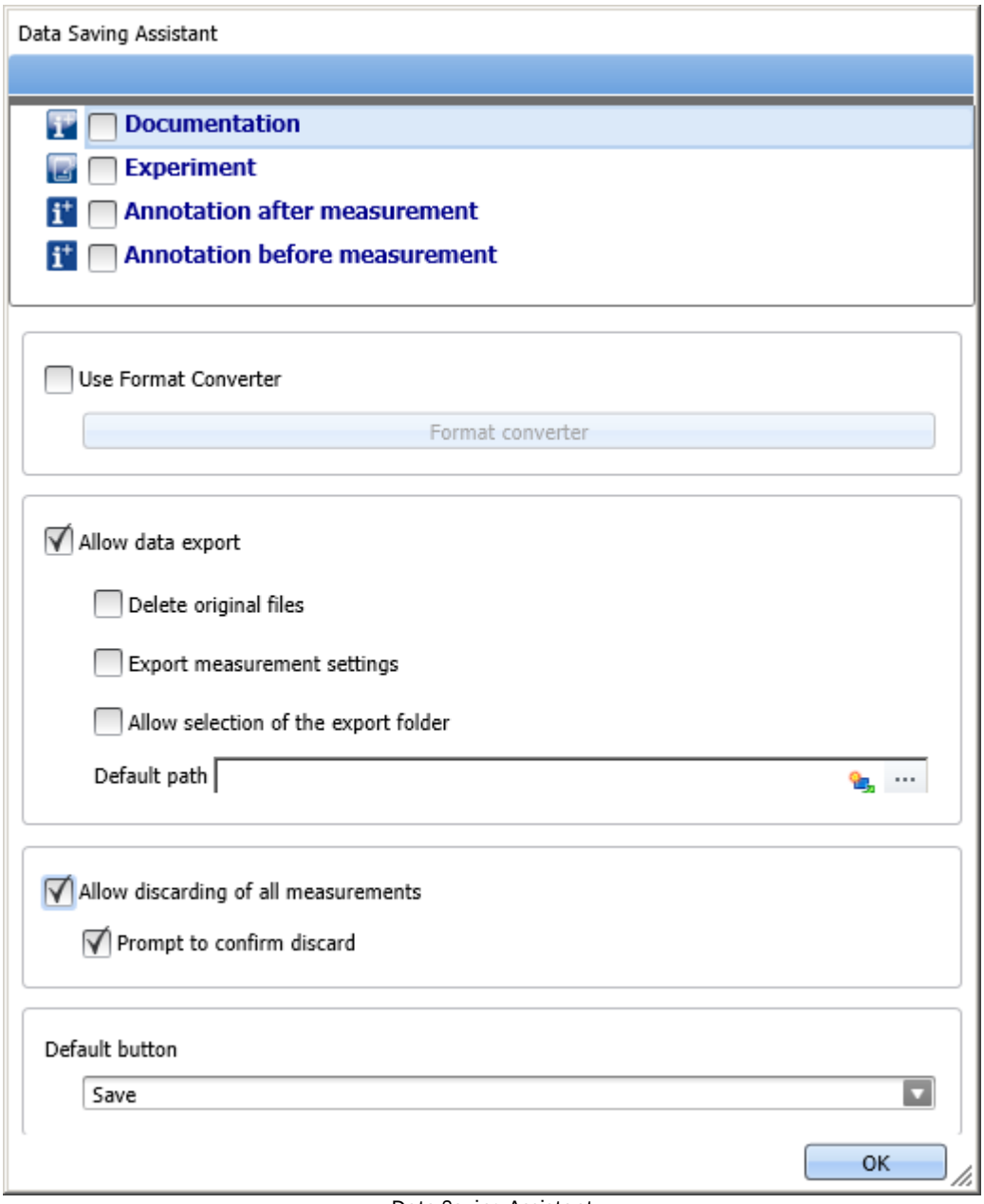
Audio-Channel Recording command

Parameter	Description
Variable	Report channel (audio), to which the recording is to be saved; on this topic, see also the section Report Channel 1675.
Sample rate	Rate at which the recording (microphone) is sampled; the higher the rate, the higher the recording quality.
Action	• Start: the recording is saved • Stop: the recording is stopped • Automatic: With USB headsets which come with a mute function, the muting is automatically recognized as an interruption. If muting is deactivated, recording resumes automatically.

16.2.3.4 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* 1611 (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 62). 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

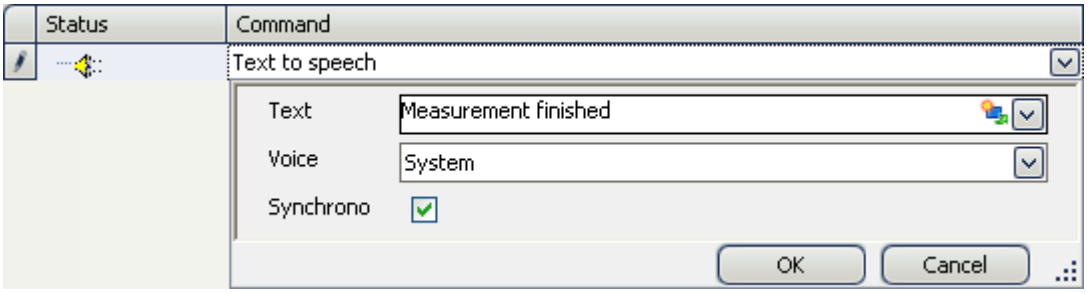
Here you set which button is "pre-selected". If the commando is executed, the selected action can be run by clicking on the Enter-key.

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.

16.2.3.5 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.

16.2.3.6 SQL command

The SQL command offers the ability to communicate with relational databases from within imc STUDIO by means of the database language SQL. It is possible to process data from a database within imc STUDIO using one of the supported adapters, as well as to save other data and values to a database.

Supported database systems

Previously, SQL commands supported three database adapters:

- OleDB (MS Access databases *.mdb)
- MS SQL (Microsoft SQL Server as of Version 7)
- Oracle (Oracle Server as of Version 8)

Notes on the use of Oracle databases

On the PC on which the SQL command is run, a version of the Oracle client software which match the database server. Otherwise, it is not possible to set up a connection with the Oracle server.

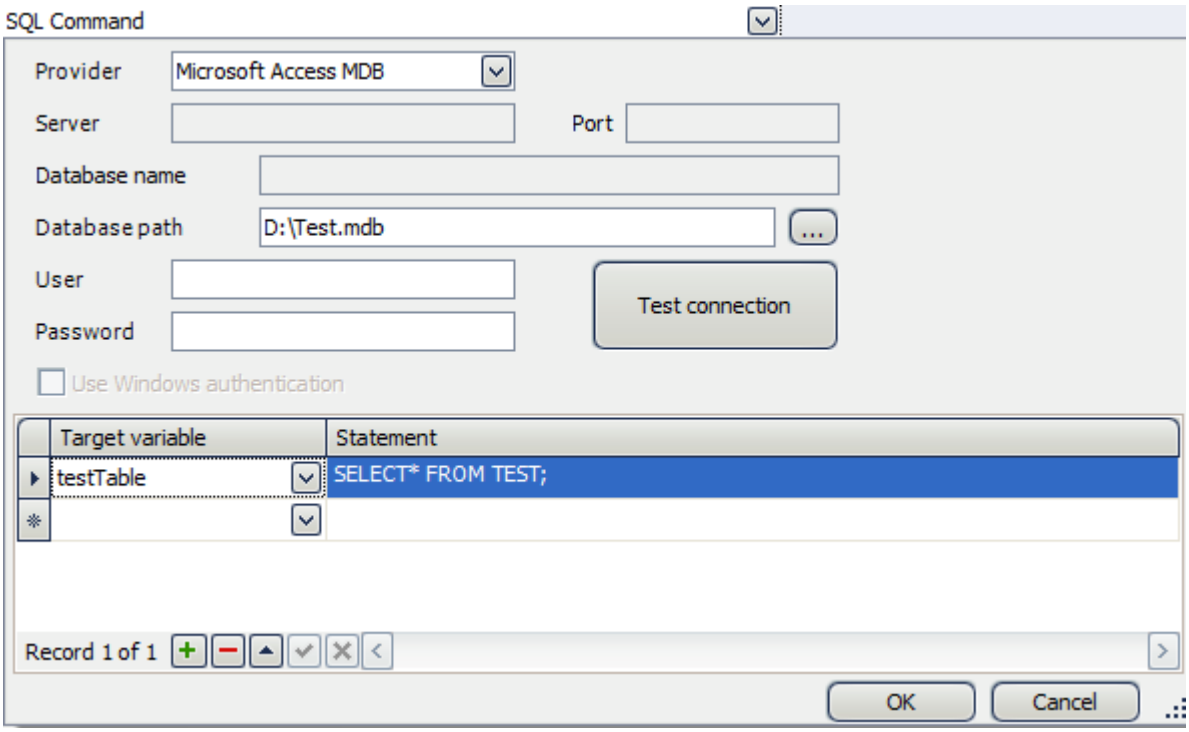
Supported SQL-commands

Except for a few limitations (see note below), all SQL commands which the respective adapter supports can be used on the database by means of the command.

Note

It is not possible to use SQL commands, e.g. selections, on data tables which were created in the imc STUDIO. Data selection, deletion and other changes can only be used on tables within the database.

Configuration



Command "SQL Command"

Parameter	Description
Provider:	Provider of the database accessed: • Microsoft Access MDB • Microsoft SQL Server • Oracle Server
Server:	IP address and name of the server instance; instead of the IP address, it is also possible to state the server's name: <Server Address>\<Instance Name>. Providing this specification is only possible with Microsoft SQL Server and Oracle Server.
Port:	If the database system's port on the server differs from the respective default, it must be specified separately here. Providing this specification is only possible with Microsoft SQL Server and Oracle Server.
Database name:	Name of the database to be accessed. Providing this specification is only possible with Microsoft SQL Server and Oracle Server.
Database path:	Absolute path of the access database; it is also possible to specify a UNC-path, e.g. \\Computer\test\data.mdb; by means of ..., it is also possible to select the database by means of the file selection dialog. Providing this specification is only possible with Microsoft Access MDB.

Parameter	Description
User:	User name for registering with the database (if necessary); the user name is saved with the experiment, with secure encoding, and is only visible in plain text in the configuration dialog.
Password:	Password for registering with the database (if necessary); the password is saved with the experiment, with secure encoding, and is only visible in plain text in the configuration dialog.
Use Windows authentication:	Some database systems support the usage of the current Microsoft Windows user's data (name and password) for log-in on the database server. It is required that the current account is enabled for log-in on the database server. It is unnecessary to explicitly specify the user name and associated password. The Windows user logged in is always determined at runtime (time at which the command is called) and refers to the current Windows user, not the current imc STUDIO user. None of these data (name and password) are saved with the experiment.
Test connection:	Uses the data supplied to verify whether a connection can be made to the database.
Target variable:	see the section *Target variable*
Statement:	see the section *Statement*

Target variable

All values returned by SQL-commands at the time they are called can be saved in user-defined variable 1671. The following types of user-defined variables can be used for this purpose:

- Data tables
- Text
- Numerical

Which variable type to select depends on what the particular SQL command returns, and on how many values you wish to save. For instance, if you with to save only a single value even though the SQL-command returns multiple values or a table, select the data type Text or Numeric. If the data type Data Table is selected, then the SQL command must actually return a table. With the data type Data Table k, it is not possible to save single values.

If the target variable is a single value (numerical or text) and the SQL-command does not return a data table, then the value in the first row in the first column is always assigned.

Note

- The specified target variable must exist at the time of the call.
- If the SQL-command's return value is not required, then the box *Target variable* must also be left empty.
- It is also possible to select variable of types other than the above mentioned ones, but these can not be used for saving the return values.

Command

Here, the SQL-command is entered. This can consist of a single but arbitrarily extensively nested command which is valid for the selected database system-provider. The validity and required syntax can vary with the provider as presented in the respective database system documentation.

Any arbitrary amount of commands can be entered. All these commands use the configured connection and are processed successively.

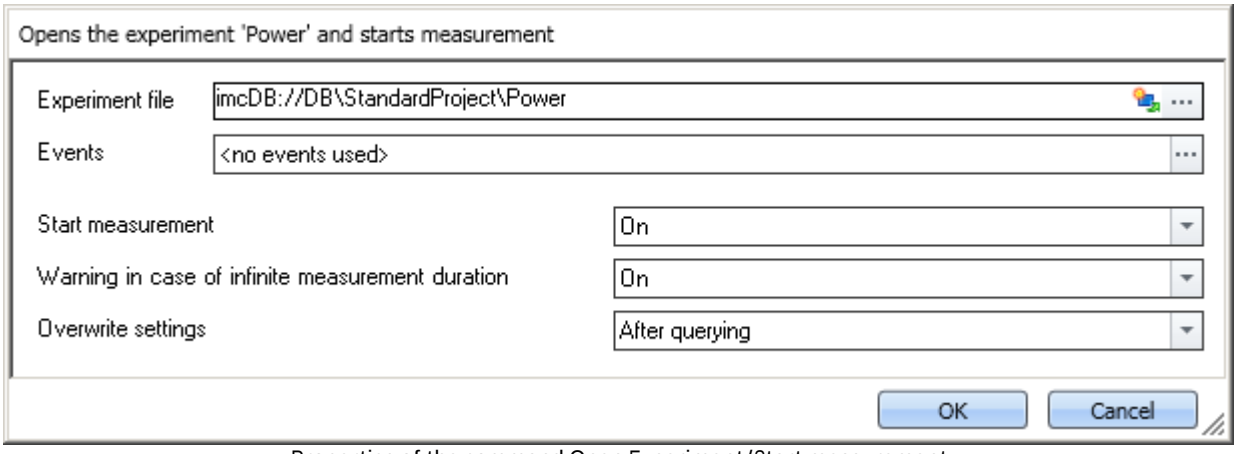
Commands can contain placeholders 1655, allowing values to be inserted dynamically at runtime. Thus, it is possible to transfer data at runtime from imc STUDIO to databases.

Note

If a certain SQL-command is not valid for the adapter selected, or cannot be run for other reasons, a corresponding error message is posted in the log book and the running of commands is interrupted.

16.2.4 Open experiment/Start measurement

The command Open Experiment/Start measurement enables starting of the measurement from the Sequencer.



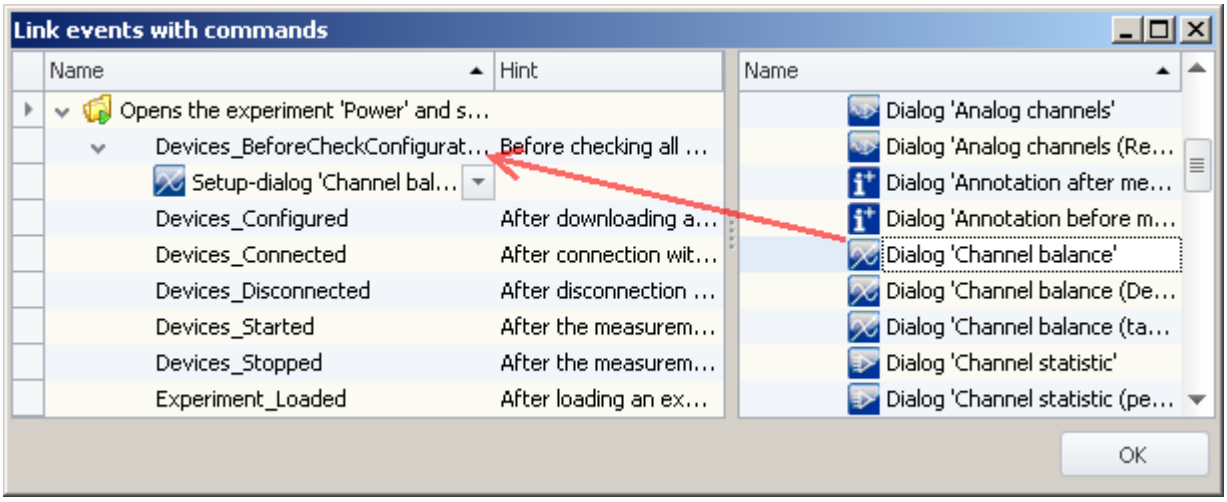
Properties of the command Open Experiment/Start measurement

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*.

Events

Just as there events for each measurement, there are also events which can only be configured for this measurement.

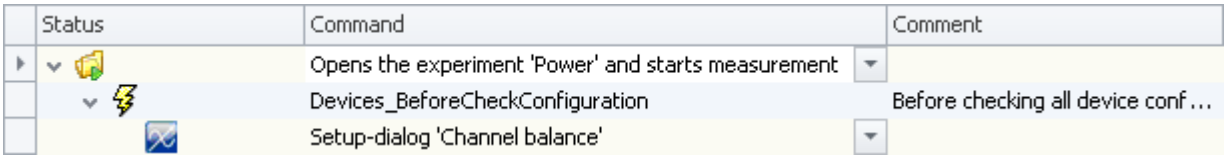
Using the parameter: *Events* opens an additional window:



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:

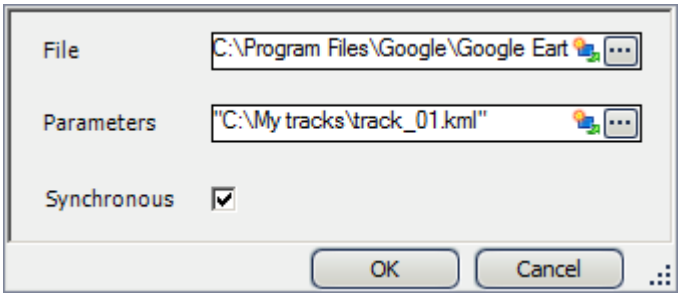


Example: Command associated with an event in the Sequence table

16.2.5 External calls

16.2.5.1 Execute file

The command Execute file starts a Windows program.



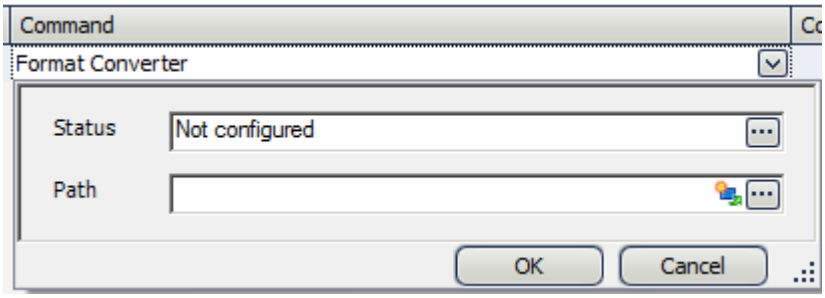
Execute: In the example, Google Earth is started to display a test drive

Parameter enables transfer of Start parameters.

16.2.5.2 Format converter

The Format Converter offers the user the ability to convert measured data to a different format, e.g. into EXCEL data format.

The following image shows the Configuration dialog:



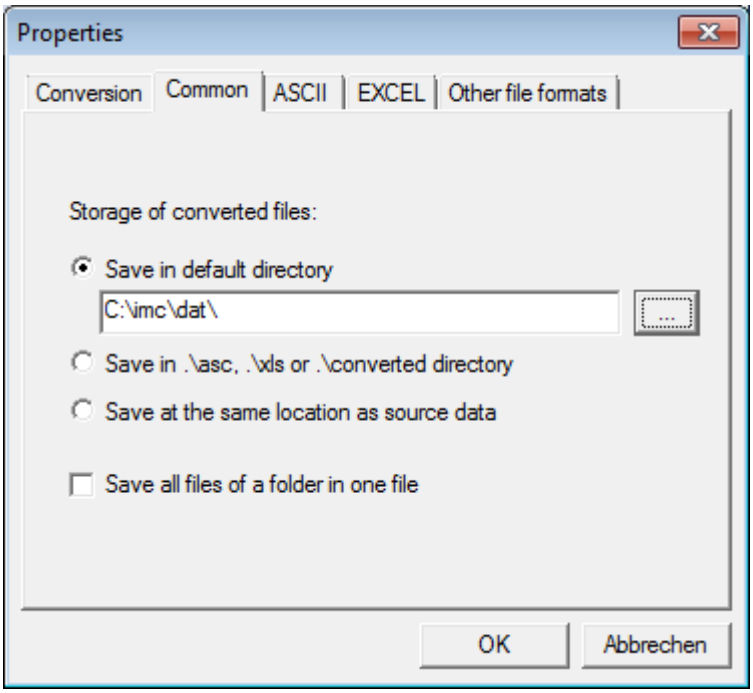
Configuring the command: Format Converter

Status

Immediately upon opening the command configuration, the status is *Not configured.*

Using the button ⋯ belonging to the box *Status*, you open a dialog in which all conversion settings are made:

- Conversion: To which format are the measured data to be converted?
- Common: where to save the converted file?
- ASCII: What format template is to be used and what should the converted file's extension be? (only relevant if *ACSII* is selected for *Conversion*)
- EXCEL: What format template is to be used and what should the converted file's extension be? (only relevant if *EXCEL* is selected for *Conversion*)
- Other file formats: Here, it is possible to call additional external programs for converting measured data (only relevant if *Other file formats* is selected for *Conversion*)



Setting the Format Converter

Path

- Here, one measurement data file (*.dat, *.raw) is specified, so you can select whether only the individual file is to be converted or whether all measurement data files in the folder are to be converted.
- If a folder is specified here, the all measured data files in the folder are converted:

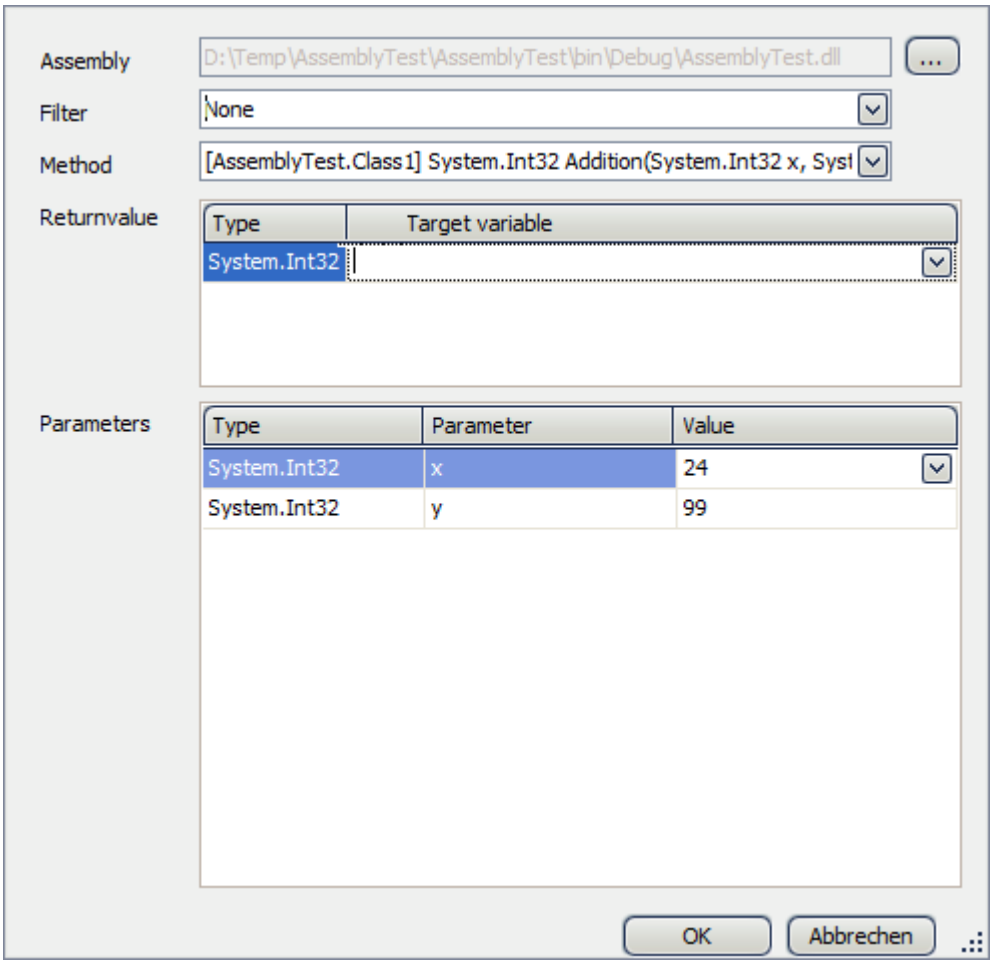
- If the option *Save all files of a folder in one file* on the page *Common* is selected (see the section *Status*), then all measurement data files, located in the specified folder, are converted to a single target file.
- If the option is not selected, then for each file of measured data which the folder contains, the data are converted to a separate file.

16.2.5.3 Running a COM Method

This command enables you to to use methods from external DLLs in imc STUDIO.

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.



Example of an configured Run COM method command

Assembly: Clicking on the button ... causes a *.NET Framework Assembly* to be selected which usually is of the type *.dll*. These DLLs, however, differ from classical DLLs from *unmanaged* code in the ability of their content to be accessed by means of reflection.

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*. If a method which is desired is not listed, it is not directly visible and thus not accessible by means of reflection.
Return value:	See the section *Return value*
Parameters:	See the section *Parameters*

Notes on the use of WinAPI-DLLs

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.

Return value

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").
Target variable:	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. *Remark: Normally, methods ALWAYS HAVE ONLY ONE return value. In the special case of a compiled MatLab-DLL, however, here it is possible to offer more than one. For this reason, the input box for the target value takes the shape of a table (and not, for a single line, as might be expected).*

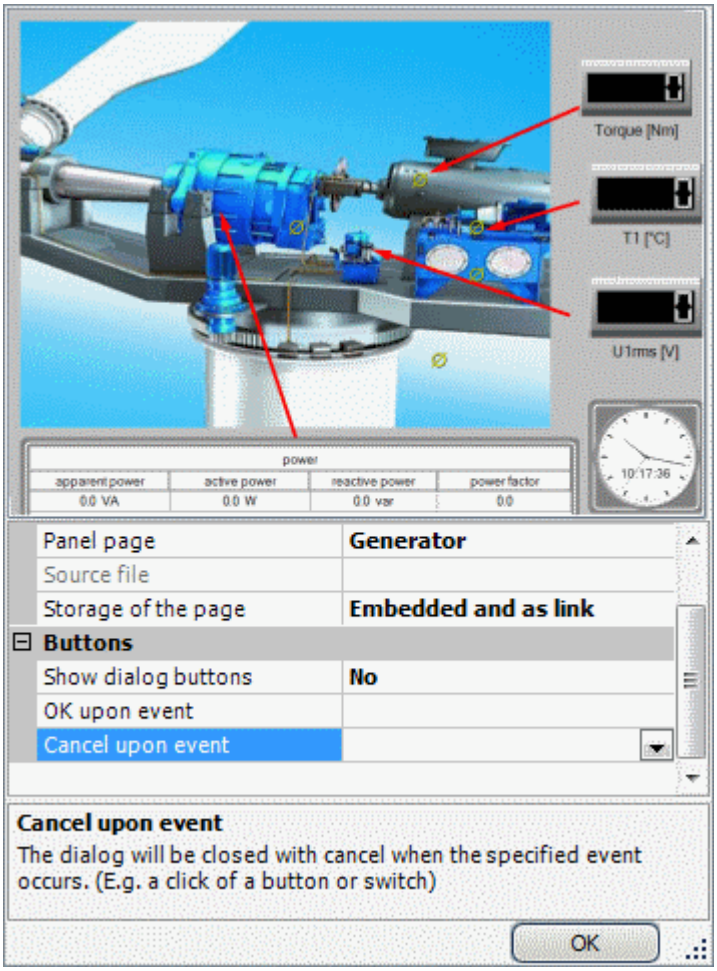
Parameters

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 1649. 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.

16.2.6 Panel

16.2.6.1 Show Panel page as a dialog

A Panel page can be displayed as a dialog. Add the command Panel page as a dialog in the Sequence Table.



Properties of the command "Panel-page as dialog"

General

Dialog title:	Title of the dialog to be displayed
Panel page:	Among the available Panel-pages, the page to be displayed must be selected; by means of ... , you can select an 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.
Storage of the page:	*Embedded:* The Panel-page is embedded in the experiment. This means that it is still available even once the page in the plug-in Panel has been deleted. *As link:* Only a reference (link) to the existing Panel-page is saved. *Embedded and as link:* Saves according to both options above.

Buttons

Show dialog buttons:	For the setting *No*, the dialog can only be closed by means of the "X" in the upper right corner, unless it is closed by the option "OK upon event" (see next item). Closing by clicking on "X" is interpreted as Cancel , which causes the subsequent commands to not be performed. For *Yes*, an area with buttons appears under the dialog. By means of these buttons, it is also possible to close this dialog.
OK upon event:	The dialog closes as in response to OK when the event selected occurs. Here, it is possible to make logical combinations of all events belonging to the selected Panel-page, for example, whether an existing rocker switch is switched on or off.
Cancel upon event:	Analogous to *OK upon Event*, except that if the event occurs, the dialog as if Cancel had been clicked, which means that subsequent commands will not be performed.
OK button	This field is only enabled if *Show dialog buttons* is set to *Yes*. Here you can set whether the button is visible (*Visible* box) and the button caption can be entered (*Label* box): In the *Active* field, you can set whether the button should always be active or whether it only becomes active once all entries on the page are valid. For more on this topic, see the segment Creating Mandatory Fields.
Cancel button:	This field is only available if *Show dialog buttons* is set to *Yes*. Here you can set whether the button is visible (*Visible* field) and the button caption can be entered (*Label* field).

16.2.6.1.1 Creating mandatory fields

The function for making the OK-button active only if all entries are valid only affects entries originating from the Setup's metadata columns. Only these columns can be designated as mandatory boxes. It is not possible to use this function to verify any other entries, such as entries made in Widgets.

Whether or not a metadata column is designated as a mandatory field exclusively affects its behavior in the command *Panel-page as Dialog* and not its behavior in Setup.

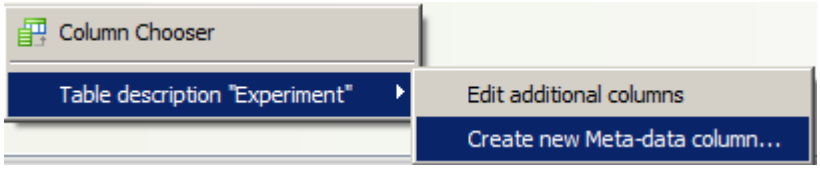
To illustrate the workings of the function *Active if all inputs are valid*, consider the following example.

Example: Creating a Panel page as a dialog with a mandatory field

In this example, a new metadata column *Tester* is added to the already existing Setup-page *Experiment* and declared as a mandatory field. Next, a Panel-page is created on which the Setup-page *Experiment* is displayed. Finally, this Setup-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 field.

Open the page *Experiment* in the Setup

Over the white space, open the context menu (by right-clicking the mouse) and select the *Table description "Experiment"* > *Create New Metadata Column...*



Context menu of the Experiment page

When the *Collection editor* opens, click on the *Insert button.* A column with no title initially opens.

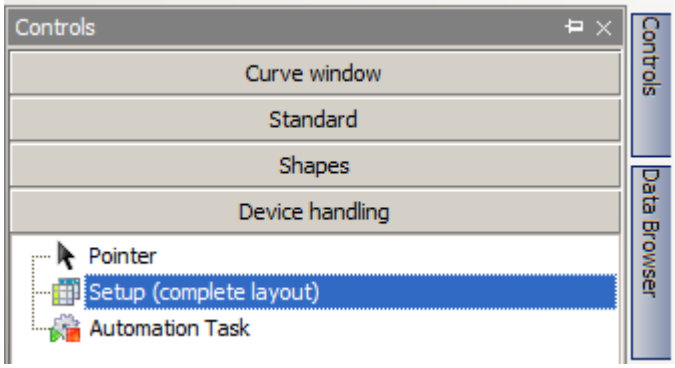
Enter the following properties:

- Title: Tester
- Type: Single-line text (unchanged)
- Mandatory field: ☑

All other settings remain intact. Close the dialog with OK. The new column is now present on the page *Experiment*.

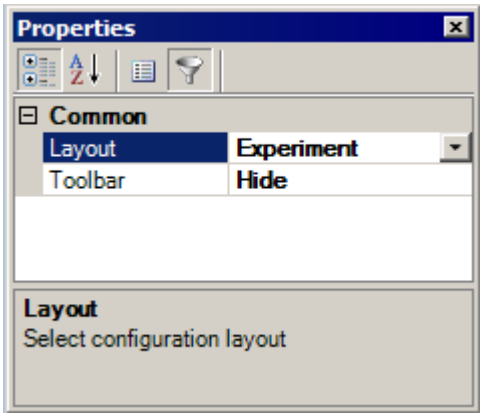
Next, go to Panel.

In the tool window Widgets in the control *Device handling*, select Widget *Setup (complete layout)* and use Drag&Drop to move it to the Panel (name of the page: *Page 1*).



Device handling - Widget Setup (complete layout)

Open the Widgets 's properties and select *Experiment* as the Layout.



Properties of Widgets Setup (complete layout)

Note

If the new column is not visible, perform the following steps: to display it, open the Panel-page's context menu in the Widget and select the menu item Column Selection. There, select the Tester column and use Drag&Drop to move it to the Panel page.

Next, go to the Sequencer and a the command *Panel-page as Dialog* to it.

Configure the command:

- Panel-page to be displayed: Page 1
- 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 *Tester* is assigned a valid value (here: text).

Note

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.

Example: Defining the existing column Short description as a mandatory field

In order to access existing metadata columns, open the tool window *Table descriptions* in the Setup via the menu *Extra*. This tool window is not available in the standard version of imc STUDIO.

Next, from the table descriptions displayed, select the one containing the metadata column, in this example *Experiment*, since that is where the column *Short description* is located.

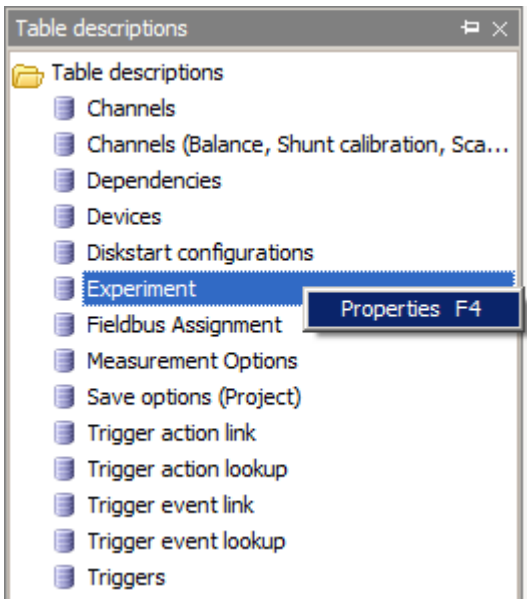
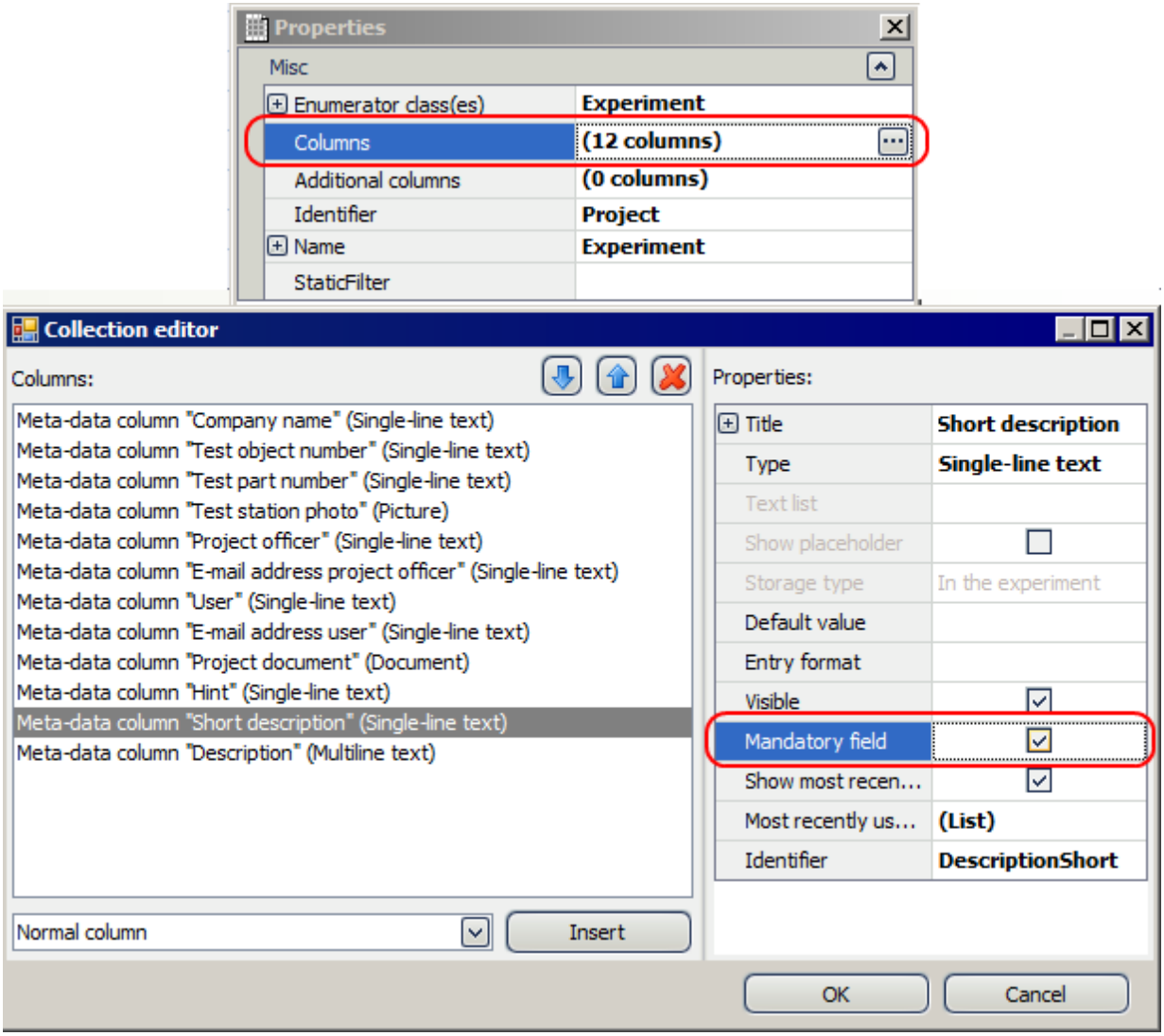


Table Descriptions tool window

Open the Properties dialog by means of the context menu (right-click the mouse). Alternatively, press the F4 key. In the Properties dialog, left-click the mouse over *Columns* and then click on ... to open the *Column editor*, which presents the properties of the individual metadata columns.



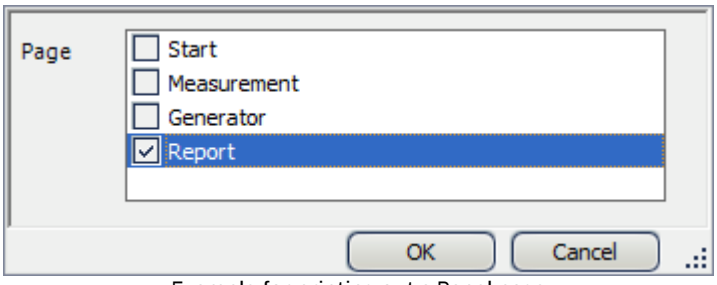
Properties and list editor for the table Experiment

In the list editor, select the column *Short description,* place a check in the box for *Mandatory field*, and finally close the dialog by clicking OK.

Now, if you use the table *Experiment* on a Panel-page which is displayed as a dialog, the field *Short description* is a mandatory field.

16.2.6.2 Print Panel page

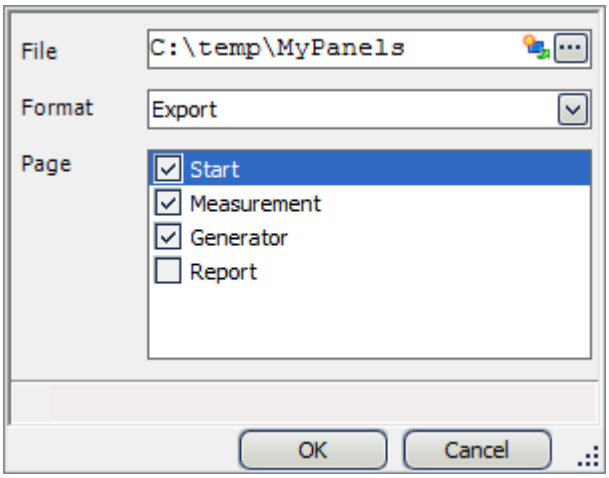
You can print out Panel pages. By this means, it is easy to compose reports. To do this, create Panel pages which are displayed as a report.



Example for printing out a Panel page

16.2.6.3 Export Panel page

You can export Panel pages to a specified folder.



Example of exporting Panel page

16.2.6.4 Import Panel page

You can import a Panel page which was previously exported in DBV format.

16.2.6.5 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.
```

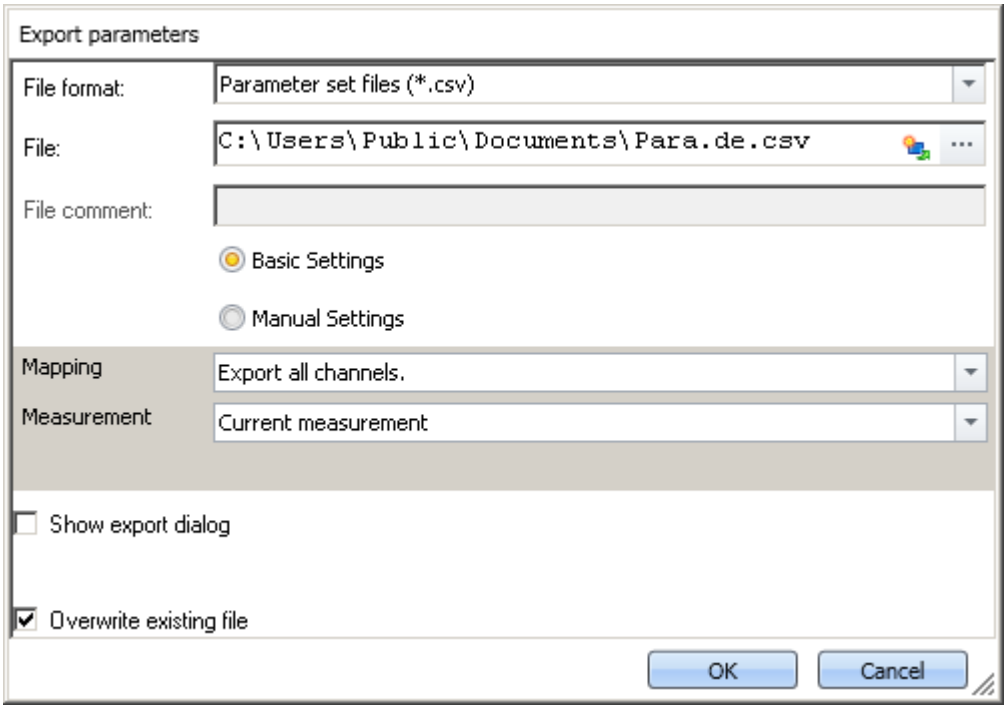
16.2.7 Parameter set

16.2.7.1 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.



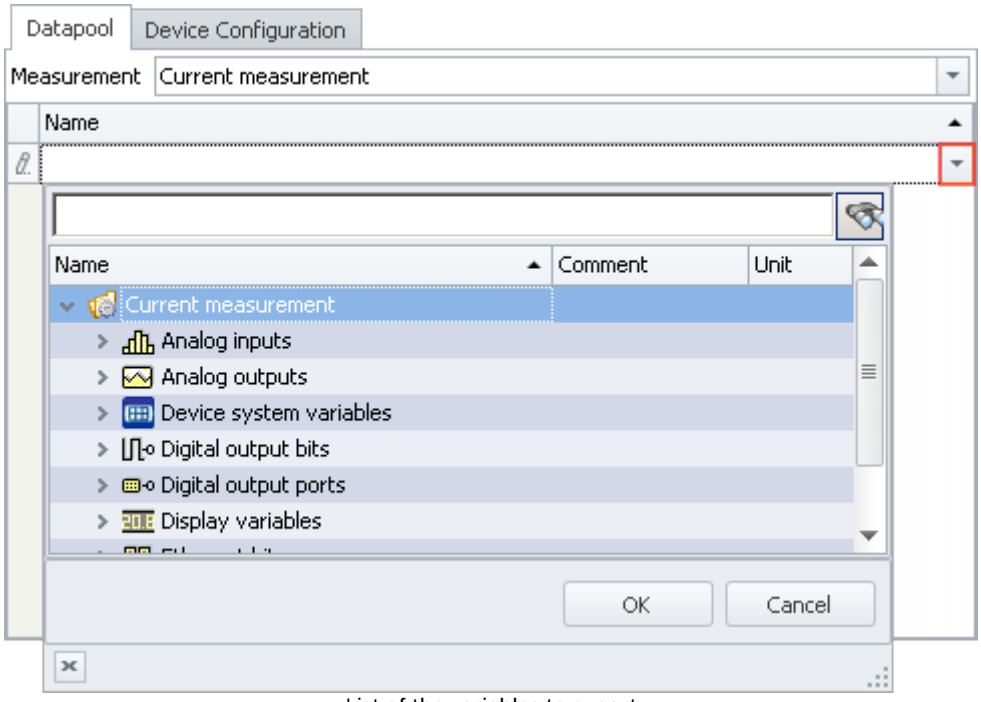
Command: Export parameters

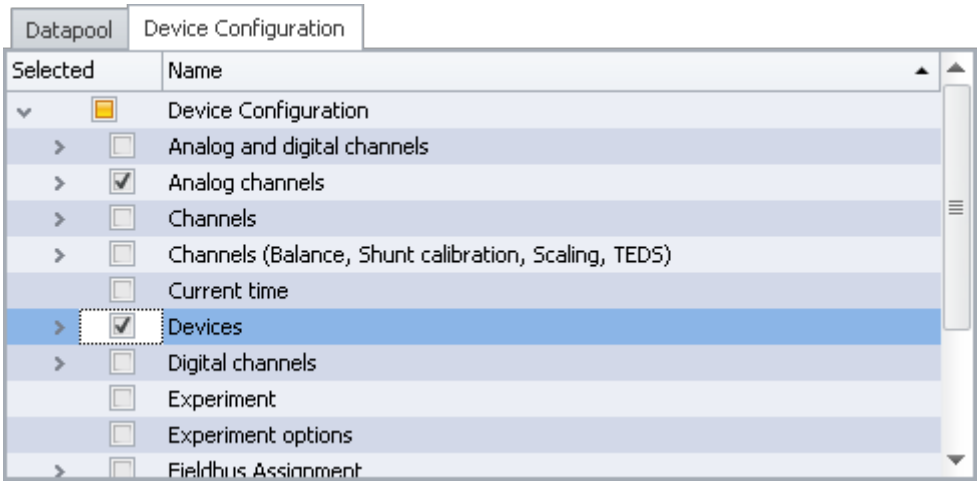
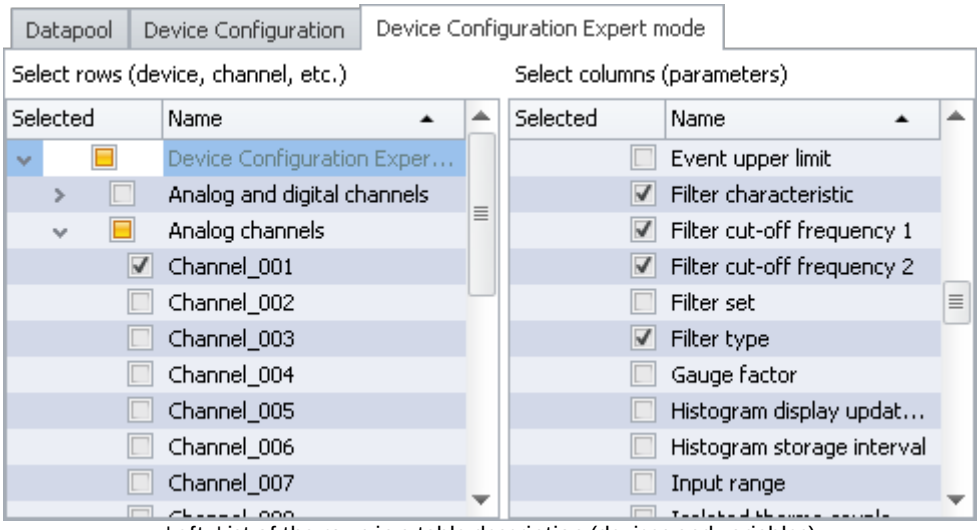
Option	Description
File format	Selection of the export format. For more information, see the description of the parameter set (File formats 241).
File	Specifies the target file including the pathname.
Basic/Manual Settings	In order to configure the export, you can choose between basic and manual settings. • Basic Settings 1627: Here you find prepared export settings, for instance, one where only all values of the analog channels are exported. • Manual Settings 1628: 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 244).

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* 139 ( ) 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 63 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.

Note: Important parameters are always exported

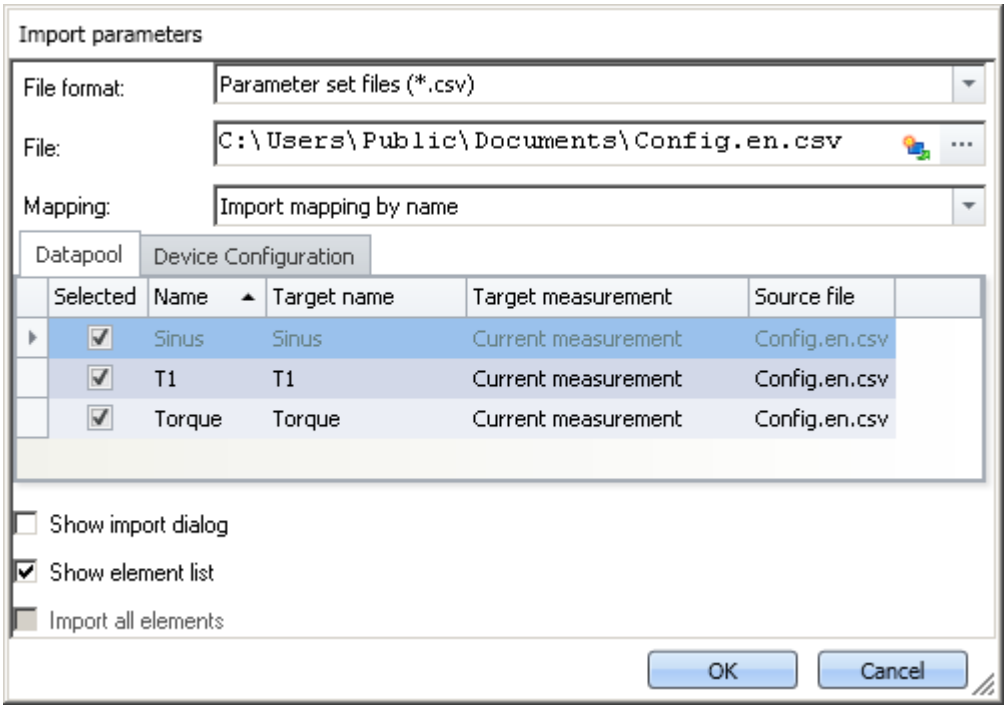
Certain important parameters, which are necessary for assignment purposes when importing, are always included in the export process. Even if they are not selected, the following parameters are always included in the export of channel parameter:

Device name, Device identifier, Channel name, Name.

16.2.7.2 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.

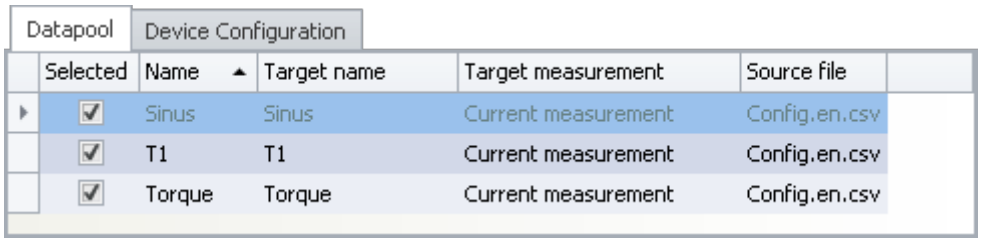
The command: "Import parameters" allows automated import of the parameters at desired times and according to previously adjusted settings. You can also select from which file to import which variable's values and/or settings.

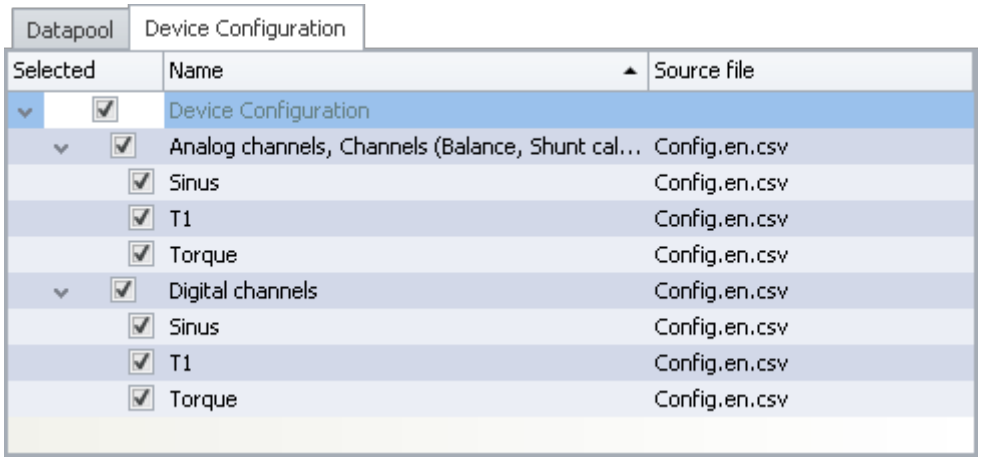


Command: Import parameters

Option	Description
File format	Selection of the file format. For more information, see the description of the parameter set (File formats 241).
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* 1634.
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.  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. If the file format xml was selected for the export, the source table description is located in the file. The parameter set import derived from this, via which table description it is possible to import. As long as the file does not state from which table description the export came, there is a choice of possible table descriptions. The table descriptions displayed may not match the original table description. 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.

Notes: 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 counters, ... This does not apply to single value variables, such as Display-variables, DACs, ... It is possible to import to these.
- For importing variables' values, instead of the command: "*Import parameter*", the command "*Import variables*" should be used. This command is specially designed for variable values.
  Using the command: "*Import parameter*" can cause unintended alteration of the device control.

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

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.
Import of channels according to connector and device serial number	For all channels/variables (parameters of the table description: Channels), the settings are imported. The assignment is made according to the connector designation and the device serial number. Note: When using multiple devices, this selection is preferable to "*Import of channels according to connector*", 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.
Import of channels according to connector	For all channels/variables (parameters of the table description: Channels), the settings are imported. The assignment is made according to the connector designation. Note: 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.
...	Select a user-defined mapping. See *Mapping - Modifications of the default behavior* 246.

FAQ

Question: Why do my changes to the parameter set have no effect following the import?

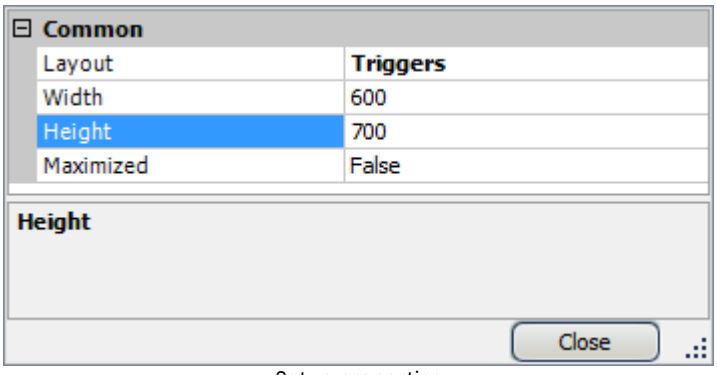
Answer: There are multiple possible reasons:

- 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* 1634).
- 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.

16.2.8 Setup

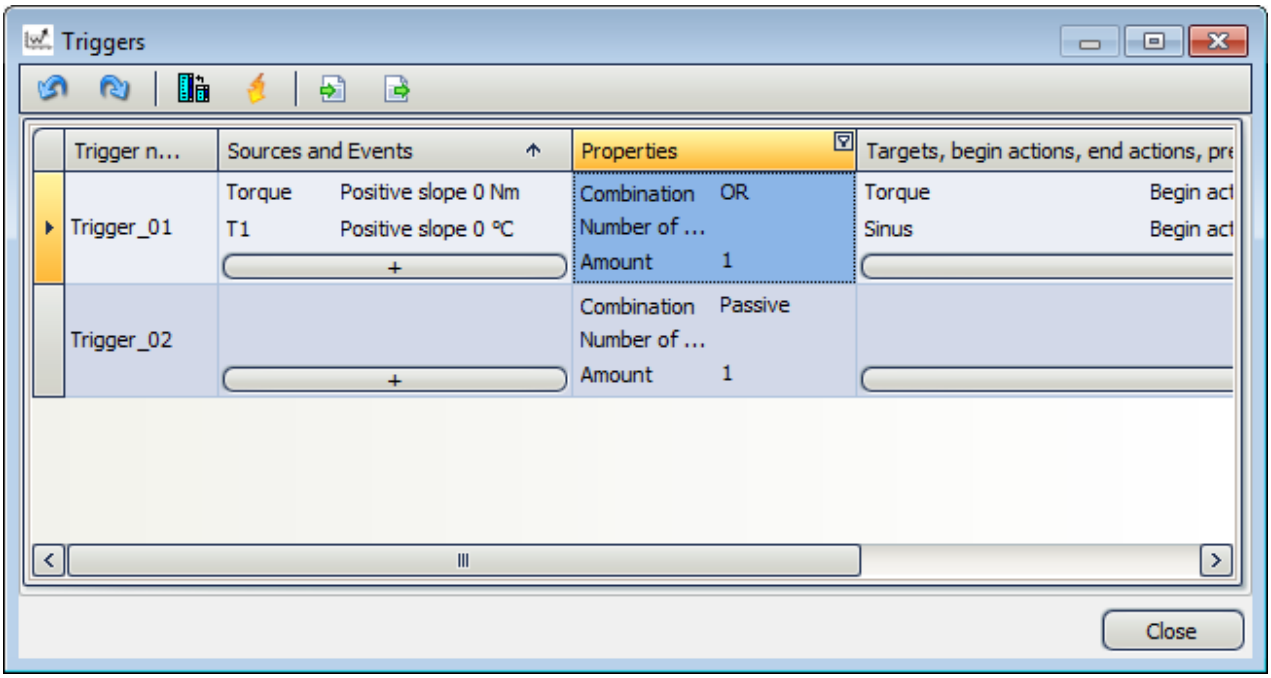
16.2.8.1 Display Setup dialog

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*.



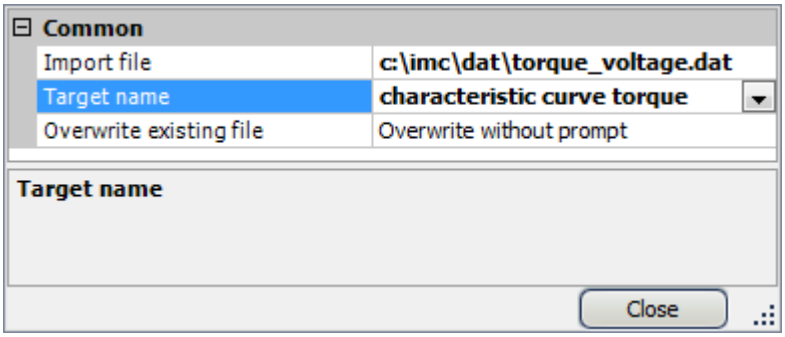
Example Trigger dialog, started by means of the command Setup

16.2.8.2 Import synthesizer-/controller configuaration

Load a configuration for a synthesizer or controller.

16.2.8.3 Import supplemental file

This command enables the import of files which were used by imc Online FAMOS.

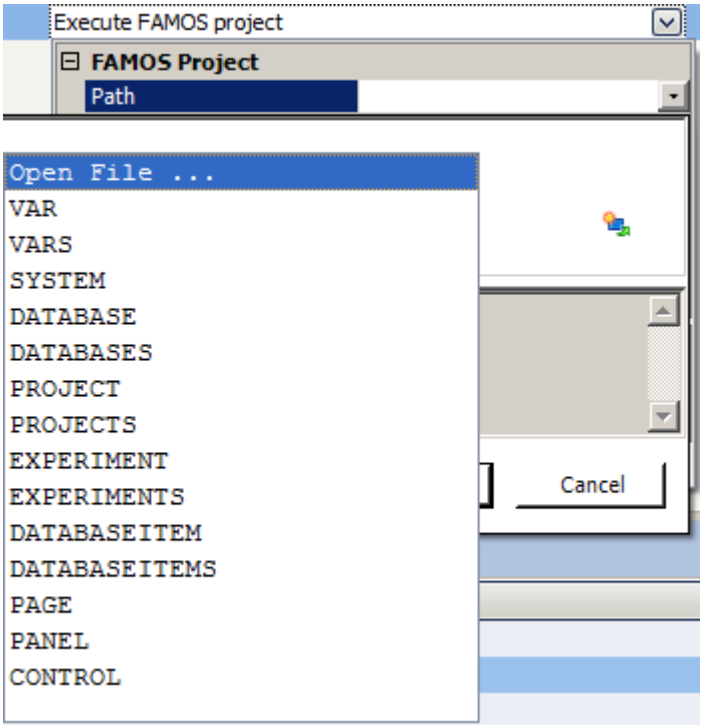


Example of a characteristic curve used in imc Online FAMOS with the Charact() function

16.2.9 Scripts/imc FAMOS

16.2.9.1 Execute imc FAMOS project

A imc FAMOS project is selected and started:

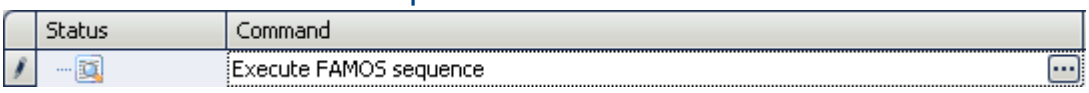


Command: Execute imc FAMOS project

Click on the symbol and select *Open File...*

In the project folder, select the file *Default.FamosPrj*.

16.2.9.2 Run imc 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

The prerequisite is installation of imc FAMOS on the same (see *Technical data sheet* > *Additional imc Software Products* 1702).

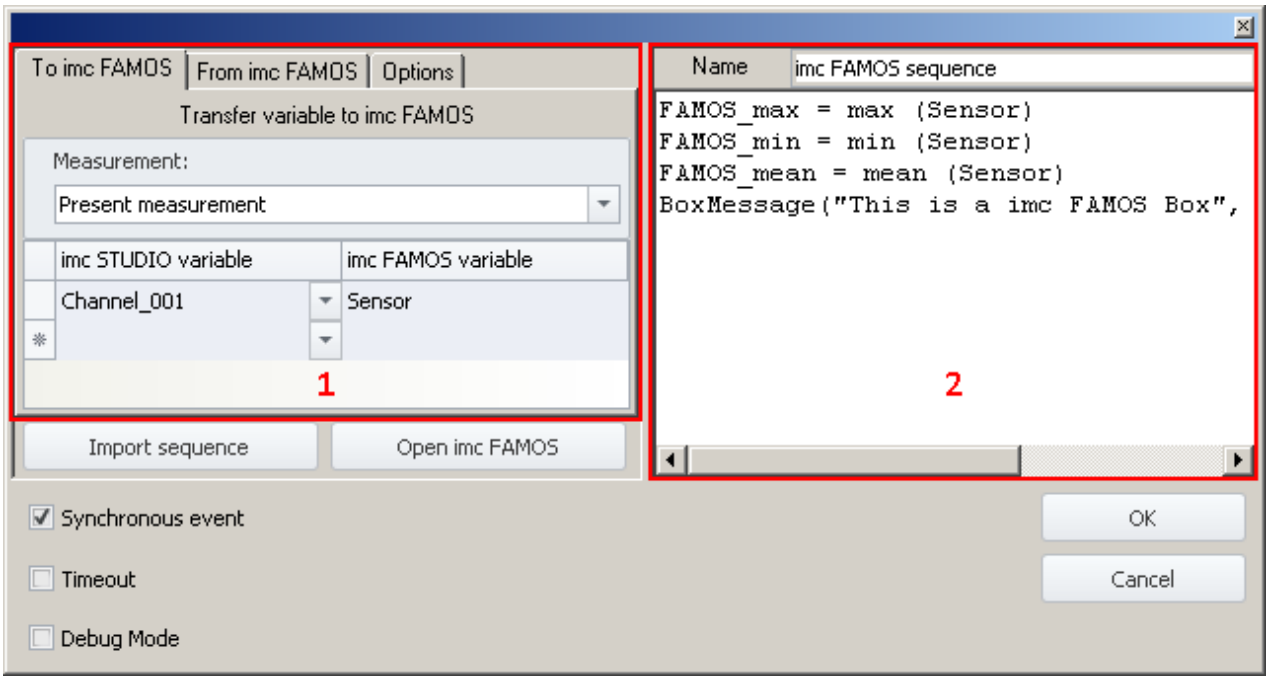
Reference

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.

16.2.9.2.1 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 1132 (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 1132 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 1132 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*.

Checkbox: 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.

Checkbox: 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.

Checkbox: 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.

Note

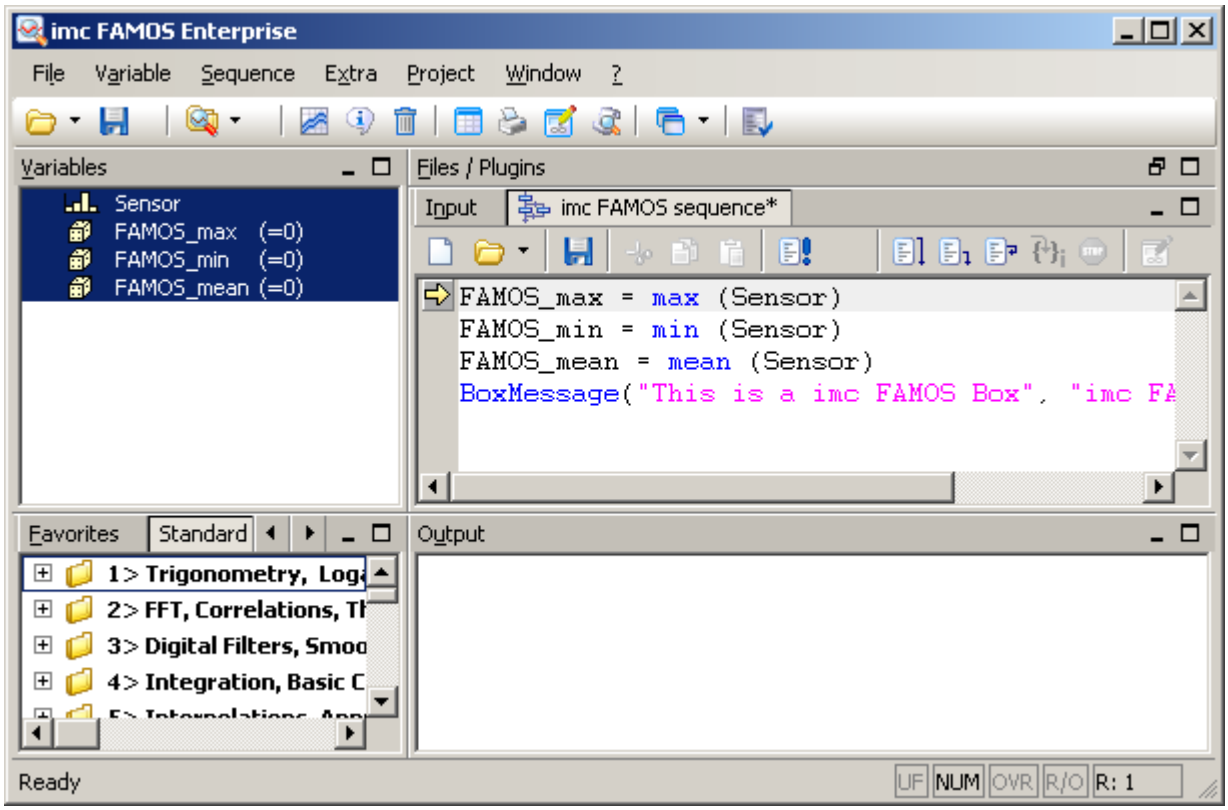
- Please note that in Debug Mode, the command needs longer to execute:
  - With the option: *Synchronized event* active: The subsequent commands wait until the imc FAMOS-command has concluded.
  - With the option: *Synchronized event* deactivated: Required results may not exist in time since the command 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.

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 1131 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 1213 .

Note: Variable not present

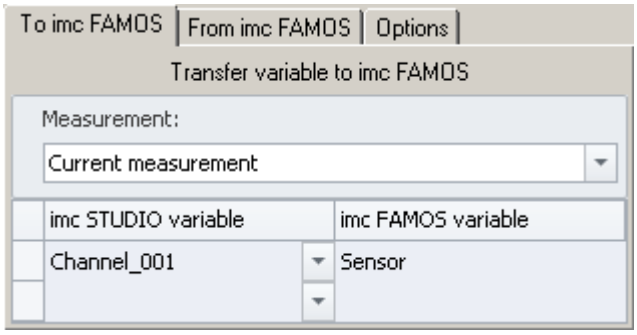
If a variable does not exist in imc STUDIO when it is to be transferred to imc FAMOS, an "empty" variable is created in imc FAMOS as a substitute.

- Data type: Normal data set
- X-Delta: 1
- Total size: 0

16.2.9.2.2 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 1131).

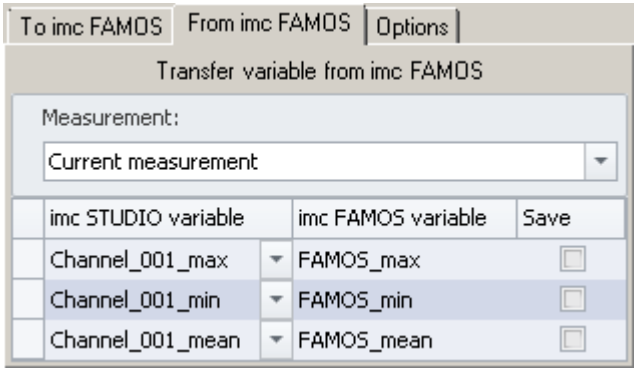
In the example shown, the measured variable *Channel_001* is passed to the imc FAMOS sequence as the variable *Sensor*.

Measurement: Selecting the source

All variables defined are transferred from the measurement selected.

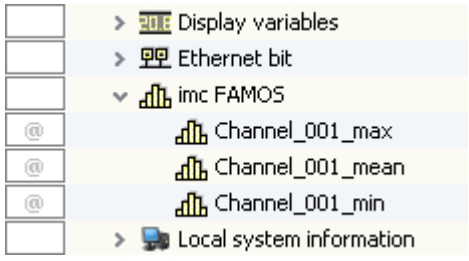
- Current measurement: The current measured data are transferred. Note that in this case, only the amount of circular buffer memory set for the curve window display is available for use. This may be only a fraction of the entire measurement.
- Last concluded measurement: If the measured data are saved, the last measurement saved is loaded automatically. The associated variables are passed to the imc FAMOS sequence.
- Measurement#<No>: The variables belonging to the measurement with the respective measurement number are passed.

Receiving the sequence's results ("From imc FAMOS")



The variables calculated by imc FAMOS are transferred to imc STUDIO along with the transfer table.

The results appear in the Data Browser:



Without saving:

The result appears in the Data Browser, but is not saved.

Measurement: Selection of the target

- 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 concluded measurement and Measurement#<Nr>: The result appears temporarily in the Data Browser along with the last/selected measurement. The result is not saved and is deleted as soon as the measurement is closed (un-loaded).

With saving:

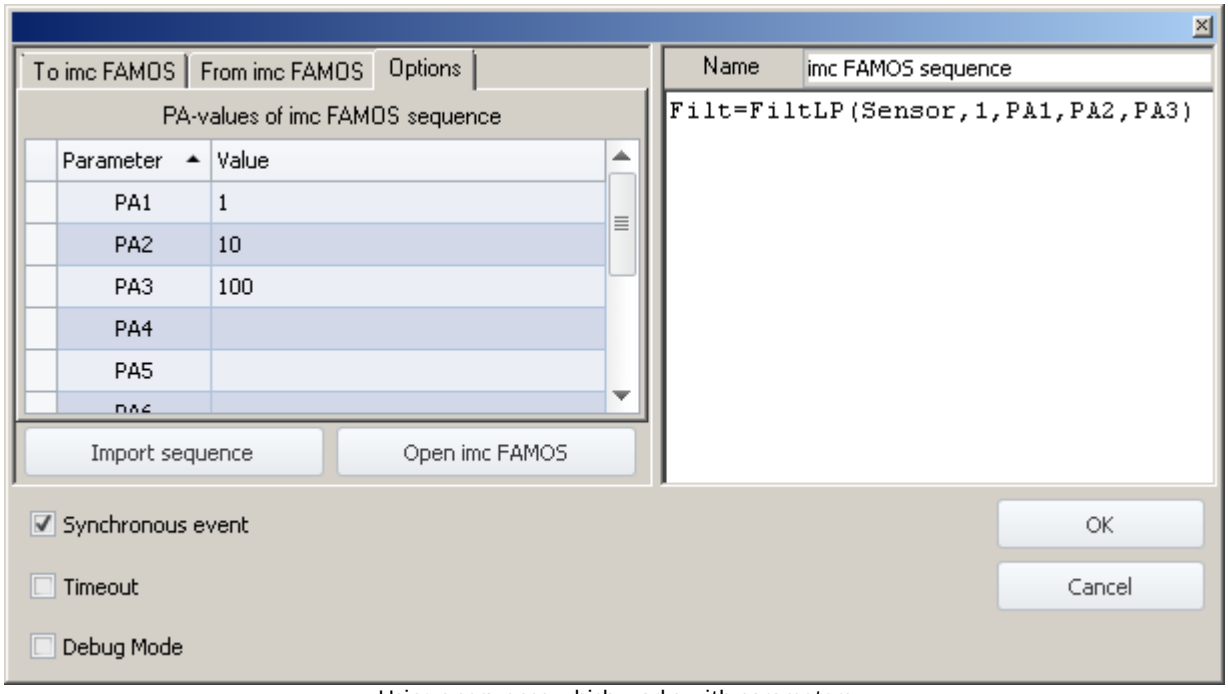
The result appears in the Data Browser and is stored as a.dat-file.

Measurement: Selection of the target

- 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.
- If Current measurement is selected as target (not recommended), 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.



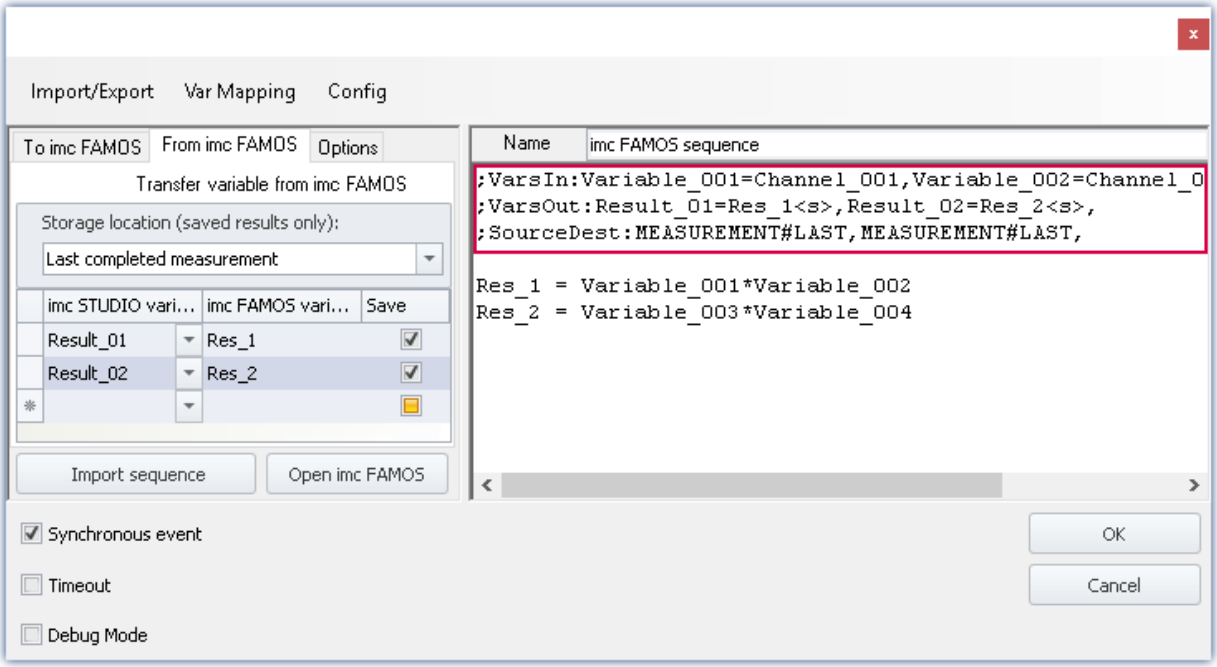
Using a sequence which works with parameters

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):

Variables: To imc FAMOS:

;VarsIn:`Variable_001`
`=Channel_001`
`,Variable_002=Channel_002,Variable_003=Channel_003,Variable_004=Channel_004,`

Description	Syntax
Start	;VarsIn:
First variable name	Name in imc FAMOS
Assignment character	=
Second variable name	Name in imc STUDIO
Separator from next assignment	,

Variables: From imc FAMOS:

;VarsOut:`Result_01=Res_1<s>,Result_02=Res_2<s>,`

Description	Syntax
Start	;VarsOut:
First variable name	Name in imc STUDIO
Assignment character	=
Second variable name	Name in imc FAMOS
Activation of data storage (optional)	<s>
Separator from next assignment	,

Source and data storage location:

;SourceDest:`MEASUREMENT#LAST,MEASUREMENT#LAST,`

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)

Examples:

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

16.2.9.3 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 4.0).

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 1312
- Run script 1317

Note: When is the command "Done"

The command is Done when the `Run()`-method has concluded.

16.2.10 Variables

16.2.10.1 Export variable

The command *Export Variable* 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. All options can be either specified, or selected manually while the Sequencer is running.

File options

- Folder: Specifies the target folder

  If no target folder is specified and the option: "Show Dialog" is not activated, a folder selection dialog appears when the command 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.

Mapping instruction

- 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 Measurement 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)

Variables

- 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.

Variables list

- 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.
- 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.

Other options

- 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 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 the command 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 Sequencer to close.
- Show file options and Show variables options

  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 und 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.

16.2.10.2 Load/Import variable

The command *Load Variable* enables creation of user-defined variables with pre-defined values (from saved files). The variable may not exist before the command is run.

The command *Import Variable* enables setting the values of an existing variable (from saved files). The variable must already exist before the command is run.

The values must be saved in a suitable file, for example one which was created using the commando: *Export Variable* 1645.

It is possible to select one or more variables to be imported or loaded. All options can be either specified, or selected manually while the Sequencer is running.

File options

- Folder: Specifies the folder where the saved files are to be imported. In the selection list, a folder or file can be selected.

  If a folder was selected, one entry appears in the Variables list for each suitable file.

  If a file is selected, it appears in the Variables list.
- Add file

  By this means, you can enter files in the Variables list which do not yet exist. E.g. for a case where the command must be configured now, even though the files to be imported do not yet exist.

  Please enter the full filename with its extension.
- Add file element

  By this means, it is possible to enter additional variables for the files in the Variables list. For example, if you have added a file, but it did not yet contain all of its variables. A file can contain either one or more variables.

  Please select the file in the Variables list and enter a variable name.

Category options

- 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 applies to all variables belonging to the command.

Variables list

- By means of the checkbox, it is possible to de-select for import a variable already entered. This means that it still exists in the list, but is not imported. It can later be activated for import again at any time.
- 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 import.
- File format: Format of the file.

Other options

- 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 not all variables to be imported. The variables in the list are imported in succession and all variables below the non-existent variable are thus not imported.

  In order that the command always imports 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 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.

- Load/Import all: If this option is activated, when the command is executed, all variables existing in the selected folder are imported.

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*.

16.2.10.3 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.

Other options

- 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 not all variables to be deleted. The variables in the list are deleted in succession and all variables below the non-existent variable are thus not deleted.

  In order that the command always deletes 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 Sequencer to close.

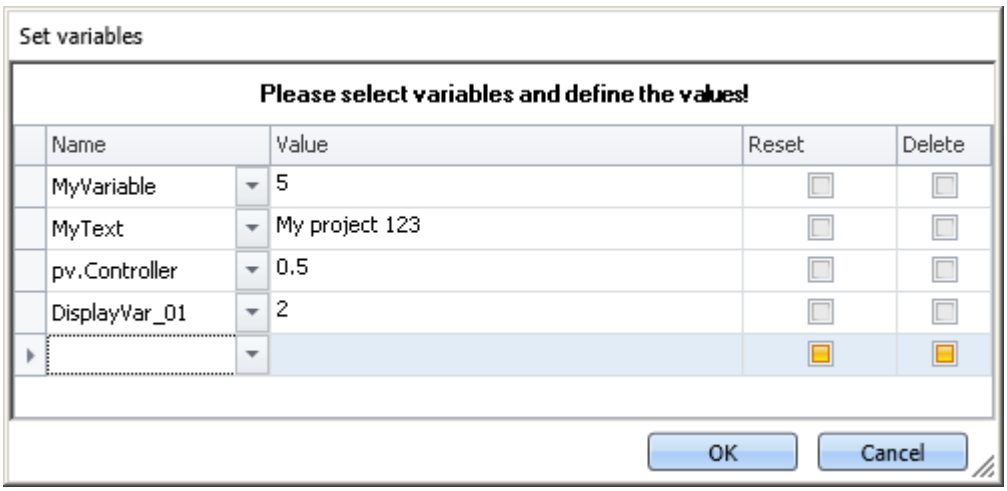
- Show dialog

  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 deletion currently performed.

16.2.10.4 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 write text entries to report channels 1675 and 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 1655 information.

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.

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".

Delete (should not be used)

If the checkmark: *Delete* is set, the variable is deleted. The value entered in the box *Value* is ignored.

No device variables are deleted, nor is any error message posted, when an attempt is made to delete a device variable.

Note

To delete variables, please use the command: Delete Variable 1648. Using this command, it is possible to preclude special cases such as the deletion of illegal variable types.

16.3 Command Sequence

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 plug-ins:

- Sequencer
- Panel
- Automation

Sequence table

Name	Hint
Checkbox1	
Switched Off	Occurs when the switch has been switched off
#01 Execute menu action: Stop (Stop measureme...	Execute menu action: Stop (Stop measurem...
#02 Run imc FAMOS sequence	
#03 Browse in workspace - Panel: Panel/Report	
Switched On	Occurs when the switch has been switched on
#01 Execute menu action: Start (Start measurem ...	Execute menu action: Start (Start measurem..
#02 Browse in workspace - Panel: Panel/Messung	

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 plug-in used.

The structure of the sequence table belonging to the plug-in Sequencer is described in the corresponding manual (see Sequencer > Sequence table 1191).

In the other plug-ins, commands can be linked with specific events. This is done in the sequence table of the dialog: *Event Dialog* 1651.

Creating a sequence of commands

A sequence consists of multiple commands positioned one below the other. (See: Creating a sequence of commands 1652)

Reference

The working of the individual commands is described in the chapter Command Reference.

16.3.1 Event Dialog

Certain *events* can be linked with *commands*. The commands are performed upon occurrence of the event.

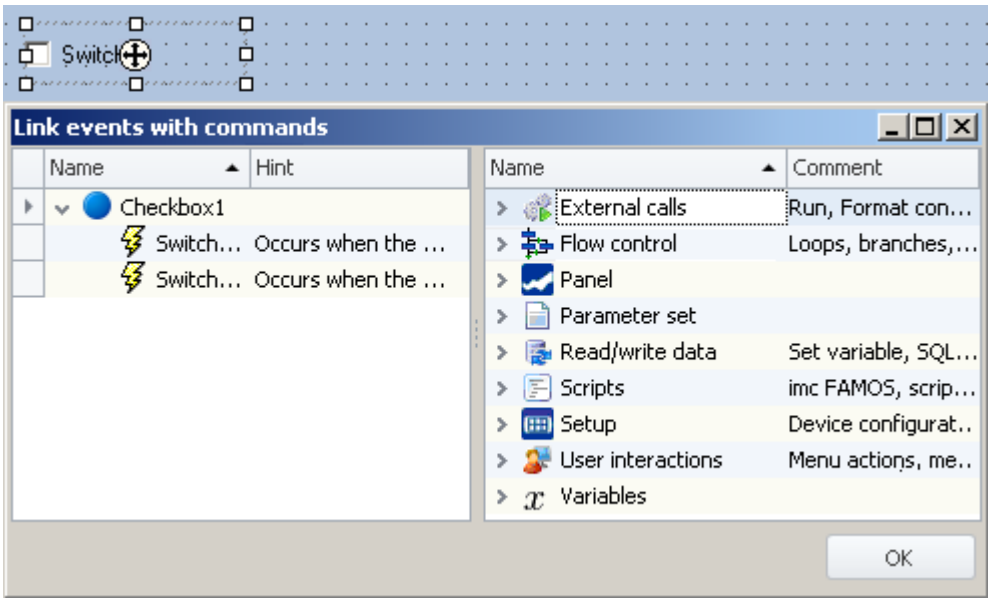
For configuring the event, the event dialog.

Events can, for example, occur in the plug-in Panel when a switch is activated or in the plug-in Automation upon entering or exiting a signal state.

Example

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)



"Event Dialog":
Events belonging to a switch from the group Standard

16.3.2 Creating a sequence of commands

The following examples are illustrated graphically in the Event dialog 1651. However they also apply to the sequence table of the plug-in: Sequencer.

To add a command to the sequence table, there are two options:

- by Drag&Drop 1653
- by the context menu 1654

You can add a command at various positions:

	Before the current position
	Below or within the current position: groups 1192 (Sequencer) or events 1651 (following also called group)
	After the current position

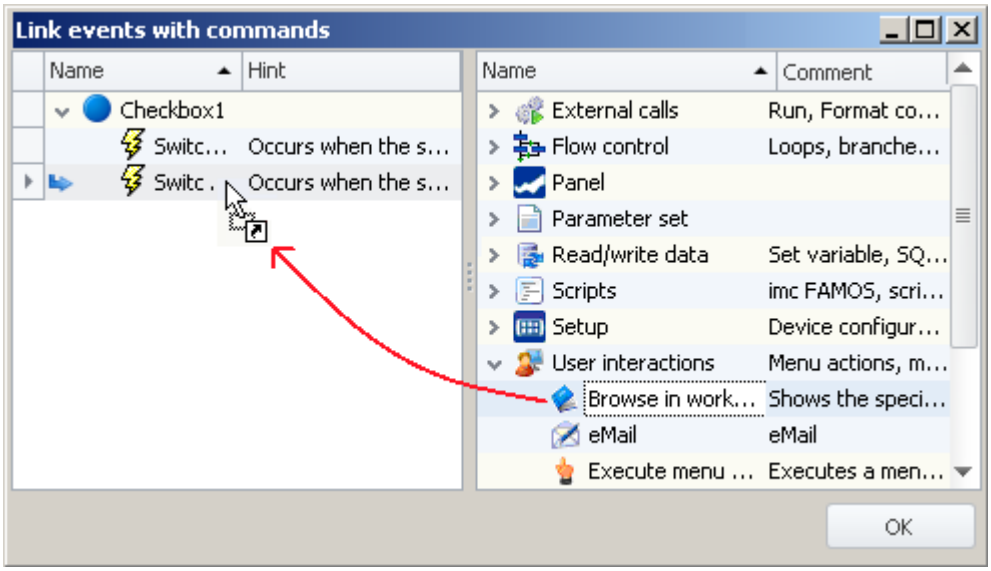
Adding a command by Drag&Drop

Commands can be added by means of Drag&Drop

- in the plug-in Sequencer from the tool window Commands
- in the event dialog from the right side of the dialog.

Adding commands to a group

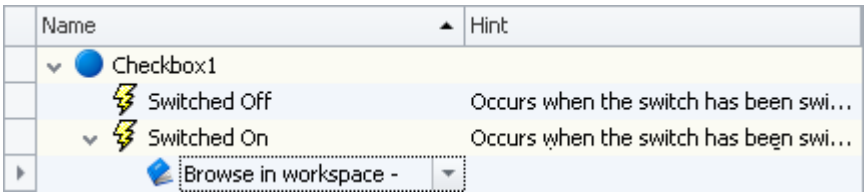
- Use the Drag&Drop technique to move the desired command to the desired position, as seen below:



Example from the "Event Dialog": The command: "Browse workspace"
linked with the event: "Switched On" by means of Drag&Drop

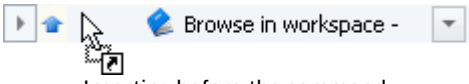
To the left of the even the position is specified ( )

- Release the mouse button. The command is added within the group.

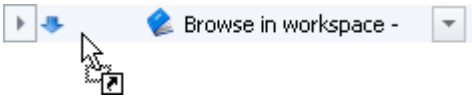


Adding commands before or after existing commands

Here, you can insert additional commands *before* or *after* the 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"

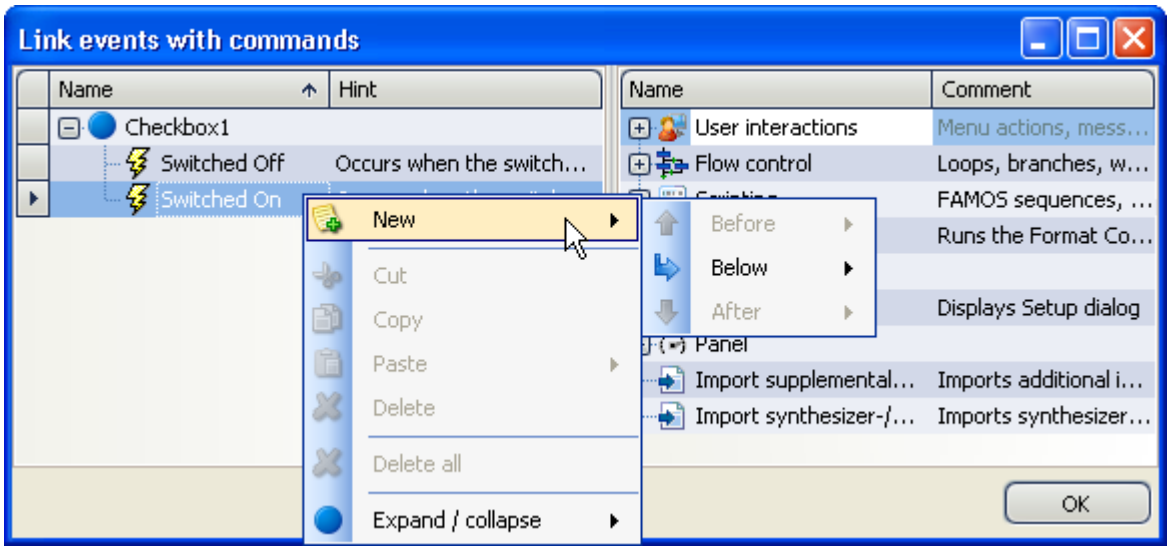
- Release the mouse button. The command is inserted *before*/*after* the existing command.

Adding a command by the context menu

Commands can be added at various positions by means of the context menu.

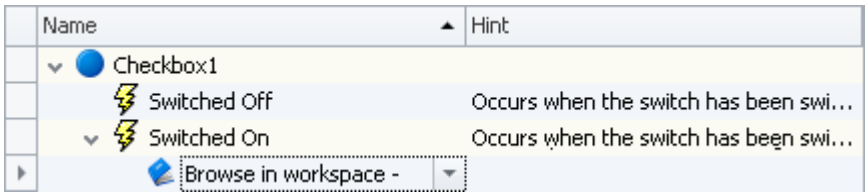
Adding commands to a group

- Open the context menu of the desired position and select *New*, as shown below:



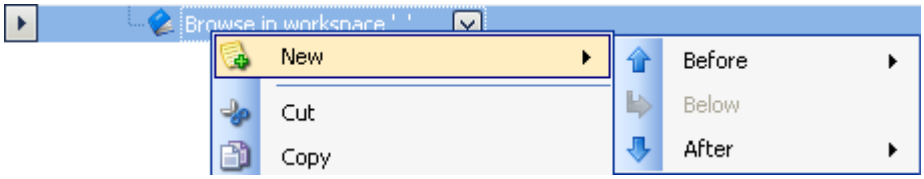
Example from the "Event Dialog": New command
linked with the event: "Switched On" y means of the context menu

- Select the position *Below* ( )
- Select the desired command. The command is added within the group.



Adding commands before or after existing commands

Here, you can insert additional commands *before* or *after* the command. To do this, open the command's context menu and select *New*, as shown below:



Inserting before or after the command:
"browse workspace"

- Select the position *Before*/*After* (/)
- Select the desired command. The command is inserted *before*/*after* the existing command.

Deleting, copying, moving commands

Deleting a command

To delete selected commands, highlight them and select the context menu item *Delete*.

To delete the entire sequence, select the command *Delete all*.

Cutting (moving), copying, pasting

To cut commands or to copy them to other locations, you can use either the context menu or drag them using the Drag&Drop technique to the desired location:

- Drag&Drop: Cut + Paste
- Drag&Drop + <CTRL>: Copy + Paste

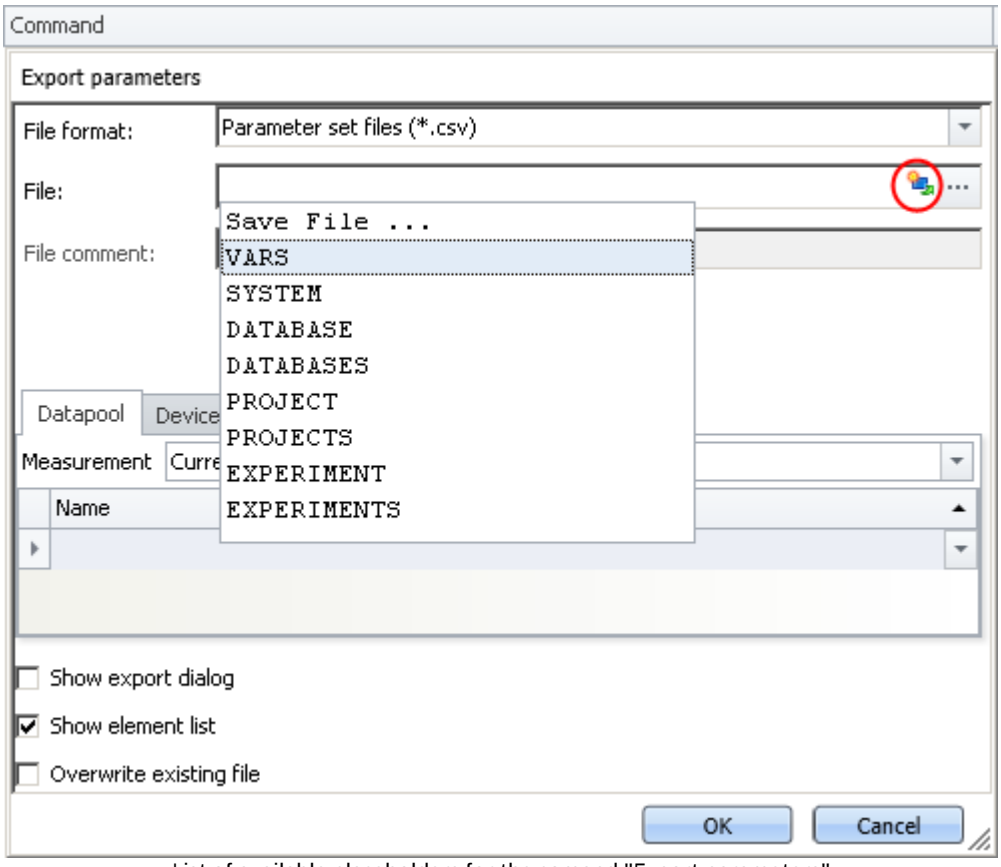
16.4 Placeholders

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 833. A list of all known placeholders is presented in the section list of placeholders 1657.

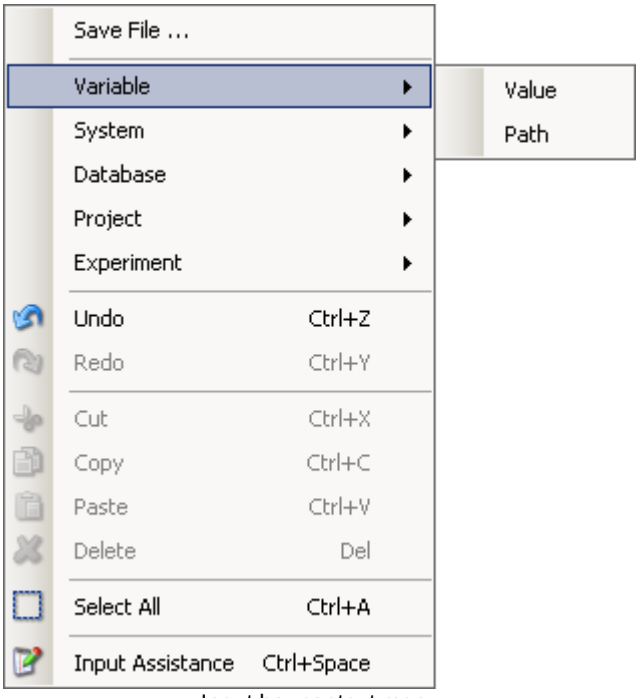
Placeholders are offered where the placeholder symbol ( ) is displayed. Not all placeholders can be used at all positions. To obtain a list of available placeholders, click the left mouse button over the placeholder symbol ( )



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 1657 of known placeholders
- Formatting 1668 of the placeholders
- Environment variables of the OS 1670 supported by imc STUDIO

16.4.1 List of placeholders

The following placeholders are described in this section:

- CONTROL 1657
- DATABASE/DATABASES 1658
- EXPERIMENT/EXPERIMENTS 1660
- PAGE 1663
- PANEL 1663
- PROJECT/PROJECTS 1663
- SELCONTROL 1665
- SETUP 1665
- SYSTEM 1666
- VARS 1667

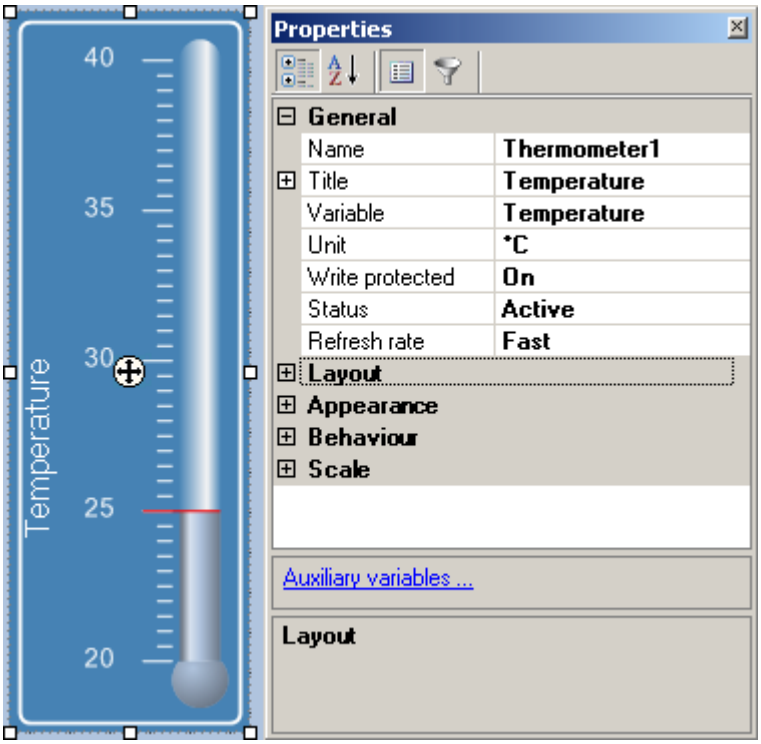
16.4.1.1 CONTROL

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

16.4.1.2 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*.

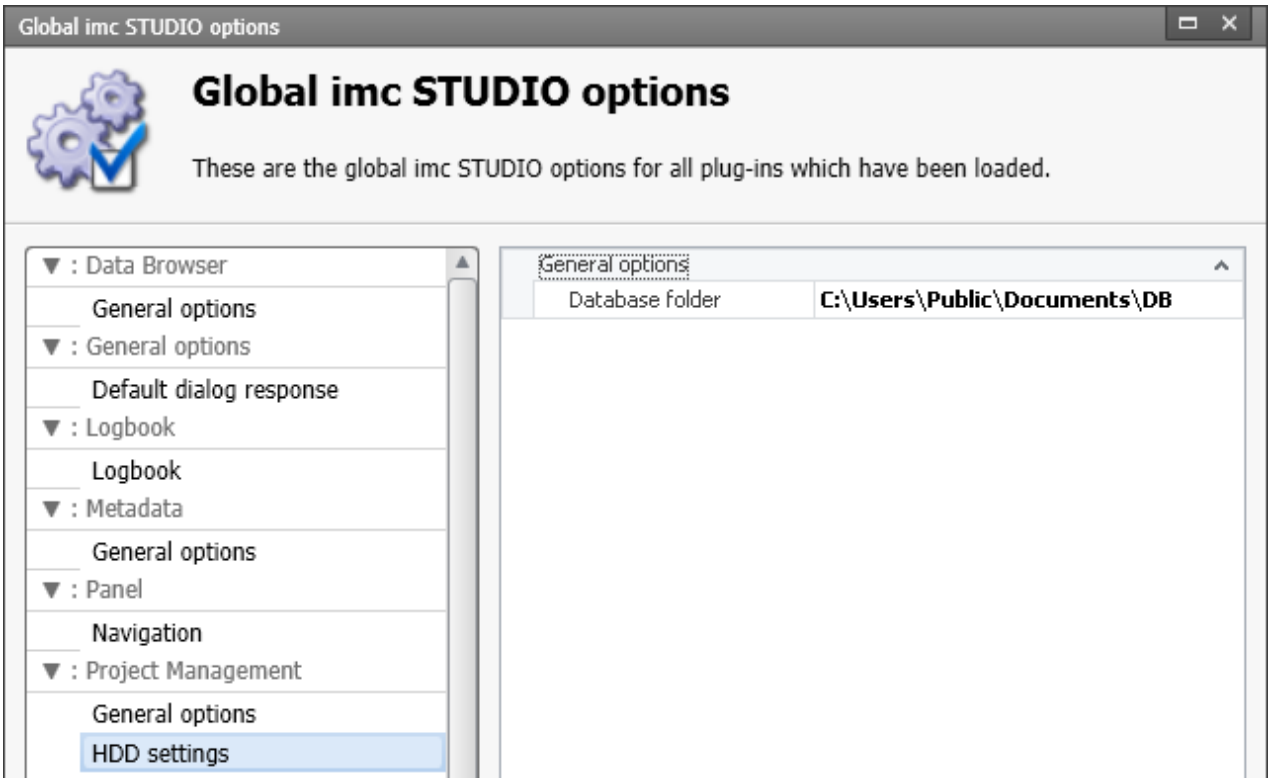
Note

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>

Example

The workings of the two placeholders is illustrated by the following to examples.

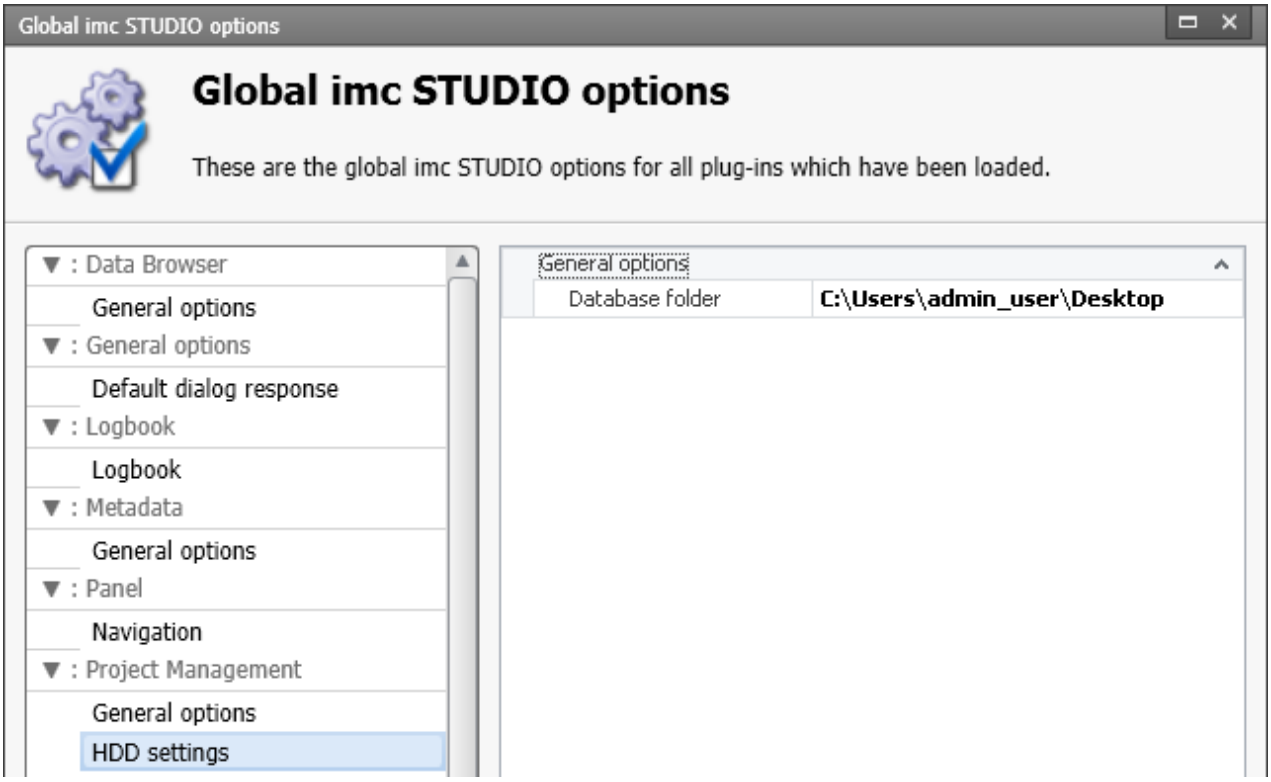
The first example pertains to the unchanged imc STUDIO default settings, here under Windows 7:



Default database path under Windows 7

- <DATABASE.NAME> = DB
- <DATABASE.PATH> = C:\Users\Public\Documents\DB

The second example shows the resolved placeholders when the database path is changed:



Changed database path

- <DATABASE.NAME> = Desktop
- <DATABASE.PATH> = C:\Users\admin_user\Desktop

To obtain the same values by using the placeholder DATABASES, the call must be structured as follows:

- <DATABASES["Desktop"].NAME> = Desktop
- <DATABASES["Desktop"].PATH> = C:\Users\admin_user\Desktop

16.4.1.3 EXPERIMENT/EXPERIMENTS

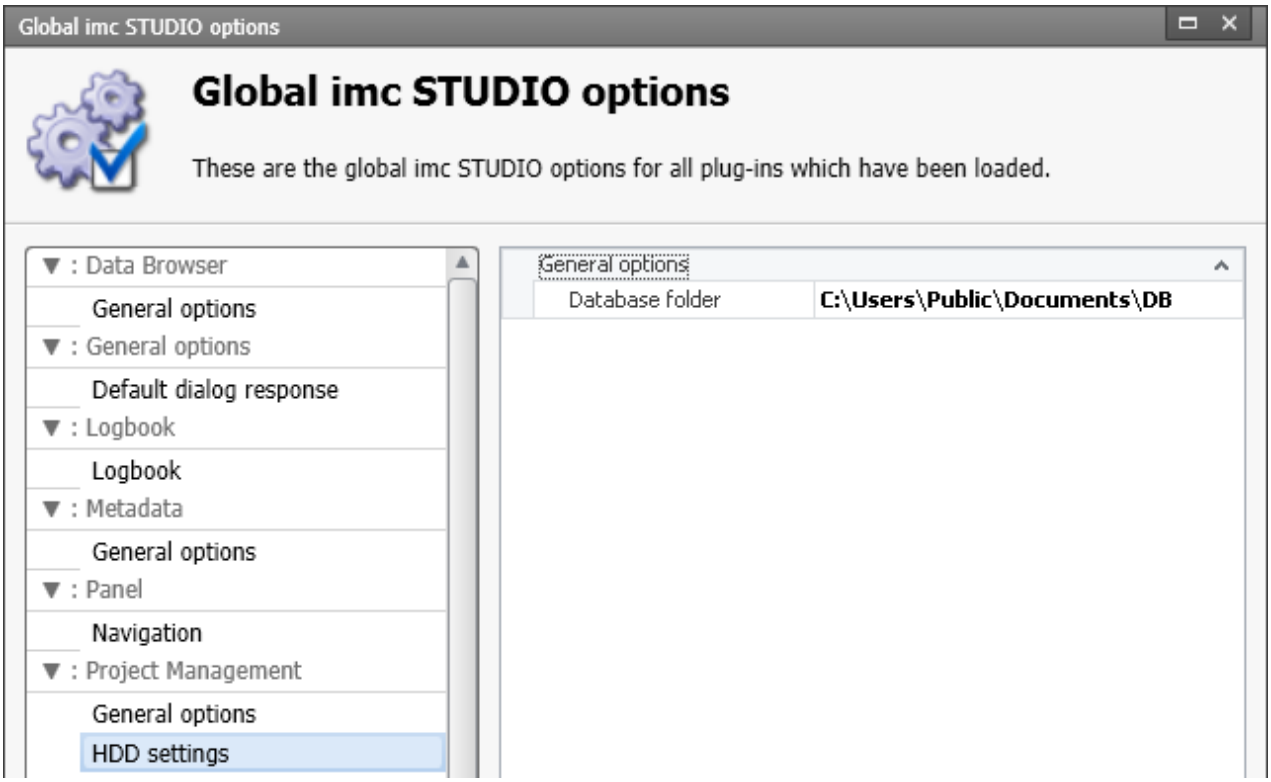
Name	Context menu	Description
CFG	Experiment - Configuration path	Path to the configuration folder (*config* folder) of the experiment
NAME	Experiment - Name	Name of experiment
PATH	Experiment - Path	Path to the experiment's saving folder
SETTINGS	Experiment - Configuration file	Experiment file including 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 7:



Default database path under Windows 7

The project's name is *StandardProject,* the experiment is called *My_Experiment*.

With these settings, the placeholders listed above have the following values:

- <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:

- <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.

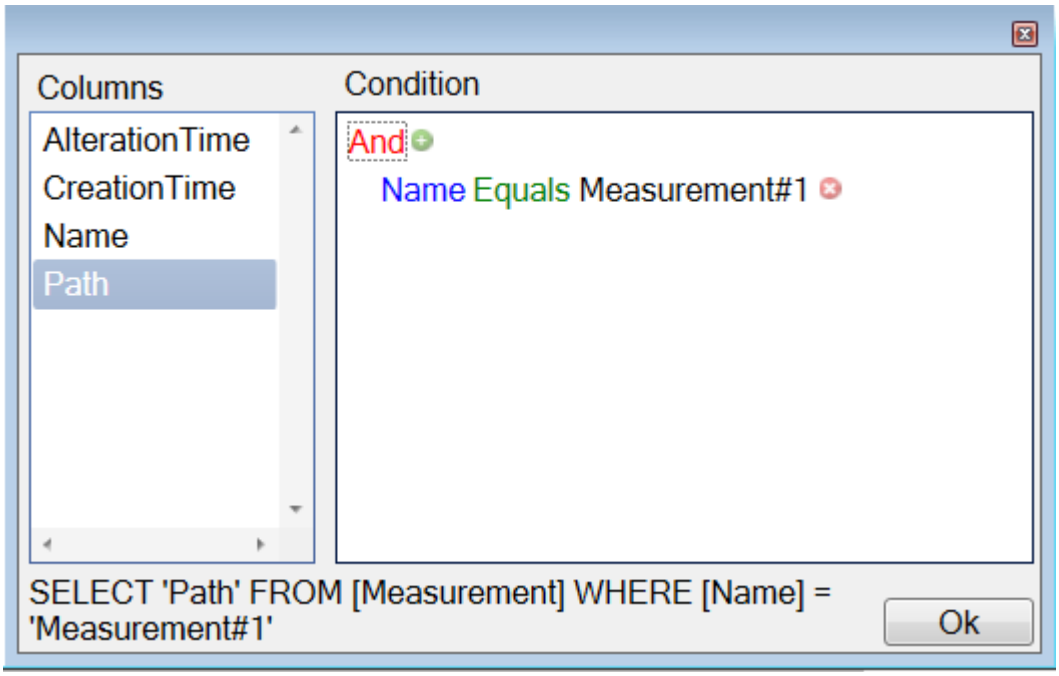
Name	Context menu	Description
CFG	-	*Not supported.*
NAME	-	*Not supported.*
PATH	Experiment - Path	Path to the storage directory of the configuration file ".imcStudio"/".imcExp"
SETTINGS	-	*Not supported.*

16.4.1.4 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:

- 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).

16.4.1.5 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.

16.4.1.6 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.

16.4.1.7 PROJECT/PROJECTS

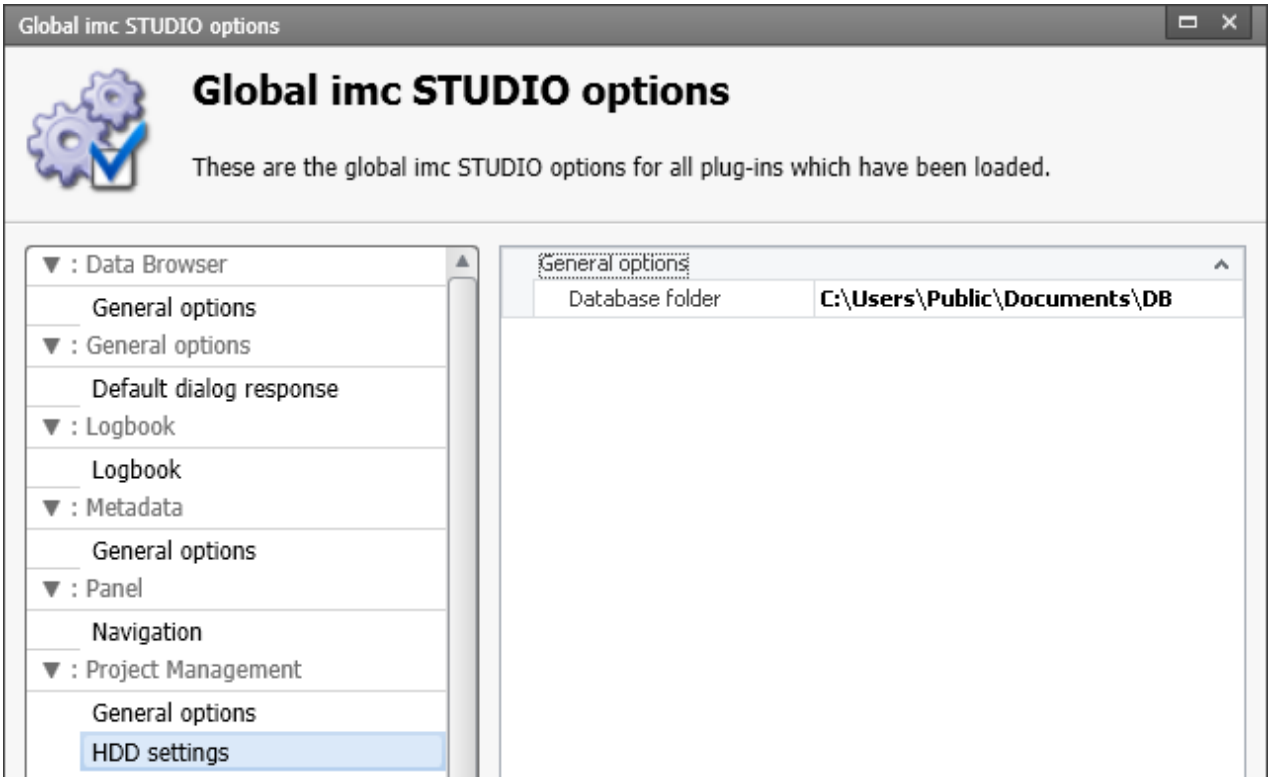
Name	Context menu	Description
CFG	Project - Configuration path	Path to the project's configuration folder (*config* folder)
NAME	Project - Name	Name of the project
PATH	Project - Path	Path to the project's saving folder
SETTINGS	Project - Configuration file	Settings for the project (*.imcAppSettings 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 7:



Default database path under Windows 7

The project's name is *StandardProject.*

With these settings, the placeholders listed above have the following values:

- <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:

- <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

16.4.1.8 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

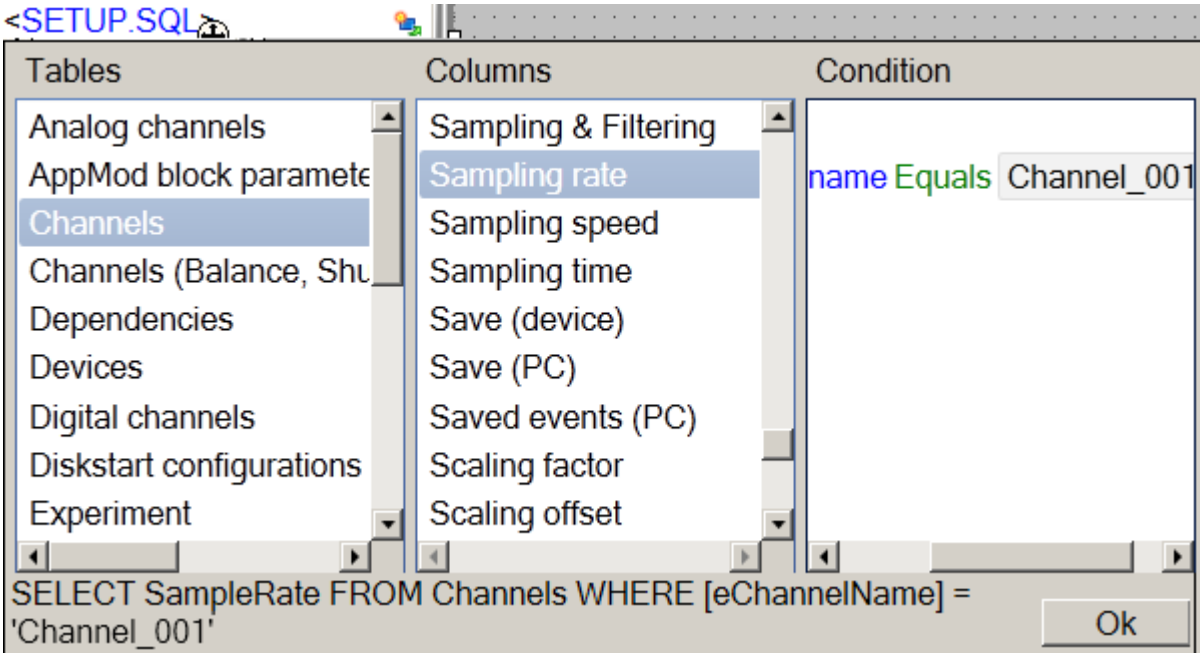
In which case, <SELCONTROL.VARS> = Channel_001,Channel_002,Channel_003.

16.4.1.9 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:

- 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 chapter Table description and complete layout 228.

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.

16.4.1.10 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 1668 and the time 1669 are provided in the section Formatting 1668.

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 1668.

16.4.1.11 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
- Path naming: Continuously numbers
- 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:

- <VARS["myFFT"].COMMENT> = Result of FFT
- <VARS["myFFT"].FILE> = myFFT (01).RAW
- <VARS["myFFT"].NAME> = myFFT
- <VARS["myFFT"].PATH> = C:\Users\Public\Documents\DB\StandardProject\My_Experiment\00000001
- <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 1669.

16.4.2 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 1668
- SYSTEM.DATE 1668
- SYSTEM.TIME 1669
- VALUE 1669

16.4.2.1 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.

16.4.2.2 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 13th, 2011 (operating system English (USA)), the use of the formatting string is illustrated.

- <SYSTEM.DATE> = 7/13/2011
- <SYSTEM.DATE("yyyy-MM-dd")> = 2011-07-13
- <SYSTEM.DATE("yy-M-d")> = 11-7-13
- <SYSTEM.DATE("dd_MM_yy")> = 13_07_11

16.4.2.3 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.

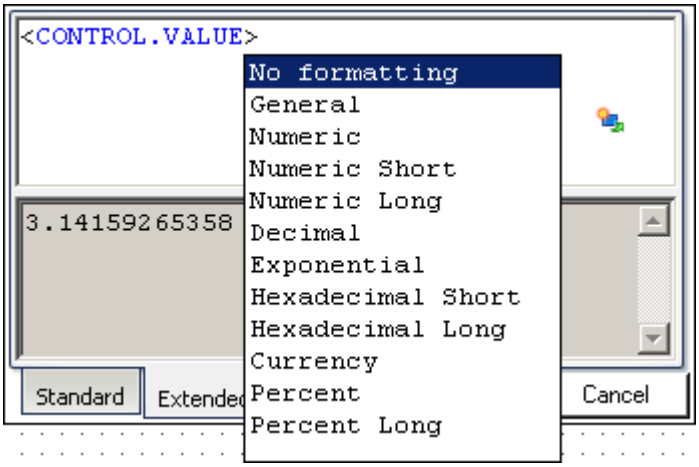
Example

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

16.4.2.4 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

16.4.3 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*.

Notes for 64-bit systems

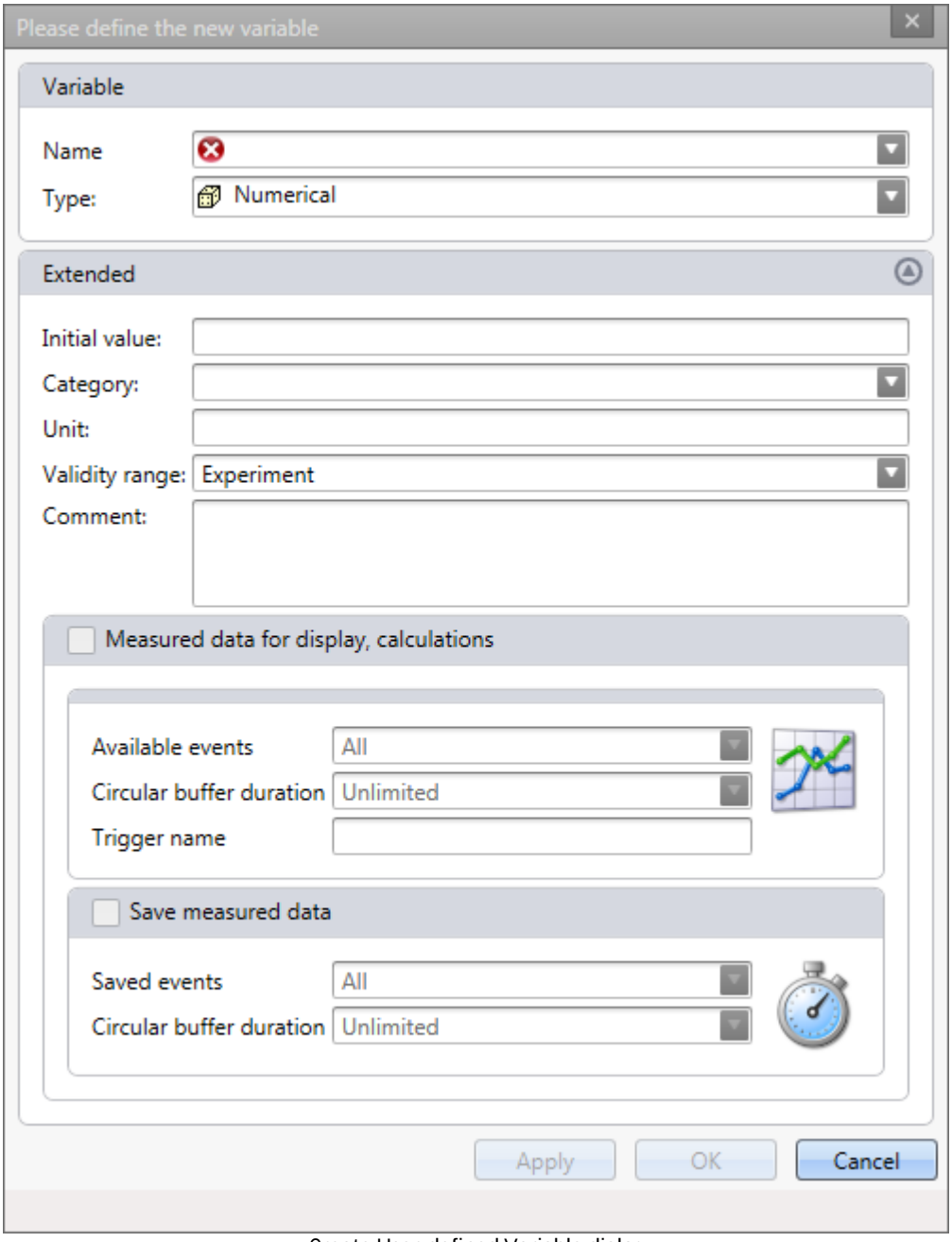
- With 64-bit systems, note that the variables ProgramFiles and CommonProgramFiles (under Microsoft Windows) refer to the respective 32-bit folders. This is because imc STUDIO is a 32-bit program and the variable values returned by the operating system refer to the running program (imc STUDIO). In this case, the operating system returns values to 32-bit paths.
- If you need the 64-bit paths, use the variables ProgramW6432 and CommonProgramW6432.
- For the reason described above, the interface does not support the variable PROCESSOR_ARCHITECTURE for 64-bit systems.

16.5 User-Defined Variables

imc STUDIO provides the user with the ability to create and use user-defined variables.

16.5.1 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* 1649. A dialog opens which helps you in setting up the variable.



Create User-defined Variable dialog

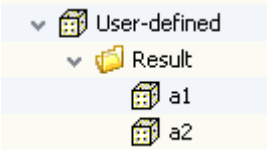
The dialog contains two regions. *Variable* contains an input box for the name and a selection box for the type. In the lower region *Extended*, you can specify additional properties such as the initial value and unit.

Variable

Name

Here, enter a name for your variable. All alphanumeric characters are permitted.

To join multiple variables together in a group, you can use the character '.' in the variable's name, e.g. *Result.a1, Result.a2* returns the following display in the Data Browser:



Display in the Data Browser

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.

Type

For your user-defined variable, you can choose among various types:

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.
Data table	In order to save data which you import from a database by means of SQL, use this data type. The data table is empty when created. Here, you can't specify any initialization value.
Channel	Creates a channel. In these, you can save contiguous, time-referenced data. The channel is empty when created; here, no initialization value is possible. Information on the data storage options is provided under *Extended*.
Report channel	Saves textual or audio commentary; for more information on their use, see the section Report Channel 1675.

Extended

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 1649), 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.

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.

Unit

You can assign a unit to your variables.

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!

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.

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, for example.

Measured data for display, calculations

These settings are only relevant if you wish to create a variable of either of the types Channel or Report Channel.

For the type Report Channel, this property is permanently and unchangeably selected.

A description of the settings for *Saved events* and *Circular buffer duration* is presented under Setup in the chapter Data Transfer 174.

Report channels must always be associated with a trigger, while channels can be associated with a trigger. Ensure that you use a a name which also exists in the Setup.

Save measured data

These settings are only relevant if you wish to create a variable of either of the types Channel or Report Channel. Additionally, the option *Measured data for display, calculations* must be selected in order to be able to select this option.

In order to save your measured data (channels) on the PC, select this option. The saved data are stored in the same folder as the data from your other saved channels(e.g. analog channels).

A description of the settings for *Saved events* and *Circular buffer duration* is presented under Setup in the chapter Data Transfer 174.

16.5.2 Report Channel

Report channels provide the ability to save textual or audio commentary as a channel, along with your measured data. Report channels can be displayed like any other channel, e.g. in the curve window.

Creating a Report Channel

To create a Report channel, first create a user-defined variable of the type Report Channel.

Note: Trigger name

- Report channels are always associated with a trigger and can only be written to while the respective trigger is active.
- If you don't enter a trigger name, then when you click on *OK* or *Apply*, then *Trigger_48* is automatically suggested. In the default case, this is the start-trigger of the first device. However, it may be in some circumstances that this trigger has been renamed and that no *Trigger_48* exists. Ensure that the trigger specified here exists and is active when data are written to the Report channel!
- You can associate the Report channel with any arbitrary trigger, on the condition that at least one additional active measurement channel is dependent on this trigger.

If you wish to save the Report channel along with your measurement data, then when you create it, select the option *Save Measured Data*. The measurement settings (e.g. storage location, interval data saving) will be applied to you measurement channel.

Note: Data storage

Report channels are by nature PC-side variables, which means they can only be saved on the PC; saving to the device is not possible!

Be aware that a report channel can contain either text or audio commentary, but not both. How this channel is used is determined with the first entry in the channel, i.e., as soon as you write text into the channel,only texts can be saved in this channel. For this reason, when creating the channel, you do not need to specify whether the channel is to contain either text or audio commentary later.

Writing to a report channel

While data are being written to the report channel, the trigger to which the channel was assigned when it was created must be active. You can add the text/audio commentary either via special Widgets, or use commands:

Widgets

For text-based report channels, there is the Widget *Text Input for Report Channel*, which is found in the category *Input, Output.* For audio-based report channels, there is the Widget *Audio* (without any category).

For a more detailed description of the Widgets, see the chapter Panel 813.

Note: Context

Report channels containing text entries can additionally save a context along with the text entry. The context is typically the channel(s) in which the event to which the comment refers occurred. How to use the context is described under the Widgets Text Input for Report Channel 1074.

Commands

It is possible to write to text-based report channels by means of the command Set Variables 1649.
For Audio-based report channels, the command Record audio channel 1609 is available.

FAQ

Question: At what point in time are the entries written to the channel?

Answer: If neither the Widget nor the curve window is linked to the Navigation bar 1093, the entry is always issued the time stamp of transmission (initiation of the command, clicking on the Widgets).

If both are linked with the Navigation bar, it is possible to set by means of the Navigation bar's settings 1101 whether the behavior described above is desired or whether the entry is to be inserted at a marker (Slider 1096).

Searching for, and subsequent changing of entries

Note: Navigation

In order to be able to navigate through the channel entries, the channel must be linked with the Navigation bar. Explanations of this topic, as well as of navigation through the entries, are presented in the chapter Panel in the section Navigation Bar 1093.

Note: Post-processing

In order to edit the report channels belonging to concluded measurements, the respective channels must be checked out by means of the *Data Browser* (context menu > Check Out). Next, make the desired changes. In order for the changes to be saved, the channels must be checked in after the editing (context menu > Check In).

The next paragraph describes subsequent editing of report channels. More detailed information on the Navigation bar, the slider and the navigation capabilities are presented in the chapter Panel in the section Navigation Bar 1093.

In order for the channels to be edited, the following three conditions must be met:

- The channel to be edited must be linked with a corresponding *Widget*; symbolic measurement names (Measurement #x) are also possible.
- The channel to be edited must be linked with the Navigation bar.
- The Widget linked with the channel must also be linked with the Navigation bar.

FAQ

Question: Why do the measurements need to be checked in/out?

Answer: Saved measured data are write-protected. By means of checking out, a writable copy of the original data is created which can then be edited. By means of checking in, this copy is saved and displayed as a new, valid version. However, since the editing is performed on a copy, the original state can be restored at any time.

Changing entries

The changing of an entry always consists of deleting the existing entry and inserting a new entry at the same location. Direct changing is not possible.

Inserting entries

To insert an entry, first use the Navigation bar to set where the entry is to be inserted. The insertion position of the new value corresponds to that of the slider. Once the slider is at the desired location, apply the linked *Widget* to insert the value.

Deleting entries

Open the dialog *Search in Variables* (see the chapter Panel 813). There, highlight the entry to be deleted using the mouse, and click on the Delete key. For this purpose, the Widgets does not need to be linked with the Navigation bar.

17 Technical Data Sheets

Current technical data sheets and further information can be found on www.imc-berlin.com/download-center.

17.1 imc STUDIO

imc STUDIO is the common framework uniting various imc software components (plug-ins) to a modular system. Specific combinations of plug-ins comprise Editions (product packages), each designed for particular applications.

Editions

The following editions are available for imc STUDIO, and each include a certain basic package of plug-ins/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

Additional optional or individually licensable plug-ins can also be integrated.

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

Installation varieties

For imc STUDIO there are various installation varieties, parts of whose configuration options can differ substantially.

Installation variety	Required licenses	Description
Default variant: imc STUDIO	One imc STUDIO Edition	imc STUDIO allows configuring of measurement devices, performance of measurements, and monitoring of instantaneous readings. Data can be viewed and edited live at one work station.
For special applications: imc STUDIO Monitor	One imc STUDIO Edition + imc STUDIO Monitor	imc STUDIO Monitor makes it possible to connect with one or more measurement devices during a running measurement, particularly for the purpose of monitoring the instantaneous readings. Data can be viewed and edited live on multiple workstations.

Available plug-ins/components

Below is a table correlating plug-ins and their functions to the individual editions.

Available plug-ins/components	Description	imc STUDIO Edition			MON *
	●: included ○: optional ⦿: subset of full feature set	STD	PRO	DEV	
imc STUDIO	Framework	●	●	●	●
	User administration: Access control, roles and rights management		●	●	●
Setup	Device configuration of imc devices	●	●	●	
	Device configuration of 3rd-party (non-imc) devices (e.g. Video or 3rd-party devices, provided appropriate license available)	○	○	○	
Monitor	Multi-client monitoring and visualization				●
Panel	Display of measurement data	●	●	●	●
Widgets: Standard	Gauges and control elements (basic selection)	●	●	●	●
Widgets: Automotive, Industrial, Designer, Aviation	Extended selection and styles		●	●	●
Project Management	Display of saved measurement data in the Data Browser	●	●	●	
	Projects and Experiment templates		●	●	
Sequencer	Batch generator, workflow automation		●	●	●
Automation (1)	Design of real-time test station automation			●	
Scripting	Integration of custom specific user code incl. accessing external hardware (actuators, lab equipment and devices etc.)			●	●
Third Party Device Interface (2)	Integration of devices from manufacturers (Third Party Devices) into imc STUDIO			○	
Data Processing	User interface for configuring mathematics functions DLL-interface for custom specific processing of data streams	●	●	●	●
imc Inline FAMOS (2)	Real-time analysis on data streams (PC based platform and license)	○	○	○	○
Power Quality	Expansion package for power quality analysis conforming to EN 50160 (IEC 61000-4-30 Class A)	○	○	○	○
Bus Decoder (2)	Extension package for decoding of fieldbus log channels	●	●	●	●
Powertrain Monitoring (2)	Extension package for monitoring powertrains	○	○	○	○
Video	Synchronized video acquisition	○	○	○	
imc SENSORS	Sensor database integration	○	○	○	
Sensors	Expansion package for Setup for setting up imc STUDIO sensors and administering the imc SENSORS sensors.	●	●	●	

*: In the appropriate Edition, the function is included in the installation variety imc STUDIO Monitor
1: Requires additional device based license imc Online FAMOS Professional
2: Available for imc STUDIO 5.0R3

17.1.1 Licensing

License activation is performed using the imc LICENSE Manager. An additional second activation is allowed. Find details in the imc LICENSE Manager documentation.

The Edition imc STUDIO Runtime is a restricted version, free of charge. The trial demo version (offering full functionality of imc STUDIO Developer Edition) is for free as well. Both require activation.

17.1.2 System Requirements

Supported operating systems
Windows 10
Windows 8 / 8.1
Windows 7 (32 bit; recommended: 64 bit)
Windows Vista (32 bit) as of SP1
Windows XP (32 bit) as of SP3

Minimum requirements for the PC [1]	Recommended configuration for the PC [2]
Hyper-threading or Dual Core processor with 2 GHz clock speed	Quad Core processor with 2 GHz clock speed or higher
2 GB RAM / 4 GB RAM for Windows 7 and 8 (64 bit)	3 GB RAM (32 bit) / ≥ 8 GB RAM (64 bit)
10 GB free hard disk space (NTFS format)	10 GB free hard disk space (NTFS format)
For installation of the software via DVD an appropriate drive is needed	For software installation via DVD an appropriate drive is needed
Color graphics (16 bit color depth)	True color graphics (32 bit color depth)
Display resolution 1280 x 768	Display resolution: 1280 x 1024 or more
	64 bit operating system

1 A system with minimum requirements is not adequate for connection with multiple devices and complex design tasks with the imc STUDIO Developer. Use such systems preferably only for data monitoring purposes.

2 The requirements for the PC's configuration increase with the number of devices involved and the scope of the Data Processing-calculations to be performed.

17.2 Plug-in imc STUDIO Setup

imc STUDIO Setup is the integrated user interface for the complete configuration of all measurement parameters along with its saving in a system configuration. This user interface can be adapted to individual needs. This provides the ability to link particular settings options to the user's specific level of authorization. Similarly, any interface elements which are not needed can be hidden. As a result, the training required of a user to operate routine experiments is kept to a minimum.

All familiar hardware properties of imc measurement devices are completely supported.

Association of imc STUDIO Setup functions with the imc STUDIO editions

Functionality	Description	imc STUDIO Edition			
	●: included ○: optional ⦿: subset of full feature set	RUN	STD	PRO	DEV
Setup	Device configuration of imc devices	●	●	●	●
	Device configuration of 3rd-party (non-imc) devices (e.g. Video 1695 or 3rd-party devices, provided appropriate license available)	○	○	○	○
	Customizing of configuration tables		●	●	●
	Combined parameters, meta data, pictures, doc etc.			●	●
Layout designer	Free design of custom Setup menus (GUI)				●
Meta data	Augmenting data with descriptive meta information	⦿	⦿	●	●
	Defining custom meta data (additional table columns for experiments and channels)			●	●
imc Online FAMOS	Real-time analysis on data streams (device based platform and license)	○	○	○	○
Sensors	Expansion package for Setup for setting up imc STUDIO sensors and administering the imc SENSORS sensors.		●	●	●
imc SENSORS	Sensor database integration	○	○	○	○

Special functions and applications:

- Uniform operating software for imc's Ethernet-compatible measurement devices (see: Supported imc measurement devices)
- Operation of multiple synchronized devices, networked via Ethernet. The maximum device count depends on the computer’s memory capacity and on the operating system (64- or 32-bit). System configurations having over 30 devices can be achieved
- Setup automatically recognizes the measurement system's capabilities and offers correspondingly adapted configurations (low training requirements - high measurement reliability)
- Setting up a system configuration ("Experiment") is possible without even having a measurement device physically available ("offline")
- Configures auto-start for autonomous autarkic measurement operation (Diskstart/Autostart)
- Supports sensor recognition by means of TEDS conforming IEEE 1451.4.

Supported imc measurement device groups:	
• imc CRONOS*compact* • imc CRONOS*flex* • imc CRONOS-PL/-SL as of 07/2005 • imc C-SERIES	• imc BUSDAQ*flex* • imc BUSDAQ with serial numbers 13xxxx and higher • imc SPARTAN with serial numbers 13xxxx and higher

Basic functions	Description
Channel settings	• All inputs and outputs of a measurement system can be set using one single user interface (analog inputs/outputs, digital inputs/outputs, field-bus channels, virtual channels, etc.) • Per-channel configuration (e.g. name, sampling interval, measurement duration, input range, characteristic curve correction, filters, and much more.) • Capture of imc CRONOS*flex* channel data in 24-bit • Opening independent curve windows, which are not connected with a Panel page
Data processing	• Data saving can be set for each channel separately • Saving of measured data in a different file format (imc Format Converter,e.g. ASCII, EXCEL and more) • Storage location on the PC and / or the device or on a network server • Each trigger event can be saved to a separate data file • Channels can also be parameterized for internal processing only (data not saved) • CAN Log data in the file format: Vector(CANAlyser) possible
File Manager	• Enhances the Windows Explorer® • Enables copying and deleting of files and folders from the devices internal storage to a PC.
Trigger-Machine	• Either directly started or triggered measurement • Starting and/or stopping by trigger • 48 independent triggers supported • Pre-triggers adjustable • Various definable events (thresholds, time-in-range, signal edges, etc.) • Logical conjunctions of multiple events can form complex trigger conditions • Number of trigger releases freely selectable (multitrigger) • Event-driven digital output
imc Messaging	• Devices having the associated interface (e.g. imc CRONOS*compact*) are able to send text messages in response to particular events. • Available triggering events include all signal transitions in the virtual bits and network bits. • A specific target can be set for each of these message texts. The available message types are email and faxes, and any combination of these. • To send a FAX, a modem supporting the G3-Fax Version 2 or 2.0 is required. The fax machine must also be compatible with one of these two standards. For most modern machines, this should be not present any problems. Emails can be sent either via a modem or via a network interface, if an appropriate server can be reached via the network. With the help of services available through the Internet, it is also possible to forward emails as SMS or fax messages. • If a GSM modem or a GSM mobile is used, the SIM card does not need to be activated.
Balancing and taring function	• Setting of the scaling and balancing performed on a per-channel basis and the results are displayed for the current experiment. • Export / import for adjustment settings for all or selected channels
Sensor Recognition	• Readout of sensor information from TEDS • Adoption of sensor information in the sensor database (imc SENSORS)

Optional functions	Description
Sensor Database imc SENSORS (optional)	For the purpose of editing sensor information, the turnkey, system- independent sensor database imc SENSORS can be integrated. For more information on the sensor database, see the "imc SENSORS" data sheet.
Application-oriented functions (optional)	• Device display (internal display or with hand-held terminal) - Display configuration / Display Editor • Timer start • Autostart / Diskstart • Synchronized measurement with multiple devices. Master device generates a DCF77 or IRIG-B(2) signal. • Real-time clock (DCF 77(1), GPS(1), NTP(2), PTP(3) or IRIG B002(2) radio clock for synchronization of sampling clocks and absolute time) • Support of GPS-receiver for capturing geo-position data • Time zones and switch between daylight saving and standard time • Exchange of display variables via the network • imc Online FAMOS: for device based immediate real-time signal analysis (digital filtering, control commands, closed-loop control, FFT, order-tracking) as well as real-time control. • Synthesizer and PID-controller module • Process vector • Synchronous Tasks, imc Online FAMOS Professional necessary

1 : supported by devices as of group 2 and higher

2 : supported by devices as of group 5 and higher

3 : supported by devices as of group 7 and higher with the suffix "-GP"

Supported interfaces	Description
Connection with devices	• Ethernet (LAN) • Modem, external modem for PPP remote control (analog, ISDN, GSM) • WLAN (54 MBit) • Device configuration via FTP • imc REMOTE LinkSecure: Device access by means of imc LINK via secure https access (for imc devices of serial number 14xxx) • imc REMOTE SecureAccess: Access via a secured https-access for imc devices of serial numbers 14xxx and higher. • imc REMOTE Webserver: Provides platform-independent remote access to imc measurement systems (for the purpose of device operation via internet browser) • The connection to the device can optionally be password-protected • Up to 3 (SN 12xxx) or 4 (SN >13000) imc STUDIO Monitor or imc REMOTE plus 3 Microsoft-Explorer connections per device *Example: Device with SN19123 can simultaneously maintain connections with 2 imc STUDIO Monitor, 1 imc REMOTE WebServer and 1 imc REMOTE LinkSecure connections at the same time. Additionally, 3 computers can access the internal data carrier via the Microsoft-Explorer.*
Fieldbusses	• e.g. CAN + protocols, CAN FD, ARINC, LIN, FlexRay, AFDX, XCPoE, J1587, MVB, IPT • imc CANSAS configuration via imc STUDIO Setup, Vector database import (optional)

Data formats	Description
Displayvariable	Float $\pm 10^{30}$
imc Online FAMOS	Local variable: Float $\pm 10^{30}$ PV-variable as Float: $\pm 10^{30}$ PV-variable as Integer: $\pm 2^{31}$

Maximum channel count per device

Type	CRC-400(GP), CRSL, CRPL, C-SERIE(-N), SPARTAN, BUSDAQ, BUSDAQ*flex*	CRFX
All active channels in total incl. their monitor channels	512	
Active Analog inputs incl. their monitor channels	198	128(2)
Analog inputs (active + passive)	240 (+240 Monitor channels)	
Field-bus channels (active + passive) incl. their monitor channels	1000	
Incremental counter channels	16 (+16 Monitor channels)	
DIO-Ports + DAC-Ports (modules with analog outputs)	16(1)	
Process vector variables	800	

1 : Examples: one DO-16 module correspond to one DIO-Port /
one DI8-DO8-ENC4-DAC4 module correspond to 2 DIO-Ports + 1 DAC-Port

2: Incl. flex-output channels of type DAC-8 and Flex-DIO-Port of type DO-16, DI-16.

Additional imc STUDIO software options for devices:

Components	Description
imc DEVICES	Device Driver, Firmware and Web-Server. Also: alternative but limited and restricted form of operating software. Comprises a subset of functions such as storage, trigger configuration, messaging etc.
imc Online FAMOS (Professional)	imc Online FAMOS offers a large number of real-time functions for pre-processing. The pre-processing is performed by a digital signal processor (DSP) in the device.
Online class-counting	Extension for imc Online FAMOS: Class-counting and Rainflow counting for fatigue analysis
Online order tracking	Extension for imc Online FAMOS: order tracking analysis of rotating machinery
Vector database interface	Import of *.dbc CAN configuration files
ECU protocols for CAN Interface	Support for complex ECU protocols (CAN-Bus)
imc CANSAS configuration software	Assistant for the configuration of imc CANSAS modules.

Components	Order code	CRFX, CRC, CRSL, PL, C-SERIES(-N)	SPARTAN, BUSDAQ, BUSFX
imc DEVICES		●	●
imc Online FAMOS	DEV(1)/OFA	●	○
Update of imc Online FAMOS on imc Online FAMOS Professional	DEV(1)/OFA-UP	○	○
Online class-counting	DEV(1)/ONLKLASS	○	○
Online order tracking	DEV(1)/ONORDER	○	○
Vector database linkage	DEV(1)/VEC-DATB	○	○
ECU protocols for CAN Interface	DEV(1)/ECUP	○	○
imc CANSAS configuration software	CAN/CONSOFT	○	○

●: included ○: optional --- : not available in imc STUDIO

1 : DEV is to be replaced with the device's order code abbreviation.

17.3 Plug-in imc STUDIO Panel

The imc STUDIO Panel provides, in addition to the familiar imc curve window, a wide scope of new graphical display possibilities.

It is possible to create report pages for documentation of measurement and analysis results.

Association of imc STUDIO Panel functions with the imc STUDIO editions

Functionality	Description	imc STUDIO Edition			
	●: included ○: optional ⦿: subset of full feature set	RUN	STD	PRO	DEV
Panel	Customized visualization of data, including printable pdf reports with measured data		●	●	●
	Individual design and layout, repository		●	●	●
Widgets	Gauges and control elements (basic selection)		●	●	●
	Extended selection and styles (Automotive, Industrial, Designer, Aviation)			●	●
Full screen view	Hidden menus and restricted access (via full screen Panel)			●	●
Data Browser	Management of stored data: browse, search and filter		●	●	●
	Structuring according to meta data		●	●	●
Panel control elements	Interactive launch of actions/commands via Panel control elements (Sequencer commands 1687 such as calling Panel pages, applying imc FAMOS functions to channels, ...)		●	●	●
Runtime functionality	Execute but not modify all functions (incl. Widgets, full screen)	●	●	●	●

Functions

	Description
Special functions and applications	• The layout of the report pages can be designed freely and be exported in PDF (report mode). • Compositions of imc STUDIO Widgets in freely configurable pages (dialog mode) • Special imc STUDIO Widgets can be assigned to commands. (e.g. starting measurement or a imc FAMOS analysis)
Basic functions	• Creation of multiple pages in which imc STUDIO Widgets (display and control elements) such as curve windows, potentiometers, scales, state indicators can be positioned in any arrangement. • Synchronized navigation through the data sets in multiple curve windows along one scaled time axis

Data Browser	Description
Display and navigation through data	• Structured according to channel category or sorted by name • Searching and filtering • Navigation by means of tree diagram, selection of either fixed or variable channels for visualization purposes
Current measurement	• Displays all channels and variables belonging to the current measurement
Saved measurements	• Loads and processes saved measurements • Displays all channels and variables belonging to the saved measurements Note: Not available in the installation variety imc STUDIO Monitor
User defined variables	Declaration of variables • Numerical, Text, Data table, Channel, Report channel
Display	• All available channels and variables can be linked with Widgets • Opening of independent curve windows not associated with the Panel • Opens an independent values window with the current values of certain variables
Export / Import	Export variable • individually or all in one file Import variable • imports value to an existing (device or user-defined) variable Load variable • Creates a variable with the properties of the variable to be imported

Widgets	Description
Widgets	• imc Curve window • Standard control elements such as state indicators, edit boxes, numeric inputs, tables, buttons, switches etc. • Extended Widgets are available in four pre-defined styles (Automotive, Industrial, Designer, Aviation) • Control elements such as state indicators, edit boxes, potentiometers, thermometer, bar meters, tables, buttons, switches etc. • A repository is available in which settings for the Widgets can be saved
Skin	• Selection among multiple pre-defined skins for Widgets • Definition of one's own skin possible
Extra functions	• Copying and pasting of Widgets • Loading and saving of curve window configurations • Multi-selection of Widgets and various options for orientation and anchoring • Widgets can be grouped

Panel page	Description
Extra functions	• Loading and saving, copying and pasting of pages • Zooming to optimal possible display • Adapting the page to the respective monitor size • Access rights definable for each page separately

17.4 Plug-in imc STUDIO Sequencer

imc STUDIO Sequencer is the plug-in to create an automated measurement workflow. A sequence of actions is designed by means of a graphical Editor in an action table. Alternatively, command sequences can be coupled with events. Global, pre-defined system events as well as user-defined events can be used.

For the purpose of performing analysis by means of imc FAMOS, interaction between imc STUDIO and imc FAMOS is possible. The prerequisite is installation of an imc FAMOS version of 6.1 or higher, as a Runtime, Professional or Enterprise Edition (see Additional imc software products 1702).

Association of imc STUDIO Sequencer functions with the imc STUDIO editions

Functionality	Description	imc STUDIO Edition			
	● : included ○ : optional ⦿ : subset of full feature set	RUN	STD	PRO	DEV
Sequencer	Batch generator, workflow automation			●	●
Runtime functionality	Execute but not modify	●	●	●	●
External applications	Launch external code (DLL) and executables (exe)			●	●
imc FAMOS	Create analysis sequences, extensive post processing	○	○	○	○
imc FAMOS Runtime	Execute automated analysis sequences (instant post processing)		○	○	○

17.5 Plug-in imc STUDIO Automation

imc STUDIO Automation allows the implementation of real-time control for test stand automation. Definition and configuration of the control structures and routines is performed on the PC with graphic oriented environment provided by this plug-in. The resulting routines are automatically compiled to code which is directly executed on the measurement device itself. Execution involves the real-time capable platform imc Online FAMOS.

For the purpose of performing analysis by means of imc FAMOS, interaction between imc STUDIO Automation and imc FAMOS is possible. The prerequisite is installation of an imc FAMOS version of 6.1 or higher, as a Runtime, Professional or Enterprise Edition (see Additional imc software products 1702).

Measurement device requirements
The same hardware prerequisite apply as for imc STUDIO Setup.
An extra device option is required: imc Online FAMOS Professional

Association of imc STUDIO Automation functions with the imc STUDIO editions

Functionality	Description	imc STUDIO Edition			
	●: included ○: optional ◉: subset of full feature set	RUN	STD	PRO	DEV
Automation	Design of real-time teststand automation				●
Runtime functionality	Execute but not modify	●	●	●	●
imc FAMOS	Create analysis sequences, extensive post processing	○	○	○	○
imc FAMOS Runtime	Execute automated analysis sequences (instant post processing)		○	○	○

Functions

	Description
Special functions and applications	• Real-time capable process control (state-based control model) • Trace info (tracking of current states during execution) • Limit value monitoring (background supervision of thresholds) • Exception raising and error handling • Graphically oriented definition of the state model (Drag & Drop) • Additional integration of PC interactions (running Sequencer commands, calling Panel pages and applying imc FAMOS functions to channel data) • Quick and easy design of individual user interfaces (GUI) by means of Drag & Drop
Features	• Graphical display of the task flow • Up to five parallel, synchronized tasks can be performed per measurement device, in real-time with selectable cycle intervals of 100 µs to 1 s.

17.6 Plug-in imc STUDIO Scripting

The imc STUDIO Scripting is an imc STUDIO plug-in which provides a programming interface (C#, .NET). It comes with the editor SharpDevelop as the development environment.

Along with the ability to freely program routines, Scripting also gives the user access to all major functionalities of imc STUDIO, such as:

- Setup: reading and writing of device and channel configurations,
- Panel: access to the Panel and the Widgets,
- access to the Data Browser: creation, reading and writing of variables,
- processing of channels measurement data with imc FAMOS functions,
- execution of menu actions,
- running Sequencer commands,
- response to events.

The following mechanisms are provided for running scripts:

- as a command
  - in the Sequencer
  - to Widgets
  - in response to events
- in the background
  - linked to a Panel page
  - linked to the experiment or the project

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 scripts (also in DLL format) and to import them.

Association of imc STUDIO Scripting functions with the imc STUDIO editions

Functionality	Description	imc STUDIO Edition			
	●: included ○: optional ⦿: subset of full feature set	RUN	STD	PRO	DEV
Scripting	Integration of custom specific user code incl. accessing external hardware (actuators, lab equipment and devices etc.)				●
Runtime functionality	Execute but not modify scripts	●	●	●	●

17.7 Plug-in imc STUDIO Third Party Device Interface

Using the plug-in imc STUDIO Third Party Device Interface,it is possible to integrate devices from other manufacturers (3rd-party devices) into imc STUDIO and run them in the imc STUDIO system.

For this purpose, a C#-script is implemented which models the 3rd-party device's properties. There is a template which simplifies the process of seamlessly integrating the 3rd-party device and its channels into the existing settings menus and setup tables. In particular, this means that these devices/channels appear in the lists of devices and channels in the imc STUDIO Setup. This thus provides uniform operation style and configuration management. The script must additionally implement the interface to the 3rd-party device in the sense of a data driver.

The plug-in 3PDI is specially suited to enhancing a system configuration consisting of imc hardware with supplemental specialty devices and data sources. However, there is also a license available specifically for operation exclusively with 3rd-party devices, without the use of any imc devices.

Besides the developer framework and the licenses to run one's own self-provided scripts, ready-made implementations are also available for purchase.

All runtime licenses (for running 3PDI scripts) can categorically be operated with any edition of imc STUDIO.

A 3rd Party Device Management is available in all imc STUDIO editions, except Runtime.

The advantages of the 3PDI script are:

- integrated and uniform operation via the Setup page
- consistent format for measured data
- synchronized data

Requirements
C# programming skills (for an unmanaged data transfer also C++)
The 3rd-party's interface must be known

Association of imc STUDIO Third Party Device Interface functions with the imc STUDIO editions

Functionality	Description	imc STUDIO Edition			
	● : included ○ : optional ⦿ : subset of full feature set	RUN	STD	PRO	DEV
Run 3PDI scripts	Usage of 3rd-party devices implemented by 3PDI script. Required License: 3PDI (inclusive/exclusive)	○	○	○	○
Edit 3PDI scripts	Create and edit 3PDI scripts				●
3rd Party Device Management	Add 3PDI scripts (the execution may depends on additional licenses)		●	●	●

Already implemented templates and operative third-party devices

Device	Additionally licenses required for running	Description
AudioDevice	none	With the 3rd-party script "AudioDevice", it is possible to the computer's audio devices (such as the microphone input) as a data source.
ChannelLoader	none	The "ChannelLoader"-script enables files in the imc-format to be played back as a signal during a measurement.
FunctionSimulator	imc STUDIO 3PDI-inclusive or exclusive	The template "FunctionSimulator" makes various signal types (sine, cosine, trapezoid, square wave, ...).
SimplePollDevice and SimplePushDevice	imc STUDIO 3PDI-inclusive or exclusive	These 3rd-party scripts are executable templates and can be expanded accordingly.
AgilentInfniiVision DSCO6014L (Digital Scope)	imc STUDIO 3PDI-DigitalScope	Integrates digital oscilloscopes, namely of the series Agilent InfiniiVision DSO 6014L.
fos4x	imc STUDIO 3PDI-fos4x	Supports devices of the manufacturer fos4x for measurements with fiber-optic sensors and optical strain gauges (Fibre Bragg, FBG).

17.8 Plug-in imc STUDIO DataProcessing

imc STUDIO DataProcessing is the interface for user-defined DLLs for the purpose of data stream processing. The following plug-ins require Data Processing:

- imc Inline FAMOS
- imc STUDIO PowerQuality
- imc STUDIO BusDecoder
- imc STUDIO Powertrain Monitoring

Association of imc STUDIO DataProcessing functions with the imc STUDIO editions

Functionality	Description	imc STUDIO Edition			
	● : included ○ : optional ◉ : subset of full feature set	RUN	STD	PRO	DEV
Data Processing	DLL-interface for custom specific processing of data streams	●	●	●	●
Data analysis	Functions package				
imc Inline FAMOS 1692 (2) 1691	Real-time analysis on data streams (PC based platform and license)	○	○	○	○
Power Quality 1693	Powergrid analysis (EN 50160, IEC 61000-4-30)	○	○	○	○
Bus Decoder 1694 (2) 1691	Extension package for decoding of fieldbus log channels	●	●	●	●
Powertrain Monitoring 1694 (2) 1691	Extension package for monitoring powertrains	○	○	○	○

2: Available for imc STUDIO 5.0R3

17.9 Functions Package imc Inline FAMOS

Processing and analysis of measured data during a running measurement

imc Inline FAMOS is a functions package for Data Processing 1691.

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:

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 Trigger_48 (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.

License:

A license must be purchased in order to use imc Inline FAMOS. In contrast to imc Online FAMOS, this license is not bound to the device used, but rather to the imc STUDIO installation on the PC.

Requirements to the device firmware (imc DEVICES)

If writing DAC with imc Inline FAMOS is required, it is absolutely necessary using firmware version 2.8R7 build 15th August 2015 or newer.

Enhancement Kit

Overview	Description
imc Inline FAMOS ClassCounting	This expansion kit for imc Inline FAMOS contains not only familiar functions for endurance strength analysis but also some very practical new ones. Diverse counting and classification procedures, particularly Rainflow analysis, are available. A license must be purchased in order to use imc Inline FAMOS ClassCounting.

imc Inline FAMOS ClassCounting	Description
Rainflow	The Rainflow matrix and the Residue are calculated on the basis of a loading time function. Numerous options are provided in order to be able to adapt the algorithm to the user's needs.
Standard class-counting procedure	Some class-counting procedures conforming to DIN 45667 (such as level-crossing, Histogram) are available. Particularly helpful are functions for determining one- and two-dimensional histograms. Even one- and two-dimensional revolution class counts can be calculated.
TrueMax procedure	Even in a correctly sampled signal, extreme values do not always coincide with samples. However, for a Rainflow analysis, correct capture of these extremes value is crucial. Simply use the procedure specially designed for this application.

17.10 Functions Package Power Quality

Expansion package for power quality analysis conforming to EN 50160 (IEC 61000-4-30 Class A)

imc STUDIO PowerQuality is a function package for Data Processing 1691.

This plug-in allows the calculation of power quality indicators according to EN 50160 (IEC 61000-4-30 Class A). The calculations are performed by the PC on the basis of the currently measured data while measurement proceeds. This utilizes the PCs computation resources.

The following functions are available:

- Single-phase power calculation
- Three-phase power calculation (delta- and star-configuration).
- Saving of the result channels

Each of the functions provides a number of calculated characteristic indicators:

- Power
- RMS
- Frequencies
- Harmonic distortion
- Flicker
- High-frequency components
- Interharmonics

The results returned can be displayed on Panel pages and saved with the associated measurement data.

17.11 Functions Package Bus Decoder

Expansion package for decoding Field-bus log channels

imc STUDIO BusDecoder is a package of functions for Data Processing 1691.

This plug-in allows either all or individual measurement channels belonging to a log-channel to be decoded/extracted. A log-channel can be a logged Field-bus communication ("Logfile").

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. 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.

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.

17.12 Functions Package Powertrain Monitoring

The component imc STUDIO Powertrain Monitoring has been developed in close cooperation with the company GfM (Gesellschaft für Maschinendiagnose mbH), experts in machine and bearing 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 on the basis of characteristic values and a Advanced Diagnosis on the basis of a frequency selective search of kinematic pattern.

A configuration for a specific gear type is set up in the imc STUDIO project, from where it can be distributed to different test locations.The actual application is run in the experiment by assigning the inputs to the physical measurement channels in accordance with the configuration selected. This makes it possible to use the same configuration on different measurement systems if multiple test locations are driven with the same powertrain type.

For Powertrain Monitoring, an extra license is required, which is available from imc Meßsysteme GmbH. It can be combined and run with various base-editions of imc STUDIO.

17.13 Plug-in imc STUDIO Project Management

imc STUDIO Project Management administers all files accruing from both the configuration and data acquisition processes. A database is created in which all configurations (experiments) and measured data are stored. Settings applicable across the experiment boundaries, as well as the experiments themselves are saved in projects. These associated settings are available for all experiments belonging to the respective project. Creating multiple projects allows a clear structure to be maintained.

Basic functions:

- Management of multiple projects and their respective settings
- Export/Import of projects or selected experiments
- Creation of multiple experiment templates with arbitrary default settings

Association of imc STUDIO Project Management functions with the imc STUDIO editions

Functionality	Description	imc STUDIO Edition			
	●: included ○: optional ⦿: subset of full feature set	RUN	STD	PRO	DEV
Project management	Experiment templates			●	●
	Administration and grouping of experiments within projects			●	●
	Display of meta data while selecting experiments		●	●	●
	Measurement storage area		●	●	●

Functions

	Description
Basic functions	• Management of multiple projects and their respective settings • Export/Import of projects or selected experiments • Creation of multiple experiment templates with arbitrary default settings • Display of meta data while selecting experiments
Measurements	• Enables displaying of saved measurements in the Data Browser • Measurement storage area: Allows the measurements to be saved in varying (including variable-dependent) folders.

17.14 Plug-in imc STUDIO Video

imc STUDIO Video is the imc STUDIO plug-in for capture and processing of video data. Simultaneously and in synchronization with data capture by imc measurement devices, video data from cameras are recorded. The cameras must be connected to the operating PC on which imc STUDIO is running.

The achievable data rate depends on the PC's performance.

Association of imc STUDIO Video functions with the imc STUDIO editions

Functionality	Description	imc STUDIO Edition			
	●: included ○: optional ⦿: subset of full feature set	RUN	STD	PRO	DEV
Video	Synchronized video acquisition	○	○	○	○

17.14.1 Specifications and Notes

Data capture:

- Two measurement channels per camera are available simultaneously: the main channel and the monitor channel. Among other things, this means that you may apply different settings for the sampling rate and triggering.
- Main channel for high-speed capture and storage, e.g. for triggered snapshots.
- Duplicated monitor channel at low-speed sampling rates, e.g. for untriggered, continuous long-term measurements.

Visualization:

- A video-Widget (Window) is provided to display video data on Panel pages.
- Multiple such video windows can be placed on the pages of the Panel.
- Each video window can be assigned to either a camera's main or monitor channel.
- In the video window, data are displayed even before release of the trigger.

Trigger:

- The main- and monitor channels can each be assigned to different imc measurement device triggers.
- The imc measurement device's triggers are also the triggers for the camera. This means that video channels are triggered at the same time as the associated imc measurement device channels.
- Pre-trigger: As for the imc measurement device channels, it is also possible to configure pretriggers for video channels. This means that the data recorded can also include images relating to situations prior to the trigger events.
- Pre-trigger duration: 0 sec to 10 minutes.

Synchronization:

- Automatic synchronization of the video- and measurement data.
- The achievable accuracy depends on the workload of the entire system.
  Up to dt = [1 frame duration + 20 ms] is achievable.
- The device must be connected with the PC by an Ethernet line of at least a 100 MBit/s , with a maximum of 1 hub or switch in between. The connection must remain intact for the duration of the entire measurement.

Advisory notes:

- For stable and reliable operation we recommend the use of tested and approved camera models and selected software drivers according to the tables: *Supported Cameras* 1698. imc can only grant support for these combinations listed.
- Systems with cameras from different manufacturers are not recommended, since the drivers could also interact uncontrollably.
- For maximum cable length between the control PC and camera, please refer to the respective technical specs for the camera.

Data throughput / frame rate:

- The data transfer rate is specified as the frame rate (typically 60 fps)
- The frame rate is based on pictures of the size 640 * 480 pixels with 1 Byte per Pixel in Bayer encoding, meaning 300 kByte per frame. This results in 17.5 MByte per second being continually written to the data storage medium.
- With the computer equipped accordingly, in Bayer format up to 100 fps can be achieved for 640 * 480 pixels or 200 fps for 320 * 240 pixels.
- The data transfer rate stated is aggregate. When multiple cameras are used, they split the transfer rate. Thus, a camera with 60 fps has about the same transfer volume as two cameras with 30 fps apiece. One camera at full resolution of 640 x 480 generates about the same data volume as four cameras with 320 x 240 resolution.
- With other encoding (such as RGB instead of Bayer), the data volume is increased threefold. This means the achievable frame rate is reduced to one third.
- With triggered data recording (all video channels not assigned to any 1-Trigger), the achievable frame rate is cut in half due to higher demands on the system made by the circular buffer memory used.

Prerequisites for achieving maximum frame rate:

- Windows 7 operating system. The performance of XP and Vista is lower in certain areas.
- Hard drives: Solid State Disk (SSD) or 3.5” SATA hard drives (at least 5400 revolutions per minute) configured as Raid 0 [1]. Please note that 2.5” hard drives are much slower. Particularly in notebooks, slower hard drives are often installed.
- The data carrier may only be filled to a maximum of 70%. Note that writing to a harddisk that is almost full results in significantly reduced writing speed.
- The data carrier may not be fragmented. Note that high writing speed is only achieved if the write head is not forced to excessive displacements.
- Hard drive controller: This must allow data throughput in write mode. Please note that during measurement operation mode, not only video files have to be written, but other data as well!
- Processor: Quadcore with 2.4 GHz (or in case of using Intel I7, two processor cores should be sufficient).
- Interface to camera: 1 GBit Ethernet, Firewire A or B or USB as of Version 2.0.
- No virus scanner for video files.
- No backup tool (or synchronization tool) in use during the measurement.
- No additional programs running on the computer. Also services such as hard drive defragmentation or file indexing may not be running during measurement.

[1] A RAID system consists of multiple hard drives connected together in Stripe-Mode (RAID 0). This increases the capacity as well as the data throughput. It is also possible to connect more than two hard drives, but eventually the hard drive controller imposes limits on the data throughput.

Crucial parameters for optimum frame rate performance:

- The maximum frame rate is determined by the camera’s properties
- The performance capacity of the interface to the camera, e.g. 400 MBit/s for Firewire A
- The hard drive controller's and its PC interface and driver
- Processor and mainboard chip set
- The hard drive's maximum writing speed
- Compression

17.14.2 Supported Cameras

Supported and tested cameras:

- Connector type: 1394-Fire Wire A

Camera		Driver		Operating system	Limitations / Remarks
Manufac.	Model	Manufac.	Version		
Imaging Source	DFK 21BF04	Imaging Source	4.1.1.1	Win XP/ Win Vista/ Win 7	
	DFK 31AF03-Z2			Win 7	
Allied Vision	Marlin F-033C	Allied Vision		Win XP/ Win Vista/ Win 7	Few camera settings available

- Connector type: 1394-Fire Wire B (A compatible)

Camera		Driver		Operating system	Limitations / Remarks
Manufac.	Model	Manufac.	Version		
Basler	Scout scA640-120fc	Basler		Win XP Win Vista Win 7	No recognition of plugging/unplugging In order to achieve the full frame rate with the camera, it is necessary to increase the camera's packet size to 8192. This is done under the heading Transport-Layer in the camera Properties, by means of the manufacturer-specific application (tool). Please contact your camera manufacturer for detailed information on the camera settings and other settings.

- Connector type: Gigabit-Ethernet

Camera		Driver		Operating system	Limitations / Remarks
Manufac.	Model	Manufac.	Version		
Imaging Source	DFK 21BG04.H	Imaging Source	1.0.0.513	Win XP/ Win Vista/ Win 7/	No recognition of plugging/unplugging Exposure automatically regulated in response to frame rate
Basler	Scout scA640-120gc	Basler		Win XP/ Win 7	No hot-plug recognition
	Ace acA640-90g				Sensor: 1/3" Sony ICX424 PoE (Power over Ethernet) No hot-plug recognition
	Ace acA640-100g				Sensor: 1/4" Sony ICX618 PoE (Power over Ethernet) No hot-plug recognition
	Ace acA645-100g			Win 7	Sensor: 1/2" Sony ICX414 PoE (Power over Ethernet) No hot-plug recognition
Flir	A35 f = 9mm with SC	Flir	1.9.1.0	Win 7	Infrared camera To operate the camera, the eBus SDK (v4.0.6) is required. You can obtain it from the imc Hotline. The color map's colors (heatmaps) are not calibrated to the temperatures. This means that it is not possible to deduce the exact temperature from the color.

- Connector type: WLAN

Camera		Driver		Operating system	Limitations / Remarks
Manufac.	Model	Manufac.	Version		
GoPro	Hero 4 Black	-	-	Win 7	Only one GoPro camera at a time can be used in an experiment. In order to operate the camera, a special driver package is required. This driver package can be obtained from the imc Hotline. The video recordings are not synchronized to the measurement readings. However, Videodata can be retroactively synchronized with the help of imc FAMOS,for example. No pretriggering is possible for video recording with the GoPro. Recordings are not automatically saved to the imc STUDIO database. The preview of GoPro is delayed by how much depends on the network. At some resolutions, no preview is possible (see GoPro user's manual).

- Connector type: USB 2.0

Camera		Driver		Operating system	Limitations / Remarks
Manufac.	Model	Manufac.	Version		
Microsoft	LifeCam Cinema	Microsoft	3.20.240.0 (XP)	Win XP/ Win Vista/ Win 7/	M-JPEG compression
	LifeCam HD-6000 for Business		6.1.7600. 16543	Win 7	
	LifeCam Studio		6.1.7600. 16543		
Logitech	HD Webcam C615	Microsoft [3]	6.1.7601. 17514	Win 7	M-JPEG compression
	QuickCam Pro 9000 Webcam		5.1.2600. 2180 (XP)	Win XP/ Win Vista/ Win 7	M-JPEG compression Tilt / pan listed but no functionality
	C600 Webcam		6.1.7600. 16385 (W7)	Win 7	M-JPEG compression

[3] Please use the supported and tested Microsoft driver, do not use the Logitech driver

17.14.3 Frame Grabber and Compression

Frame grabber-Driver table

Frame grabber		Connector type	Driver		Operating system	Limitations / Remarks
Manufac.	Type		Manufac.	Version		
Imaging Source[4]	DFG/ USB2-it	USB 2.0/ PCI/ PCIe	Imaging Source	1.1.0.3	Win 7	Video format with color min. 2 Byte per Pixel (compression recommended)
Enciris Technologies	LT-102-PCIE	PCIe	Enciris	1.40	Win 7	Hardware Video compression WV1 (H264 similar)

[4] product type is not available any more

Compression table

Manufac.	Type	Version	Operating system	Limitations / Remarks
Pegasus Imaging	PICVideo (Motion-JPEG)	3	Win 7	Note: external license necessary (demo version is not enough) as default a compression of ca. 1:12 will be set

17.15 imc STUDIO Monitor

The imc STUDIO Monitor plug-in makes it possible to connect with one or more measurement devices during a running measurement, particularly for the purpose of observing the current measurement data. Data can be viewed and edited live on multiple workstations.

Measurement device requirements
The same hardware prerequisite apply as for imc STUDIO Setup.
For the purpose of connecting with imc STUDIO Monitor, the devices must additionally have at least 32 MB of internal device memory available.

imc STUDIO Monitor is an independent installation variety of imc STUDIO. This means it is installed and started as a separate application. For Monitor, separate licensing is required accordingly (an edition of imc STUDIO + imc STUDIO Monitor).

imc STUDIO Monitor has a different scope of functions than imc STUDIO. The following functions/components are categorically not available (no matter which edition):

Available plug-ins/components	Description
Setup	Device configuration of imc devices
	Device configuration of 3rd-party (non-imc) devices (e.g. Video or 3rd-party devices, provided appropriate license available)
Project Management	Display of saved measurement data in the Data Browser
	Projects and Experiment templates
Automation (1) 167	Design of real-time test station automation
Third Party Device Interface (2) 167	Integration of devices from manufacturers (Third Party Devices) into imc STUDIO
Video	Synchronized video acquisition
imc SENSORS	Sensor database integration
Sensors	Expansion package for Setup for setting up imc STUDIO sensors and administering the imc SENSORS sensors.

Association of imc STUDIO Monitor functions with the imc STUDIO editions

Functionality	Description	imc STUDIO Edition			
	●: included ○: optional ⦿: subset of full feature set	RUN	STD	PRO	DEV
Monitor	Multi-client monitoring and visualization		●	●	●
Runtime functionality	Execute but not modify	●	●	●	●

Functions

Basic functions	Description
Accessing of device	At most, • three imc STUDIO Monitor PCs can be directly connected with one device (SN 12xxx), or • four imc STUDIO Monitor PCs can be directly connected with one device (as of SN 13xxx).
Channel settings	Settings per channel/variable • Activate visibility (monitors) • circular buffer memory for the visualization

Storage settings	Settings for all visualized channels
	• Activate data saving
	• Folder designation
	  • Date and time
	  • Consecutive numbering
	• Interval saving (optional with a selected maximum count)

Writing rights	Writing to device variables optionally possible

17.16 Additional imc Software Products (optional)

Product	Description
imc FAMOS	Some imc STUDIO plug-ins (Automation 1102, Sequencer 1189, Panel 813) are able to integrate imc FAMOS for data analysis purposes (executing sequences). imc FAMOS is a imc software for complex analysis, display and presentation of signals on the PC ("offline"): www.imcfamos.com imc FAMOS must be separately purchased and licensed for the PC. Details on imc FAMOS are available in the software documentation. To execute ready to run sequences for analysis in imc STUDIO, an installation of imc FAMOS version of 6.1 or higher, as a Runtime, Professional or Enterprise Edition is required.
imc Online FAMOS Professional	Some imc STUDIO plug-ins require devices having imc Online FAMOS Professional. imc Online FAMOS 739, or its Professional version, is the software which processes data within the measurement device ("online"). imc Online FAMOS Professional must be purchased and licensed along with the device.
imc SENSORS	imc STUDIO Setup can use the imc SENSORS database. imc SENSORS is a ready-to-go, universal database application for administering and editing sensor information. In particular, the entries in the sensor's technical data sheet as well as its calibration values are processed and managed. Along with these values for smart sensors (TEDS) defined in IEEE 1451.4, selections of additional sensor properties can be entered. The prerequisite is installation of an imc SENSORS version of 1.3R3 or higher.

17.17 imc Online FAMOS

imc Online FAMOS is an expansion for imc measurement devices, offering a large number of real-time pre-processing and signal analysis functions. The mathematical analysis functions are performed by a dedicated signal processor (DSP) embedded in the measurement device.

imc Online FAMOS consists of:

- an editor in which the user enters the algorithm in the PC
- a compiler, that is initiated in the background, to generate the executable program code for the DSP
- the DSP-based runtime system on the measurement device

imc Online FAMOS processes the input channels of the system and makes available the results of computation in the form of virtual channels. imc Online FAMOS can not only work with channels, but is also able to read the digital inputs of the system, digital and analog outputs can be set, the virtual bits and display variables can be edited. Thus not only tasks of evaluation, but also control problems can be realized.

Especially when using appropriate pre-processing (such as averaging), imc Online FAMOS is able to effectively reduce the data rates and thus the transfer bandwidth (between device and PC), as well as memory requirements.

Data rates achieved in the process are not subjected to the limitation of up to just two different sampling rates within the system.

imc Online FAMOS is device-based and can thus also be deployed in stand-alone applications without any PC.

By means of assignments of complex calculations applied to "virtual bits", which can also be used in the trigger machine, it is possible to achieve complex trigger conditions, for example on the basis of FFT-spectra.

imc Online FAMOS processes continuous data streams from measurements currently being conducted. This means that the analysis results can provide valuable feedback while a test is actively in progress, as a way of controlling and possibly influencing the test procedure. In consequence, the scope of applications far surpasses that of classical "Post-Processing".

Special advantages and applications:

- Hugh inventory of integrated functions from many areas of measurement engineering analysis
- simple, intuitive syntax
- Formula Assistant for aid in entering and parameterizing functions
- Color-coded syntax and integrated formula-writing help
- Hierarchical Favorites-list
- Assignment of channels to Display variable for output on imc Display
- Interaction with other system components such as digital input/output, LEDs, analog outputs etc.
- The analysis can optionally be performed in an event-oriented manner and thus with real-time capability. In the "Professional" version, command sequences can be composed to include "control commands", the execution of which can be interrupt-driven and true-to-cycle. Conditional command sequences are also supported.
- Virtual channels are parameterized by means of the imc STUDIO GUI. They are handled and administered in the same way as any other, "conventional" measurement channels.

System requirements and authorization:

- imc Online FAMOS is a device option which must be authorized for each device separately.
- Authorization is accomplished by entering an activation code which can also be entered later by the user.
- imc Online FAMOS is supported by all imc measurement devices.

Basic functions
Extensive mathematics and calculation methods
Basic calculation functions
Basic mathematical functions (trigonometry, logarithms etc.)
Logic and comparison functions
Digital filters: ready-to-go low-pass, high-pass and band-pass filters, freely defined digital filters (FIR, IIR), smoothing filters, hysteresis filters, median filters, ABC-Assessment
Vibration-weighting conform ISO 2631-1, DIN 45671-1, ISO7505
Acoustics: Sound Pressure Level, LEQ
Transitional Recording (data reduction for analog data)
Numerical differentiation and integration
Statistics functions (mean value, sum, standard deviation etc.)
Resampling
FFT: complex spectrum, invers FFT
Measurement engineering functions such as characteristic line correction
Rosette-calculation
Input/output
Reading of input channels
Creating virtual channels
Import/export of virtual bits, e.g. for use with the trigger machine
Import/export of Display variables, e.g. for display on the device display
Import of digital inputs
Export of digital and analog outputs
Switching of the LEDs and of the buzzer
Process vector variables of the GPS interface can be processed directly
Status of system variables, e.g. synchronization, remaining memory on device drive
Paging through device display in multi-page Display configurations
imc Online FAMOS with control commands
Loops, conditions with interlacing
Timer
Controlled interval storage
Current time and date as function
Control functions

Languages

- German, English, Japanese completely supported; French, Chinese, Korean partially

Documentation

- Component of imc STUDIO documentation and functions reference in PDF format
- Integrated online help in German, English and Japanese

imc Online FAMOS Enhancement Kits

- Online Class-counting
- Online Order-tracking analysis

17.18 imc Online FAMOS Professional

imc Online FAMOS Professional is a package for implementing control loops in test rigs. It enables monitoring, open- and closed-loop control processes. Additionally, imc Online FAMOS Professional provides a substantial performance boost in virtual channel computation for data logging applications without control.

With imc Online FAMOS Professional you have:

- Performance enhancement
- Synchronous Tasks
- Direct access to the Process Vector

Performance enhancement:

- In Assembler optimized set of functions
- Use of very efficient On Chip RAM

Synchronous tasks:

- Interrupt-driven real-time processing: accurate to a single cycle
- Max. cycle rate up to 10 kHz, with guaranteed latency and reaction time
- Up to 5 synchronous tasks in different sampling rates
- Controller (PID, two-point)

Direct access to the Process Vector:

- Direct access to the signal conditioners with 10 kHz and minimum latency: "mirrored" instantaneous values of the analog channels as so-called process vectors.
  - for closed-loop control
  - for display with imc STUDIO Widgets
- Independent of the trigger machine, i.e. values from triggered channels can be processed already after preparation before release of the trigger.
- Additional user-specific variables can be created
- Status of system variables such as synchronization, current supply, hard drive

System requirements and authorization:

- imc Online FAMOS Professional is a device option which must be authorized for each device separately.
- Authorization is accomplished by entering an activation code which can also be entered later by the user.

17.19 imc Online FAMOS Class-Counting

imc Online FAMOS Class-Counting is the package for statical evaluation of signal plots in accordance with DIN 45 667. This entails analysis of the measured data with regard to instantaneous values, extreme values, and cyclical oscillations by tallying their respective frequencies of occurrence. The advantage of all class-counting procedures is that the data storage space requirements for the statistics result is defined at the outset. The disadvantage is loss of information due to reduction of the high-resolution measurement range into a specified number of classes. Furthermore, the tallied events' reference to time is lost.

Class-counting is primarily used to measure stress and fatigue states and to perform the analysis and prognostication of associated endurance strength, which are based on these measurements. In this field, the Rainflow method has become established, which counts both small and large oscillations with their amplitudes and mean values. An interpolation which approximates the true extreme values is available both as the Rainflow function (ClRainFlowTM) and as an individual function (ClTrueMax).

Single-channel functions:

- Rainflow with and without residue, with and without approximation of extreme values. The results are represented as a 3D matrix with amplitude and mean value, or start and target classes.
- Range-pair procedure: Classifies the amplitudes of the oscillations as a histogram.
- Time-at-Level procedure: The time period during which the measured data remain within a class, as a histogram.
- Extreme value approximation: Better approximation of the extreme values by means of interpolation.

Two-channel functions:

- Revolutions histogram: tallies the revolutions in the classes of the class-counting channels as a histogram.
- Two-channel revolutions histogram: Both the class-counting and the RPM-channel are classified as a 3D matrix.
- Two-channel histogram: Two channels are classified in a 3D matrix.

System prerequisites and activation:

- imc Online FAMOS Class-counting is a device option and is activated separately for each device.
- Activation is accomplished by entering an activation code by the user after purchase.
- As an supplemental package, it requires the possession of a valid device license for imc Online FAMOS.

17.20 imc Online FAMOS Order-Tracking Analysis

imc Online FAMOS Order -Tracking Analysis is the package for analyzing rotating machinery in real time.

Order-tracking analysis is a method of analyzing noise or vibration coming from rotating machinery. In contrast to frequency analysis, it plots the energy content of the noise not over the frequency but rather over the order, i.e. the noise is "normalized" to the current RPM value.

Functions:

- Order spectrum from the time plots of the oscillation and either RPM- or pulse signal (e.g. from rotary encoders) from which the RPMs are derived.
- Spectrum
  - Spectrum of the oscillation for a specified RPM-range.
  - Spectrum plot over RPMs
- RPM-based signal representation: The time histories of a vibration signal and of the RPM are evaluated to determine the vibration behaviour in reference to the RPMs. The desired RPM-range is subdivided into classes of equal width.
- Order lines: Order lines as a function of the RPMs (RMS value, or complex).
- Angle sampling: Sampling of an vibration signal is referenced to the rotation angle, for a given RPM-value or pulse signal
- Determination of frequency lines: The magnitude and phase of a periodic signal is determined, optionally with approximation.

Preparatory functions:

- Smoothing: Low-pass filtering is applied to an oscillation signal, where the cutoff frequency is RPM-dependent.
- Pulse duration measurement
  - The time between two pulses is measured.
  - The course of the RPM signal is derived from the pulse signal.
- Addition of an angle to an angle signal.

System prerequisites and activation:

- imc Online FAMOS Order-Tracking Analysis is a device option and is activated separately for each device.
- Activation is accomplished by entering an activation code by the user after purchase.
- As an supplemental package, it requires the possession of a valid device license for imc Online FAMOS.

18 Last Changes

18.1 in Doc. Rev. 3.7

Panel

Chapter	Amendments
Properties - Widget	Many of multiple Widgets properties used have been described.
Properties - Pages	The properties of the Panel pages have been described.

Scripting

Chapter	Amendments
Device functions	Code for list of selected device refactored.
Logbook	Getting entries with specific sender via linq-Methods

Shared Components

Chapter	Bug-fix
Datapool Remoting	Documentation has been deleted since Datapool Remoting is not present in imc STUDIO.

imc Online/Inline FAMOS Function Reference

Chapter	Bug-fix
GetHistoValue While	Wrong function name in the example corrected: "Histogramm" --> "ClHistogram".

Command Reference

Chapter	Bug-fix
Execute menu action	A complete new description of the command has been prepared; the old description did not correspond to the present behavior.
Run script	A complete new description of the command has been prepared; the old description did not correspond to the present behavior.

18.2 in Doc. Rev. 3.6

Setting Up

Chapter	Amendments
Product Configuration / Licensing	• Brief description of changes to the edition and the components • Information on the distinction between "Editor" and "Engine".

imc STUDIO - General

Chapter	Amendments
What is saved where?	Every setting, option and parameter is saved. In this chapter is a list of the important areas and where they are saved.

Setup - Advanced Device Functions

Fieldbusses

Chapter	Amendments
CAN - Extras - menu 425	The option: "Treatment of duplicate channel names" now offers the ability to not adopt these channels: "Don't accept in CAN-Assistant". • The channel with the lowest ID, in other words, with the highest priority, is adopted. The option: "Channels with invalid properties" now offers the ability to not adopt these channels: "Don't accept in CAN-Assistant".

Storage Options and Directory Structure

Chapter	Amendments
Designation of the Measurement Data Folders	Notes on various approaches in numbering the measurement folders have been supplemented (storage on the PC and storage on device).
Data File Format	Compressed format for many Field-bus variables

Synchronization

Chapter	Amendments
Externe Zeitgeber: PTP	PTP parameter table adapted: Up to the present time, the parameter "User description" has been pre-filled automatically with the device designator. Now the parameter after the device selection is empty and can be modified manually accordingly.

imc Online FAMOS

Chapter	Amendments
LED-flashing during measurement	New: During a running measurement, LED 6 flashes.
Types of Variables	New: Here is a list of the usual variable types which you can define in imc Online FAMOS. Descriptions of the individual variable types, which previously appeared in various locations, have been collected together in this chapter.

Automation

Chapter	Amendments
Tutorial	Tutorials have been updated.

Scripting

Chapter	Amendments
Device functions	Manufacturer parameter added.
Panel and Widget functions	Table functions added.
Datapool functions	Hint for calling Datapool in other script types.
Logbook	Hint for calling Logbook in other script types.

Chapter	Bug-fix
Panel and Widget functions	Missing "using" added.
RS232-Interface	Argument in ReadLine method removed.

Third Party Device Interface

Chapter	Amendments
General	Chapter Agilent Scope added.
Add a new script	Hint for other templates added.

Chapter	Bug-fix
Device control	Argument in ReadLine method removed.

Shared Components

imc Online/Inline FAMOS Function Reference

Chapter	Amendments
VectorStatic	New: Access to vectors via variables
For-Loop	New: Modifiable end of a For-loop

Command Reference

Chapter	Amendments
imc FAMOS Transfer Table 1132	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.
Import parameters	Import of the measurement range Rounding imprecision

18.3 in Doc. Rev. 3.5

Setup - Advanced Device Functions

Synchronization

Chapter	Amendments
External Clocks: PTP Master/Slave setup Separate synchronization	New description: PTP-parameter and configuration.
Internal Clocks - Device Clock	Revised content: SyncRTC.

Scripting

Chapter	Amendments
Inline FAMOS functions	Hint about return value *Result* added.

Chapter	Bug-fix
Inline FAMOS functions	Set source code with *.ifa.xml path corrected.

18.4 in Doc. Rev. 3.3

Setting Up

Chapter	Amendments
Device Connection / Network / Firewall	The description has been amended: Which programs may not be blocked by the firewall for correct operation.
Installation: User-defined	The "imc FAMOS COM interface" has been removed from the dialog for configuring imc Shared Components. This component is no longer optionally available. This will not affect the running of imc FAMOS sequences under imc STUDIO.

Setup - Device Configuration

Chapter	Amendments
Timestart 154	Detailed description of the prerequisites for starting all devices simultaneously.
Default Values 107	New
Enumeration Classes	New
Sensors	Completely revised
Tutorial	Examples of data storage
Diskstart 99	New: Autorepeat mode
Additional Parameters 176	Bridge supply (On/Off) [eBridgeSupplyOnOffAction]

Setup - Connect the device

Chapter	Amendments
Device connection via LAN 307	New: LAN-Configuration with Device Interfaces

Setup - Advanced Device Functions

Synchronization

Chapter	Amendments
External Clocks: PTP	New description: Synchronization via PTP.

Chapter	Bug-fix
Several	Revised content and bug fixing. Particularly of the chapters: Definition of terms, the NTP timer, Phase offset correction, Master/Slave structure without external timer and "What happens when?": Measurement with one device and Measurement with multiple devices.
Measurement controlled by PC	Variable names were spelled incorrectly in the description text: IsSynch_... -> IsSync_...

Panel

Chapter	Amendments
Fullscreen	New: The capabilities of the fullscreen, as well as of the new title bar, have been described.

Automation

Chapter	Amendments
User-defined variables	Description of the default and the persistent variables has been amended.
Source Code - Find and Replace	New

Video

Chapter	Amendments
Camera: GoPro Hero 4	New: Installation steps and notes for the camera GoPro Hero 4.

Scripting

Chapter	Amendments
General	• Format and designs • Index extended
Script Properties	• Activity Scope added. • Description of Separate proxy class library amended.
Regenerate proxies	Additional information added.
Device functions	• Remove data from the storage device / removable disk.

Chapter	Amendments
	• Hint, due to ExecuteAction added • Trigger in individual folders • Measurement state added.
Channel functions	• Hint, due to ExecuteAction added • Balance at device start added
Datapool functions	• Access to channels from the last measurement • Example for user-defined channels added. • DataTable added.
Curve window	• Curve window actions and scroll mode added • Curve window kit: add channel • Curve window kit: y-Axis settings added
Inline FAMOS functions	• New functions for private tasks added. • Example added
Commands	Adding methods to the execution of menu actions added
Panel Script	• Explanation about bool parameter removed. • Headlines for Widget access.
Context Script	Adding/Removing actions and events added.
Event Script	Table for parameter description added.
Device Parameters	eDiskStartOption added.

Chapter	Bug-fix
General	Translation
Device functions	IntervalCount was mentioned twice and therefore replaced by IntervalTime

Third Party Device Interface

Chapter	Amendments
General	• Overview of tutorials added • Images edited (frame) • New chapter: eRange Parameter
AudioDevice	Notes to certain devices is now in one box.
FunctionSimulator	Note about trigger and time zone added.

Chapter	Bug-fix
General	• Image size corrected • Missing translations
AudioDevice	• Remarks to certain devices were German • Image of channels was showing device page

Shared Components

imc Online/Inline FAMOS Function Reference

Chapter	Amendments
Several	• New functions for imc Inline FAMOS have been described: ReplaceFirstValues0 ReplaceFirstValuesN SkipFirstValues • With the functions ABCRating, ExpoRMS and SoundPressureLevel, the parameter "Time Constant" has been expanded (only for imc Inline FAMOS) • New functions for imc Online FAMOS have been described: GetLastError
FFT-functions	Extended window width for imc Inline FAMOS has been described.
ReSample	The description of the Funktion: "ReSample" has been updated.

Chapter	Bug-fix
General information about the Accu* functions	The Accu*-functions were indicated as "only available in imc Online FAMOS". This has been corrected to "imc Inline FAMOS".

18.5 in Doc. Rev. 3.2

Panel

Chapter	Amendments
Zooming Panel pages	New function: The zoom can be applied to all loaded pages simultaneously.
Adapt/automatically adapting the size of pages and Widgets	New function: The functions can be applied to all loaded pages simultaneously.

Scripting

Chapter	Amendments
General	• Formats • Overview added to the tutorials • Texts adapted to changes in the software.
Panel and Widget functions	• Delete Panel page added. • Rename Panel page added.
Datapool functions	• Groups and categories are better described. • Create XY-signal added. • Query for empty channel when accessing channel data added
Logbook	• Hint added to the example • Example "Last entry" removed.
Script options	• Chapter renamed • Texts were changed and descriptions added
Context Script	Example due to automatic saving of all active channels added.

Third Party Device Interface

Chapter	Amendments
General	• "Assistant" was renamed to "Management".

Shared Components

Command Reference

Chapter	Amendments
Miscellaneous	Adapt user interface texts
Show message box	Description completely revised

18.6 in Doc. Rev. 3.1

Setting Up

Chapter	Amendments
Update and Compatibility	Notes on the SQL-placeholder have been updated. Instead of square brackets, apostrophes are now used
System requirements	Win 10 added.

Setup - Device Configuration

Chapter	Amendments
Download and Reconfigure 93	FAQ supplemented

Chapter	Amendments
Parameter set and Mapping	New
Metadata and Additional Columns	New

Scripting

Chapter	Amendments
General	Chapter Restore script changes added.
Menu bar and context menu	• Context menu "Add" described. • Regenerate proxies dialog was changed.
Export and Import	Hint due to DLL-Export and Resources added.
Channel functions	• Structure changed • Get balance action state
Logbook	Clear logbook entries.
imc FAMOS functions	Dialog License violation added.
Script options	• Options for the Script storage were extended and described. • New option for code completion added.

Chapter	Bug-fix
General	• Icons in the tables of context corrected. • Format • Term Panel was replaced at some places by Panel page.
Curve window	Toggle was not working.

Third Party Device Interface

Chapter	Amendments
General	• Chapter structure changed • AudioDevice and ChannelLoader do not need any 3PDI license.
AudioDevice	• Note about disconnecting audio devices added. • Device specific notes added.

Shared Components

imc Online/Inline FAMOS Function Reference

Chapter	Bug-fix
Rosette 1	Correction if the calculation: vorher: $\sigma_v{}^2 = \sigma_1{}^2 + \sigma_2{}^2 + \sigma_1\sigma_2$ nachher: $\sigma_v{}^2 = \sigma_1{}^2 + \sigma_2{}^2 - \sigma_1\sigma_2$

Command Reference

Chapter	Amendments
Export parameters	• Description completely revised • Function descriptions newly added: Overwrite existing file Device Configuration Expert mode • Additional notes
Import parameters	• Description completely revised • Descriptions newly added: Important note to ensure that all parameters can be imported. Notification that it is not possible to import to channels belonging to the current measurement. Notification that for import of variables, the command: "Import Variable" is recommended.

Placeholders

Chapter	Amendments
MEASUREMENT	• New
VAR/VARS	• Descriptiof PROPS added • Description of VAR added
VALUE	• Formatting with comma instead of decimal point described

18.7 in Doc. Rev. 3.0

Setting Up

Chapter	Amendments
Installation - Preparation	Supplemented information on the update/a parallel installation.
Recommended Virus Scanner Settings	Previously: "Jiggling Display and Heavy Burdening of the Processor". Renamed and paths modified to reflect the current version.
Update and Compatibility	Notes regarding the update and compatibility!

Chapter	Bug-fix
Installation Step by Step > User-defined	The installation path for the installation of imc FAMOS can not be set separately, as previously described.

imc STUDIO - General

Chapter	Amendments
Project Menu	Revised: • Experiment and project management • Importing/exporting
Views	Functioning and content of the views have been explained in greater detail. The description of importing/exporting views has been amended.
Options	The description of the option: "Virtual device clock" has been updated. The storage location of the options has been described.
User Administration	New: Import/export of the User Administration
Setting Up 20	The description of the commissioning procedure has been moved to its own separate section

Chapter	Bug-fix
Logbook 69 > Filtering the Logbook Options > Logbook	The filter is called "Information" and not "Messages", as previously stated.

Setup - Device Configuration

Chapter	Amendments
Module properties 117	Working with CRFX modules - Deleting and exchanging modules.
imc CRONOSflex Modules 161	Information on and characteristics of modular devices (e.g. CRFX)
Incremental Encoders 229	Detailed description of the measurement modes (excerpted from the device manuals)
Transferring an experiment to other devices	Describes how experiments can be started by other devices.
Additional Columns	Column type: Parameter Set Import column described Parameters and settings of all supplemental files described New examples created/existing examples revised
Sampling & Preprocessing 172	File type (24-bits) added
Supplemental Files 105	Management of supplemental files (characteristic curves, Messaging-files)
Storage 150	Note on the saving interval count: Measurements are not deleted while they are loaded in the program.
Creating a Diskstart configuration 101	It is not necessary to rename the ume-file to a ums-file. When a Diskstart configuration with Autostart is created, a ums-file is generated automatically.
Channel definition 163	A note on the maximum count of characters in names has been added.
Sensor Database - imc SENSORS 128	New
TEDS - Sensors 201	Updated and amplified.
Additional Pages	The description of how to insert existing Setup pages has been updated.
Tutorial	Updated and amplified.

Chapter	Amendments
Firmware 142	A description of the compatibility between different firmware versions and on the parallel use of multiple firmware versions has been added.
Balance - Tare and Bridge 178	Notes revised to indicate that taring/bridge balancing is now also possible during a running measurement.
Other parameters (Configuring Channels and Variables) 176	Additional parameters described: • Connector • Enumerated channel number • Current value
Configuring triggers 197	Note added on automatic deletion of the pretrigger setting, when a trigger assignment is deleted from a channel.
Download and Reconfigure 93	Note supplemented: What happens with multiple "writers" wish to write their respective values to the variables during initialization (Download).
Import of an imc DEVICES Experiment	New

Chapter	Bug-fix
Creating a Diskstart configuration 101	The description stating that in a Diskstart experiment, any PC can connect with the running measurement even if the experiment is not known, has been deleted.

Setup - Connect the device

Chapter	Amendments
Complex Networks	The description of the monitoring of devices has been updated to the current software version.
Connection via LAN	Hardware descriptions updated to the current version.
Connecting via LAN in four steps 296	The description of how to configure the LAN-settings has been updated for the new integrated user interface: Device interface configuration.

Setup - Advanced Device Functions

Fieldbusses

Chapter	Amendments
Several	Structural revision and adaptation to the menu guidance for imc STUDIO 5.
imc HiL 500	The documentation has been relocated to an external PDF file.
imc Application Module 499	The documentation has been relocated to an external PDF file.
SPI-Bus Interface 574	New

Storage Options and Directory Structure

Chapter	Amendments
Several	Structural revision and adaptation to the menu guidance for imc STUDIO 5.
Save Trigger Events in individual Files	Supplemental notes on saving channels with the trigger combination: 1-Trigger.

Chapter	Amendments
Designation of the Measurement Data Folders	Supplemented notes on matching the naming of folders both in the device and on the PC.

Chapter	Bug-fix
Data Storage and Directory Structure	When the maximum file size of 2 GB is reached, then instead of creating a new file with a suffix in the same folder, as previously described, the system creates a new measurement folder with a suffix. The new file, without any suffix, is saved to the new folder having the same name.
Saving Interval on the Internal Device Storage Medium	The chapter was missing.

Device Hard Disk, removable drive

Chapter	Amendments
Several	Structural revision. Some descriptions have been partially revised and updated.
Partition and File System	Notes on the limitations/restrictions applicable to the respective file system. Description of the use of a data storage medium in various devices has been revised.

imc Messaging

Chapter	Amendments
Several	Structural revision and adaptation to the menu guidance for imc STUDIO 5.

imc Online FAMOS

Chapter	Amendments
Several	• Structural revision • Headings and index entries have been added/revised • The term "Operations box" replaced with "Editor" • Syntax-highlighting has been revised

Chapter	Bug-fix
Brief Introduction Setting System Outputs	A scaling for the DAC channels is taken into account.

Miscellaneous

Chapter	Amendments
Several	Structural revision and adaptation to the menu guidance for imc STUDIO 5.
Miscellaneous > Multiple triggering files in imc FAMOS	The chapter was removed since the processing of event-based data sets is already described in the imc FAMOS user’s manual.
Tuning, Tips and Tricks > Jiggling display and heavy burdening of the processor	Filepaths have been changed to reflect the conditions in the current version.
FAQ > Device Hard Drive	Amended answers for the issue: "The system won't recognize the data storage drive"

Chapter	Amendments
	• Information added on which file systems are supported. • Additional error source: Multiple data storage units plugged in simultaneously.

Panel

Chapter	Amendments
Widgets - Operation and Properties	Tool Window: Widgets separated from the Widget-condition.
Curve window	Curve window documentation incorporated here.
Data Browser 818	Description completely revised.
Grouping Widgets	New
Properties - Widget	Many properties described.
Properties (general)	Window structure described.
Aligning to grid	New
Zooming Panel pages	New
Repository	New

Automation

Chapter	Amendments
imc FAMOS automation (data cutting)	imc FAMOS Dialog - New functions described: • Debug mode • Variable transfer when the imc FAMOS Editor is open. Transfer Table - Revised functions: • Measurement selection • saving of the results

Video

Chapter	Amendments
Supported Cameras Frame Grabber and Compression	The technical spec sheet has been included in the software documentation. For this reason, this chapter has been omitted since its content is also presented in the Technical Data.

Scripting

Chapter	Amendments
General	• The chapter *Script* was renamed to *Script (general)*. • General setting of parameters was described. • Table design was remodeled. • The chapter Curve window was added. • The chapter *Experiment Parameters* was added. • The chapter Inline FAMOS functions was added • The structure was revised. • The chapter Script options was added.
Menu bar and context menu	• A note when regenerating proxies due to the imc FAMOS Math Provider was added. • The dialog for changing the script scope / script storage was changed. • The script options, the script types and the script storage are now described in a table. • The script options are now mentioned when creating a script. • Add links to graphics. • Proxy regenerator added • Hint for saving removed. • The properties for Panel scripts were changed. • Save with Sequencer was deleted. It is not possibly anymore. • Information added for changing script scope: Projects and experiment have to be saved. • The chapter <%TARGETTITLE%> was sectioned. • The chapter *Default dialog response* added.
Import and Export	• The Script-Import-Dialog has changed and therefore the image was updated. • It is possible to export more than one script at once. • Hint due to Experiment > Save as and script editor added. • Import in the script added.
Debugging	• Images were added and edited. • table of key short-cuts added
Important Information	• Important Information due to *hooking* and *unhooking* events were added.
Device functions	• Query the selection state of a device was added. • Diskstart: Setting of default dialog response in the options and via Scripting added. • Assignment of device object after selection. • Section *Storage* was added. • List of selected devices (serial number) added.

Chapter	Amendments
Channel functions	• The description of getting values for the parameter *eUserUnit* via *getValue* and *CValueArray* was added. • Note due to PC saving was removed. • Example for saving active analog channels added. • Get display value of input range added. • The section about Saving data / Data transfer was edited and amended. • Strain gauge paragraph added. • Tables for sample time and range for UNI2-8 are now in a popup-window. • Writing sensor information (TEDs) was added.
Logbook	• Example of listing entries by sender added.
imc FAMOS functions	• Explicitly mentioned, that a imc FAMOS Runtime license must be activated.
Commands	• The menu action *Reconfigurate* was added.
Panel Script	• The properties (**`Run()`** and **`Stop()`** method) for Panel scripts were changed. • Important Information due to *hooking* and *unhooking* events were added. • The Panel page properties due to the Script name were added. • Information, when a Panel page is already linked to a Panel Script, was added. • A bool variable was added to the example to hook and unhook the event. • Setting text for standard widgets added. • New methods were added.
Device and channel settings	• The list of parameters and values are now shown in a table.
Channel calculation	• The list of parameters and values are now shown in a table.
Curve Window and channel selection	• The tutorial can now be used without imc FAMOS commands. • The tutorial uses the new functions Initializes and Dipose of the Panel script.
Datapool functions	• Changing the category added. • Access to complex variables added.
RS232-Interface	• This tutorial was added.
Using the script clipboard	• This tutorial was added.

Chapter	Bug-fix
General	• Not working toggle were fixed. • Formatting and spelling mistakes were corrected. • The term Script scope was changed by the term Storage Scope. • Frame of C#-code tables remodeled.
Debugging	• The toggle button was shown in the PDF version for step 2.
Device functions	• DevSetup.BeginTransaction() was mentioned twice, when summarizing device actions with DevSetup.BeginTransaction() and Dev.EndTransaction(). • The plural "s" was missing at Parameters in some lines of the C#-Code.
Channel functions	• The plural "s" was missing at Parameters in some lines of the C#-Code. • There were problems when getting the channel name via connector. Therefore the code was changed and advice to use *foreach* instead was added.
Datapool functions	• Dispose() of DmChannel was missing.
imc FAMOS functions	• Missling line (cmp_channel) was added.

Chapter	Bug-fix
Device Parameters	• Not implemented values of parameter eDiskStartAction were grayed out.
Script	• Missing image was added.
Device and channel settings	• The parameter Sample time was changed to Sample rate. • A placeholder was not resolved.
Channel calculation	• The parameter Sample time was changed to Sample rate.
Curve Window and channel selection	• The items of the ComboBox with analog channels was missing.

Third Party Device Interface

Chapter	Amendments
All	New

Shared Components

Chapter	Amendments
Curve Window	Documentation on the curve window has been moved to the Panel-documentation.
imc Online/Inline FAMOS Function Reference 1440	Documentation incorporated here.

imc Online/Inline FAMOS Function Reference

Chapter	Amendments
Several	• Structural revision. Some descriptions have been partially revised and updated. • Identifier for functions which can only be used in imc Online FAMOS or imc Inline FAMOS. • New functions for imc Inline FAMOS have been described: AccuLength AccuMax AccuMean AccuMin AccuRMS AccuStDev SpecThirds

Chapter	Bug-fix
Several	Corrections have been added.

Command Reference

Chapter	Amendments
Data Saving Assistant	• The option: "Keep original files" has been *inverted*. Now, activating the option explicitly deletes the files instead of retaining them. New text: "Delete original files" • The checkbox "Execute without acknowledge" has been described.
Set variables	• Behavior when using quotation marks described in more detail. • *Reset* and *Delete* have been described.
Export Variable	• The option "Show dialog" has been subdivided into the options: "Show file options" and "Show variable options". In this way, some options can be protected. • New functions: "Always overwrite existing files"
Miscellaneous	Descriptions updated: Browse in workspace, eMail, Open Experiment/Start measurement
Set measurement number	The following functions have been added: • Deselect
Evaluating the last dialog's answer	Behavior described for the cases: • starting the Sequencer • no response provided yet
Execute imc FAMOS sequence	imc FAMOS Dialog - New functions described: • Debug mode • Variable transfer when the imc FAMOS Editor is open. Transfer Table - Revised functions: • Measurement selection • saving of the results

Chapter	Bug-fix
Data Saving Assistant	The Data Saving Assistant only lists the measurement folders which had been created since the start of the last measurement. And not, as incorrectly stated in previous descriptions, the measurement folders created since the last start of the Assistant.
Set variables	It had previously been stated that variables written inside of quotation marks are resolved if they appear within angle brackets (<>). This is not true; a complete placeholder must be used in order for a variable.
Logbook entry	imc STUDIO 3.0 posted a notification dialog when an error message of the category "Fatal" occurred. Since imc STUDIO 4.0, this has been eliminated. However, the function is still described along with the command: "Logbook entry". Now the corresponding description is eliminated also.

Placeholders

Chapter	Amendments
EXPERIMENT/EXPERIMENTS	Note added on the deviating system behavior in absence of the component: imc STUDIO Project Management

Chapter	Bug-fix
EXPERIMENT/EXPERIMENTS	Error in the examples of placeholder resolution.

18.8 in Doc. Rev. 2.2

Setting Up

Chapter	Amendments
System requirements	Win 8 added.
Installation - Preparation	Update description from imc STUDIO 3.x and 4.x.
Installation	Installation description updated
Product configuration / Licensing	Updated and amplified with Edition Runtime.
Info / Version Information	Updated: New Configuration dialog.
Start	New: Start with new, simplified views described.
Command Line Parameters	Updated: Starting an experiment from the data base.
Installing additional languages	New

imc STUDIO - General

Chapter	Amendments
Extra Menu > Options	"Variables Options" is now called "Save current Measurement Data".

Setup - Device Configuration

Chapter	Amendments
Configuring Channels and Variables 159	The channel Setup pages have been splitted into channels and variables
Configuring Channels and Variables 159 and Configuring triggers 182	Trigger event dialog has been removed Description was moved to Setup page Trigger
Ribbon	Ribbon revised partly
Documentation (Experiment Description)	Experiment page renamed to Documentation page

Setup - Advanced Device Functions

Storage Options and Directory Structure

Chapter	Amendments
Targeted Data Saving, or Saving Subsequent to Measurement	New: Save current measurement

Chapter	Amendments
Measurement Storage Area	New

Device Hard Disk, removable drive

Chapter	Amendments
Several	Structural revision and adaptation to the menu guidance for imc STUDIO 5.

Synchronization

Chapter	Amendments
Several	Structural revision and adaptation to the menu guidance for imc STUDIO 5.

imc Display Editor

Chapter	Amendments
Several	Structural revision.

imc REMOTE WebServer

Chapter	Amendments
imc REMOTE WebServer	New

Panel

Chapter	Amendments
Ribbon	Ribbon revised partly
Editing Widget-elements	Example: It has been pointed out that the Widget consists of two standard meters

Video

Chapter	Amendments
All	New

Scripting

Chapter	Amendments
All	Revised and many amendments made

Shared Components

Command Reference

Chapter	Amendments
If / Switch / Loop-While Evaluating the last dialog's answer	New: Evaluating the last dialog's answer

Chapter	Amendments
	New: Command description: • Stop Sequencer • Open Logbook viewer • Delete Panel page • Export Variable • Import Variable • Load Variable • Delete Variable • Data Saving Assistant

18.9 in Doc. Rev. 2.1

imc STUDIO - General

Chapter	Amendments
Navigation pane	Updated: Show and hide.
Tool Windows > Operation	Tool window chooser: Showing/hiding the tool window.
Tool Windows > Logbook 69	Revised completely.
User Administration	Removed: Login without user administration (no longer necessary).
Extra Menu	Options amplified.

Scripting

Chapter	Amendments
All	New

Index

(

( ) Opening Closing Bracket 1444
(%) Modulus 1441
(-) Subtraction 1440
(*) Multiplication 1441
(/) Division 1441
(;) Semicolon 1444
(^) Power 1441
(+) Addition 1440
(<) Less than operator 1443
(<=) Less than or equal operator 1443
(<>) Inequality operator 1442
(=) Equal Operator 1442
(=) Equal Sign 1441
(>) Greater than operator 1443
(>=) Greater than or equal operator 1444

*

*.msg 687

<

<auto> (curve window) 892

1

1/3-octave, octave labeling 910
16-bit integer 172
1-Trigger 185
  Trigger event 192

2

24-bit mode 172
2-point calibration 181
2-point controller 1588
2-point scaling 165

3

3D 910
  + color (=4D) 991
  4D 991
  Color palette 1030
  Display 959
  Extras 1033
  Isolines 1032
  options 1027
  Perspective 1031
  Rotate 1034
  surface 1027
3D-bars 1027
3rd Party Device Management 1414

A

A2L 456
A2L import
  XCP 420
  XCPplus 420
A2L-file XCPoE
  creating 582
  loading 584
ABCRating 1445
Abs 1445
Absolute date, time (curve window) 1012
Absolute time
  Display for synchronized start 655
accelerometers (triaxial) 128
Access codes 704
Access rights of imc user group 79
Accessing the data carrier 617
AccuLength 1446
AccuMax 1446
AccuMean 1447
AccuMin 1447
AccuRMS 1448
AccuStDev 1448
Acos 1448
Acquired
  Point 165
Acquisition time
  UPS 114
Acquisition time after power fail (s) 114
Action
  Definition 186
  Setting bits 192
  Starting data recording 192
  Stop recording of data 192
Actions and targets
  Trigger 192
Activate saving (imc STUDIO Monitor) 1235
activating map mode retroactively 983
Adapt
  Menu (curve window) 1061
  toolbar (curve window) 1061
Adapt Panel page to window size 1083
Adapt Panel page to window size (keep aspect ratio) 1083
Adapt Widgets to the size of the Panel page 1083
adapting maps 981
Adaption channel selection
  More Channels in Curve Window 1003
Add device 296, 300
Adding a device (New...) 622

Addition 1440
Additional Columns 211
Combined column 213
export 82
import 82
Metadata column 219
Parameter set column 221
Parameter set import column 225
save 80
Additionally required runtime environments 1680
Adopting sensor information in the sensor database 128
Advanced Configuration
Interface configuration 308
Advanced Diagnosis 1278
AFDX-Assistant 369
AFDX-Bus
Assistant 369, 370
channel parameter 380
configuration 379
Configuration file list 377
configuration list 379
copy channel 378
cut channel 378
defining search filters 380
delete channel 378
edit channel 377
find channel 378
paste channel 378
Starting the Assistant 370
AFDX-Bus Channel
Byte position 376
create 372
data word 375
destination 374
function state 376
IP address 374
IP address mask 374
LSB value 376
MAC address 374
number format 375
number of Bits 375
Port 374
Scaling 376
settings 374
Source 374
start bit 375
start byte 375
Unit 376
value 376
virtual link 374
Y-Offset 376
AFDX-Bus Configuration file 371
create 371
load 371
save 372
AFDX-Bus Interface 369
Agilent Scope
Agilent 1417, 1419
Channel parameters 1419
Channel settings 1419
Device parameters 1419
Device settings 1419
IO Library Suite 1417
Keysight 1417, 1419
License 1417
Settings 1419
Aligning Widgets 840
Allow data export 1611
Allow discarding of all measurements 1611
Allow login of anonymous users 74
Allow selection of the export folder 1611
alternative IP and FW-Update 306
Amount 195
Amplifiers
Address 117
Module address 117
Modules 117
Permanently installed amplifiers 117
Analog channels 160
Analog inputs 160
AND 1449
Trigger event 192
angle measurement 235
Anonymous users
Logging allow 74
Logging prohibit 74
Appearance
Property - Widget 845
Apply changes
Curve Window 1043
Arangement
Widgets 840, 841
ARINC-Bus
Assistant 345
BCD data word 351
BCD parameter 367
BNR data word 349, 357
BNR parameter 366
Bus speed 364
Channel parameters 364
Configuration 363
Configuration file list 361
Configuration list 363

ARINC-Bus
copying label definition 362
cutting label definition 362
deleting label definition 362
DSC data word 354
DSC parameter 368
editing label definition 361
Equipment-ID 348, 356
Label (Rx) 348
Label (Tx) 356
Label-number 348, 356
Mixed configurations 356
pasting label definition 362
Protocol channel 364
Receiver Channel (Rx) 348
Rx 363
Searching label definition 362
Starting the Assistant 346
Transmitter Channel (Tx) 356
ARINC-Bus Assistant 345
ARINC-Bus Bus speed
All High 364
All Low 364
High 364
Low 364
ARINC-Bus configuration file
Creating 347
idb 347
idb2 347
Loading 347
Save 347
ARINC-Bus Interface 344
Arithmetic mean 798
ARON measuring circuit 784
Arrangement
Axes 1014
Configuration (curve window) 1042
Arrangement (Widgets)
Align Bottom 840
Align left 840
Align Right 840
Align top 840
Cascade 840
To Background 840
To Foreground 840
Arrow keys (move range in curve window) 1073
As presettings
Curve Window 1043
ASCII display (curve window) 931
Asin 1449
Assignment 173
Assistant EtherCAT-IF 464
Asynchronism 805
At defined time 155
Atan2 1449
ATZ 325
Audio 1076
AudioBoardThirds 1450
AudioDevice 1415
Auto: installation type 29
auto-blinking of LED_06 742
Auto-IP 307
Auto-IP (DHCP + APIPA) 306
Automatic loading on demand 820
Automatic timed start 155
Automation elements 1159, 1179
Autorepeat mode 99
Autosize
Property - Widget 846
Autostart 99
Available Waveforms
Context menu 1000
More Channels in Curve Window 1000
Avoidance of data overflow 626
Axes
Arrangement 1014
labeling 910
Scaling 1005
Text 1016
Axes List 996
Symbols 996
Axes Navigation Bar (curve window) 1073

B

Back (curve window) 904
Background
Property - Widget 845
Background color
Property - Pages 1082
Background image
Property - Pages 1082
Background image layout
Property - Pages 1082
Background plain colored
Property - Pages 1082
backround picture (curve window) 979
Balance action
execute via Scripting 1377
Balance by Display 679
Balancing 178
Bar meter (curve window) 910, 935
Base Diagnosis 1272
Basic options

Basic options
J1587-Bus 517
Baud rate 325
BCD data word 351
Beeper outputs 753
Begin action 192
Begin=0 192
Begin=1 192
End=0 192
End=1 192
start 192
stop 192
BitAnd 1450
Bitmapped options
J1587-Bus 514
Bitmask
Property - Widget 846
BitNot 1450
BitOr 1450
Bits
LIN 536
BitXor 1450
Blink duration 1159
Blinking duration 1167
Blinking LED 1159, 1167
Block Comment 755
Block Editor 1140, 1155
BNR data word 349, 357
Border
Background 845
boundaries (curve window) 879
boundary intervals (curve window) 879
Boxplot (curve window) 973
Branching 1159, 1167, 1173, 1179
Break Condition 1119
Breakpoint
Scripting 1320
Bridge 178
Bridge by Display 679
Bridge factor N 167
Bridge resistor 164
Bridge supply 176
Bridge supply (On/Off) 176
Browse in workspace
Layout 1599
Page 1599
Parameters 1599
Buffer duration 592, 593
Buffer size 593
Buffer time
increase 794
Bus Decoder 575, 1269
Button
Storage medium slot 618

C

Cable compensation without sense 182
Calculation of power
Basics 1265
Call by Call Internet access
GSM 705
Land lines 704
Camera 1292
Configuration 1292
Search 1292
Search for Devices 1292
Camera drivers 1292
CAN
blob 407
bus decoder 407
decode all channels 407
CAN channel
Data formats 393
CAN channel not appearing in the imc Online FAMOS 802
CAN configuration CBA
Load 420
Save 420
CAN database import
enable 116, 117
CAN FD 386
Activate 398
CAN Monitor channel 407
CAN protocol
CANAlyser-MDF-Format 591
CAN-1 Protocol 401
CANALYSER 408
CANAlyser-MDF-Format 591
CAN-Bus 382
A2L 456
Analog CAN channel 413
Assignment 426
Assistant 395
Balancing a bridge module via CAN 428
cabling 383
CAN protocol 388
CCP 453
Channel - Error Handling 419
Channel - State 417, 418
Channel - Validity 417, 418
Channel configuration 426
CURVE 456

CAN-Bus 382
Diagnostic On CAN 449
DiagOnCAN 449
Digital CAN channel 417
D-SUB - pin configuration 383
ECU configuring 437
Editing a CAN configuration 397
Error handling 388
Expert settings 400
Format 388
GMLAN 455
ID 388
imc STUDIO as a CAN analyzer 436
Initialization 393
Intel format 413
J1939 411
KWP2000 443
KWP2000 for TP2.0 449
Log-channel 417, 427
MDF 408
Message log 408
Messages 387
Monitor channels 427
Motorola format 413
Number of CAN nodes 386
Number of samples per bit 400
OBD-2 450
Options 425
Quick Tutorial 396
Receiving a message 408
Safety note 384
Sample location in bit time 400
Sampling interval 426
Sampling time 426
Send with imc Online FAMOS 758
Sending messages 409
Sensor initialization 393
Shunt calibration 431
Slope detection 400
Specifications 387
Terminator 384
Time stamp 388, 426
Transfer rate 385
UDS 456
VAL_BLK 456
Validity 401
XCP 455
Y-adapter 383
CAN-Bus (Menu)
Edit 422
Extras 425
File 420
Insert 424
CAN-Bus Definition
Baud rate 398
CAN FD 398
Connection 398
Format 398
Nodes 398
Terminator 398
CAN-Bus high speed 382
CAN-Bus Interface
Ordering information 382
CAN-Bus low speed 382
CAN-Bus Synchronization
CAN-1 Protocol 401
CAN-Bus Validity
Acknowledge 402
Deactivate device if no measured value x s 405
Designation 407
Diagnostic Message (DM) 403
Errorframes 405
J1939 403
Monitor channel 407
Support of Wake On CAN 405
Wake on CAN 405
CAN-Bus-Data (curve window) 931
Cancel
Curve Window 1043
Cancel Button
Show message box 1602
Text (Button caption) 1602
Visible (Hide) 1602
Visible (Show) 1602
CAN-database 420, 425
CAN-FD (ISO 11898-1:2015) 382
CAN-message
Sensor delay time 412
CanMsg 1584
CanMsg.GetData 1585
CanMsg.SetData 1586
CanMsg.Transmit1_S1 1586
CANopen object folder EtherCAT-IF- 460
CANopen over EtherCAT-IF 459
CAN-TimeStampASCII 798
Capabilities FlexRay 483
Capabilities MVB-Bus 540
Cascade Widgets 841
CBA import 420
CCP 390, 453
ccv file
loading 856
saving 858

CE Certification 13
Centered
Background image 1082
Certain parts from data (curve window) 1035
Certificates 13
CF card (Compact Flash) 619
CFast Storage medium 620
Change requests 14
changing the scale 1053
Channel 1143
Access via Scripting 1338
user-defined 1338
Channel action
execute via Scripting 1377
Channel definition (Setup page) 163
Channel file name 587
Channel info
More Channels in Curve Window 1004
Channel name 755
Channel name for the log 527
Channel number 176
Channel numbering 117
Channel options
J1587-Bus 513
Channel parameters
get/set via Scripting 1327, 1377
Channel playback 1416
Channel properties
get/set via Scripting 1377
Channel table 160
ChannelLoader 1416
Channels 160
Access via Scripting 1338
add to SimplePollDevice 1435
add to Third Party Device 1410
Parameters (Scripting) 1327
Properties (Scripting) 1327
Saving (Scripting) 1327
Channels with invalid properties 425
Charact 1451
Characteristic curve
creating 125
Characteristic curves
export 82
Characteristic lines 772
CiA Draft Standard 382
Circular buffer operation 599
Class Learning Time 1273
Class-counting 790
ClHistogram 1452
Clipboard 1044
curves 902
clipboard format (curve window) 1056
Clipboard settings 1044
Font 1044
Line style 1044
Line width 1044
plot 1044
Print 1044
printout 1044
Ticks 1044
ClLevelCrossing 1453
ClMarkov 1454
Clock 99
Clock types
DCF77 632
GPS 632
IRIG-B 632
NTP 632
PTP 632
RTC 632
SyncRTC 632
Close folder on command 616
CloseSaveInterval 1455
CloseSaveInterval() 616
ClRainFlow 1456
ClRainFlowRes 1457
ClRainFlowTM 1458
ClRainFlowTMRes 1459
ClRangePairCount 1460
ClRevolutionsHistogram 1460
ClRevolutionsMatrix 1461
ClTrueMax 1461
ClTwoChannelHistogram 1461
Cluster FlexRay
Properties 489
Cluster measuring FlexRay 497
Cluster size 608
CodeRange 1462
CoE 459
Coldstart FlexRay 498
Collection Editor 211
Color
Title 844
Color components 1050
Color grandient
Background color 1082
color map (curve window) 910, 991
Display 946
general 949
ISO-lines 954

color map (curve window) 910, 991
measure mode 873
Waveforms with associated x-,y-,z-variables 957
color of Y-axis 1017
Color palette 991
3D 1030
Color spectrum 946
Coloring
Property - Widget 845
Colors (curve window)
Copy 1050
Curves 1050
Screen/Printer 1050
Column captions 227
Column description
save 80
Column descriptions
export 82
import 82
Column ID
CurrentValueDisplay 177
eBridgeSupplyOnOffAction 176
eEnumeratedChannelNumber 176
ePlugInName 176
Combination mode 230
Combinations of events 192
Combined column 213
Combined event 187
Combined Parameters 214, 223
Command
Delete Variable 1648
Execute imc FAMOS sequence 1637
Export Variable 1645
If 1594
imc FAMOS project 1637
imc FAMOS sequence 1637
Import Variable 1647
Load Variable 1647
Open Logbook viewer 1603
Panel page entfernen 1625
Set variables 1649
Stop Sequencer 1596
Command Help 1138
Command line parameters 43
Command name 1351
Command Reference 1592
Command Sequence 1650
Commands
copy 1195
copying 1655
delete 1195
Deleting 1655
execute via Scripting 1351
move 1195
moving 1655
Run script 1644
CommandWindow 1138
CommandWindow: Command Help 1138
CommandWindow: Formula Assistant 1138
Comment 163
Property 770
Title - Default Source 844
Comment a line 755
Comment several lines 755
Communicator
create scroll link 905
Transfer to imc FAMOS 859
Compact Flash 619
comparator conditioning
incremental encode 233
Comparison of data 195
Compiler
Error CS0029 1403
Error CS0234 1403
Error CS0525 1403
Error CS1729 1403
Complete layout 228
Complete layouts
export 82
import 82
Components
Product configuration 36
COM-Port 325, 1401, 1409
Computations (Local Limits) 1119
Conditional AND
Trigger event 192
Conditions (Break Condition) 1119
Configuration
Email 688
FAX 694
Configuration (curve window)
Arrangement 1042
loading 856
Menu 909
saving 858
Configuration EtherCAT-IF 460, 465
Configuration file 347
Configuration variant I (no mapping) 464
Configuration variant II 464
Configuration with mapping EtherCAT-IF 465
Configuration without mapping EtherCAT-IF 464
Configure via FTP - Syntax 713

Connect 91
PPP 331
Connect (curve window) 904, 905
cross-section with 3D 906
connect device 296, 298
Connected sensors 122
Connected to sensor 204
Connecting via LAN in four steps 296
Connection
to Device 41
Connection failed - transfer configuration 60
Connection status
get/set via Scripting 1374
Connector 176
Constraints FlexRay 483
Constraints MVB-Bus 540
Context menu
Available Waveforms 1000
Curve window 1072
Context menu imc Inline FAMOS
Editor 1252
Functions list 1253
Help window 1253
Variables list 1253
Context menu imc Online FAMOS
Channels directory tree 773
Editor 774
Functions directory tree 774
Help box 774
Context Script
Activity Scope 1365
create 1312, 1314
Parameter changed (Event) 1365
Status changed (Event) 1365
Value changed (Event) 1365
Continuous numbers
Path naming 603
CONTROL 1173, 1657
Control Commands
Trigger 199
Control of synchronization 659
Control sync state - PC 660
Control units 437
Control units (ECU) 390
Control units in imc Online FAMOS 440
Controller functions 1588
Coordinate system size 910
Coordinates (map) 990
Copy & Paste 1072
Copy data to the PC 621
Correction 164
Cos 1462
Counter 229, 1167, 1594
Counting loop 1121
Coupling 164, 167
Coupling is DC 164
Create a process vector variable in imc Online FAMOS 756
Create an Access Point 315
CreateVChannel 1462
CreateVChannelInt 1463
Creating a new sensor 125
Creating the columns with the Collection Editor 211
CRFX Modules 161
Cross sections
Lines 1025
Cross-section with 3D
connect 906
CtPID 1589
CtPID.Calc 1590
CtTwoPos 1591
CtTwoPos.Calc 1592
cumulative measurements 232
Current device configuration
Interface configuration 308
Current PC configuration
Interface configuration 308
Current value 177
Currently not reachable
Interface configuration 308
CurrentValue 1463
CURVE 456
Curve configuration
loading 856
saving 858
Curve window
Add channel 1342
Axes 1342
changing the scale 1053
clipboard 902
clipboard format 1056
construction 995
context menu 1072
Date/time absolute 910
Days/hours/minutes relative 910
Display of synchronized start 655
Double-click on empty areas 1056
Enable horizontal measurement cursor 1056
Fill mode 1035
Font 1053
Global Handling 1056
Graphic export 1056

Curve window
history 1053
import a picture or map as backround 980
Introduction 854
legend 976
link 1053
Max. delay [ms] 1053
Measure 868
Measure window 868
More Channels 995
Navigation in x-direction 1056
New coordinate systems 996
New x-axis 996
New y-axis 996
Oscilloscope mode 1035
PDF orientation 1056
Rezoom 1056
Scripting 1342
scroll links to tables 905
Scroll mode 1035
selection mode 1053
settings 1053
Size of measurement point 1056
sound output 1053
Start delay [ms] 1053
Stretch mode 1035
Time-Shift in the ccv 1053
Transfer to imc FAMOS 1056
Undo 1053
Updating 792
XY plots 1002
y-Axis 1342
Zoom 1056
Curve Window - "Free floating" 1077
Customer Support 13
Cut sound
sound output (curve window) 1067

D

DAC
imc Inline FAMOS Scaling 1250
Scaling 1250
DAC does not output every value 802
Data formats
CAN-Bus 393
J1587-Bus 511
Data overflow 592, 593, 605, 626, 793, 794
avoidance 626
Buffer duration 592
Buffer size 593
on the internal (removable) hard drive 593
Storage medium 626
SysLog-file 626
Data processing on the PC 594
Data rate 627
reduce 794
Data Saving Assistant 1609
Allow data export 1611
Allow discarding of all measurements 1611
Allow selection of the export folder 1611
Default button 1612
Default path 1611
Delete original files 1611
Deleting a measurement 1611
Discard measurement 1611
Export measurement settings 1611
imc Format Converter 1610
Measurement comments 1610
Parameter set 1610
Prompt to confirm discard 1611
Setup pages 1610
Data storage 603
Data stream 586
Data transfer 621, 1413
FTP access 623
LIN-Bus 521
Storage medium 620
Data type
16-bit integer 172
24-bit mode 172
Float (24-bit mode) 172
Data types
Memory Requirement 608
Data Browser 818
Access via Scripting 1338
Check in 823
Checkout 823
current measurement in the Data Browser 819
Measurement number 821
measurements in the Data Browser 819
Navigation mode 822
Navigation step size 822
renaming measurements 819
saved measurements 819
Update to revision 823
Data Browser columns
Category 819
Comment 819
Event time 819
Metadata columns (optional) 819
Name 819
Numbering column (#) 819
Unit 819
Database conversion 23

DATABASE/DATABASES 1658
Dataoverflow 800
DataTable
Access via Scripting 1338
Date (format)
imc Messaging 698
UDP Status Monitoring 698
UDPNoise 698
Daylight saving time 630
Days/hours/minutes relative (curve window) 1012
dB 1464
DBC
J1939 411
DBC import 420
DCB 425
DCF77 155
External clocks 633
Interrupted reception 155
Synchronization 646
D-component 786
D-Controller
imc Online FAMOS 785
Deactivate saving (imc STUDIO Monitor) 1235
Deactivating the process vector variables 763
Debug
Breakpoint 1320
Scripting 1320
Decimal places
Property - Widget 846
Default 1581
Default button
Data Saving Assistant 1612
Default configuration
FTP 694
Default directories in imc Online FAMOS 772
Default path
for export 1611
Default printer (curve window) 861
Default Source
Title 844
Default Value 220, 1147
Default values 107
Applied condition 107
Default value 107
Element type 107
Enumerator class 107
Group name 107
Parameter 107
Parameter identifier 107
Row scope 107
Definitions FlexRay 482
Definitions MVB-Bus 539
Definitions XCP
CCP 580
XCP 580
XCPoE 580
DelayBuffer 1464
Fill 1465
Next 1466
SetSize 1467
DelayBuffer.Fill 1465
DelayBuffer.Next 1466
DelayBuffer.SetSize 1467
DelayLine 1468
Delete
Experiment 54
Project 54
Set variables 1649
Views 80
Delete original files 1611
Delete Variable 1648
Show dialog 1649
Treat error as warning 1649
Variable name 1648
Variables list 1648
Deleting a measurement
Data Saving Assistant 1611
Deleting sensor information from channel 124
Deleting the process vector backup file 765
Delta-connection 785
Delta-x 770
Delta-z 770
Demo: installation type 29
Description 148
Design Mode 832
Designation data folders 603
Design-Modus
(de)activate via Scripting 1351
DEV001.MSG 687
Deviation of system time from NTP 763
Device
add 296, 300
connect 296, 298
connection 41
Properties 113
Device actions
execute via Scripting 1351, 1374
Device configuration 113
Device connection via LAN 307
Device group 88, 337
Device information
Interface configuration 308

Device Interfaces 308
Device is known 140
Device list 296, 300
Device overview 88, 337
Device parameters
get/set via Scripting 1374
Device properties
get/set via Scripting 1374
Device Search (imc STUDIO Monitor) 1235
Device selection 140
Devices
add to Third Party Devices 1410
Parameter (Scripting) 1323
Parameters (Scripting) 1374
Properties (Scripting) 1323, 1374
Devices deselected 60
Devices/Channels
Tool Window 1236
Device-specific ESI
EtherCAT-IF 469
DFilt 1468
DHCP 305, 307
DHCP ClientID 301, 306
DHCP Hostname 306
Diagnostic channel (P)PID 194 507
Diagnostic Message (DM) 403
Diagnostic On CAN 449
DiagOnCAN 449
Dialog response 1354
Diff 1469
differential measurement procedures 232
Digital bits
Trigger target 192
Digital channels 160
Digital events
Signal = 0 190
Signal = 1 190
Signal change: 0 -> 1 190
Signal change: 1 -> 0 190
Digital inputs / outputs (bits and ports) 160
Digital output
Trigger target 192
Dimension Lines 886
DIN-EN-ISO-9001 13
DirClosed 605, 606
Directory Structure 603
Disable if this column is empty 214
Discard measurement
Data Saving Assistant 1611
Disconnect 91
PPP 331
Disconnect from running measurement 60
DiskFreeSpace 1470
DiskFreeTime 1471
DiskRunDir 1472
Diskstart 99
execute via Scripting 1323
selection via Display 669
Display
3D 959
backlight 667
Balance 679
bar meter 673, 935
Browse 677
color map 946
Combobox 672
Compatibility mode 665
Connector 114
Curve window 675
Diskstart selection 669
Display Demo 667
Display editor 666
display mode 667
Display variables 664
exit 667
External display 664
FAQ 800
Firmware update 667, 682
Function keys 676
GPS connector 114
Input control 671
Input/ Output control 671
Integrated display 664
Keyboard 668
Line chart 675
Needle meters 674
Number (curve window) 927
Output control 671
Overview 664
page timer 667
Push button 678
Recognize the Display type 664
Refresh rate 664
Set values 678
Shunt calibration 681
Static text 670
System menu 667
Text selection 672
Time 671
Toggle button 678
Type of display 114
waterfall diagram 921

Display (curve window) 910
Display before trigger events 199
Display editor 666
Add page 667
Delete page 667
Export configuration 667
Import configuration 667
Display editor menu
Device 667
Edit 667
Extras 667
File 667
Object 667
Display for synchronized start
Absolute time 655
Curve window 655
Display format 214
Display format parameter 214
display options
third-octave axes 965
Display Setup dialog 1635
Display type
recognize 664
display uncertainty 1023
Display update interval
Histogram 175
Display_variables 161
Display-GPS connector 114
DisplaySetButton 1472
DisplaySetPage 1472
Display-variables 160
LIN 536
distance measurement 235
Division 1441
DM-services 403
DNS-server-address 301
Dock
Property - Widget 845
Dock (Widgets)
Below 842
Fill 842
Free 842
Left 842
Right 842
Top 842
Docking
Tool window 69
Documentation 19
Documentation (Setup page) 148
Domain 301
Double-click on empty areas (curve window) 1056
DoWhile 1121
Download 93
Drag & Drop (Sequencer) 1194
Drag&Drop
in curve window 1072
Drive Element 1287
Drive elements
Belt Gearing 1279
Clutch 1279
Drive Shaft 1279
Gear Drive 1279
Planetary Gear 1279
Rolling Bearing 1279
Shaft 1279
Steady Frequency 1279
Steady Order 1279
Three-Phase Motor 1279
DSC data word 354
DST 630
DSUB-9
FlexRay 484
LIN 519
DSUB-Plug
EMD (MVB-Bus) 545
ESD (MVB-Bus) 546
Duration 172, 1143
Duration of events 191
Dwell time
Definition 186
for Trigger events 191
max 191

E

ECAT-IF 458
ECU 504, 518
Clear Diag Information 1474
Cmd Return 1474
Properties in common 457
Read Trouble Codes 1473
Start Session 1473
Stop Session 1473
ECU configuration 437
ECU configuring 437
ECU Control units 390
ECU Monitor channel 407
ECU requirements 390
ECUClearDiagInformation 1474
ECU-protocol
enable 116, 117
ECU-protocols 390
ECUReadTroubleCodes 1473

ECUStartSession 1473
ECUStopSession 1473
ECU-version
Querying 504
edge (incremental encoder) 235
Edit
Menu (curve window) 864
signals (curve window) 874
Edit box color
Property - Widget 846
Edit EtherCAT-IF 466
Edit menu 56
Else 1579
Email 706
Configuration 688
Message Types 685
via LAN 706
via modem 707
E-Mail 13
Embedded 214
EMD connection (MVB-Bus) 545
Enable
ECU-Protokolle 116, 117
imc CANSAS 116, 117
imc Online FAMOS 116, 117, 742
imc Online FAMOS Professional 116, 117
imc STUDIO Monitor 116, 117
Online-Klassierung 116, 117
Online-Ordnungsanalyse 116, 117
Enable horizontal measurement cursor 1056
End online mode (curve window)
Scroll mode 1035
Entering range 189
Entry format 220
Enumerated channel number 176
Enumeration classes 254
Environment variables of the OS 1670
Equal 1475
Equal Operator 1442
Equal Sign 1441
eRange 1412
Error 69
Error handling (Exception & Error Events) 1119
Error Message 103 807
Error/Exception 1143
ESD connection (MVB-Bus) 546
ESI
EtherCAT-IF 469
ESI device-specific
EtherCAT-IF 469
ESI EtherCAT-IF 469
ESI save
EtherCAT-IF 469
EtherCAT-IF 458
CANopen object folder 460
Configuration with mapping 465
Configuration without mapping 464
Device-specific ESI 469
ESI 469
ESI device-specific 469
ESI save 469
Inputs 467, 468
Mapping 468
Outputs 466, 468
PDO 468
PDO add 467
PDO Create automatically 467
PDO delete 467
Prerequisites 459
RxPDO 468
TxPDO 468
Xml 469
EtherCAT-IF Assistant 464
EtherCAT-IF configuration 460
EtherCAT-IF general 459
EtherCAT-IF Menu Configuration 465
Accept changes 465
Export 465
Import 465
New 465
EtherCAT-IF Menu Edit 466
Add PDO 467
Create PDOs automatically 467
Delete PDOs 467
Inputs 467
Mapping 468
Outputs 466
EtherCAT-IF Menu ESI 469
Save device-specific ESI 469
Save general ESI 469
EtherCAT-IF Menu Show 468
Inputs 468
Inputs/Outputs 468
Outputs 468
PDO 468
RxPDO 468
TxPDO 468
Xml 469
Ethernet for Control Automation Technology 459
Ethernet-Bus Interface 469
Evaluating the last dialog's answer
If 1598
Switch 1598

Evaluating the last dialog's answer
While Loop 1598
Event 195
Definition 186
Event (curve window) 1035
Event history (imc REMOTE) 739
Event Script
create 1312, 1314
Measurement storage area 1372
Event type
Digital 190
Level 189
Range 189
Events
Open Experiment/Start measurement 1615
Property - Widget 846
Trigger 187
events counting 229, 234
Events from digital signals 190
Events from virtual bits 190
EventTyp 1143
Excel-Display (curve window) 931
Exception & Error Events 1119
Exception handlers 1179
Exceptions 1153
Execute imc FAMOS sequence 1637
Execute menu action
Resume data storage 611
Suspend data storage 611
Exiting range 189
Experiment 54, 83
Definition 595
Delete 54
Export 54
Import 54
new 55
save 55
save as 55
Experiment file
Open Experiment/Start measurement 1615
Experiment from template
new 55
Experiment has changed 60
Experiment parameters
get/set via Scripting 1383
Experiment template 54, 84
Export 54
Import 54
new 54
Preferred 55
Experiment, open
Automatically upon starting imc STUDIO 43
With a specific imc STUDIO installation 43
EXPERIMENT/EXPERIMENTS 1660
ExpoRMS 1476
Export
Experiment 54
Experiment template 54
Interface-configuration 310
Measure window 870
Project 54
Scripting 1318
Video 1301
Export all channels
Mapping 246
Export all channels and settings
Mapping 246
Export DBC 420
Export measurement settings 1611
Export of views, metadata columns, sensors, ... 82
Export Panel page 1625
Export Variable 1645
Add variables 1646
Always overwrite existing files 1646
File format 1646
File options 1645
Filename 1646
Folder 1645
Mapping instruction 1645
Measurement 1645
Measurement name 1646
Save all in a single file 1645
Save to one file per file format 1645
Show file options 1646
Show variables options 1646
Treat error as warning 1646
Variable name 1646
Variables 1646
Variables list 1646
ExpressCard Storage medium 620
Extended Format
CAN 398
Extended+ Format
CAN 398
External display 664
Extra menu
Administration 58
Login 58
Logout 58
Metadata Assistant 58
Options 58
Start (Sequencer) 58
Startup behavior 58

Extra menu
Stop (Sequencer) 58
User Access Rights 58
Extras (curve window)
3D 1033
Lines 1023
Menu 1060
Menu and Toolbar 1060

F

Factor 165
Property - Widget 846
Factor-offset scaling 165
Factory calibration 178
Falcom A1, A2D-1, A2D-2 332
FAQ
Compiler error 1403
imc Messaging 702
Scripting 1403
User-defined measurement storage area 616
FAQ synchronize devices 655
Fast loading
Property - Pages 1082
FAT12 801
FAT16 608, 801
FAT32 608, 801
Fatal 69
Fault codes
J1587-Bus 515
FAX
Configuration 694
Message Types 684
FFT 1411, 1477
FFTAmplitudePhase 1477
FFTAverage 1478
FFTInverse 1479
FFTRealImaginary 1479
FIBEX FlexRay 496
FIBEX-file FlexRay 497
FIBEX-Plus FlexRay 496
Field bus
Analog inputs 160, 173
Digital inputs / outputs 173
Digital inputs / outputs (bits and ports) 160
Field-bus
General notes 338
Operating methodology 338
Sampling 340
Timestamping 342
Field-bus channels
GPS 586
File
Menu (curve window) 855
File Extensions 811
File format
CANAlyser-MDF 591
idb 347
idb2 347
imc FAMOS 590
Windows Video File (AVI) 1301
File header 607
File name for channel 587
File system 627
File System FAT16/FAT32 624
Filesize (maximum) 624
Fill mode
Curve window 1035
FiltBP 1480
FiltBS 1480
Filter
instable state 1480
Filter coefficients 772
Filter digital
Limits 797
filtering (curve window) 876
FiltHP 1481
FiltLP 1481
Firewall 302
Security alert 41
Firmware update
blocking / unblocking 146
Display 667, 682
Logbook 143
password 146
Fix Axes 904
Fixed-point (curve window) 927
FlexRay 482
capabilities 483
Cluster 489
Cluster measuring 497
Cluster with missing Coldstart-node 498
Coldstart 498
constraints 483
Definitions 482
DSUB-9 484
FIBEX 496
FIBEX-file 497
Fibex-Plus 496
Frame Trigger 492
Frames 491
Frame-Triggerings 492
Import of FIBEX-Plus files 496

FlexRay 482
KeySlotID 497
Loading FIBEX-files 496
Menu 487
Node mode 497
Overview window 486
PDU 496
Pin configuration 484
Properties of Clusters 489
Properties of Frames 491
Properties of Frame-Triggerings 492
Properties of Signals 493
Properties Window 486
Rules for the KeySlotID and node mode 497
Settings 487
Signals 493
FlexRay-Assistant
Introduction 485
Float (24-bit mode) 172
floating-point notation 927
Floor 1481
Flush right (curve window) 927
Font
Property - Widget 845
Title 844
Font (curve window) 1053
Last value as number 927
For 1580
For changed message clock rate 425
Format Converter 1616
format marker (curve window) 892
Format of numerical variables
imc Messaging 697
UDP Status Monitoring 697
UDPNoise 697
Formatting 625, 627
Formula Assistant 1138
Forward (curve window) 904
four dimensions display 991
Frame Trigger FlexRay
Properties 492
Frames (MVB) 554
Frames FlexRay
Properties 491
Frame-Triggerings FlexRay
Properties 492
Freely positioning
Tool window 69
Frequency 237
frequency bands
third-octave display 965
Frequency error
Synchronization 640
From the Variable
Range 846
FTP
Configure 707
Configure via FTP - Syntax 713
Default configuration 694
FTP access
Data transfer 623
Fullscreen 1086
Automatically upon starting imc STUDIO 43
Launch 1086
Fullscreen Mode
Panel 814, 815
Function 172
Function dialog
imc Online FAMOS 777
Function keys display 676
Function Reference
imc Online FAMOS 1440
Functions 1159, 1179
FunctionSimulator 1425
FW-Update and alternative IP 306

G

Gauge factor 167
GearRatio 1482
General terms and conditions 13
Generate default values from selection 107
GetDateTime 1482
GetDuration 1484
GetHistoValue 1484
GetHistoValue2 1485
GetLastError 1486
GetSampleCount 1488
GetSamplingTime 1489
GetSoftwareVersion 717
Global buffer duration 593
Global Handling (curve window) 1056
Global RAM 796
GMLAN 455
GMW 3110 Version 1.5 455
GPS 767
connector 114
External clocks 634
Synchronization 645
GPS + DCF Master/Slave
Synchronization 642
GPS channels 586
GPS data as XY-plot 986

GPS Master/Slave
Synchronization 642
GPS-Display connector 114
Graphic export (curve window) 1056
Greater 1489
Greater than operator 1443
Greater than or equal operator 1444
GreaterEqual 1489
Greek characters (curve window) 1069
Grid
Curve Window 1026
Grid (curve window) 910
small x-ticks (-1 = auto) 910
small y-ticks (-1 = auto) 910
Grid (Panel)
Realign to Grid 1085
Snap to Grid 1085
Grid size
Property - Pages 1082
Group
Device overview 88, 337
Group (Widgets)
Create Group 843
Dissolve Group 843
Enter Group 843
Leave Group 843
GSD files 558
GSD folder 558
GSM-Modem 325
Guarantee 13

H

Hand-Terminal
Typ 114
Hard drive 88, 337, 617
System requirements 20, 1680
Hardware-Handshaking 330, 331
harmonic cursor 899
Headline
Show message box 1602
Heavy burdening of the processor 793
Height
Property - Pages 1081
Help 19
Hide disabled columns from caption 214
Hide incomplete measurement files 606
Hiding
Tool window 68
HighLowRatio 1489
high-pass (curve window) 876
HiL 205
Histogram display update interval 175
Histogram storage interval 175
history (curve window) 1053
Homepage 39
Hookup and Circuiting
LIN-Bus 519
Host 302
Hostname (DHCP) 306
Hotline 13
Hot-Plug
FTP access 623
Storage medium 618
Hyst 1490

I

IBC node addressing mode 405
I-component 786
I-Controller
imc Online FAMOS 785
idb 347
idb2 347
Identifier 213
Identity 74
iDiv 1490
IEEE FLoat 763
If 1579
Command 1594
Condition 1594
Else 1594
IF-Config 307, 325
If-instruction 1123
imc CRONOSflex
Module address 117
imc CRONOSflex Modules 161
imc ECAT-Slave-IF 458
imc REMOTE
Event history 739
imc REMOTE WebServer 720
Activation 722
Adapting the design 727
Bar meter 736
Certificate 738
Close a session 724
Curve diagram 737
Designer 725
display language 738
Experiment change 726
Files 730
Grid 731
Language Selection 724
LED 735

imc REMOTE WebServer 720
Monitor 725
Numerical input/output 737
OEM identifier 738
Operator 725
Panel 731
Pointer instrument 735
Press key 734
Shapes 734
Standard - Text area 733
Starting measurement 727
Stopping measurement 727
System requirements 721
Toggle switch 734
User Role 725
imc REMOTE WebServer ume.zip 726
imc Software License Agreement 14
imc STUDIO's bus decoder 418
imc Systems
Formatting 625
imc Application Module 499
imc CANSAS 103
enable 116, 117
imc DEVICES
Installation 34
imc FAMOS 1127
Access to functions via Scripting 1346
Additional imc software products 1702
Current measurement 1132, 1641
Debug Mode 1129, 1638
From imc FAMOS 1132, 1641
Installation 35
Last concluded measurement 1132, 1641
Measurement 1132, 1641
Measurement number 1132, 1641
Open 1129, 1638
Options 1132, 1641
Synchronous event 1129, 1638
Timeout 1129, 1638
To imc FAMOS 1132, 1641
Transfer table 1132, 1641
imc FAMOS project 1637
imc FAMOS sequence 1637
imc FAMOS-Sequence 1127
imc FAMOS-Window 1127
imc Format Converter
Data Saving Assistant 1610
Installation 31
imc HiL 205
imc HiL Interface 500
imc Inline FAMOS 1243
adding channel names 1250
adding functions 1250
Apply 1250
Check - Syntax 1250
Comparison: imc Online FAMOS 1244
Context Menu 1252
Data flow schematic 1244
Opening the Editor 1249
Ribbon 1242
Saving the source text 1250
Scope of functions 1248
Supplemental Files 1252
Syntax check 1250
System prerequisites and limitations 1247
User interface 1249
imc Inline FAMOS Context menu
Editor 1252
Functions list 1253
Help window 1253
Variables list 1253
imc LICENSE Manager 37
imc Messaging 682
Basics 682
Configuration 687
Configuration file 687
Error sources 702
FAQ 702
Format date/time 698
Format of numerical variables 697
Message types 684
Network bits 699
Recognizing events 685
Syntax 696
System requirements 684
Technical Specifications 704
Templates 704
UDP Status Monitoring 700
UDPNoise 700
Variables 695
imc Messaging Configuration
Email 688
FAX 694
imc Messaging Message Types
Email 685
FAX 684
imc Online FAMOS 739
Additional imc software products 1702
Analog output 753
Beeper 753
Capabilities 743
Class-counting 790
Configuration and device equipment 742

imc Online FAMOS 739
Context menu 773
Control units 440
DAC 753
Default directories 772
Digital Inputs 752
Digital output 753
Directories 769
Display variable 753
enable 116, 117, 742
Ethernet bit 753
Example - Simplified motor simulation 760
Export as Text File 768
External program editor 769, 772
FAQ 802
Filter digital - Limits 797
Formula Assistant 775
Function dialog 777
Function Reference 792
Global RAM 796
How does imc Online FAMOS work? 740
ini-Examples 780
LED 753
Local channels 751
Local single value 751
Local variable 751
maximaum stack 791
Menu 768
Menu Edit 769
Menu Extras 769
Menu File 768
Menu Window 770
Nesting depth 791
ONLFAMOS.INI 780
Polling Digital Inputs 752
Process vector 763
Reference 792
Save to Configuration 768
Schematic 740
Single value 751
Starting 747
Time Base for Channels in a Formula 755
Trigger 753
Virtual bit 753
imc Online FAMOS / imc Inline FAMOS
Comparison 1244
comparison example 1246
imc Online FAMOS Context menu
Channels directory tree 773
Editor 774
Functions directory tree 774
Help box 774
imc Online FAMOS Formula Assistant 775
Nested expressions 775
User interface 775
Using the Formula Assistant 775
Writing multiple formulas 775
imc Online FAMOS Professional 746
Additional imc software products 1702
Controller 785
D-Controller 785
enable 116, 117
I-Controller 785
P-Controller 785
PI-Controller 785
PID-Controller 785
System requirements 741
imc Online FAMOS program is not saved 802
imc Online FAMOS with Control Command 744
Array 756
Comparison operators 757
Create a process vector variable 756
Create a varialbe 756
Data array 756
Single value 756
SingleValueChannel 756
imc Online FAMOS with control commands
Activate control commands 745
Comparison operators 745
Control commands 745
Control functions 745
imc SENSORS
Additional imc software products 1702
Adopting sensor information in the sensor database 128
Importing sensor data 128
Installation 33
Reset channel's sensor information 203
supported sensors 204
imc Shared Components
Installation 32
imc STUDIO
Help 12
Installation 33
Start 39
imc STUDIO Automation 1102
imc STUDIO DataProcessing 1241
Ribbon 1242
imc STUDIO Monitor
enable 116, 117
imcDB://DB 53
imcSyslog 605
Immediately 154
Import

Import
Experiment 54
Experiment template 54
Project 54
Scripting 1318
Video 1301
Import / Export
Menu 51
import a picture or map as backround (curve window) 980
Import DBC, A2L, CBA 420
Import mapping by name
Mapping 246
Import of channels according to connector
Mapping 246
Import of channels according to connector and device serial number
Mapping 246
Import of views, metadata columns, ... 82
Import Panel page 1625
Import Variable 1647
Add file 1647
Add file element 1647
File format 1647
File options 1647
Folder 1647
Import all 1648
Show dialog 1648
Target variable name 1647
Treat error as warning 1648
Variable name 1647
Variables list 1647
Importing sensor data 128
Importing the database
J1587-Bus 508
ImportRoot 227
Increment
Property - Widget 846
Incremental counter does not go to zero 801
incremental encoder 229
comparator conditioning 233
edge 235
maximum input range 232
scaling 232
start edge 235
stop edge 235
time measurement 235
Incremental encoder inputs 160
incremental encoders
combined measurement 237
index-channel 234
Inequality operator 1442
Information 69
Init 1151
Init / Terminate 1151, 1159, 1167, 1173, 1179
Initial Action 1119
Initialization
CAN-Bus 393
Sensor 393
Initialization (Initial Action) 1119
Initialization script 325
Initializing a sensor 434
Inline FAMOS
private Task 1349
Scripting 1349
Input range 164, 165, 167, 803
Inputs
EtherCAT-IF 467, 468
Insert Widget 834
Inside of range 189
instable filter functions 1480
Installation 21
imc DEVICES 34
imc FAMOS 35
imc Format Converter 31
imc SENSORS 33
imc Shared Components 32
imc STUDIO 33
McAfee 25
PROFIBUS Configurator 558
Step by step 26
Update 21
User account control 25
Installation type
Auto 29
Demo 29
User-defined 31
Integral 1490
Integral2 1490
IntegralFFT 1491
IntegralP 1491
IntegralP2 1492
Integrated display 664
Interface
imc HiL 500
Interface configuration
Device Interfaces 308
Interface-Configuration 310
Export current settings 310
Internal storage media 617
Internet Explorer 801
Internet Settings for maps 990
Interrupted reception 155

Interval
Property - Widget 847
Interval (Timeout)
Show message box 1602
Interval funktions 1492
Interval saving 600
IntervalFrom1Level 1494
IntervalFromLevels 1495
IntervalFromPulse 1496
IntervalMax 1496
IntervalMean 1497
IntervalMin 1497
IntervalMult 1498
IntervalResample 1499
IntervalRMS 1499
IP address
of the PC 296
IP-address 301, 328
configure 296, 299
of the devices 296, 299
of the PCs 296
IRIG-B
External clocks 633
Synchronization 646
ISO 11519-2 (CAN-Bus low speed) 382
ISO 11898 (CAN-Bus high speed) 382
ISO-9001 13
isolated synchronization 662
Isolated thermo couple 164
Isolines
3D 1032
ISOSYNC 662
Issues
Storage medium 628
IsSyncronized 1500

J

J1587_EcuVersion 504
J1587-Bus
Basic options 517
Bitmapped options 514
Channel 513
Data formats 511
ECU 518
Fault codes 515
Importing the database 508
Measurement channels (PID) 512
Request options 514
J1587-Bus Assistant 508
J1587-Bus Interface 501
J1939 411
CAN 403
DBC 411
Jiggling display 46, 793
JKFlipFlop 1500

K

KiRoad 569
Konfiguration übertragen 1072
KWP2000 390, 443
KWP2000 for TP2.0 449
KWP2000 TP2.0 390

L

Labeling
Lines 1021
LAN 301
Device connection 307
Language installing 45
Last value as number (curve window) 910
Display 927
Font 927
Launching in Fullscreen mode 1086
Layout 57
Browse in workspace 1599
Customize 57
Reset 57
Tool windows 57
Layout-Repository
export 82
import 82
LCP Expansion 328
LED 745
Storage medium slot 618
LED button
Storage medium slot 618
LED output 753
LED_06 blinking automatically 742
legend
curve window 976
legends (line) 1022
LEQ 1501
Less 1501
Less than operator 1443
Less than or equal operator 1443
LessEqual 1501
Level - Trigger-Event type
Negative slope 189
Positive slope 189
Signal < Level 189
Signal > Level 189
Licensable plug-ins 36

License 36
Limit monitoring 1179
Limit value monitoring
Channel 1143
Disable 1143
Duration 1143
Error/Exception 1143
Event Type 1143
Lower Limit 1143
Rank 1143
Upper Limit 1143
Limitations
Storage medium 628
Limited Warranty 13
LIN
Bits 536
Display-variables 536
Prozess vector variables 536
sending variables 536
LIN Sleep 537
LIN WakeUp 537
LIN-Assistant 523
Limitations 524
Properties 523
LIN-Bus
Assistant 523
Baud rate 527
Data transfer 521
DSUB-9 519
Frame (message) properties 531
Frames 526
Hookup and Circuiting 519
Logging 521
Master 521
Master-Breaks (Duration) 527
Master-Breaks Delimiters (Duration) 527
Monitoring 521
Overview window 525
Pin configuration 519
Polling cycle time 527
Properties window 525, 527
Signal 526
Slave 521
Sleep-WakeUp 527
Testing the configuration 526
Wakeup-Breaks (Duration) 527
LIN-Bus Interface 518
line color depend on magnitude 991
Lines (curve window) 1018
Color 1018
Cross sections 1025
Extras 1023
Labeling 1021
Line structure 1018
Line thickness 1018
Line type 1018
Lines 1018
Printer / Screen: 1018
Symbol 1018
Symbol size 1018
Line-Shift 904, 907
Reset 907
link (curve window) 1053
Link with several curve windows 906
linking a map with time data 985
Lissajous figures
curve window 972
List of devices 296, 300
Ln 1501
Load
Video 1301
Views 80
Load CAN configuration CBA 420
Load dialog (curve window) 856
Load Variable 1647
Add file 1647
Add file element 1647
Category options 1647
File format 1647
File options 1647
Folder 1647
Load all 1648
Show dialog 1648
Target variable name 1647
Treat error as warning 1648
Use different category 1647
Variable name 1647
Variables list 1647
Local channels 751
Local Limits 1119
Local single value 751
Local variable 751
Localtime 630
Log channel 402
Log off
Logging a user off 77
Log on
anonymous user 77
registered user 77
LogAnd 1502
Logbook 69
Access via Scripting 1344
Autoscroll 72

Logbook 69
Category 69
Code 69
Deleting messages 71
Entries 1344
Export 1344
Filtering 70
Firmware update 143
Hiding duplicates 71
Location 69
Logbook-Viewer 72
Message 69
Message copying 71
Options for the Logbook 72
Search 70
Sender 69
Sending Email 72
Showing duplicates 71
Logbook (curve window) 931
Logbook category
Error 69
Fatal 69
Information 69
Warning 69
Logbook viewer
Opening by command 1603
LogNot 1502
LogOr 1502
LogXor 1502
Long name
Title - Default Source 844
Long-term measurement 793
Loop 1159
For Loop 1594
While Loop 1594
Loops 1121
Loops: Counting loop 1121
Loops: DoWhile 1121
Loops: While 1121
Lower 1502
Lower Limit 1143
low-pass (curve window) 876

M

MAC-address 333
Magnitude (curve window) 927
Mail servers
imc Messaging 705
Main window 42, 1103
Making a device known 140
Manage Projects
Menu 51
Mandatory field 220, 223
Map 980
cache 990
curve window 984
from Internet 986, 987
link 985
mode 981
provider 987
settings 990
streched 989
map coordinates adjust 980
Mapping 246
EtherCAT-IF 468
Export all channels 246
Export all channels and settings 246
Import mapping by name 246
Import of channels according to connector 246
Import of channels according to connector and device serial number 246
Marker (curve window) 882
connecting line 892
defaults 892
delete 892
Delete all markers 897
extra functions 897
Markers at Min/Max 897
Markers on all lines 897
move 890
new line 884
placeholders 892
position 892
precision 892
Select al lines 897
set 883
text 892
marker precision 892
Master-Send-List
Aktivate 576
Deaktivate 576
Matrix (curve window) 1040
Max 1503
max dwell time 191
max. delay (sound) 1068
Max. delay [ms] (curve window) 1053
Maximaum stack imc Online FAMOS 791
Maximum
Range 846
Maximum amount of variables in imc Online FAMOS 741
maximum input range
INC-channels 232
McAfee 25

MDF 591
Mean 795, 1503
Measure (curve window)
1/dx (frequency) 868
dx 868
Slope 868
Slope per decade 868
xl 868
xr 868
xr/xl 868
yl 868
yr 868
yr/yl 868
yr-yl 868
Measure context menu
Channels List 870
Clipboard 870
Expand List 870
Export Curve Segment 870
Place marker with left click 870
Send Curve Segment... 870
Measure mode
Curve 867
Measure window (curve window) 867, 868
color maps 873
context menu 870, 874
Date/ time display 868
Days/ hours/ minutes display 868
Measurement
Automatic loading on demand 820
Definition 595
loading 820
Loading... 820
start/stop via Scripting 1323, 1351
unloading 820
Measurement comments
about Data Saving Assistant 1610
Measurement cursor (curve window) 867
Measurement devices as a PPP-Server 322
Measurement folder structure 612
Measurement mode 172
Encoder 169
Measurement mode (Setup page) 164
Measurement modes for encoder inputs 229
Measurement number
Assigning 821
Locking 821
Measurement procedure for incremental encoder channels 169
Measurement process states 1573
Measurement running 60
Measurement running - reconnect not possible 60
Measurement status
get via Scripting 1374
Measurement storage area 612
Measurement storage path 612
MEASUREMENT.SQL 1661
Median3 1504
Median5 1504
Memory cards 88, 337
Memory Requirements for Measurement Data 607
Memory volumes 619
Menu
Edit (curve window) 864
Extras (curve window) 1060
File (curve window) 855
FlexRay 487
Options (curve window) 1044
Menu (curve window)
Adapt 1061
Configuration 909
Toolbar 1061
Menu actions
execute via Scripting 1351
Menu and Toolbar (curve window) 1060
Mercator-projection 989
Message
Synchronization 413
Message Types
Email 685
FAX 684
Metadata 54
Metadata column 219
Metadata type
Date 219
Directory 219
Document 219
Logical value 219
Multiline text 219
Picture 219
Single-line text 219
Text from list 219
Time 219
Metadata-column
export 82
import 82
Metafile
transferring curve windows 902
MFBEthernet
Introduction 469
MFBEthernet-Bus
Assistant 470

MFBEthernet-Bus
Channel parameters 481
Configuration 477
Configuration file 471
Configuration file list 475
Configuration list 477
Connection settings 478
copying channel 476
creating channel 472
Creating new Ethernet message 478
cutting channel 476
Defining search filters 481
deleting channel 476
editing channel 476
Message 478
pasting channel 476
Search filters 481
Searching channel 476
Starting the assistant 470
MFBEthernet-Bus channel
Creating 472
Data word 473
Error handling 475
LSB-value 474
Number of bits 473
Numerical format 473
Occurrences 474
Scaling 474
Start bit 473
Start byte 473
Unit 474
Value (Error handling) 475
Y-Offset 474
MFBEthernet-Bus configuration file 471
Creating 471
Loading 471
Save 472
MFBEthernet-Bus Interface 469
MFBEthernet-Bus-Assistant 470
Min 1504
Minimum
Range 846
Minimum accuracy [ms] 65
Minimum length of
Password 74
User name 74
Mode 164, 167
Modem 321, 322
Connect 331
Disconnect 331
Instructions 334
Null modem cable 330, 331
PC as a PPP-client 328
Modify values (curve window) 904
Module 161
Address 117
exchange 117
Properties 117
removing 117
Module address
Assign manually 117
Reassign 117
Module numbering 117, 161
Modulus 1441
Modulus of elasticity E 167
Monitor 1086
Activate saving 1235
Analog inputs 160
Deactivate saving 1235
Device Search 1235
Digital inputs / outputs (bits and ports) 160
Incremental encoder inputs 160
PROFIBUS 564
Ribbon 1235
System Prerequisites and Limitations 1235
Monitor channel
CAN 407
ECU 407
Monitor channels 160, 199
Monoflop 1504
MonoflopRT 1505
More Channels in Curve Window 995
Adaption channel selection 1003
Apply changes 1043
As presettings 1043
Available Waveforms 1000
Cancel 1043
Channel info 1004
OK 1043
Presettings 1043
Set as default 1043
Waveform Preselection 1001
Waveforms in axes list 1002
Most recently use values 220
mouse operation
Drag&Drop 1072
move area in curve window 1072
Move markers
curve window 890
msg 687
Multi-Event 795
Multifunction Vehicle Bus (MVB) 540
Multiline comment 755

Multilingual title
Title 844
Multimaster Synchonisation 649
Multimeter (curve window) 927
Multiple triggering 195
Memory requirements 609
Multiplication 1441
Multi-section Messages 505
MVB
Frames 554
Protocol- and signal-channel name 553
Signals 555
MVB-Bus
Activation 542
Cabling EMD 543
Cabling ESD 544
capabilities 540
Connection 542
constraints 540
Definitions 539
DSUB-Plug EMD 545
DSUB-Plug ESD 546
EMD cabling 543
EMD connection 545
ESD cabling 544
ESD connection 546
General description 540
Overview window 549
Password 540
Prerequisites 540
Properties Window 549
Starting the assistant 548
Topology 542
MVB-Bus Interface 539

N

Name 163
Property 770
Property - Pages 1081
Property - Widget 844
Show message box 1602
Title - Default Source 844
User-defined variables 1147
Names (curve window) 927
Navigation Bar 1093
Navigation bar (curve window)
Axes 1073
Navigation in x-direction (curve window) 1056
Navigation mode (Data Browser)
Default 822
Over events 822
Over measurements 822
Over measurements and events 822
Navigation Pane 42, 67
Navigation step size (Data Browser) 822
Navigator 1103
Necessary system component
Installation 31
Negative slope 189
Nesting depth imc Online FAMOS 791
Net Bits 160, 162
Net_bits 161
Network 801
Firewall 41
Network bits
imc Messaging 699
Network drive 601
network name (device hostname) 306
New
Experiment 55
Experiment from template 55
Experiment template 54
Menu 51
Project 54
New device configuration
Interface configuration 308
New experiment 55
from template 55
Menu 51
save 55
save as 55
Next 1151, 1152
Next-Branching 1159, 1167, 1173
Next-skips 1151, 1152
NMEA 767
No new devices found 60
noise (curve window) 876
NorthCorrection 1505
NOT 1506
NTFS 801
NTP
External clocks 634
Synchronization 647
NTP Master/Slave
Synchronization 643
Null modem cable
Making 330
Null modem with Hardware-Handshaking 330
Null modem without Hardware-Handshaking 331
Number (curve window)
Display 927
Number of folders (max.)
Storage medium 628

Number of last intervals 599
Number of symbols (0 = every sample) 910
Number of trigger releases 195
Number trimming (useful when scrolling) 910
NumberOfPulses 1506

O

OBD-2 450
octave labeling
  display options 965
Offline cache (map) 990
Offset 165, 178
Offset (curve window) 876
OK
  Curve Window 1043
OK Button
  Show message box 1602
  Text (Button caption) 1602
  Visible 1602
  Visible (Hide) 1602
  Visible (Show) 1602
On Screen Display (OSD) 1296
OnAlways 1574
OnCanMessageReceive1_S1 1588
OnECUCmdReturn 1474
OnECUCmdReturn_ECU_001 1506
OnInitAll 758, 1573
Online classification
  enable 116, 117
Online FAMOS
  possiblities 742
Online-Order tracking
  enable 116, 117
Online-Trigger 199
OnMeasureEnd 1575
OnMeasureStart 1574
OnPowerOff 1578
OnSyncTask 1575
OnTimeout 1159, 1173, 1179
OnTimer 1577
OnTriggerEnd 199, 758, 1576
OnTriggerMeasure 199, 1577
OnTriggerStart 199, 758, 1576
Open
  Menu 51
Open experiment
  Menu 51
Open Experiment/Start measurement
  Events 1615
  Experiment file 1615
  Overwrite settings 1615
  Start measurement 1615
  Warning in case of infinite measurement duration 1615
Open imc FAMOS 1129, 1638
Opening Closing Bracket 1444
Opening the imc Inline FAMOS Editor 1249
Opening the Power Quality Editor 1254
Operating System
  System requirements 20, 1680
Operation of the Display editor 666
Operations 1152
Optimize the network 335
Optional components 36
Options 43, 58, 513
  3D 1027
  Dialog response 1354
  Menu (curve window) 1044
  Presettings (curve window) 1053
  Save current measurement data 611
  Save current measurement data as 611
OR 1507
  Trigger event 192
Ordering information
  CAN-Bus Interface 382
Orientation
  Property - Widget 845
Oscilloscope mode
  Curve window 1035
OtrAngleAdd 1507
OtrEncoderPulsesToRpm 1508
OtrFrequLine 1510
OtrFrequLine2 1511
OtrFrequLine3 1511
OtrOrderSpectrum 1512
OtrOrderSpectrumP 1514
OtrPulseDuration 1517
OtrResample 1517
OtrResampleAngle 1520
OtrResampleFromRpm 1521
OtrRpmComplexOrder 1522
OtrRpmOrder 1525
OtrRpmPresentation 1527
OtrRpmPresentVector 1528
OtrRpmSpectrum 1529
OtrSynthSin 1529
OtrTrackingLowPass 1530
Outputs
  EtherCAT-IF 466, 468
Outputting a waveform to a DAC module 432
Outside of range 189
Overview Drive Elements 1287

Overview window 862
Overwrite autostart configurations 60
Overwrite existing file 60
Overwrite settings
Open Experiment/Start measurement 1615

P

P3 time out 443
PAGE 1663
Browse in workspace 1599
Property - Pages 1081
Repository 828
Page size
Property - Pages 1081
PAGE.NUMBER
Formatting 1668
PANEL 1663
Access via Scripting 1334
Add Windows.Forms 1334
Repository 828
Tables 1334
Video 1296
Panel Fullscreen Mode 814, 815
Panel page
Adapt Page Size 1083
Deleting by command 1625
Repository 828
Zoom 1085
Panel page as dialog 1620
Panel Script 1358
create 1312
Data Browser 1358
New Widgets 1358
Variables 1358
Widgets 1358
PANEL.PAGECOUNT
Formatting 1668
Parameter set column 221
Parameter set columns
export 82
import 82
Parameter set import column 225
Parameterization of imc CANSAS modules 116, 117
Parameterizing bridges 802
Parameters
Browse in workspace 1599
Channels (Scripting) 1377
Devices (Scripting) 1374
Experiment (Scripting) 1383
Passive
Trigger event 192
Password 74, 77
Minimum length 74
MVB-Bus 540
Path naming 603
P-component 786
P-Controller
imc Online FAMOS 785
PDF orientation (curve window) 1056
PDO
EtherCAT-IF 468
PDO add
EtherCAT-IF 467
PDO Create automatically
EtherCAT-IF 467
PDO delete
EtherCAT-IF 467
PDU FlexRay 496
Peakhold (curve window) 927
Peaks 797
Peaks (curve window) 874
Period comparison (curve window) 910, 1035, 1040
Periodic (curve window) 1040
Periodic calculations 1167
Persistent 1147
Persistent Loop Count 1122
Perspective
3D 1031
Phase offset correction
Synchronization 640
PI-Controller
imc Online FAMOS 785
Pictogram
Property - Widget 845
PID
J1587-Bus 512
options 513
PID 192 505
PID 194
Diagnostic channel 507
PID 448 505
PID query 502
PIDcontroller 1588
PID-Controller
imc Online FAMOS 785
Pin configuration
CAN 383
FlexRay 484
LIN 519
Pinning
Tool window 68
Place (map) 990

Placeholder
CONTROL 1657
DATABASE/DATABASES 1658
EXPERIMENT/EXPERIMENTS 1660
PAGE 1663
PANEL 1663
PROJECT/PROJECTS 1663
SELCONTROL 1665
SETUP 1665
SYSTEM 1666
VARS 1667
Placeholder Formatting
PAGE.NUMBER 1668
PANEL.PAGECOUNT 1668
SYSTEM.DATE 1668
SYSTEM.TIME 1669
VALUE 1669
Placeholders 1655
Placeholders for measurement folder structure
SETUP.SQL 614
STORAGE.FOLDERNAME 614
STORAGE.MEASUREMENT 614
VARS.VALUE 614
Places left of point (curve window) 927
Places right of point (curve window) 927
Platzhalter
MEASUREMENT 1661
Plug-in 42
Info 45
Version information 45
Points 172
polar plot 960
Poll 1531
Polygon needle
Property - Widget 845
Ports 302
Position
Property - Widget 845
Title 844
Positive slope 189
post checked loop 1121
Power 1441
Power fail 114
Power measurements 782, 783
Power Quality 1253
(Mean) Root mean square 1266, 1267
2 kHz - 9 kHz 1266, 1267
Active power 1266
Active power(mean value) 1266
adding functions 1257
Apparent power 1266
Apply 1257
Basics 1265
Check - Syntax 1257
configuring result quantities 1257
Context Menu 1265
Flicker 1266, 1267
Frequency 1266
Harmonic distortion 1266, 1267
Harmonics 1266, 1267
Higher frequency contents 1266, 1267
Interharmonics 1266, 1267
Opening the Editor 1254
Output parameters 1262
Performing calculations 1264
Reactive power 1266
Ribbon 1242
Root mean square 1266, 1267
Syntax check 1257
User interface 1254
Power1 1531
Power2 1532
Power3 1533
Powertrain Monitoring 1269
PPP 312, 321, 331
Connect 331
Disconnect 331
Instructions 334
PC as a PPP-client 328
Setting up 334
Preferred experiment template 55
Prerequisites
EtherCAT-IF 459
Prerequisites MVB-Bus 540
Prerequisites XCP 580
Presettings
Curve Window 1043
Presettings (curve window) 910
options 1053
Pretigger
Definition 186
Pretrigger 197
Print
Curve Window 859
Print Panel page 1624
Printer
Setup (curve window) 861
Printer preview 1059
printer settings 1044
Problems
Storage medium 626
Procedure 1159, 1167, 1173, 1179

Process configuration 90
Process vector 763
backup file 765
Process vector backup file 765
Process vector variable
As monitor channels 768
Create 756
Deactivating 763
Name 763
restore 765
Restoring 765
Update rate 767
Process_vector_variables 161
Product
Configuration 36
Edition 36
Info 45
Licensing 36
Version information 45
Product configuration 36
Changing 36
Product improvement 14
PROFIBUS
Baudrate 564
Highest Address 564
Loading the configuration 564
Monitor 564
Name 564
Transferring configuration 564
PROFIBUS Assistant 565
Channel settings 567
Connection 566
Menu 565
Message settings 566
PROFIBUS Configurator
GSD files 558
GSD folder 558
Installation 558
Main window 559
Master 560
Menu 559
Slave 560
PROFIBUS Interface 558
Project 54, 83
Delete 54
Export 54
Import 54
new 54
Project has changed 60
Project Menu
Import / Export 51
Manage 51
Manage Projects 51
New 51
New experiment 51
Open 51
Open experiment 51
Save 51
Save as 51
Save current Measurement Data 51
Save current Measurement Data as 51
Save experiment 51
Save experiment as 51
Save Project 51
PROJECT/PROJECTS 1663
Prompt to confirm discard 1611
Properties
Device 113
Properties of moldules 117
Properties of virtual channels
Comment 770
Delta-x 770
Delta-z 770
Name 770
x-offset 770
x-unit 770
y-unit 770
z-offset 770
z-unit 770
Protection against changing 1086
Protocol- and signal-channel name (MVB) 553
Protocol display (curve window) 931
Prozess vector variables 160
LIN 536
PSID 507
PTP
External clocks 635
Synchronization 648
PTP Master/Slave
Synchronization 644
Pulse spikes in the signal 797
pulse time 236
PulseDuration 1533
PulseFrequency 1534
PulsePhase 1534
Push button 1173
Push-button
Switching behavior 846
PushDeviceDriver 1424
pv.State.ExternalPower 763
pv.State.SyncTimeDeviviation 763
PWM mode (INC4) 237

Q

Quality Management 13
Querying the ECU-versions 504

R

Radio Clock 155
Radio modem 332
Radius
  Scale position 847
RAM Buffer 800
RAM buffer duration 592, 593
RAM buffer time 587
RAM size 88, 337
Range 1412
  Property - Widget 846
Range - Trigger-Event type
  Entering range 189
  Exiting range 189
  Inside of range 189
  Outside of range 189
Range & Scaling (Setup page) 165
RangeMax 1534
RangeMin 1535
Rank 1143
Reaction Time 1273
Read sensor information 203
Ready for measurement
  Interface configuration 308
ReadyForPowerOff 1535
Real time 1167
Realign to Grid 1085
RealTimeClock 630
Receiver
  GPS 767
Reconfigure 93
RecordEvent 1536
RecordText 1536
Red 1537
Reference Literature and Standards 483, 539, 580
Refresh rate
  Property - Widget 844
Register in Network 328
Released 192
ReplaceFirstValues0 1537
ReplaceFirstValuesN 1538
Report Channels
  Access via Scripting 1338
Report Generator 859
Report template
  Property - Pages 1082
Repository 1103, 1179
  Page 828
  Panel 828
  Panel page 828
  Widget 828
Reprocessing signals (curve window) 874
Request options
  J1587-Bus 514
ReSample 1538
Reset
  Set variables 1649
  Slave Pointer 935
Reset channel's sensor information 203
Reset Slave Pointer... 904
Resistor 167
Restart measurement automatically 661
Restore 1539
  Process vector variable 765
  Views 80
Restoring process vector variables 765
Resulting sampling rate 172
Resulting sampling time 172
Resume 1153
Resume data storage 611
Resume-Branch 1179
Rezoom (curve window) 1056
Ribbon 57
  Customize 57
Ribbon Monitor
  Activate saving 1235
  Deactivate saving 1235
  Device Search 1235
Rights
  Denied 79
  Disabled 79
  Enabled 79
  Full access 79
  Hidden 79
  Write protected 79
Ring (circular) buffer storage 587
Ring buffer 594, 800
Ring buffer memory 793, 794
RMS 781, 1539
RoaDyn 569
  all channels 571
  assistant 570
  CLK 570
  prerequisites 570
  TRG 570
Rosette1 1540
Rosette2 1541

rosettes (strain gauge) 128
Rotate
3D 1034
Round 1542
Router 301, 312
RPM 238
RS232 1401, 1409
RSFlipflop 1542
RTC 630
RTF 595
RunAutoBalance 1542
RunAutoShuntCalibration 1542
RxPDO
EtherCAT-IF 468

S

Safety note
CAN-Bus 384
Sample count 172
Sample delay 805
SamplesGate 1542
Sampling & Preprocessing (Setup page) 172
Sampling rate 172
Sampling time 172
Sampling time, large 795
Save
Experiment 55
Menu 51
Views 80
Save as
Menu 51
Save CAN configuration CBA 420
Save current measurement data 610
Menu 51
Options 611
Save current measurement data as 611
Menu 51
Options 611
Save dialog (curve window) 858
Save experiment
Menu 51
Save experiment as
Menu 51
Save measurement data
Saving subsequent to measurement 610
Targeted saving 610
Save Project
Menu 51
Save trigger events in individual files 604
Save/load view settings
additional columns 80
column descriptions 80
metadata columns 80
parameter set columns 80
table descriptions 80
window arrangements 80
Saving
activate 1235
Cluster size 608
Data types 608
deactivate 1235
to the internal drive 592
Saving interval 598, 599
Storage medium 598
Saving process vector variables 765
Saving subsequent to measurement 610
Sawtooth 1543
Scale angle
Property - Widget 847
Scale center
Range 846
Scale decimal places
Property - Widget 847
Scale position
Property - Widget 847
Scaled
Point 165
scaling 803
incremental encoder 232
lines 1023
x-axis 1006
Scaling with offset 804
ScanDisk
Storage medium 628
Script 1312, 1314, 1356
create 1312, 1314
execute 1356
run 1356
Script Editor 1312, 1314
Language 1312, 1314
Script name
Property - Pages 1081
Script options 1390
Script Scope 1312, 1314
Scripting 1309
abstract base class 1319
AbstractScript.cs 1403
Activity Scope 1315, 1365
Advanced code completion 1390
Axes 1342
Balance 1327
base class 1319

Scripting 1309
Breakpoint 1320
Changes 1355
Channel 1338
Channel calculation 1397
Channel parameters 1327, 1392
Channel properties 1327
Channel selection 1398
Channel state 1327
Channels 1338
Clipboard 1354
Commands via Scripting 1351
complex variable 1342
COM-Port 1409
Computer variable 1342
Connector 1327
Context menu 1310
Context Script 1314, 1365
Coupling 1327
Curve window 1342, 1398
Data Browser 1338
DataTable 1338
Debug 1320
Device actions 1323
Device parameters 1323
Device properties 1323
Device selection 1323
Device system variable 1342
Dialog response 1354
Diskstart 1323, 1374
Duration 1327
Edit 1317
Editor 1320
Error 1344, 1403
Error CS0029 1403
Error CS0234 1403
Error CS0525 1403
Error CS1729 1403
Event Script 1314, 1372
Execute 1317
Export 1318
FAQ 1403
Filtering 1327
get/set Device properties 1323
imc FAMOS functions 1346
imc FAMOS Kit 1319
imc FAMOS Math 1319
Import 1318
Import 4.0R1 1403
Inline FAMOS 1349
Input range 1327
IntelliSense 1390
Introduction 1312, 1314
Logbook 1344
Measurement storage area 1372
Menu 1310
Menu actions via Scripting 1351
Meta data 1327
Metadata 1392
Module type 1327
Offset 1327
Open 1317
Options 1315, 1390
Panel 1334
Panel Script 1312, 1314
Parallel-Istallation 1403
Parameter changed (Event) 1365
Parameters 1385
ParameterValues 1323, 1327
Preprocessing 1327
private Task (IFA) 1349
Provider 1319
Proxy class 1319, 1403
Range 1327
Regenerate proxies 1319
Restore 1355, 1390
RS232 1409
Run 1317
Run script 1644
Sample time 1327
Scale 1327
Scaling 1327
Script 1312, 1314, 1356
Script clipboard 1402
Script Editor 1317, 1320
Script Scope 1312, 1314
Script storage location 1390
Sensitivity 1327
Separate proxy class library 1315, 1354
SerialPort 1409
Setting values 1385
start/stop Measurement 1323
Status changed (Event) 1365
Storage scope 1315, 1316
Strain gauge 1327
Subversion 1390
SVN 1390
System clock 1342
Tables 1334
TED 1327
Third Party Device Script 1407
Tool window 1310
Tutorial 1391, 1392, 1397, 1398
Type Library Script 1314, 1369

Scripting 1309
Unit 1327
Value changed (Event) 1365
Variables 1338
Widget 1334
Working copy 1390
XY channel 1338
y-Axis 1342
Scroll mode (curve window) 1035
Curve window 1035
End online mode 1035
Scroll-mode 793
Search CAN channel, message or node 422
Search for devices 96
Search for devices by IP/DNS 96, 312
Security alert 41
Seed/Key 443
Seed-Wert 443
Segment (curve window) 1035
SELCONTROL 1665
Select device with explorer 621
Select mode (curve window) 881
Selected 140
Selected x-unit (curve window) 910
Semicolon 1444
Send channels with imc Online FAMOS 410
Send channels with process vector variables 411
Send Curve Segment to FAMOS!:context menu 870
Sending an Email 706
sending variables
LIN 536
SendMessage 1543
Sensitivity 165
Sensor Database
Importing sensor data 128
Sensor delay time (CAN message) 412
Sensor information
writing to channel 123
Sensor initialization 393
Sensors 122
export 82
Sensors (Menu)
<< (level higher) 124
>> (indentation) 124
Add to imc SENSORS 124
Create a new characteristic curve 124
Delete characteristic curve 124
Delete Filter 124
Edit characteristic curve 124
Filter list hidden 124
Find 124
Import Filter 124
Open imc SENSORS 124
Standard layout 124
Sensor-TEDS 201
Sequence table 1191
Sequencer 1189, 1202
Commands via Scripting 1351
Conditional branches 1216
create scripts 1312, 1314
Else 1219
Event handling 1197
For 1216
If 1219
imc FAMOS 1213
Output box 1204
Show dialog 1208
Show message box 1204
start/stop via Scripting 1351
Starting 1196
Switch-Case 1219
User-defined events 1199
While 1218
Sequencer stop
about commands 1596
sequencer: context menu 1193
sequencer: tool window Commands 1190
Sequences: Persistent Loop Count 1122
Serial Interface 1401
SerialPort 1409
Service 13
Set as default
Curve Window 1043
Set markers (curve window) 883
Set the root directory 828
Set values
Display 678
Set variables
Delete 1649
Reset 1649
SetPoint 786
Setting System Outputs 753
Setting up a PPP-device 334
Settings
FlexRay 487
Settings (curve window) 1053
SETUP 1665
Printer (curve window) 861
Setup data in Panel 1665
Setup data in Sequencer 1665
SETUP.SQL 1665
Shell extension 621, 801

Show EtherCAT-IF 468
Show Grid 1085
Property - Pages 1082
Show in curve/values window 1077
Show message box
Cancel 1602
Headline 1602
Interval (Timeout) 1602
Name 1602
OK 1602
Size 1602
Text 1602
Timeout-action 1602
Voice 1602
Show most recently use values 220
Show Panel page as dialog 1620
Show placeholder 220
Show Tabs 1086
Show x=0 (trigger) 910
Showing
Tool window 68
Shunt calibration 180
Shunt calibration by Display 681
SID / PSID 507
Signal < Level 189
Signal = 0 190
Signal = 1 190
Signal > Level 189
Signal change: 0 -> 1 190
Signal change: 1 -> 0 190
signal lcp open 325
Signals (MVB) 555
Signals FlexRay
Properties 493
SimplePollDevice 1429
add channel 1435
add Display variable 1437
Extend 1435, 1437
SimplePushDevice 1432
Sin 1544
single signal counter 234
Single value 751
single value (curve window)
window 930
Single-phase power measurement 783
single-signal 234
SingleValueChannel 1544
Size
Show message box 1602
Size automatic (curve window) 927
Size of measurement point (curve window) 1056
skb files 443
Skin
Property - Pages 1082
SkipFirstValues 1547
Slave Pointer
Reset 935
Sleep (LIN) 537
SlopeClip 1547
Small Ticks (curve window) 1006
Smo3 1547
Smo5 1548
smoothing (curve window) 876
Snap to Grid 1085
Property - Pages 1082
so files 443
Software
Installation 21
Uninstall 21
Update 21
Solid
Background color 1082
sound output (curve window) 1053, 1064
cut sound 1067
directly 1068
Toolbar 1064
volume 1064
SoundPressureLevel 1548
space curve 1027
Special characters 755
Specific Values 1287
SpecThirds 1549
Spectra 1287
Speed 237
SPI Assistant 575
16-Bit Word Structure 576
Chipselect 579
error handling 579
Master-Send-List 576
Menu 575
SPI-Bus Interface 574
Hardware prerequisites 575
Software prerequisites 575
Spikes 797
Splash screen 39
Spline (curve window) 876
Spread angle
Scale angle 847
Sqrt 1550
Standard display (curve window) 910
Standard Format
CAN 398

Standard time 630
Standard-Gateway 328
Standardization of intervals 600
Start angle
  Scale angle 847
Start button 185
Start by Trigger 192
start delay (sound) 1068
Start delay [ms] (curve window) 1053
start edge (incremental encoder) 235
Start measurement
  Automatically upon starting imc STUDIO 43
  Open Experiment/Start measurement 1615
Start option 99, 154
  At defined time 155
  Automatic timed start 155
  Immediately 154
Start page 39
Start Sequencer
  Automatically upon starting imc STUDIO 43
Starting imc Online FAMOS 747
Starting imc STUDIO 39
StartTimerPeriodic 1582
StartTimerSingle 1583
Startup behavior 76
State 1119, 1159, 1167, 1173, 1179
State description 1159, 1167, 1173, 1179
State of the external power 763
State: Break Condition 1119
State: Exception & Error Events 1119
State: Initial Action 1119
State: Local Limits 1119
State: Time tracks 1156
State: User Timeouts 1139
States: If-instruction 1123
Status 163
  Property - Widget 844
STD 630
StDev 1550
Step 1159, 1167, 1173, 1179
Stop by Trigger 192
stop edge (incremental encoder) 235
Stop Sequencer
  Command 1596
StopTimer 1583
Storage 1323
  System requirements 20, 1680
Storage interval
  Histogram 175
Storage location (PC) 594
Storage medium
  Age 628
  Button 618
  CF card 619
  CFast 620
  Compact Flash 619
  Data transfer 620
  Defective sectors 628
  ExpressCard 620
  FAT16/FAT32 624
  File system 624
  Filesize (maximum) 624
  Hot-Plug 618
  Issues 628
  LED 618
  Limitations 628
  Manufacturer 628
  Memory volumes 619
  Number of folders (max.) 628
  Possible problems 626
  Problems 626
  ScanDisk 628
  Slot 618
  USB 619
Storage options 587
Storage type 220
Strain Gauge (Setup page) 167
strain gauge rosettes 128
Stretch
  Background image 1082
Stretch mode
  Curve window 1035
STri 1551
Structure of the trigger machine 185
Subnet 306
Subnet-mask 301
Subtraction 1440
Sum 1551
Sum2 1552
Supplemental files
  Characteristic curves 105
  Filter data 105
  imc Online FAMOS source code 105
  Messaging configurations 105
  Synthesizer directory structures 105
Supply 164
Surface
  3D 1027
Suspend data storage 611
Swapping the storage medium 617
Switch 1580

Switch 1580
Switching behavior 846
Switch - Case 1596
Switching behavior
Property - Widget 846
synchron (sound) 1068
Synchronisation
get/set via Scripting 1374
Synchronism 805
synchronity re-establish - automatical 661
Synchronization 629, 630, 805
CAN-1 Protocol 401
CAN-Bus 401
Constraints 663
Control 659
DCF77 646
Field-bus channels 339
Frequency error 640
GPS 645
GPS + DCF Master/Slave 642
GPS Master/Slave 642
imc CANSAS 401
imc CRONOS PC 663
IRIG-B 646
Master/Slave setup 640
NTP 647
NTP Master/Slave 643
Phase offset correction 640
Precision 663
PTP 648
PTP Master/Slave 644
Seperate device 644
Synchronization with other devices
No external timer 641
synchronize devices FAQ 655
Synchronized recording 413
Synchronous start 99, 157, 630
Synchronous Task 1140
Synchronous Tasks 746
SyncOverload 1553
SyncRTC 630
Internal clocks 632
Syntax
UDP Status Monitoring 700
UDPNoise 700
Syntax-Check 750
syslog 605
SYSTEM 1666
System prerequisites and limitations
imc Inline FAMOS 1247
Monitor 1235
System requirements
Hard drive 20, 1680
imc Online FAMOS Professional 741
Minimum requirements for the PC 1680
Operating System 20, 1680
Recommended configuration for the PC 1680
Storage 20, 1680
Windows 20, 1680
SYSTEM.DATE
Formatting 1668
SYSTEM.TIME
Formatting 1669

T

Table (curve window) 910
Table description 228
Table descriptions
export 82
import 82
save 80
Table display (curve window) 931
Tables (curve window) 931
Tan 1553
Tare by Display 679
Targeted saving 610
Targets
Trigger 192
Taring 178
Task 1159, 1167, 1173, 1179
Task Editor 1140
Task Management 1140
TCP/IP 307
with PPP 321
TCP/IP, PPP via a router 312
Technical Data Sheet
Automation 1688
Bus Decoder 1694
Data Processing 1691
imc Inline FAMOS 1692
imc Online FAMOS 1703
imc Online FAMOS Class-Counting 1707
imc Online FAMOS Order-Tracking Analysis 1708
imc Online FAMOS Professional 1706
imc STUDIO 1678
Monitor 1701
Panel 1685
Power Quality 1693
Powertrain Monitoring 1694
ProjectManagement 1695
Scripting 1689
Sequencer 1687

Technical Data Sheet
Setup 1681
Third Party Device Interface 1690
Video 1696
Technical Specifications
imc Messaging 704
TEDS 201
Read sensor information 203
Reset channel's sensor information 203
supported sensors 204
Telephone numbers 13
Terminate 1151
Terminate condition 1159, 1167, 1173, 1179
Terminator
CAN 384
Terminator in device
CAN 398
Testing the configuration
LIN-Bus 526
Text
Axes 1016
Show message box 1602
Text display (curve window) 931
Text format
Property - Widget 846
Text input for report channel 1074
Text list 220
Text Off/On
Property - Widget 845
TextAdd 1554
TextFormatE 1555
TextFormatF 1556
TextFormatH 1557
TextFormatI 1558
The device has been shut down inaccurate in the past 93
The measurement process
Trigger 199
Third Party Device 1424
add Channel 1410
AudioDevice 1415
ChannelLoader 1416
Data transfer 1413
FunctionSimulator 1425
Logbook 1413
SimplePollDevice 1429
SimplePushDevice 1432
Simulator 1424
virtual 1424
Third Party Device Script 1374
create Script 1407
Third Party Devices
add Device 1410
eRange 1412
Range 1412
Third Party Device Interface 1407
Third/ octave labeling (curve window) 1012
third-octave axes
display options 965
Three-phase power measurement 785
Tick distance
Property - Widget 847
Ticks (curve window)
small 1006
Tile
Background image 1082
Time Base
imc Online FAMOS 755
Time Base for Channels in a Formula 755
Time counter
GPS 767
Time measurement 230, 235
Time offset 155
Time resolution between Field-bus and analog channels 339
Time Stamp ASCII (curve window) 931, 932
time stamped 1411
Time to suspend 405
Time tracks 1156
Time zone 630
Time zones 649
Timed start 155, 630
FAQ 806
Timeout-action
Show message box 1602
Timer function 1159, 1179
Timer functions 1582
Timerfunction 1173
Time-Shift in the ccv 1053
Timestamp - date time (measurement number)
Path naming 603
Timestamping
Field-bus 342
Timestart 154
Start option 154
Synchronous start 157
Tip
Avoid large sampling times 795
Title 213
Property - Pages 1081
Property - Widget 844
Tolerance 1558

Tool Window 42, 67
Devices/Channels 1236
Docking 69
Freely positioning 69
Hiding 68
Operation 68
Pinning 68
Showing 68
Tool window chooser 68
Tool windows 57
Toolbar
Sound output (curve window) 1064
Toolbar (curve window) 1060
Traffic light 1173
Transfer experiment to other devices 255
Transfer Options
Transfer to imc FAMOS 1057
Transfer rate
CAN 385
Transfer to imc FAMOS 1056
from curve window 859
Transfer Options 1057
Transferring curves
using clipboard 902
TransRec 1558
Transverse strain coeff. ν 167
Tree structure
Storage on the PC 595
triaxial accelerometers 128
Trigger 806
Definition 186
Event-controlled digital outputs 201
Multiple triggering 195
Multiple-triggered signals 795
Virtual channels from imc Online FAMOS 198
Trigger events 187
Trigger machine 185
Trigger note with LED 745
Trigger_48 185
Triggered
Foreground Measurements 200
Troubleshooting 807, 808
TSA-Kit 932
Twin Window (curve window) 863
Two point calibration 181
two signal encoder 234
Two-phase power measurement (ARON) 784
Two-Point controller 1588
two-point scaling calculated 804
two-point scaling via amplifier adjustment dialog 805
Two-position controller 789
two-signal 234
TxPDO
EtherCAT-IF 468
Type 1147
Type Library Script 1369
create 1312, 1314

U

UAC 25
UDP Status Monitoring 700
Format date/time 698
Format of numerical variables 697
Syntax 696, 700
Variables 695
UDP Status Monitoring test 701
UDP Status Monitoring.exe 709
UDPNoise 700
Format date/time 698
Format of numerical variables 697
Syntax 696, 700
Variables 695
UDPNoise test 701
UDS
Dynamic lists 456
UMC 595
ume.zip (WebServer) 726
UMI 595
UMU 595
Unable to find device 302
uncertainty (display) 1023
Unchanged
Background image 1082
Undo (curve window) 874, 1053
UnEqual 1563
Uninstall 21
Unit 165, 167
Property - Widget 844
Untriggered
Background Measurements 200
Untriggered Measurements 185
Unzoom (curve window) 866
Update
Backup Application settings (imc STUDIO 3.x) 23
Backup View settings 23
Database 23
Restore View settings 23
Update rate
of process vector variables 767
Updating
Curve window 792
Upper 1563

Upper Limit 1143
UPS
Acquisition time 114
USB storage medium 619
Use DHCP 306
User
adding 74
deleting 74
log off 77
log on 77
removeing 74
switch 77
User account control 25
User administration 43, 73, 74
activating 74
deactivating 74
log on 77
Startup behavior 76
User access rights 79
User group 73, 74, 79
imc Administrators 73, 74
imc Advanced Users 73, 74
imc Developers 73, 74
imc Standard Users 73, 74
User name
Minimum length 74
User Timeouts 1139
User type
Active Directory domain accounts or groups 74
imc STUDIO internal users 74
Local computer accounts or groups 74
User-defined
Title - Default Source 844
User-defined characteristic curve 122
creating 125
User-defined measurement folder structure 612
User-defined measurement storage area
FAQ 616
User-defined measurement storage path 612
User-defined variables 1167, 1671
Access via Scripting 1338
User-defined variables (Automation)
Default Value 1147
Name 1147
Persistent 1147
Type 1147
User-defined: Installation type 31
UTC 630

V

VAL_BLK 456
VALUE
Formatting 1669
Value display 214
Value representation
Property - Widget 845
Variable
Property - Widget 844
Variable display format 214
Variables
Access via Scripting 1338, 1358
imc Messaging 695
UDP Status Monitoring 695
UDPNoise 695
VARS 1667
VectorChannel 1564
VectorChannelSet 1564
VectorFromFile 1564
VectorizeAndSkip 1565
VectorizeOverlapped 1565
VectorStatic 1566
Version
Get version with GetSoftwareVersion 717
Version information 45
VibrationFilter 1567
Video 1292
Configuration 1292
Export 1301
Import 1301
Load 1301
Playback 1296
Recording 1296
Trigger 1295
Video camera
Search 1292
Search for Devices 1292
Video Compression 1303
Video Preview 1292
Video settings
Changing by means of the Panel page 1296
Video-PC
VNC connection to the video-PC 1307
Windows remote connection to the video-PC 1307
Video-Technical data
Compression 1701
Frame grabber 1701
Notes 1301, 1696
Specifications 1301, 1696
Supported Cameras 1698
Video-Widget
On Screen Display (OSD) 1296
View menu

View menu
Choose Tool Windows 57
Customize Ribbon Menu 57
Delete View 57
Load 57
Reset Window Arrangement 57
Restore 57
Ribbon 57
Save View 57
Save View as 57
User-defined buttons 57
View settings export/import 82
Viewing saved measurement data 616
Views 57
Delete 80, 81
Load 80, 81
Restore 80, 81
Save 80, 81
Save as 81
Views, metacolumns, sensors, ... 82
Virtual bits 160
Trigger target 192
Virtual channels 160, 751
Properties 770
Trigger 781
trigger machine 198
Virtual channels and the trigger machine 781
virtual device 1424
Virtual device clock
Minimum accuracy [ms] 65
Virtual_bits 161
Virus scanner 46, 793
VIsAnyGreater 1569
Visible 213
Property - Widget 845
Visible in printout
Property - Widget 844
Visual comparison of data 195
VMax 1569
VMaxV 1569
VMean 1569
VMeanV 1570
VMin 1570
VMinV 1570
Voice
Show message box 1602
volume sound output 1064
VRedV 1570
VRMS 1571
VSum 1571
VValueAtXValue 1571
VXValueOfMax 1571
VXValueOfMin 1571
VXValueWithYValue 1571

W

Wait
Command 1597
Wake on CAN 405
WakeOnLAN 702
WakeUp (LIN) 537
Warning 69
Warning in case of infinite measurement duration
Open Experiment/Start measurement 1615
Warranty 13
Waterfall diagram (3D) 910
display 921
Waterfalls 910
WaveCom Fastrack 333
Waveform Preselection
More Channels in Curve Window 1001
Waveforms in axes list
More Channels in Curve Window 1002
While 1121, 1581
While-Loop 1159
Widget
Access via Scripting 1334, 1358
Repository 828
Video 1296
Widgets 817, 1074
Aligning 840
Arrangement 840
Cascade 841
Dock 842
Grouping 843
insert 834
new 834
Realign to Grid 1085
Snap to Grid 1085
Width
Property - Pages 1081
window arrangements
export 82
import 82
Load 80
Save 80
Window size
Panel 1083
Windows
Firewall 41
Security Alert 41
Shell extension 621

Windows
System requirements 20, 1680
User account control 25
Windows 8 / 8.1 and adhoc con. 315
Windows Video File (AVI) 1301
WindRoseCorr 1572
Wiring 164
WLAN 313
Connection setup possibilities 314
PC via Access Point - Device via Access Point 314
PC via Access Point - Device via integrated WLAN 314
PC via integrated WLAN - Device via integrated WLAN 314
WLAN and Windows 8 / 8.1 315
Write protected
Property - Widget 844
Write-protected 213
Writing sensor information 204
Writing sensor information to channel 123

X

X-axis 173
automatic 1006
changes 1056
fixed range 1006
lin/log 1006
markings 1006
rounding 1006
scaling 1006
Scaling the Axes 1005
XCP 455
XCP on Ethernet 580
XCP on UDP/IP 580
XCP onTCP/IP 580
XCPoE
A2L-file creating 582
A2L-file loading 584
Channel settings 586
Definitions 580
ECU-settings 585
Master 583
Master assistant 583
Node settings 585
on Ethernet 580
on UDP/IP 580
onTCP/IP 580
Prerequisites 580
Slave 581
triggert channels 581
Variables 581
XCPoE Assistant
Slave 582
X-link with curve windows 905
Xml
EtherCAT-IF 469
XmlRpcCmd 707
x-offset 770
x-unit 770
XY data
complex data 972
different sampling rates 972
Lissajous figures 972
measurement cursors 972
polar plots 972

Y

Y-adapter
CAN-Bus 383
Y-axes stacked 910
Y-axis
color 1017
line thickness 1017
small ticks 1017
Y-connection 785
y-unit 770

Z

Z-axis 921, 959
zeitgestempelt 1411
zero pulse 234
z-offset 770
Zone representation
Property - Widget 846
Zones
Property - Widget 846
Zoom
Background image 1082
Zoom (curve window) 865, 1056
Zooming Panel pages 1085
z-unit 770