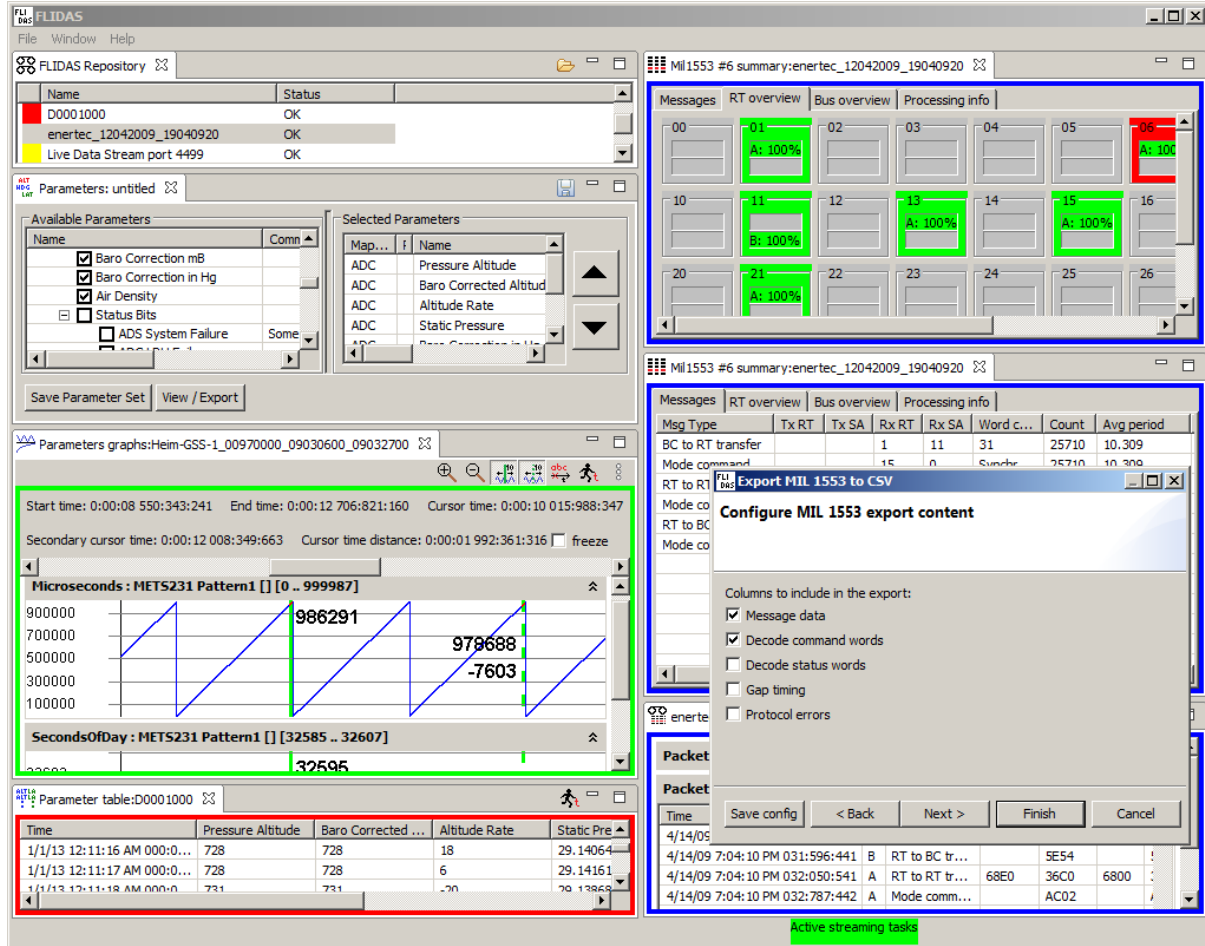


FLIDAS - Flight Data Studio

User Manual



Data Bus Tools GmbH
Rosenheimer Landstrasse 139
85521 Rierning
Germany

Tel: +49 89 62 70 79 61
Fax: +49 89 88 90 95 15
Email: info@databustools.de
Support-
email: support@databustools.de
Website: www.databustools.de

00101101000101
10100100111007
1001011001007

**Data
Bus
Tools**

Table of Contents

1	FLIDAS Overview	15
1.1	Outlook	15
1.2	How to Use This Manual	15
1.3	System Requirements.....	15
1.4	Limitations / Features out of Scope	16
1.4.1	Supported IRIG 106 Specification Version	16
1.4.2	Data Streaming Performance and Reliability	17
1.5	Versions	17
1.5.1	Feature Details	19
2	Getting Started	26
2.1	Installing FLIDAS	26
2.1.1	Associating CH10 Files with FLIDAS	27
2.2	CH10 Tutorial	27
2.3	Basic Tutorial.....	27
2.4	Parameter Tutorial	31
2.4.1	Creating an ICD	32
2.4.2	Creating an ICD Mapping.....	36
2.4.3	Create Parameter Sets	37
2.4.4	Instrument Panels.....	40
2.4.5	Added flexibility by ICD mapping files	42
2.4.6	How to continue learning.....	43
2.5	Data Streaming Tutorial.....	43
2.5.1	Receive a Live Data Stream to Verify Proper Reception	44
2.5.2	Record a Data Stream	45
2.5.3	See Live Statistics from a Live Data Stream	47
2.6	Features by Application Area	48
2.6.1	CH10 File-Based Features	48
2.6.1.1	Features for CH10 File Debugging	48
2.6.1.2	Protocol Level Analysis.....	49
2.6.1.3	Parameter Level Analysis.....	50
2.6.1.4	Data Presentation.....	50
2.6.1.5	Test Data Generation	50
2.6.1.6	Working with Distributed CH10 Multiplexer Sources.....	51
2.6.2	Live Data Stream-Based Features.....	51
2.6.2.1	Features for CH10 Stream Debugging	51
2.6.2.2	Protocol Level Analysis.....	51

2.6.2.3	Parameter Level Analysis	52
2.6.2.4	Data Presentation	52
2.6.2.5	Data Publishing.....	52
2.6.2.6	Working with Distributed CH10 Multiplexer Sources	53
3	Concepts.....	54
3.1	CH10 Sources	54
3.1.1	CH10 Files	54
3.1.2	Live Data Streams.....	55
3.2	Packets	56
3.3	Packet Entries.....	56
3.4	Channels and Subchannels.....	56
3.5	CH10 Data Types.....	57
3.6	TMATS Attributes	57
3.7	Recording Events	58
3.8	Time Markers.....	58
3.9	Exporters	59
3.10	Packet Entry Viewers.....	60
3.11	Protocol Viewers.....	60
3.12	CH10 Data Type Filters	60
3.13	Preferences	60
3.14	Properties	61
3.15	Views	61
3.16	Wizards	62
3.17	Error Logging.....	62
3.18	ICDs	62
3.19	Parameters.....	63
3.20	Parameter Sets.....	63
3.21	ICD Mapping	64
3.22	Parameter Viewers	65
3.23	Parameter Sampling	65
3.23.1	Sampling Modes	65
3.23.2	Initialization Values	69
3.24	Data Streaming.....	70
3.24.1	Data Stream Tasks	70
3.24.2	Real-time and Reliability	71
3.24.3	IRIG 106 CH10 Data Streaming Format Support.....	71
3.25	Live Processors	72

3.26	Parameter Live Processors	72
3.27	Instrument Panels	73
3.28	Distributed CH10 Multiplexer Sources	73
3.28.1	Currently Supported Distributed Architectures	74
3.28.2	Tips and Tricks	75
3.28.3	Ongoing Discussions in IRIG 106 Standardization Committees.....	75
4	Reference	77
4.1	Help System	77
4.2	Views	77
4.2.1	Repository View	77
4.2.1.1	Active Data Stream Tasks	79
4.2.2	Packet View	80
4.2.3	Packet Entry View	81
4.2.4	Channel View.....	83
4.2.5	TMATS View	85
4.2.6	Recording Event Definition View	85
4.2.7	Hex Packet View.....	86
4.2.7.1	Data Manipulation	88
4.2.8	Parameter Plug-in Control View.....	89
4.2.8.1	ICD Mapping Management	90
4.2.8.2	Instrument Panel Management	90
4.2.8.3	Parameter Set Management	91
4.2.8.4	Parameter Details.....	91
4.2.9	ICD Mapping View.....	92
4.2.9.1	Channel / Subchannel ICD Mapping.....	92
4.2.9.2	Associated Instrument Panels	93
4.2.9.3	Associated Parameter Sets	94
4.2.10	Parameter Set View.....	94
4.2.11	Parameter List View	96
4.2.11.1	Filtering	97
4.2.12	Instrument Panel View.....	98
4.2.12.1	Edit Mode	100
4.2.12.1.1	Parameter assignment.....	100
4.2.12.2	Further notes	101
4.2.13	CH10 Verification View	101
4.2.13.1	General Error Indications	102
4.2.13.2	Results from Error Check	102

4.2.13.3	Further Problem Analysis and Solution Prototyping.....	103
4.2.13.4	Checks Executed.....	103
4.2.13.5	Disclaimer.....	104
4.3	Packet Entry Viewers.....	104
4.3.1	Analog Packet Entry Viewer	104
4.3.2	PCM Packet Entry Viewer.....	105
4.3.3	UART Packet Entry Viewer	106
4.3.4	TSPI / CTS Format 0 and 2 Packet Entry Viewer	107
4.3.5	TSPI / CTS Format 1 Packet Entry Viewer	107
4.4	Preferences.....	107
4.4.1	General Preferences	107
4.4.2	Packet Entry Viewers Preferences	109
4.4.3	Default Directories and Files Preferences.....	109
4.4.4	User Map Preferences	110
4.4.5	Subchannel Number Preferences	111
4.4.6	TMATS Default Preferences	111
4.4.7	Data Streaming Preferences	112
4.4.8	User Defined Time Representations Preferences	115
4.4.9	Predefined Time Channel Offsets.....	116
4.4.10	Parameter Plug-in Preferences	117
4.4.11	GUI Scaling (Troubleshooting) Preferences.....	118
4.5	Properties	118
4.5.1	General CH10 Source Information Properties.....	118
4.5.2	General File Information Properties	120
4.5.3	General Live Data Stream Information Properties	121
4.5.4	General TMATS Attributes Properties	121
4.5.5	CH10 Data Type Filter Properties	122
4.5.6	Time Marker Properties	122
4.5.7	General Channel and Subchannel Properties	124
4.5.8	General Packet Properties	125
4.6	CH10 Data Type Support.....	128
4.6.1	ARINC 429	128
4.6.2	MIL-STD 1553	128
4.6.2.1	MIL-STD 1553 16PP194.....	129
4.6.3	Ethernet.....	129
4.6.4	ARINC 664 (AFDX) / UDP Payload.....	130
4.6.5	MPEG Video.....	130

4.6.6	UART.....	131
4.6.7	Analog.....	132
4.6.8	Discrete	133
4.6.9	PCM.....	134
4.6.10	IRIG 106 Chapter 7 PCM	136
4.6.11	CAN Bus.....	137
4.6.11.1	DATARec4 CAN Bus Support.....	137
4.6.12	Fibre Channel FS Layer	137
4.6.13	Fibre Channel PH Layer	138
4.6.14	Message Data	138
4.6.15	TSPI / CTS (GPS NMEA-RTCM)	138
4.6.16	TSPI / CTS (EAG ACMI).....	139
4.6.17	TSPI / CTS (ACTTS).....	139
4.6.18	Time	140
4.6.19	TMATS	141
4.6.20	Recording Events	142
4.6.21	Recording Indices	142
4.6.22	User Defined Computer-Generated Data.....	142
4.6.23	Partly Supported CH10 Data Types.....	143
4.6.23.1	Image Data.....	143
4.6.23.2	Still Imagery	143
4.6.23.3	Dynamic Imagery	143
4.6.23.4	IEEE 1394 Physical Layer.....	143
4.6.23.5	IEEE 1394 Transaction.....	144
4.6.23.6	Parallel	144
4.6.23.7	MJPEG.....	144
4.6.23.8	MJPEG 2000.....	144
4.6.23.9	PCM DQE.....	144
4.7	Wizards	145
4.7.1	Channel Selection	145
4.7.2	Subchannel Selection	146
4.7.3	Exporter Target File Selection	146
4.7.4	Exporter Target Directory Selection	146
4.7.5	Time Slice Selection	147
4.7.6	Time Representation	148
4.7.7	CH10 File Selection.....	150
4.7.8	Live Data Stream Selection	150

4.7.9	Sampling Mode Selection	150
4.7.10	CSV Properties.....	151
4.7.11	CH10 Data Type Filters	151
4.7.12	UDP Properties	151
4.7.13	CH10 Data Stream Format.....	152
4.7.14	MATLAB File Properties	152
4.7.15	Pseudo Random Bit Stream Type Selection	153
4.7.16	IENA Properties.....	153
4.7.17	iNET-X Properties	154
4.7.18	IRIG 106 Chapter 7 Properties	155
4.7.19	Data Representation	156
4.7.20	Time Channel Offsets	156
4.7.21	WAV File Properties.....	157
4.7.22	Safran CH10 File Segmentation	157
4.8	Exporters	158
4.8.1	TMATS to Text File	159
4.8.2	Filtered CH10.....	159
4.8.3	MSID Channel Splitter	159
4.8.4	Replace TMATS in CH10	160
4.8.5	MPEG Video to MPEG File	161
4.8.6	Replay Audio from Analog.....	161
4.8.7	Replay Audio from CVSD.....	161
4.8.8	MPEG TS Replay (UDP Streaming)	161
4.8.9	Analog to WAV	162
4.8.10	Analog to MAT.....	163
4.8.11	Ethernet to PCAP.....	163
4.8.12	PCM Throughput Payload	164
4.8.13	Payload to File	164
4.8.14	General CSV Exporting	165
4.8.14.1	ARINC 429 to CSV	165
4.8.14.2	MIL-STD 1553 to CSV	165
4.8.14.3	PCM to CSV.....	167
4.8.14.4	CAN Bus to CSV.....	168
4.8.14.5	Analog to CSV.....	168
4.8.14.6	Discrete to CSV	169
4.8.14.7	UART to CSV	170
4.8.14.8	NMEA 0183 to CSV.....	170

4.8.14.9	Message Data to CSV	171
4.8.14.10	Fibre Channel to CSV	171
4.8.14.11	Sampled parameters to CSV	172
4.8.14.12	ARINC 429 Message-Based Parameters to CSV	172
4.8.14.13	IENA to CSV	173
4.8.14.14	iNET-X to CSV	174
4.8.14.15	TSPI / CTS to CSV	175
4.8.15	Sampled Parameters to MAT	175
4.8.15.1.1	Exporting to Multiple MAT Files	176
4.8.16	CVSD to WAV	176
4.8.17	PCM Numeric Parameter to WAV	177
4.8.18	Byte Array Parameter to Binary File	177
4.8.19	IRIG 106 Chapter 7 Exporting	177
4.8.19.1	IRIG 106 Chapter 7 to CH10	177
4.8.19.2	IRIG 106 Chapter 7 to Ethernet PCAP	178
4.8.19.3	IRIG 106 Chapter 7 to IP PCAP	178
4.8.20	Further Exporters	178
4.9	Protocol Viewers	178
4.9.1	Communication Summaries	178
4.9.1.1	Messages Page	179
4.9.1.2	Special Events Page	180
4.9.1.3	Processing Info Page	181
4.9.1.4	Other Pages	181
4.9.1.5	ARINC 429 Communication Summary	181
4.9.1.6	Ethernet Communication Summary	183
4.9.1.7	MIL-STD 1553 Communication Summary	185
4.9.1.8	UART Communication Summary	188
4.9.1.9	MPEG Transport Stream Summary	189
4.9.1.10	CAN Bus Communication Summary	190
4.9.1.11	PCM Communication Summary	191
4.9.1.12	Message Data Communication Summary	192
4.9.1.13	Time Communication Summary	193
4.9.1.14	Recording Event Communication Summary	193
4.9.1.15	IENA Communication Summary	194
4.9.1.16	iNET-X Communication Summary	195
4.9.1.17	CH7 Communication Summary	197
4.9.2	NMEA 0183 Protocol Viewer	199

4.9.3	Navigation Data View.....	201
4.9.3.1	The Sequence of Navigation Data	202
4.9.3.2	The Map.....	203
4.9.3.3	Instrument Panel	203
4.9.3.4	Control Panel	203
4.9.4	PCM BERT Viewer	204
4.9.5	UART Text Protocol Viewer	205
4.9.6	Recording Gap Viewer	206
4.9.6.1	Gap Detection Rules	207
4.10	Parameter Viewers	208
4.10.1.1	Performance.....	208
4.10.2	Parameter Table Viewer	209
4.10.3	Parameter Graph Viewer.....	209
4.10.3.1	Stripes	210
4.10.3.2	Curve Marks	211
4.10.3.3	Time Information.....	211
4.10.3.4	The Cursors.....	212
4.10.3.5	Zoom.....	212
4.10.3.6	Time Markers	213
4.10.3.7	Printing	213
4.10.3.8	Performance.....	213
4.11	ICD Formats.....	214
4.11.1	CSV ICD Formats.....	214
4.11.1.1	ARINC 429 Simple CSV ICD Format	218
4.11.1.2	MIL-STD 1553 Simple CSV ICD Format	220
4.11.1.3	PCM Simple CSV ICD Format	222
4.11.1.4	Analog Simple CSV ICD Format	224
4.11.1.4.1	Full Live Decoding Parameters.....	226
4.11.1.5	Discrete Simple CSV ICD Format	226
4.11.1.6	CAN Bus Simple CSV ICD Format	227
4.11.1.7	Ethernet UDP Simple CSV ICD Format	228
4.11.1.8	Recording Events Simple CSV ICD Format	231
4.11.1.9	UART ASCII Simple CSV ICD Format	231
4.11.1.10	UART Message-Aligned Binary Simple CSV ICD Format	234
4.11.1.10.1	Limitations	236
4.11.1.11	Message Data Binary Simple CSV ICD Format	236

4.11.1.12	User Defined Computer-Generated Data Binary Simple CSV ICD Format	238
4.11.2	NMEA 0183 on UART or TSPI Hardcoded ICD Format	240
4.11.3	PCM TMATS ICD Format	242
4.11.3.1	Additional Notes on Features	243
4.11.3.1.1	D-Group Features	243
4.11.3.1.1.1	Bit Masks.....	243
4.11.3.1.1.2	Measurement Locations	243
4.11.3.1.1.3	Word and Frame Intervals.....	243
4.11.3.1.1.4	Fragmentation and Transfer Orders	243
4.11.3.1.1.5	Parity	244
4.11.3.1.1.6	Simultaneous Sampling.....	244
4.11.3.1.2	C-Group Features	244
4.11.3.1.2.1	Binary and Floating Point Formats	244
4.11.3.1.2.2	Data Conversion Types	245
4.11.3.1.2.3	PCM Time	245
4.11.3.1.2.4	Pair Sets.....	245
4.11.3.1.2.5	Discretes (Enumerations)	246
4.11.3.1.2.6	Digital Voice	246
4.11.3.1.2.7	Comments.....	246
4.12	CH10 File Loading	246
4.13	Selecting Time Markers or Recording Events	248
4.14	Default Table Actions	248
4.14.1	Saving Individual Column Setups	248
4.14.2	Table Cell Copy	248
4.14.3	Table Exporting	248
4.14.4	Table Font Switching	249
4.14.5	Expanding / Collapsing a Tree.....	249
4.15	CH10 Packet Body Exporting.....	249
4.16	TMATS Overriding	249
4.17	Gauges.....	250
4.17.1	Clock.....	250
4.17.2	Replay Control	250
4.17.3	Simple Value.....	251
4.17.4	Table	251
4.17.5	Bar Graph.....	252
4.17.6	Dial.....	252

4.17.7	Compass Rose.....	253
4.17.8	Attitude Indicator.....	253
4.17.9	Altimeter	254
4.17.10	Airspeed Indicator.....	254
4.17.11	Map.....	255
4.17.12	Strip Chart	256
4.18	Maps.....	257
4.19	Data Streaming.....	257
4.19.1	Create a Live Data Stream	257
4.19.2	Record a Data Stream	259
4.19.3	Publish a Data Stream.....	260
4.20	Analyzing Data Stream Problems.....	261
4.20.1	Data Flow.....	261
4.20.2	Analysis: Try with Minimum Setup	262
4.20.3	Analysis: What to Check	263
4.21	Live Processors	264
4.21.1	Channel Activity Live Processor	264
4.21.2	MPEG TS Replay Live Processor (UDP Streaming)	265
4.21.3	UART Replay Live Processor (UDP Streaming)	266
4.21.4	PCM Throughput Replay Live Processor (UDP Streaming).....	266
4.21.5	PCM Streams Live Processor	266
4.21.6	PCM BERT Live Processor.....	267
4.21.7	CH10 from CH7 Publishing.....	267
4.21.8	TmNS from CH7 Transmission.....	267
4.21.9	CH7 Live Processor	267
4.21.10	Analog Audio Live Replay	268
4.21.11	Raw Data Live Processors.....	269
4.21.11.1	The Raw Data Area	269
4.21.11.2	The Graph Area.....	269
4.21.11.3	PCM Raw Data Live Processor.....	270
4.21.11.3.1	The Minor Frame Area.....	270
4.21.11.3.2	The PCM Word Filter Configuration	271
4.21.11.4	MIL-STD 1553 Raw Data Live Processor.....	271
4.21.11.4.1	The MIL-STD 1553 Data Area	271
4.21.11.4.1.1	RT Overview	271
4.21.11.4.1.2	Subaddresses	272
4.21.11.4.1.3	Mode Codes.....	272

4.21.11.4.2	The MIL-STD 1553 Word Filter Configuration	272
4.21.11.4.3	The Graph Area	273
4.21.11.5	ARINC 429 Raw Data Live Processor	273
4.21.11.5.1	The Message Area	273
4.21.11.5.2	The ARINC 429 Message Filter Configuration	274
4.21.11.6	Analog Raw Data Live Processor	275
4.21.11.6.1	The Packet Entry Area	275
4.21.11.6.2	The Analog Subchannel Filter Configuration	275
4.21.11.6.3	The FFT Display	275
4.21.11.6.4	General Notes	276
4.21.11.7	Discrete Raw Data Live Processor	276
4.21.11.7.1	The Discrete Status Area	276
4.21.11.7.2	The Discrete Filter Configuration	276
4.21.11.7.3	The Curve Configuration	277
4.21.11.7.4	The Graph Area	277
4.22	Parameter Live Processors	277
4.22.1	Parameter Table Live Processor	277
4.22.2	Parameter Graph Live Processor	278
4.22.2.1	Graph Page	278
4.22.2.2	Curve Setup Page	279
4.22.3	CVSD Live Replay	279
4.23	XML CH10 Mapping	280
4.23.1	Intention of Specification	280
4.23.2	Official and Extended Implementations in FLIDAS	280
4.23.3	Operation	281
4.23.3.1	CH10 to XML Conversion	281
4.23.3.2	XML to CH10 Conversion	281
4.23.4	Samples and Other Information	282
4.23.5	Verifying Created CH10 Files	282
4.23.6	Notes on Implemented CH10 to XML Conversion	282
4.23.7	XML CH10 Mapping Commercial Additions	283
4.23.7.1	CH10 to XML Conversion Commercial Additions	283
4.23.7.2	XML to CH10 Conversion Commercial Additions	284
5	System Administration Topics	286
5.1	Portable Installation	286
5.2	Backup	286
5.3	Update	286

6	Security	288
7	FAQ.....	289
8	Open IRIG 106 Issues	292
8.1	Storage of LSB First PCM Data Words in Unpacked or Packed PCM Mode.....	292
8.2	Storage of PCM Data Words in Unpacked PCM Mode with Word Sizes Above 16 bits	292
8.3	Disabled Analog Subchannels	292
8.4	Chapter 4 time as secondary header time or intra packet time stamps	293
8.5	TSPI / CTS format 1 is ambiguous	293
9	Feedback	294
10	What's New	295
10.1	FLIDAS 4.15.2.....	295
10.2	FLIDAS 4.15.1.....	295
10.3	FLIDAS 4.15.....	296
10.4	FLIDAS 4.14.....	297
10.5	FLIDAS 4.13.....	298
10.6	FLIDAS 4.12.1.....	298
10.7	FLIDAS 4.12.....	298
10.8	FLIDAS 4.11.....	299
10.9	FLIDAS 4.10.....	299
10.10	FLIDAS 4.9.....	300
10.11	FLIDAS 4.8.....	301
10.12	FLIDAS 4.7.....	302
10.13	FLIDAS 4.6.....	303
10.14	FLIDAS 4.5.....	304
10.15	FLIDAS 4.4.....	305
10.16	FLIDAS 4.3.....	305
10.17	FLIDAS 4.2.....	306
10.18	FLIDAS 4.1.....	307
10.19	FLIDAS 4.0.....	308
10.20	FLIDAS 3.18.....	308
10.21	FLIDAS 3.17.....	309
10.22	FLIDAS 3.16.....	309
10.23	FLIDAS 3.15.....	310
10.23.1	Java topics	310
10.23.2	Other changes	311
10.24	FLIDAS 3.14.....	312

10.25	FLIDAS 3.13.....	312
10.26	FLIDAS 3.12.....	313
10.27	FLIDAS 3.11.....	314
10.28	FLIDAS 3.10.....	315
10.29	FLIDAS 3.9.....	316
10.30	FLIDAS 3.8.....	316
10.31	FLIDAS 3.7.....	317
10.32	FLIDAS 3.6.....	318
10.33	FLIDAS 3.5.....	319
10.34	FLIDAS 3.4.....	320
10.35	FLIDAS 3.3.....	321
10.36	FLIDAS 3.2.....	321
10.37	FLIDAS 3.1.....	323
10.38	FLIDAS 3.0.....	323
11	Legal	325
11.1	Free License	325
11.1.1	Third-Party Software	325
11.2	FLIDAS End User License Agreement (EULA)	325
11.2.1	License Grants	325
11.2.2	Warranty	325
11.2.3	System Environment for Upgrades	325
11.2.4	Applicable Law and Place of Jurisdiction	326
11.2.5	Changes to Contract	326
11.2.6	Third-Party Software	326

1 FLIDAS Overview

FLIDAS reads IRIG 106 chapter 10 files with flight data and allows accessing the data on the raw data, protocol and parameter level for display and export. It also allows debugging the CH10 file structure itself. FLIDAS can receive CH10 UDP data streams to display raw data, protocol data and decoded parameters for checking of proper setup and calibration of FTI equipment and as simple monitor application. Besides the free version there are four general-purpose versions

Currently there is no explicit support for TMATS configuration changes during a single recording.

FLIDAS support some types of recordings and live data streams from distributed CH10 multiplexer sources.

1.1 Outlook

Besides continuous improvements there are several areas where extensions are planned:

- More support for recordings of distributed CH10 multiplexer sources.
- Support for more CH10 data types on live data streams
- Support for more CH10 data types on Parameter level

Hard real-time replay of data to the GUI or physical lines is considered out of scope even for the future.

1.2 How to Use This Manual

This manual consists of several sections:

- The **Overview** chapter contains some initial information you should read before starting.
- The **Getting Started** chapter contains tutorials for the basic use of the program but does not explain all functionality.
- The **Concepts** chapter explains general terms and ideas but not the use of these concepts. Instead, it links you to the graphical user interface elements that work with these concepts.
- The **Reference** chapter then covers all the user interactions with the tool in detail.
- Further chapters cover topics that did not fit in somewhere else or are not directly related to the use of the tool.
- The **Legal** chapter in the end explains the conditions under which you may use the tool. If you don't agree you may not use it.

The manual offers a lot of cross linking between topics that you could use to find additional information.

1.3 System Requirements

In short:

- Windows 10 or later
- Java 17 JRE or JDK 64-bit or later (you might have 64-bit Windows but still have 32-bit Java => unsupported. Install 64-bit Java. See [installation chapter](#))

For more information or other older systems keep reading:

Most of FLIDAS is written platform independent. Only the GUI aspects are linked to specific operating systems. FLIDAS is officially available for Windows 10 or higher. Versions for other operating systems (especially Linux (X11 not Wayland)) can easily be provided on request (most of the time free of charge). You may use the free version to see if the GUI aspects work out in your environment.

To run FLIDAS officially a Java installation of at least Java 17 is needed. You will also need a 64-bit Java. Note that having a 64-bit Windows does not mean that your installed Java version is necessarily 64-bit. You might need to install a 64-bit Java in addition. If you don't know how to set it up, read the [installation chapter](#). Most Java is free and you can download it for example from www.azul.com or www.microsoft.com/openjdk.

Note: A quick test was done on Windows 7 with Java 11 and no problems were discovered. You can try on your own risk and also report problems. Liability is however limited to Windows 10 or later and Java 17 64 bit or later.

FLIDAS can arrange windows over multiple monitors if available.

It is possible to create [portable installations](#) that run from an USB stick.

For all [data streaming](#) features you will need a network adapter with IP4.

1.4 Limitations / Features out of Scope

This section shall clarify some topics that are out of scope of FLIDAS or that are limitations that might be relevant to you. Some of these topics may be addressed in future versions:

FLIDAS is not downloading the CH10 files from the recorder. CH10 files must be accessible within the environment of the computer where FLIDAS is installed.

FLIDAS does not support all [CH10 data types](#) in detail. Look up the [CH10 data type support](#) section for the level of support for each CH10 data type.

1.4.1 Supported IRIG 106 Specification Version

FLIDAS offers support for the CH10 file format version up to the 2022 version with one exception:

- Video format 0 packets from 2004 are not supported. (The only sample data available to us doesn't comply with the standard, so it was not safe to implement. However, the actual video data was properly available so you might try exporting video data on your own risk)

The 2022 support in general is implemented in this way:

- The support for the 2022 version has been implemented
- All supported CH10 data types have been checked for differences between specification versions
- CH10 sources of older versions will still show the data according to the 2022 version.
- Since all unused data bits in older standards are required to be set to 0, options not available in that specification will use the "0-value option"
- The "0-value option" has been checked to be a useful default also for older versions.
- In rare situations there is a special treatment for older versions.

Example: Assume a packet was defined to be in little-endian coding in the 2004 standard. The 2005 standard added the option to have little or big-endian coding but little-endian is coded as a 0. Since the coding bit was not used in 2004, it is set to 0 by default. The GUI will show a little-endian property also for the 2004 standard although it is not coded as an option in the CH10 source but defined by the specification instead. The fact that it is little-endian coded is however still correct.

1.4.2 Data Streaming Performance and Reliability

If you plan to use data streaming bear two one things in mind: Data streaming uses the UDP protocol which by specification is allowed to drop data all along its transmission path (routers, switches, network interfaces, operating system processing etc.) and neither the operating systems you use nor FLIDAS itself is real-time capable. The IRIG 106 CH10 specification already defines permissible delays for the data to become available on the network and the network itself gives no guarantees on delivery delays. So ultimately you have to live with the risk of data loss and unknown delays for displayed data. Never the less realistic system setups can run for hours without any data loss and perceived delays of around two seconds (or one second without channel synchronization). The points addressed here also affect competitor's software but we might be more cautious than others so the advice is not to use FLIDAS data streaming in an environment where you need guaranteed maximum display delays and cannot sustain data loss. If you do anyway make sure to test FLIDAS under your specific system environment. You can order evaluation versions if you need to.

1.5 Versions

FLIDAS is available in a **free version** and **four general-purpose commercial versions** that can be further extended by plug-ins for specific needs. The general-purpose versions have a very affordable price because Data Bus Tools focuses on earning on customizations for customer specific needs. The prices are given on the [feature details table](#) below.

Since many parts of the IRIG 106 standard are open to interpretation and may be handled differently by recorder manufacturers, you are invited to try your IRIG 106 chapter 10 files with the free version first.

The colored table below shows six colored areas of feature groups. Vertically you see the level of the data that the features are working on (CH10 structures, protocol / raw data, decoded Parameters). Horizontally you see the source of the CH10 data (files or UDP data streams).

The features given in the tables are high-level sample features to illustrate what is meant by this feature group. The detailed features are found in another [table further down the page](#).

	CH10 file-based features	CH10 UDP stream-based features
CH10 debugging level	<ul style="list-style-type: none"> analyze CH10 structures and attributes CH10 file verification ("CH10 validator") View TMATS data Find CH10 errors Read corrupted files Guided hex view of packets Limited manipulation of raw packets XML CH10 mapping to analyze and manipulate the CH10 level of files and generate low level test data Bit rate calculation Locate recording gaps 	<ul style="list-style-type: none"> Summarize incoming packets Summarize CH10 stream errors CH10 publishing CH10 recording CH10 live processing
Protocol / Raw data level	<ul style="list-style-type: none"> Raw / protocol data exports Communication summaries Inspection of raw / protocol data Detection of protocol errors Detect errors in IRIG 106 Chapter 7 streams Locate transmission pauses 	<ul style="list-style-type: none"> Live display of raw data and protocol summaries Live display of video Live replay of CH10 or TmNS / (Telemetry Network Standard, IRIG 106 Chapter 24) messages from IRIG 106 chapter 7 PCM Stream out live raw data as simple UDP stream for external processing
Parameter level	<ul style="list-style-type: none"> Decode raw data to engineering units Display / print engineering values in tables or strip charts Export engineering values Export / replay of CVSD audio Map display and replay of NMEA 0183 position data Replay of data to user defined instrument panels 	<ul style="list-style-type: none"> Live display of engineering values Live replay of CVSD audio

The table above also divides the features in four quadrants with the thick borders. For each part you need to decide if you need any of the features and this leads to the general-purpose version you will need.

The small colored tables below show the free and general-purpose versions and which feature groups are included with the color coding from the table above.

The decisions you need to take are:

- Do you need UDP data streaming (file processing is always included)? \Rightarrow take a version to the right.
- Do you need parameter decoding to engineering units (protocol / raw data and CH10 structure level is always included)? \Rightarrow take a version from the bottom.

Free Version			
a little			
a little			
Basic Version		Data Streaming Version	
Parameter Version		Parameter Streaming Version	

1.5.1 Feature Details

The following table shows the detailed features of the versions. If a feature does not explicitly refer to *data streaming* or *live* it is a file-based feature.

Feature	Free	Basic	Data Streaming	Parameter	Parameter Streaming
Price in €	0	1500	2000	2900	3600
Price in US\$ (we charge €; this price is from the last release date just for information)	0	1627	2170	3146	3906
Read IRIG 106 chapter 10 files up to IRIG 106-22	X	X	X	X	X
Read segmented .ch10 files from Safran Data System recorders	X	X	X	X	X
Display the list of packets	X	X	X	X	X
Support for recording events	X	X	X	X	X
Display the packet entries within packets of supported CH10 data types	X (data displayed in simple lines of text)	X (same data displayed but split to individual and configurable columns. For some CH10 data types more data available)			

Display the available channels and subchannels with some TMATS attributes	X	X	X	X	X
Display the TMATS setup	X	X	X	X	X
Debugging of corrupted files (missing packets, checksum errors etc.)	X	X	X	X	X
Locating packet and packet entry errors	- (Errors are shown but no aid to navigate to them)	X	X	X	X
CH10 file verification ("CH10 validation") with detailed error inspection on GUI	-	X	X	X	X
Display of all common packet properties and structured hex display of whole packet data	-	X	X	X	X
Modification of packet data within hex display	-	X	X	X	X
Replay audio from analog subchannel	X	X	X	X	X
Replay audio from CVSD	-	-	-	X	X
Video preview of MPEG transport stream	-	X	X	X	X
Display graph of analog subchannels within sequence of packets	-	X	X	X	X
Display PCM minor frames within sequence of Packets	-	X	X	X	X
Display UART subchannel as text	-	X	X	X	X
Display TSPI format 0 channel, UART subchannel or UDP packets as NMEA 0183	-	X	X	X	X
Display and replay of some NMEA 0183 position parameters (from TSPI format 0 channel, UART subchannel or UDP packets) on build-in or user defined map and instruments.	-	-	-	X	X
More detailed world map for NMEA 0183 map display	-	-	-	free on request	
Tabular display of sampled parameters	-	-	-	X	X
Strip chart of sampled parameters for display and printing	-	-	-	X	X

User defined instrument panels with gauges for live display of decoded parameters from live streams or replayed CH10 files	-	-	-	X	X
Unicode UTF-8 support for Japanese, Chinese, Cyrillic etc. letters	X	X	X	X	X
Override settings of CH10 files with external TMATS file	X	X	X	X	X
CH10 bitrate calculation for file and channels	X	X	X	X	X
PCM BERT check	-	X	X	X	X
Locate recording gaps or transmission pauses	-	X	X	X	X
XML CH10 mapping according to RCC document 123-16 Appendix P/Q (IRIG 106 Chapter 10 Programmers Handbook, moved to appendix Q/R in RCC 123-20): Convert between XML and CH10 for test data creation, bug correction etc.	X	X	X	X	X
Additional XML generation options for XML CH10 mapping	-	X	X	X	X
Support for CAN buses in DATaRec4 (Zodiac Data Systems format)	-	separate plug-in			
Support for STANAG3910 buses in DATaRec4 (Zodiac Data Systems format)	-	separate plug-in			
Distributed CH10 multiplexer sources support					
Sources must provide a common TMATS with a single R-Group (either from CH10 file / stream or external TMATS file). Live data streams need to go to a single UDP port.					
Time decoding from independent RTC on different multiplexer sources using MSID	X	X	X	X	X
Secondary header time	X	X	X	X	X
Configure time offsets between different sources	X	X	X	X	X
Split CH10 files from distributed CH10 multiplexer sources into separate files by their multiplexer source IDs (MSIDs).	X	X	X	X	X
Data streaming support					
Recording of CH10 UDP data stream	X	X	X	X	X

Publishing of CH10 file as CH10 UDP data stream	X	X	X	X	X
Reception of CH10 UDP data stream for live processing	X	X	X	X	X
Live display of statistics, bit rates and errors per channel	X	X	X	X	X
Live display of some PCM decommutation statistics	-	-	X	-	X
Live display of MPEG transport stream videos	-	-	X	-	X
Live display and graphical plot of PCM raw data	-	-	X	-	X
Live display and graphical plot of MIL-STD 1553B raw data and protocol information	-	-	X	-	X
Live display and graphical plot of ARINC 429 raw data and protocol information	-	-	X	-	X
Live display, graphical plot and FFT (magnitude only) of an analog subchannel	-	-	X	-	X
Live display and graphical plot of discrete raw data	-	-	X	-	X
Tabular live display of decoded parameters	-	-	-	-	X
Strip chart live display of decoded parameters	-	-	-	-	X
Live replay of CVSD audio	-	-	-	-	X
Live replay of analog audio	-	-	X	-	X
PCM BERT check	-	-	X	-	X
Live statistics of IRIG 106 chapter 7 PCM	-	-	X	-	X
Live replay of CH10 from IRIG 106 chapter 7 PCM	-	-	X	-	X
Live replay of TmNS / (Telemetry Network Standard, IRIG 106 Chapter 24) messages from IRIG 106 chapter 7 PCM	-	-	X	-	X
Stream out live raw data as simple UDP live stream for external processing (PCM throughput or UART)	-	-	X	-	X
Exporters (concurrent export possible)					
MPEG video to MPEG file	X	X	X	X	X
TMATS to text file	X	X	X	X	X

CH10 to filtered CH10 file (filtering time and channels. The result is no formally correct CH10 file (see manual))	X	X	X	X	X
Ethernet to PCAP	-	X	X	X	X
MIL-STD 1553 to CSV	-	X	X	X	X
ARINC 429 to CSV	-	X	X	X	X
CAN bus to CSV	-	X	X	X	X
Discretes to CSV	-	X	X	X	X
UART to CSV	-	X	X	X	X
Fibre channel to CSV	-	X	X	X	X
Message data to CSV	-	X	X	X	X
TSPI to CSV	-	X	X	X	X
PCM minor frames to CSV	-	X	X	X	X
PCM throughput mode stream to binary file	-	X	X	X	X
Analog subchannel(s) to WAV, CSV or MAT (MATLAB)	-	X	X	X	X
TSPI format 0 channel, UART subchannel or UDP packets as NMEA 0183 to CSV	-	X	X	X	X
Payload to file (from UART or message data subchannel or UDP packet) Examples are streamed video to MPEG or NMEA 0183 to text	-	X	X	X	X
IENA packets to CSV (from Ethernet or message data)	-	X	X	X	X
INET-X packets to CSV (from Ethernet)	-	X	X	X	X
IRIG 106 chapter 7 data from PCM channel to CH10, Ethernet PCAP or IP PCAP	-	X	X	X	X
MIL-STD 1553 / ARINC 429 to Airbus Defence & Space real-time test system AIDASS 2000 (also MaTE, SYGAM) TRF format (can be used for replay or analysis in AIDASS) (<i>AIDASS is a registered trademark of Cassidian Germany</i>)	-	separate plug-in			
Export all important GUI tables to CSV	-	X	X	X	X
Sampled parameters to CSV or MAT (configurable period or asynchronously on change)	-	-	-	X	X

PCM numeric parameters to WAV	-	-	-	X	X
Raw sections of PCM streams (Byte Array) to binary file	-	-	-	X	X
ARINC 429 message-based export to CSV using parameters (no sampling but export at time of occurrence. Export of protocol information and one included parameter)	-	-	-	X	X
Export CH10 file with modified TMATS setup	-	X	X	X	X
Communication summaries (message types, periods, error counts and locations etc.)					
MIL-STD 1553	-	X	X	X	X
ARINC 429	-	X	X	X	X
CAN bus	-	X	X	X	X
UART	-	X	X	X	X
Ethernet (including payload analysis for IP4 and UDP)	-	X	X	X	X
Message data	-	X	X	X	X
Video MPEG transport stream	-	X	X	X	X
Time source	-	X	X	X	X
PCM minor frames	-	X	X	X	X
Recording events	-	X	X	X	X
IENA (from Ethernet or message data)	-	X	X	X	X
iNET-X (from Ethernet)	-	X	X	X	X
IRIG 106 Chapter 7	-	X	X	X	X
Parameter decoding management (engineering unit conversion) for files and streams					
Definition of engineering unit (EU) conversions for parameters in an external ICD file (CSV text file. Can be used with Excel).	-	-	-	X	X
Definition of engineering unit (EU) conversions for parameters in a separate ASCII TMATS file. PCM only. Not all TMATS features supported.	-	-	-	X	X
Parameter decoding support for: <ul style="list-style-type: none"> • ARINC 429 • CAN bus • MIL-STD 1553B (format 1) • PCM (see manual for details) 	-	-	-	X	X

<ul style="list-style-type: none"> • Text-based UART (see manual for details) • Binary UART (see manual for details) • Ethernet UDP (see manual for details) • Analog (format 1 and 3 (I/Q recording of intermediate frequency) as well) • Discretes • NMEA 0183 (e.g., GPS, Galileo, GLONASS and Beidou GNSS) on UART or TSPI format 0. Fixed ICD with most relevant parameters from some sentences with positional information. • Message data (see manual for details) • User defined computer-generated data (see manual for details) • Recording events • Other CH10 data types on request or added with further releases. 					
Parameter decoding support for STANAG3910 in DATaRec4 (Zodiac Data System format)	-	-	-	separate plug-in	
Parameter decoding support for network data objects (NDO) in Ethernet/UDP (quite specific: Fixed sized data blocks on fixed starting position in UDP packet. Separate decoding for each block)	-	-	-	separate plug-in	
Definition of parameter sets (group of parameters used in the same test situation) and joining of existing parameter sets.	-	-	-	X	X

The features of all versions will be continuously enhanced. You can add upgrades to your procurement for a price of 35% of the current version price per year. This includes all upgrades within your version. Feature requests raised by you that make it to the general-purpose versions are provided to you this way.

2 Getting Started

2.1 Installing FLIDAS

To install simply extract the ZIP archive in a directory on your computer. There is no installation required.

Notes:

- *If you are concerned if the download was successful, check the [FAQ section](#) for the checksum verification process.*
- *All configuration data of FLIDAS will be stored in this directory so you should make sure this directory is accessible for writing (not only during installation). This might not be the case for the "Program Files" directory on newer Windows releases.*
- *The Windows native ZIP archive support is said to have problems with long file names. If you encounter problems, use a third-party tool (e.g., 7-Zip) for extracting the archive.*
- *There are two warnings for [Eclipse](#) based applications which probably also apply to FLIDAS and should be followed:*
 - *Avoid path names longer than 255 characters on Windows*
 - *Do not use spaces in the path*

When Java is properly installed, you should be able to start the FLIDAS executable in the directory where you extracted the ZIP archive. If not, you might not have a Java installation or not the right one. You will need a 64-bit Java. Note that even on a 64-bit computer Java 32-bit may be installed which cannot run FLIDAS. To solve this, you have several options:

- If you have no other software on your computer that needs the 32-bit Java, uninstall it and install 64-bit. You might need administrator privileges for this
- If you need to keep the 32-bit Java or cannot have administrator privileges:
 - Consider installing a 64-bit Java inside the FLIDAS directory like described for the [portable installation](#).
 - Alternatively install an additional 64-bit Java in any directory you like and point FLIDAS to it:
 - Open the FLIDAS.ini file inside the FLIDAS installation directory with a text editor.
 - Add these two lines before the line that says "-vmargs"
 - `-vm`
`C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe`

The text must directly start in front of the line.

- Adapt the second inserted line to have the path for the javaw.exe file within your new Java installation.

Java installation above normally means just extracting a ZIP archive. There are a lot of free Java versions out there (called OpenJDK). You can use the smaller JRE or the bigger JDK version. For FLIDAS a Java version of at least 17 is recommended. An LTS version means long-term-support. Prefer this over the other versions. An example where to download Java would be www.azul.com/downloads. Other free sources are Microsoft, IBM and others. Usually all these versions are created from the same open-source code so choose any of them.

Note: You can also create a portable installation on a USB stick without the need of a local Java installation. [Here are the procedures.](#)

2.1.1 Associating CH10 Files with FLIDAS

You can open FLIDAS automatically when you double-click a CH10 file in the Windows Explorer. To do so, right-click a CH10 file and select "Open with" (or "Open"). Then select the FLIDAS executable and use it to open the CH10 file. If you configure Windows to use this method of opening by default, you can open CH10 files with FLIDAS with a double-click on them. You still have to complete the wizard to open the file after FLIDAS has started.

2.2 CH10 Tutorial

There is a tutorial giving you some basic insight in the CH10 file format and related topics like PCM and TMATS. It is independent of FLIDAS and uses techniques not suitable for this manual. You can open it in a browser at https://www.databustools.de/ch10_tutorial/ch10_tutorial.html

2.3 Basic Tutorial

This tutorial will show you how to do the first steps in FLIDAS. It will show you how to inspect a CH10 file's content. On the way it will teach you some basic concepts of the user interface and link you to further topics somewhere else in the documentation.

Start FLIDAS with the FLIDAS executable in your installation directory.

To open a CH10 file click **File > Open CH10 File...** or use the keyboard shortcut **Alt+O**.

If you don't have a file and still want to test out FLIDAS, you could download sample files from the third-party website www.irig106.org.

After you have selected the file, a Wizard dialog pops up.

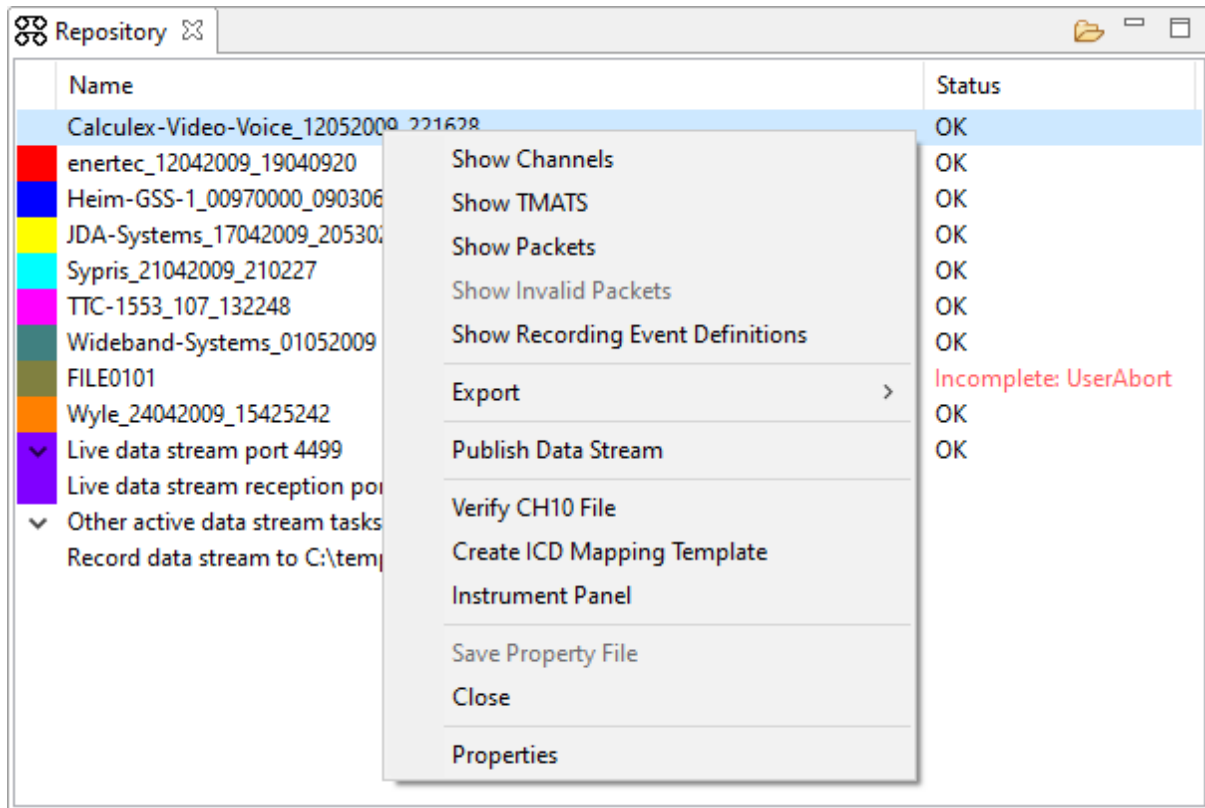
The CH10 file loading wizard controls which data is loaded from the file. It can be used to increase the performance of loading and overall handling of the file. It will also help you focus on topics important to you.

All wizards work the same way and allow toggling between a set of configuration pages with **Next** and **Back**. In some situations, you might not be able to switch to the next page or finish the dialog when there is a problem in the current or previous page.

Right now, you can just click **Finish (3)** as the following pages contain sufficient default values. The "(3)" means that you ignore the following three configuration pages. After clicking the button, the file will load and display its progress in loading.

Note that loading files from network drives is usually slower and any kind of exporting data to network drives is significantly slower compared to local drives.

Once the loading is finished, the **Repository View** opens. This view shows all files that are currently open and is the starting point for most actions on the files.



Note: The Repository View in fact shows CH10 sources which may be CH10 files but also live data streams

To show the file's content right-click it and select **Show Packets**.

The **Packet View** opens and shows a list of all packets contained in the file. The colored border around the view will help you distinguish between several open files at a time.

Time	Chan #	Type	Errors	Data
4/7/09 9:03:05 AM 961:262:900	87	MIL-STD-1553 Data, Fmt 1		Messages: 26 Time tag reference: First bit of the first word of the me...
4/7/09 9:03:05 AM 961:262:800	88	MIL-STD-1553 Data, Fmt 1		Messages: 26 Time tag reference: First bit of the first word of the me...
4/7/09 9:03:05 AM 961:262:900	89	MIL-STD-1553 Data, Fmt 1		Messages: 26 Time tag reference: First bit of the first word of the me...
4/7/09 9:03:05 AM 961:262:800	90	MIL-STD-1553 Data, Fmt 1		Messages: 26 Time tag reference: First bit of the first word of the me...
4/7/09 9:03:05 AM 961:262:800	91	MIL-STD-1553 Data, Fmt 1		Messages: 26 Time tag reference: First bit of the first word of the me...
4/7/09 9:03:05 AM 961:262:800	92	MIL-STD-1553 Data, Fmt 1		Messages: 26 Time tag re
4/7/09 9:03:05 AM 961:262:800	93	MIL-STD-1553 Data, Fmt 1		Messages: 26 Time tag re
4/7/09 9:03:05 AM 961:262:800	94	MIL-STD-1553 Data, Fmt 1		Messages: 26 Time tag re
4/7/09 9:03:05 AM 861:152:200	62	Analog Data, Fmt 1		#Subchannels: 1 Packing
4/7/09 9:03:05 AM 943:032:200	61	Analog Data, Fmt 1		#Subchannels: 1 Packing
4/7/09 9:03:05 AM 953:702:600	55	PCM Data, Fmt 1		Mode: Packed 16-bit alig
4/7/09 9:03:05 AM 953:702:600	56	PCM Data, Fmt 1		Mode: Unpacked 16-bit a
4/7/09 9:03:05 AM 967:715:000	51	PCM Data, Fmt 1		Mode: Throughput 16-bi
4/7/09 9:03:05 AM 970:303:400	52	PCM Data, Fmt 1		Mode: Throughput 16-bi
4/7/09 9:03:05 AM 975:465:200	53	PCM Data, Fmt 1		Mode: Throughput 16-bi
4/7/09 9:03:05 AM 993:702:600	54	PCM Data, Fmt 1		Mode: Throughput 16-bi
4/7/09 9:03:06 AM 001:227:102	96	Ethernet Data, Fmt 0		Frames: 72 Format: Ether
4/7/09 9:03:05 AM 735:179:000	43	Video Data, Fmt 0 (MPEG-2/H...		Little endian Payload: MF
4/7/09 9:03:05 AM 977:418:700	95	Ethernet Data, Fmt 0		Frames: 44 Format: Ether
4/7/09 9:03:06 AM 019:982:833	87	MIL-STD-1553 Data, Fmt 1		Messages: 25 Time tag re...
4/7/09 9:03:06 AM 019:982:733	88	MIL-STD-1553 Data, Fmt 1		Messages: 25 Time tag reference: First bit of the first word of the me...

By now you should have heard the word "view" a few times. The term is used in the Eclipse application framework (that is used by FLIDAS) for a window with a specific behavior. You will need to know how to arrange the views so here are some basic actions:

- Double-click the views title to resize it to fill the whole window. Double-click it again to size it normal
- Drag the title bar to another position inside the main window. The area will be split between this view and other existing views. Some indicators will show where it will be placed.
- Drag the title bar next to the title bar of another view (like the Repository View). It will then be stacked with the other view.
- Drag the line between two views to resize them

Now double-click on a packet. A view will open to show the packet entries contained in the packet

This Packet Entry View will look different depending on the situation:

- There are CH10 data types which offer further support. The others do not show any packet entries.
- The free FLIDAS version shows a very simple, unstructured list of the data, while all commercial versions have an improved representation

Below you will see a few examples:

TTC-1553_107_132248

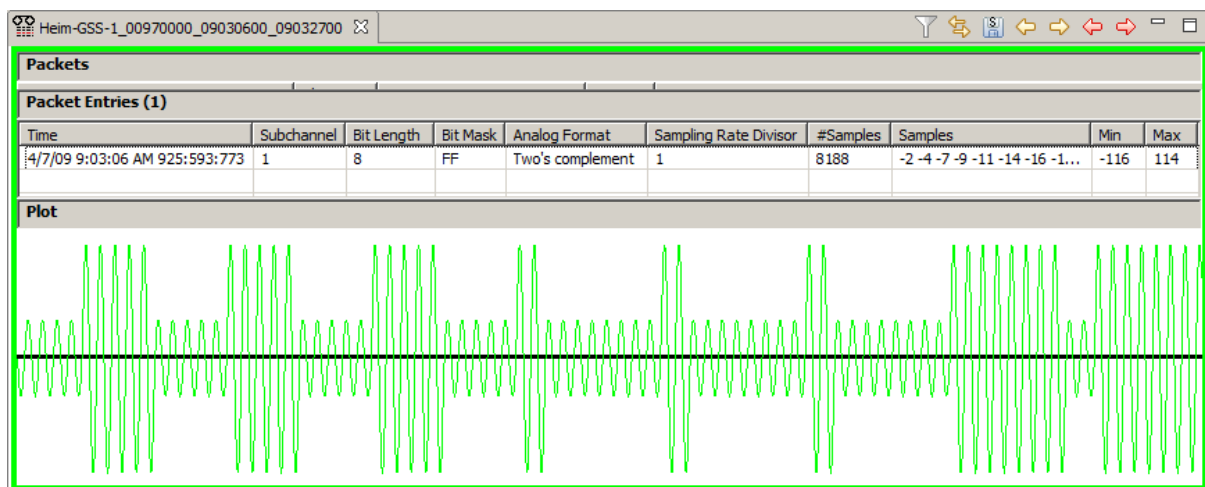
Packets																	
Time	Chan #	Type	Errors	Data													
4/17/09 1:22:48 PM 997:206:000	2752	MIL-STD-1553 Data, Fmt 1		Messages: 44 Time tag reference: Last bit of the first (command) word of the message													
Packet Entries (44)																	
Time	Bus	Msg Type	Rx Cmd	Tx Cmd	Rx St	Tx St	Rx RT	Rx SA	Rx WC	Tx RT	Tx SA	Tx WC	RxSt RT	RxSt Flags	TxSt RT	TxSt Flags	N
4/17/09 1:22:48 PM 997:206:000	B	RT to BC transfer		5E54		5800				11	18	20			11		
4/17/09 1:22:48 PM 997:660:000	A	RT to RT transfer	68E0	36C0	6800	3001	13	7	32	6	22	32	13		6	Term	
4/17/09 1:22:48 PM 998:397:100	A	Mode command with...								21	0	2			21		Ti
4/17/09 1:22:48 PM 998:458:100	A	Mode command with...								7	31	18			7		Ti
4/17/09 1:22:48 PM 998:534:100	A	Mode command with...	78						17				15				S
4/17/09 1:22:49 PM 000:017:099	A	BC to RT transfer	09						31				1				
4/17/09 1:22:49 PM 000:681:095	B	RT to BC transfer								11	18	20			11		
4/17/09 1:22:49 PM 001:135:092	A	RT to RT transfer	68						32	6	22	32	13		6	Term	
4/17/09 1:22:49 PM 001:872:087	A	Mode command with...								21	0	2			21		Ti
4/17/09 1:22:49 PM 001:933:086	A	Mode command with...		3FF2		3800				7	31	18			7		Ti

A commercial version (MIL-STD 1553 example above - a lot more columns to the right)

TTC-1553_107_132248


Packets				
Time	Chan #	Type	Errors	Data
4/17/09 1:22:48 PM 997:206:000	2752	MIL-STD-1553 Data, Fmt 1		Messages: 44 Time tag reference: Last bit of the first (command...
4/17/09 1:22:49 PM 017:322:082	2752	MIL-STD-1553 Data, Fmt 1		Messages: 46 Time tag reference: Last bit of the first (command...
Packet Entries (90)				
Time	Data			
4/17/09 1:22:48 PM 997:206:000	Bus B Type: RT to BC transfer Tx cmd: 5E54 Tx stat: 5800 Data: 0001 9DF0 07D9 006B 0000 BC28 000...			
4/17/09 1:22:48 PM 997:660:000	Bus A Type: RT to RT transfer Rx cmd: 68E0 Tx cmd: 36C0 Rx stat: 6800 Tx stat: 3001 Data: 000...			
4/17/09 1:22:48 PM 998:397:100	Bus A Type: Mode command without data Tx cmd: AC02 Tx stat: A800			
4/17/09 1:22:48 PM 998:458:100	Bus A Type: Mode command with data (tx) Tx cmd: 3FF2 Tx stat: 3800 Data: 382D			
4/17/09 1:22:48 PM 998:534:100	Bus A Type: Mode command with data (rx) Rx cmd: 7811 Rx stat: 7800 Data: 87EE			
4/17/09 1:22:49 PM 000:017:099	Bus A Type: BC to RT transfer Rx cmd: 097F Rx stat: 0800 Data: 0001 9DF5 07D9 006B 0000 BC29 00...			
4/17/09 1:22:49 PM 000:681:095	Bus B Type: RT to BC transfer Tx cmd: 5E54 Tx stat: 5800 Data: 0001 9DF6 07D9 006B 0000 BC29 000...			
4/17/09 1:22:49 PM 001:135:092	Bus A Type: RT to RT transfer Rx cmd: 68E0 Tx cmd: 36C0 Rx stat: 6800 Tx stat: 3001 Data: 000...			
4/17/09 1:22:49 PM 001:872:087	Bus A Type: Mode command without data Tx cmd: AC02 Tx stat: A800			
4/17/09 1:22:49 PM 001:933:086	Bus A Type: Mode command with data (tx) Tx cmd: 3FF2 Tx stat: 3800 Data: 382D			
4/17/09 1:22:49 PM 002:009:086	Bus A Type: Mode command with data (rx) Rx cmd: 7811 Rx stat: 7800 Data: 87EE			
Information				

The free version (MIL-STD 1553 example above)



A commercial version (analog example above)

Now arrange the Packet View and the Packet Entry View side by side (e.g., drag the Packet Entry Views tab to the lower part of the screen).

Click on the  "Link with selection" icon on the top of the Packet Entry View. Then click on a new packet in the Packet View. See how the Packet Entry View changes to show the new packet's content. This link between the selected packet and the displayed packet entries can be enabled or disabled as you have just done. The view will freeze on the shown packet entries or will stay synchronized with the latest selection in the Packet View of the same color (i.e., belonging to the same file).

Now click on a new packet in the Packet View and then hit the F1 key (or whatever key is associated with help on your system). [Online help](#) will open explaining the Packet View. You can use this on many other areas of the application too.

This tutorial ends here. There are a lot of other features to try out guided by the manual. Some major areas where you might want to start are:

- [Things you can do with almost any table](#)
- [Inspecting the channels](#)
- [Communication summaries](#)
- [Exporting data](#)
- [Parameter decoding](#)
- [Live data processing from UDP data streams](#)

A good thing to do is also to right-click on items inside the tables in the Views that you use to bring up the context menu. This shows you available actions in many places that you can then read about in the context sensitive help for the View.

2.4 Parameter Tutorial

This tutorial will show you how to work with [parameters](#) i.e., how to set up your system to decode raw data to an engineering unit level. The tutorial expects that you are familiar with the basic use of FLIDAS as explained for example in the [basic tutorial](#).

All features described here rely on the [Parameter or Parameter Streaming Version](#).

The tutorial will use PCM decoding as an example but you should be able to apply this to other CH10 data types too. A sample file is provided.

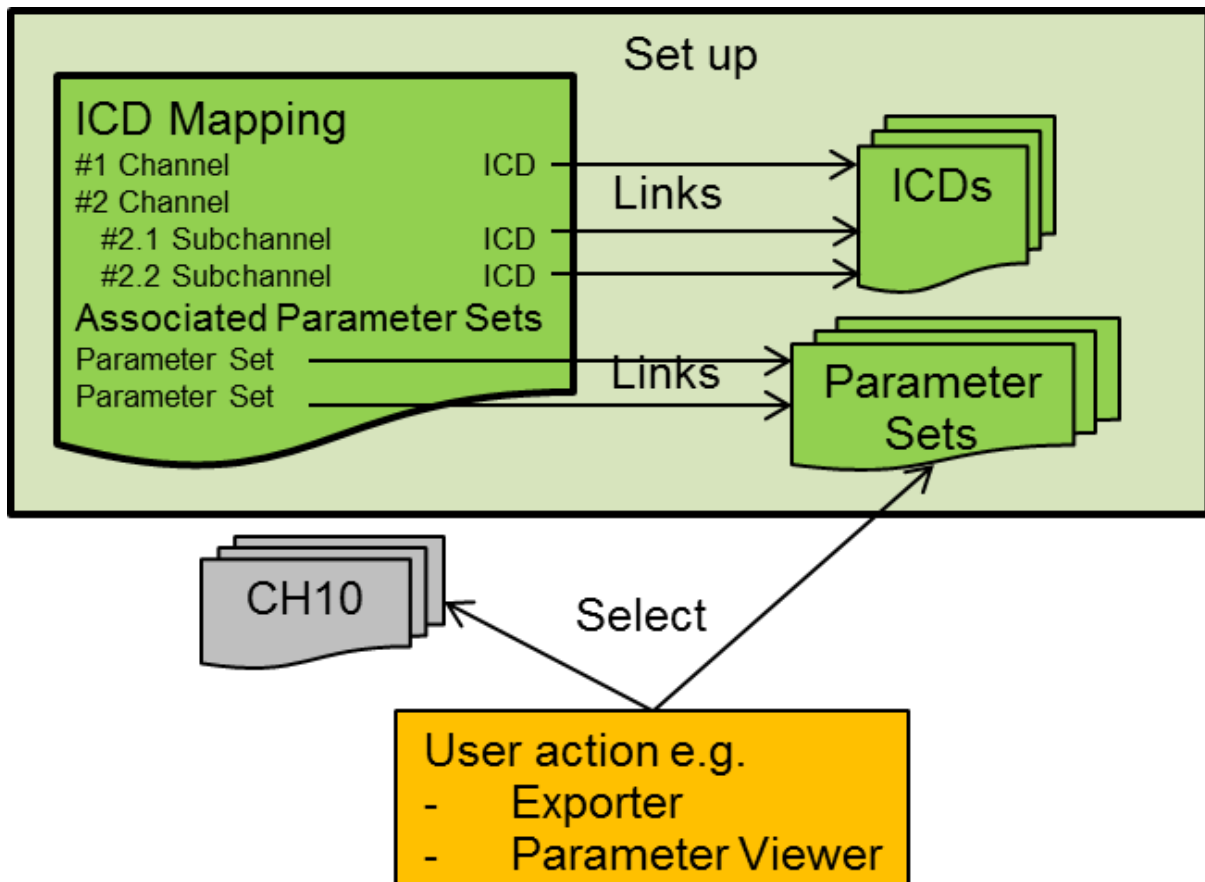
FLIDAS offers several functions that use parameters for displaying ([parameter viewers](#) and [parameter live processors](#)) and exporting ([exporters](#)). Before these functions can be used you must teach the system about the parameters that are encoded in the raw data of your avionics system.

Setting up your system consists of several steps:

1. You must manually define [ICDs](#). These are files that describe which data is encoded in the raw data of a specific source (e.g., an equipment) and how it is decoded to engineering units. *Note that it is also possible to order support for your already existing company internal file format.*

2. You must tell FLIDAS which channels and subchannels of your CH10 sources are using which ICD (i.e., which equipment or sensor is connected). This is called an ICD mapping.
3. You must select which parameters you want to use e.g., for displaying. This selection is called a parameter set. You can save parameter sets for later use or just make a one-time selection. Saved parameter sets can also be joined for use.

These three things you need to create can be seen in the green box in the picture below. This set up is intended to be used over a broader scope of CH10 sources so ideally this is a one-time set up action. Of course, your avionics system may change as well as your CH10 channel setup. Some mechanisms are implemented to limit the changes required to adapt to this.



2.4.1 Creating an ICD

With an ICD you solve the three questions about a parameter:

- What is the name and the engineering unit?
- Where in the raw data can I find the parameter? Usually this means defining a range of bits that encode the parameter value. The question is not completely solved here as it only defines the location inside the data of this type of equipment. The ICD Mapping solves the question on which channel it is recorded.
- How is the raw data interpreted to its engineering units once it is found?

ICDs are managed externally from FLIDAS. The default format for PCM and most other data types would be a CSV text file that you can create for example with Excel. PCM also supports the TMATS ICD format.

For this tutorial we will work with sample files that are included in the installation. Click **File > Export Resources** and select the **Parameter Tutorial Sample Files** and select a directory for saving. The created directory will include several files for this tutorial.

The CH10 sample file contains PCM data for this tutorial. It is an artificially created file (using the FLIDAS XML CH10 mapping). The focus was to create working test data for this tutorial and not to correctly model every other aspect of CH10. The XML file is included in the sample files to play around with it and add error cases if you want.

The following screenshot shows the general PCM decommutation settings of the channel. You can see this yourself if you show the channel properties in the Channel View.

PCM Channel Properties

Data packing option: Throughput
 PCM code: NRZ-L
 Transmission randomized: No
 Bit rate: 4000.0
 Common word length: 16
 Word transfer order: Most significant bit first
 Words per minor frame: 24
 Bits per minor frame: 400
 Minor frame sync type: Fixed pattern
 Sync pattern length: 32
 Sync pattern [hex]: fe6b2840
 Good frame syncs to synchronize: 0
 Allowed bit errors in frame syncs to synchronize: 1
 Bad frame syncs to go out of sync: 0
 Allowed bit errors in frame syncs when going out of sync: 1
 Minor frames per major frame: 4

Name	Word	MSB ...	Bits	Transfer order	Counte...	Numb...	Co...
Subframe ID counters:							
SFID	1	15	2	Most signific...	0	4	Inc

The following screenshot shows a major frame of the data and marks some parameters inside that we are going to decode. You can see this yourself if you show a PCM packet in the Packet Entry View and load enough packets inside for a full major frame.

FS	SFID	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	W18	W19	W20	W21	W22	W23
FE6B2840	1	0000	0100	0000	000A	2EE0	2710	0000	0000	0000	0000	0000	0000	0000	0110	0000	0000	0000	0000	0000	0000	0000	0000	0000
FE6B2840	2	0001	0120	0000	0014	2EFF	2610	0000	0000	0000	0000	0000	0000	0000	0130	0000	0000	0000	0000	0000	0000	0000	0000	0000
FE6B2840	3	0002	0140	0000	0020	2F03	2510	0000	0000	0000	0000	0000	0000	0000	0150	0000	0000	0000	0000	0000	0000	0000	0000	0000
FE6B2840	4	0003	0160	0000	002E	2F09	2410	0000	0000	0000	0000	0000	0000	0000	0170	0000	0000	0000	0000	0000	0000	0000	0000	0000

Some notes for some who are not too familiar with PCM or TMATS:

- The PCM minor frame has 24 words where one of them is the frame sync displayed in the "FS" column. The remaining 23 PCM data words are numbered 1 to 23.
- The subframe ID counter (column "SFID") is numbering the different minor frames within a major frame from 1 to 4. The number is encoded in the word 1 as values 0 to 3.

To decode the data, you need to define an ICD. Create a file with a *.csv extension and open it with a simple text editor first. Then copy and paste the following three lines inside:

```
V3;;;;;;;;;;
Name;Unit;Comment;Subframe ID Counter name;Frame position;Frame
interval;Word (1...n);Word Interval;MSB pos (1...n);Length;Encoding (BNR /
UBNR / BCD / Float / Enum / CVSD / ASCII / ByteArray / MIL1750AFloat /
DECDFloat);Min;Max;Enumdef;Coefficient 0;Coefficient 1
Baro Altitude;ft;;;1;;4;;1;16;BNR;;;0;2
```

Save the file and open it with a spreadsheet application like Excel. The result should look somewhat like this:

V3															
Name	Unit	Comment	Subframe ID Counter name	Frame position	Frame interval	Word (1...n)	Word Interval	MSB pos (1...n)	Length	Encoding (BNR / UBNR / BCD / Float / Enum / CVSD / ASCII / ByteArray / MIL1750AFloat / DECDFloat)	Min	Max	Enumdef	Coef-ficient 0	Coef-ficient 1
Baro Altitude	ft			1		4		1	16	BNR				0	2

The first line represents the version number of the ICD format, while the second gives the meaning of each column. The third line defines a single parameter that is encoded in the raw data. Its name is "Baro Altitude" and it is measured in feet.

FLIDAS must be able to find it in the raw data, so we must describe its position. This parameter is the one that is marked in red in the screenshot of the major frame above. You will see that the parameter is encoded in the 4th word of every minor frame. Therefore, the "Word" is set to 4 and the "Frame position" is set to the first minor frame it is found in (1). Since it repeats every minor frame the minor frame interval is left empty (this is the default and same as 1, which means 1 minor frame later the parameter is repeated). Inside this word we need to define the position. This parameter covers the whole word, which means that the MSB is found in the leftmost bit with the position number 1 and it spans over 16 bits. Therefore, MSB is set to 1 and length to 16.

Note that PCM words may also be recorded LSB first. In this case it must be declared in TMATS already and FLIDAS will reverse it before using the above parameter decoding. I.e., MSB will always be left of LSB even if the word is LSB first.

Now that FLIDAS knows where to find the parameter, the interpretation must be defined. The 16 bits are encoded in a two's complement binary encoding. Therefore, the encoding is set to "BNR". A raw value of 1 of this parameter represents 2 feet, therefore the coefficient 1 is set to 2 and the coefficient 0 stays 0 as there is no offset in the calculation. *Note that you could add further columns "Coefficient 2, Coefficient 3 ..." if needed.*

This is enough to name, locate and interpret the first parameter. We will now add further parameters to show further options.

Look at the parameter that is marked green in the major frame screenshot above. The special thing is that it occurs twice in each minor frame (supercommutation).

Create a new line in the CSV file for this parameter. Try to fill it as much as possible with this information:

- The name shall be "True Air Speed" and it shall be measured in "kts"
- You will need an additional property of the parameter called "Word Interval". You will define the first word position inside the minor frame as above, but use the word interval to indicate after how many words the parameter will be repeated inside the minor frame.
- The parameter also fills the whole 16-bit word and is encoded as two's complement. A raw value of 1 shall indicate 0.1 kts.

After you added the information to the CSV, compare to the solution below:

- Name: True Air Speed
- Unit: kts
- Frame Position: 1
- Frame Interval: empty (or 1)
- Word: 2
- Word interval: 12
- MSB pos: 1
- Length: 16
- Encoding: BNR
- Coefficient 0: 0 (or empty)
- Coefficient 1: 0.1

If you are unsure about a setting, check the description in the [PCM CSV ICD description](#).

Look at the parameter that is marked blue in the major frame screenshot above. The special thing is that it occurs only every second minor frame (subcommutation).

Create a new line in the CSV file for this parameter. Try to fill it as much as possible with this information:

- The name shall be "Target Lock" without engineering unit.
- Since it only appears in every second minor frame you must use the frame interval setting to set this up.
- The parameter is located only in the rightmost bit of the 16-bit word. It is an enumeration that displays "Not locked" if the bit is not set and "Locked" if it is set. Check the [PCM CSV ICD description](#) to see how to achieve that. You will need the Encoding and Enumdef setting.

After you added the information to the CSV, compare to the solution below:

- Name: Target Lock

- Frame Position: 2
- Frame Interval: 2
- Word: 3
- MSB pos: 16
- Length: 1
- Encoding: Enum
- Enumdef: 0=Not Locked#1=Locked
- All coefficients: empty

If you are unsure about a setting, check the description in the [PCM CSV ICD description](#).


The sample files include the finished CSV ICD file which also contains more parameters than the above.

2.4.2 Creating an ICD Mapping

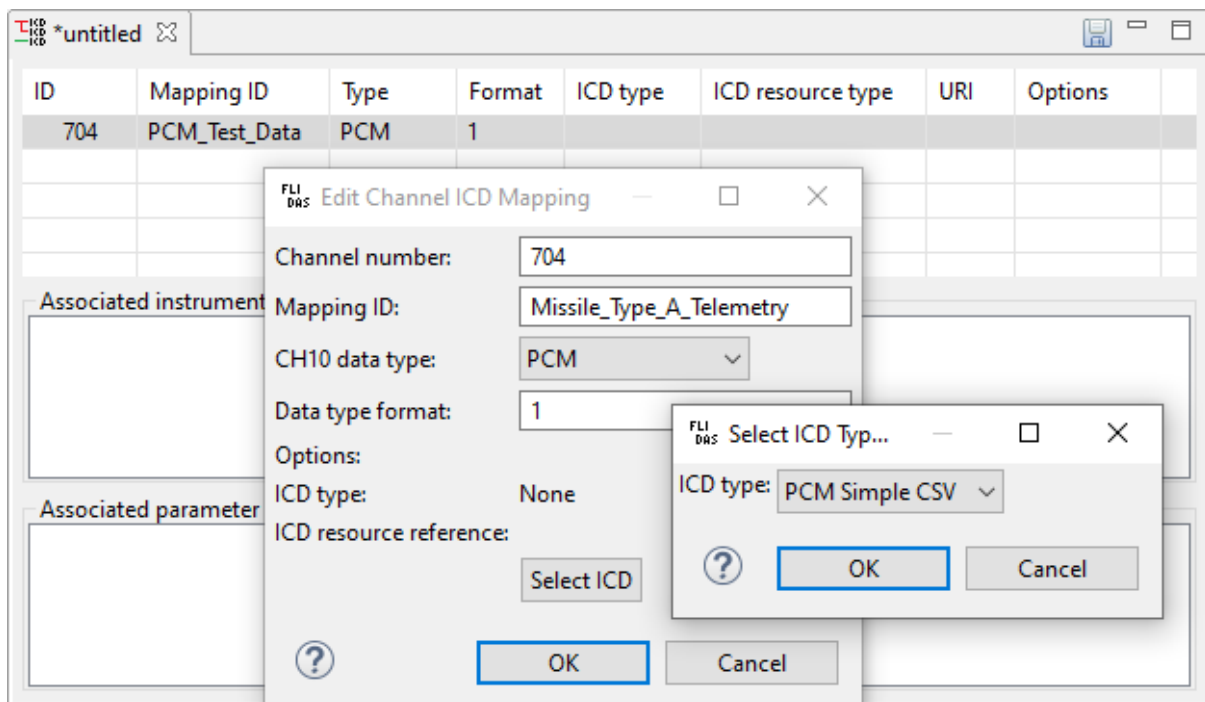
The next step is to map this ICD to a PCM channel with your ICD mapping.

Open the sample CH10 file, right-click it in the Repository View and select **Create ICD Mapping Template**.

The [ICD Mapping View](#) opens and shows an empty channel mapping for the PCM channel.

Save the new created file using the  symbol in the upper right of the view. Right now, save it in the same directory like your ICD file.

Double-click the PCM channel to edit the ICD mapping. Click at **Select ICD** to change the **ICD type** to **PCM Simple CSV**. Choose the ICD file you have created. Choose to store the ICD file name relative to the ICD mapping. This way you can move the ICD and ICD mapping file freely on your file system. If you have a fixed directory structure later for ICD files you might need to store the path absolute.



Enter "Missile_Type_A_Telemetry" as mapping ID for the PCM. This describes the source of the data on this channel. Whenever you store a selection of parameters (parameter set) these will store the mapping ID and the parameter name for each parameter. So, identical parameter names from different sources can be distinguished. The mapping ID may be any text but not contain "#".

Accept the changes with **OK** and close the ICD Mapping View. Save when asked for it.

A prepared ICD mapping file exists in the sample files.

2.4.3 Create Parameter Sets

Now that FLIDAS knows how to find and decode each parameter in the raw data, you need to define which data you are interested in. Such a selection of parameters is called a parameter set. It can be temporary just for a single display of data or it can be a meaningful group of parameters that will be used over and over in this combination. There could be a group of "Engine Parameters" and one of "Primary Flight Parameters" for example and you can also join several parameter sets for use. The parameter sets will store the name of the parameter and the mapping ID of the channel or subchannel.

To create parameter sets you first need to open the Parameter Plug-in Control View. This view controls most of the parameter-based actions. To open the view, click **Window > Show View > Parameter Plug-in Controls**.

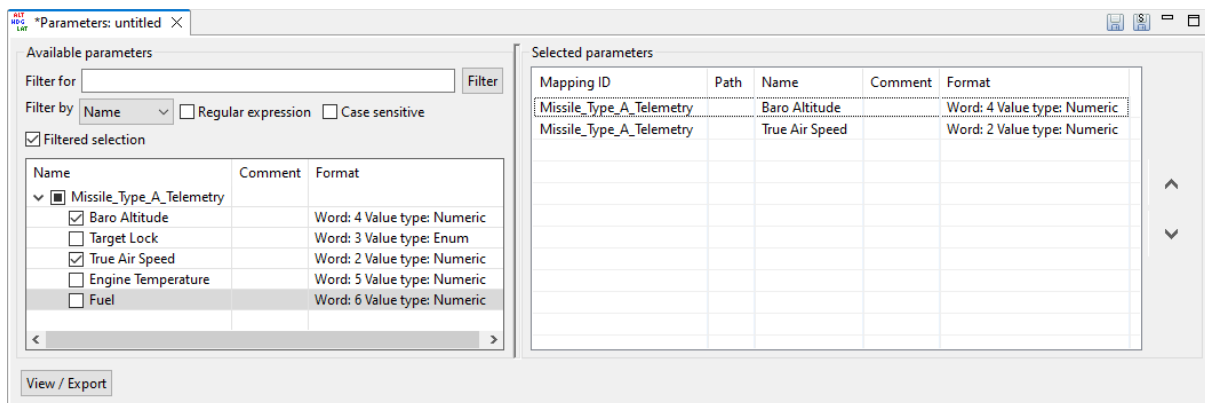
Working with parameter sets requires an activated ICD mapping since the parameter set only knows the name but not the definition of the parameters.

To activate the ICD mapping, first select it by clicking the **Select** button and choosing the ICD mapping file you have saved in the last step. It is then activated by clicking at **Activate**.

Note that you can pre-select the ICD mapping file for the next FLIDAS sessions using the Default Directories and Files Preferences.

With the activated ICD mapping you can now work with parameter sets. Create a new one with the **New** button on the lower end of the view.


The **Parameter Set View** comes up. It shows all available Parameters defined through the ICDs in your active ICD mapping and your selection of parameters.




Expand the mapping ID "Missile_Type_A_Telemetry" of your PCM channel in the **Available parameters** area to the left and select "Baro Altitude" and "True Air Speed" with the checkboxes.

You can now use this parameter set to decode the PCM channel to the GUI. Click at **View / Export > Parameter Viewers > View Parameter Table** on the lower end of the view. Select the **Asynchronous Full sampling mode** and finish the dialog.

Time	Baro Altitude	True Air Speed
6/25/19 12:00:00 AM 000:000:100	20	25.6
6/25/19 12:00:00 AM 050:000:100	20	27.2
6/25/19 12:00:00 AM 100:000:100	40	28.8
6/25/19 12:00:00 AM 150:000:100	40	30.4
6/25/19 12:00:00 AM 200:000:100	64	32
6/25/19 12:00:00 AM 250:000:100	64	33.6
6/25/19 12:00:00 AM 300:000:100	92	35.2
6/25/19 12:00:00 AM 350:000:100	92	36.8
6/25/19 12:00:00 AM 400:000:100	124	38.4
6/25/19 12:00:00 AM 450:000:100	124	40
6/25/19 12:00:00 AM 500:000:100	160	41.6
6/25/19 12:00:00 AM 550:000:100	160	43.2

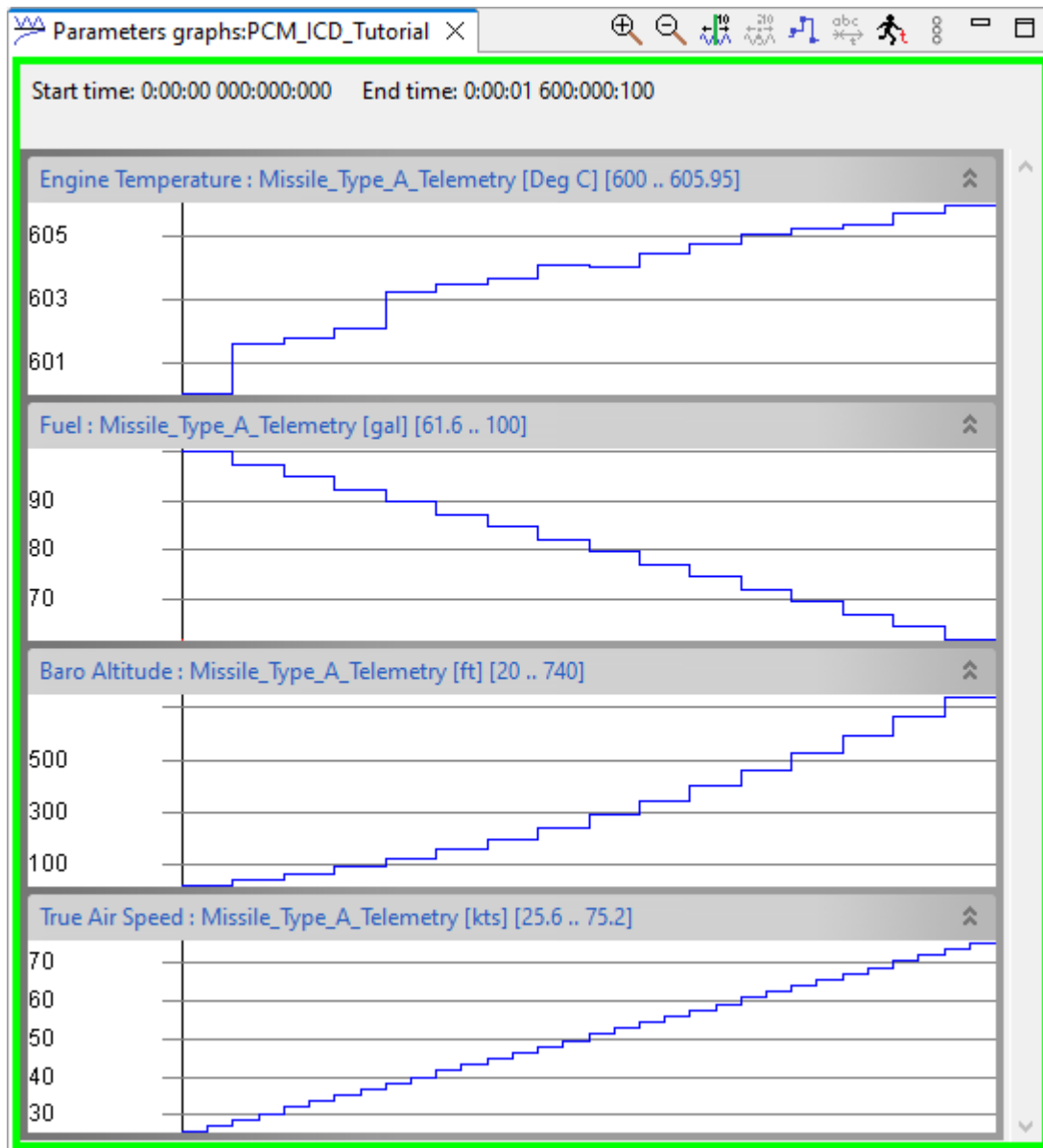
You have used a temporary parameter set to do this and this may be what you do most often. But you can also save this parameter set and reuse it. Go back to the Parameter Set View and click the  symbol in the upper right of the view. Save the file in the same directory like your other files and call it "primary_flight_parameters".

A parameter set will work with a specific combination of ICD mapping and ICD files, so you can associate it with the ICD mapping so you can quickly access the parameter set.

Go back to the Parameter Plug-in Controls View. Click at the **Edit** button for the ICD mapping. The ICD mapping will open. Right-click the area on the bottom labeled **Associated parameter sets** and select **Add**. Select your file and chose to store the path to the parameter set relative. Click the  symbol in the upper right of the view and close the ICD mapping. To see the changes, you must deactivate the ICD mapping and activate it again. In the Parameter Plug-in Controls View click **Deactivate**. A message will come up to tell you there are open parameter sets. You need to close them before deactivating because they depend on the current version of the ICD mapping. Close the parameter sets, deactivate the ICD mapping and then click **Activate** to load the new version. You will now see the parameter set that you associated with the ICD mapping in the Parameter Plug-in Control View in the associated parameter set area.

Now create another parameter set called "engine_parameters" and include the "engine temperature" and "fuel" from the included ICD sample file. If you use your self-created ICD file, use a text editor to copy the parameter definitions. Then associate the parameter set with the ICD mapping and activate the new ICD mapping.


Now you learn how to use the associated parameter sets. Show the Parameter Plug-in Control View. You should now see two associated parameter sets there. Left-click both while holding down the CTRL key to make a multi-selection. Right-click this group now and select **Parameter Viewers > Show Parameter Graphs**. On the sampling mode page use the **Asynchronous valid change sampling mode**.




2.4.4 Instrument Panels

Another important feature for working with parameters are instrument panels. You can use them to lay out gauges on a panel to visualize current parameter values. This means they either show incoming data of a live data stream or they show data of a CH10 file being replayed for the instrument panels.


In the Parameter Plug-in Control View click at **New** in the **Unassociated instrument panels** section on the bottom. A new view with an empty panel opens.

Instrument panels can only be manipulated in edit mode. When editing is disabled, they can only show current parameter values. Click at the  symbol on the top right of the view to enable editing.




To create a gauge, open the view menu by clicking at the three dots icon  at the top right of the view. Select **Create Dial**. A red box with a dial will appear. The red border means there is no parameter associated to display the values from.


To associate a parameter with the gauge, go back to the Parameter Plug-in Control View. Open the Parameter List View by clicking at **Show Parameters** on the bottom. Expand the "Missile_Type_A_Telemetry" entry to show its parameters. Move this view side by side with the instrument panel, drag the "True Air Speed" Parameter and drop it on the dial. The dial's border will turn black as the parameter is now associated with it.

To replay data from a file you need to create another gauge, a **Replay Control**. Do this and then drag it on the green bar and move it somewhere else on the panel so it does not cover the dial.

Before you can start the replay now you need to activate the instrument panel with a CH10 source. To do so click  on the top right of the view. If you have several CH10 files or live data streams open it will ask you which one to use, otherwise it just picks the only one available. The border of the view will now show the color of the CH10 file and the buttons of the Replay Control are now enabled.

Now disable the edit mode. You will see how the green bars on the gauges disappear. You can also replay data in edit mode if you want but the intent is to have a mode where you are safe from modification of your instrument panel.

To replay now you need to click the  button on the Replay Control. You should see the needle of the dial move. The data for replay is only 1.5 seconds so it will soon stop but you can use the  button to rewind to the start of the file. Since the replay is still running it will immediately run again unless you pause it with .

The instrument panel has been changed by you, so its title on the view says "*untitled". The "*" indicates the change. Use the  symbol on the top right of the view to save it.

Since the instrument panel visualizes parameters that are provided by your ICD mapping, you might associate it with the ICD mapping like you did with the parameter sets. This way you can easily open it every time you activate the ICD mapping. Edit the ICD mapping as you did before (click edit on top of the Parameter Plug-in Control View). Right-click in the area with **Associated instrument panels**, select **Add** and choose your file. Again, select to use a relative or absolute path to store like with the parameter sets. Save the ICD mapping. Deactivate the ICD mapping (you need to close open instrument panels and parameter sets as well), then re-activate it again. Now you should see your instrument panel in the **Associated instrument panels** area. You can quickly open it with a double-click.

To see the current replay time, create a clock gauge on the instrument panel. You can choose between different styles of time using the gauge menu. Right-click the green bar on the top of the clock and choose a style from the **Time Representation** submenu.

Now create a bar graph gauge and associate it with the baro altitude parameter. If you run the replay from start you will notice that the blue bar rises while the baro altitude value increases but soon it turns into a small red bar on top. This shows that the value exceeded the

maximum value configured for this gauge. Right-click the green bar on top of the bar graph and select **Set Min/Max > Value**. In the upcoming dialog you can configure which minimum and maximum value should be displayed before the display indicates the exceedance of the maximum value. Configure the maximum to 1000 and run the replay from the start of the file again.

If you use several instrument panels, the replay position of the file will be the same for all. It is enough to have one Replay Control on one instrument panel although you can also have several.

Remember that the Replay Control is only used for replaying files. If you use a live data stream the parameters of the live data stream will be displayed when the data comes in automatically. You can still have a Replay Control on your instrument panel so you can use the same instrument panel with live data streams and CH10 files.

2.4.5 Added flexibility by ICD mapping files

This section will explain how you can use the ICD mapping for flexibility in various situations. If you will always be working with simple tasks you can skip this or read it later.

Assume you have two identical equipment like two engines and both deliver data in the same format. You will only want to maintain one ICD for them but you need to connect this ICD to two channels that record the data from these two different sources. Using the mapping ID, you can assign names like "Left engine" and "Right engine". Parameter sets can identify which one you mean by keeping the parameter name and mapping ID.

Assume you have different recorders with different hardware and therefore different ICD mappings (if channel numbers or types differ). But still, you can assign the same mapping IDs "Left Engine" and "Right Engine" and can reuse your parameter sets and ICD files. *Note that the same ICD mapping can be used for a different hardware setup as well. It must only be identical in the mapped channel / subchannel numbers and types. Therefore, you might want to remove channels that you are not going to use from the ICD mapping or at least not map ICDs on them.*

Another thing that an ICD mapping will be useful for is to manage different versions of a unit under test. Let's assume you are testing an avionics software update. You have an ICD mapping that maps the ICDs of the old software and associates some parameter sets with it for different tests. There are still aircrafts flying with the old software so you want to keep the ICD mapping unchanged. You can copy the old ICD mapping and assign updated ICDs to the channels. You will keep the mapping IDs. If the parameter names in the updated ICDs stay the same, you can reuse the existing associated parameter sets.

ICD mappings can also refer to the ICDs and parameter sets by relative or absolute file names. This means you can also avoid modifying the ICD mapping from the last paragraph. If you have the ICD mapping in a directory and the ICDs and parameter sets in subdirectories, you could copy the whole directory and then update the ICD files of the copy to a new software version. If the ICD mapping refers to them by relative paths it will use the new ones automatically. You have different options here for different use cases. You could also refer to all (or some) ICDs relative and absolute to the parameter sets. This way the parameter sets will stay consistent over the avionics software versions.

2.4.6 How to continue learning

This tutorial ends here. There are a lot of other features to try out guided by the manual. The main points where to learn about the parameter features are the concepts and view descriptions:

- [Parameter concept](#)
- [Parameter set concept](#)
- [ICD concept](#)
- [ICD mapping concept](#)
- [Parameter sampling concept](#)
- [Parameter Plug-in Control View](#)
- [ICD Mapping View](#)
- [Parameter Set View](#)
- [Display of parameters as graphs](#)
- [How to organize parameters in a hierarchy within an ICD](#)
- [Instrument panels](#)
- There is a several hour sample file with real flight data and a prepared ICD and instrument panel at https://www.databustools.de/ch10_samples.html

2.5 Data Streaming Tutorial

This tutorial will show you how to work with data streams i.e., how to process data coming live from a network instead of a CH10 file. The tutorial expects that you are familiar with the basic use of FLIDAS as explained for example in the [basic tutorial](#).

All features described here are available in the [free version](#).

FLIDAS can do live processing of data coming in from the network or record it to CH10 files. It can also transmit CH10 files on the network to other applications or even transmit data within the same FLIDAS instance. All network transmission must be according to IRIG 106 CH10 data streaming with the UDP protocol on IP4. Formats 1 and 3 are supported.

While the loading of CH10 files just runs slow if your computer is not powerful, working with data streams might overload your computer because the data must be processed in the speed it comes in. Also, you must understand that the standard protocol used for data streaming is allowed to drop data so there are no guarantees on completeness. However, with some configuration and no overloading you should do well. Also, recorders are allowed to create delays of up to one second and FLIDAS will add another second if you request to synchronize the data between different channels. So, you should expect delays between real occurrences and the display in FLIDAS of one or two seconds depending on your channel synchronization choice.

The operating system you are running with FLIDAS and FLIDAS itself is not real-time capable. This means there are no guarantees that something happens exactly at a specific time. However, this should normally not be perceivable by just looking at displays.

The configuration of your operating system itself is out of scope here. Note that incoming and outgoing network traffic might need configurations on your local firewall and on your network settings if you must switch to a different subnet.

Should you have problems setting up the data streaming there is a section of the manual that helps you [analyzing data streaming problems](#).

Normally you will have an external device to transmit a data stream you are processing with FLIDAS. Within this tutorial we will use FLIDAS to transmit and receive the data so everyone can go through the tutorial even without a transmitting device. The sections covering the setup of the transmission in FLIDAS are written in *italics* so you can avoid executing the actions if you have a real device but should still read through it since you must do similar things with your transmitting device.

In this tutorial you will:

- Receive a live data stream from a CH10 file you transmit and validate proper reception
- Record a CH10 file from a CH10 file you transmit and look into the differences of both files
- See live statistics as you receive a CH10 file you transmit

2.5.1 Receive a Live Data Stream to Verify Proper Reception

First you must [create a live data stream](#). This will then be visible in the Repository View and is your starting point for most actions on the [live data stream](#).

To create the live data stream:

1. Go to the main menu and click **File > Create Live Data Stream**
2. Use the default values (IP = 127.0.0.1 and port = 4499) and click **Finish**

If you use an external device for transmitting then the configuration of the live data stream in FLIDAS must be:

- Leave the IP address empty unless your device uses IP multicast. Then enter the address.
- Configure the port address that your external device streams to
- If you use IP multicast, configure the network interface to use unless you want the operating system to choose it.
- If your hardware uses an UDP data streaming format of 3 and transmits several sources at the same time you must select a source ID on the CH10 data stream format wizard page
- On the TMATS configuration wizard page use an external TMATS file ([TMATS overriding](#)) describing your device configuration unless your device streams TMATS out on the data stream.

When you finish the live data stream configuration a window will pop up indicating it is waiting. The live data stream needs TMATS configuration before you can use it. Normally it expects this from the data stream. It will now keep on receiving until it detects TMATS on the live data stream. You can tell the pop-up window to **Run in Background** or if you want or keep it open move the window to the side.

Notice that you should now have a text **Active Streaming Tasks** with a green background in the bottom line. This reminds you about possible data stream tasks running in the background. There might be situations where you forget about them and they use up system resources without your attention.

Now you must transmit some data so the live data stream can receive something. If you have a real device, set it up to publish the data on the network. Make sure the publishing includes a time channel (within the selected source if you transmit a UDP data stream format 3). Otherwise FLIDAS cannot operate (unless you use secondary header timing). If your device does not periodically transmit TMATS information you must tell your device to transmit a TMATS setup at least one time now (or use a TMATS override file as described above).

For our testing purpose we will now publish a CH10 file with FLIDAS itself:

1. *Load any CH10 file you want.*
2. *Right-click the CH10 file in the Repository View and select **Publish Data Stream**.*
3. *Keep the configuration as it is (i.e., IP and port as in the live data stream configuration before) and finish the dialog.*

As soon as the data stream is published on the network and FLIDAS detects a TMATS setup, the live data stream will be initialized and become visible in the Repository View.

Now check for proper reception by looking at the General Live Data Stream Information Properties. Right-click the live data stream in the Repository View and select **Properties > General Live Data Stream Information**:

- There should be a number of **UDP packets processed**.
- The number of **UDP packets dropped** should be low compared to the previous number. With optimal configuration it should be close to 0. If you encounter problems you should look into the description of the Data Streaming Preferences to find additional configuration to fix the problem. Especially the **UDP receive buffer size** that you can also see in the current dialog is of critical importance.

Now you can close the live data stream. Right-click it in the Repository View and select **Close**. You cannot use the same live data stream for another reception once it has tuned into a data stream. You must create a new one if you need one.

*Possibly the publishing of your CH10 file is still running. If so, you will still see the green text **Active Streaming Tasks** at the bottom. You will see the data stream task in the Repository View when you click at the symbol in front of your CH10 file to expand the data stream tasks. This will be explained in the next chapter. Right now, right-click at the only (if any) data stream task and select **Stop**.*

2.5.2 Record a Data Stream

Now you will record a data stream:

1. Go to the main menu and select **File > Record Data Stream**
2. Select a file name for the data to record with the **Browse...** button

3. Click **Next** and configure the IP address and UDP port. You should use the same settings as in the previous chapter.
4. Click **Next** and configure the source ID if needed for your real hardware. You should use the same settings as in the previous chapter.
5. Click **Next** and make sure the option **Synchronize to TMATS and time** is on. This means the recording will skip any recorded data until it received TMATS data and a time Packet afterwards. In effect the created file will fulfill basic requirements for CH10 files. You can turn this option off to do a raw recording. You will be able to load it with FLIDAS but it is no valid CH10 file.
6. Finish the dialog

Now you should see the text **Active Streaming Tasks** with a green background in the bottom line. Open the Repository View. Normally you would see running data stream tasks organized below the CH10 sources when you click at the symbol in front of each CH10 source. Right now, there is no CH10 source because you are just recording data. Data stream tasks that are independent of a CH10 source like this are found below the item **Other active data stream tasks**. Currently you should only find one recording. So, the Repository View is the place where you can see tasks running in the background that take up system resources so you don't forget about them. Select the recording task, then right-click it and select **Show Status**. You will see statistics like those of the live data stream in the chapter before in the General Live Data Stream Information Properties. Two information in the last two lines are important now: **Waiting for TMATS: true** and **Waiting for first time packet: true**. Both indicate that the recording has not yet recorded any data because it has not seen TMATS data and time on the data stream.

As in the previous chapter you have the opportunity now to start transmitting of your real device (with TMATS and time)

For our testing purpose we can publish the same CH10 file as in the last chapter again. Repeat the steps from the last chapter to publish it.

In the Repository View look at the status of your recording by looking at its data stream task. Make sure it records something. After a while right-click the data stream task and stop it. If you don't do this at some point in time the created CH10 file will not be properly closed and any buffered data from the end might be missing.

Also stop the publishing of the CH10 file if it is still running. You will find the data stream task below its CH10 file.

Open the newly recorded CH10 file and show all packets (in Repository View right-click it and select **Show Packets**). Do the same for the original file and align both Packet Views side by side.

Assuming you had no UDP packet loss you should notice the following:

- Both CH10 files start with a TMATS packet followed by a time packet
- Your original file shows CH10 packets not in perfect time order. The CH10 standard allows recorders to wait up to a second before writing them to the file. This delay can be different on each channel. Also, when transmitting a data stream these delays are

allowed. In result an application receiving the data like FLIDAS must create additional delays to synchronize data of different channels.

- Since you published the file with FLIDAS, the CH10 packets have been transmitted in correct time order and you should see them ordered in the Packet View. If you use another transmitting device they may or may not be in time order. The specification only guarantees they never fall more than a second behind.
- On the recorded data you will see no CH10 packet before the first time and TMATS packet (neither in packet sequence nor in the time sequence). FLIDAS dropped all data before. Your original file may contain packets that are before the first time packet in time sequence (not in packet sequence).
- For every packet that is in time after the first time packet in the original file you should find a corresponding packet in the recorded file (up to the place where you stopped the recording)

What you should understand here is that if your data stream transmits TMATS only every 30 seconds and you start a recording you may lose up to 30 seconds of data. Also, if you have no TMATS or no time on the data stream you will not record at all. If you need to anyway you can use the configuration dialog when starting the recording and turn off the option to synchronize with TMATS and time. You will not get a valid CH10 file this way but can still open it with FLIDAS.

2.5.3 See Live Statistics from a Live Data Stream

Now you will learn to see live data from a live data stream. Opening the live processors you are interested in might take a couple of seconds but when we publish a recorded CH10 file with FLIDAS this is likely to have only a few seconds of data before the data stream is finished. To cope with this problem, we will create the live data stream not as in the first section of this tutorial by taking the TMATS information from the data stream. We will use an external TMATS file so we can work with an initialized live data stream before we start the publishing of the sample data. If you have a real device transmitting an unlimited data stream you can skip some of the specific preparations.

*First, we need an external TMATS file for the CH10 file you are going to publish. If you don't have one, right-click the CH10 file in the Repository View and select **Export > TMATS to Text File** and store the TMATS somewhere.*

Create a live data stream as in the first section of this tutorial. Only select the external TMATS file on the third page (unless you use a real device - in this case start it up let it publish and make TMATS available on the data stream).

The live data stream should now be immediately available in the Repository View because it doesn't need to wait for the TMATS information on the data stream.

The simplest way to see live data from a live data stream is to right-click it in the Repository View and select **Channel Activity Live Processor**. You will see a view which shows a line for each channel and some statistics about the data received so far. *To see some actual data, you must now publish the CH10 file as before. Of course, this must be the file that you provided the TMATS to the live data stream.* Once you start the publishing you should see some activity in the live processor.

Notice the colored border around the Channel Activity Live Processor. It has the color associated with the live data stream in the Repository View however the colored border is not complete as for CH10 files but has an uncolored area in the middle of each side. This distinguishes live data streams from CH10 files.

Open the Repository View and see that there is a data stream task associated with the Channel Activity Live Processor. Close the Channel Activity Live Processor by closing the view. You will see that the data stream task closes with it.

Open the Channel View of the live data stream from the Repository View with a double-click on the live data stream. It will show the same information as for CH10 files. Only the packet counts are limited to the count at the time you opened the Channel View. They will not update here.

Right-click any channel and select **Live Processor > Channel Activity**. In the upcoming dialog add some more channels and click **Finish**. A Channel Activity Live Processor just for the selected channels comes up. This is the regular way to start live processors. You can also start other types of live processors that are more specialized in specific types of data like the MPEG TS Replay Live Processor or the PCM Streams Live Processor but these are only available in the Data or Parameter Streaming Versions.

This tutorial ends here. Check out some more related topics guided by the manual:

- Live processors
- Parameter live processors
- Instrument panels

2.6 Features by Application Area

This section cross links you to specific features of FLIDAS that are useful for specific application areas. It does not point out general features that are needed everywhere. For some of these general features read the basic tutorial.

On the top level the features are separated by CH10 file or live data stream features. In some situations, you might be able to use both so check both sections.

2.6.1 CH10 File-Based Features

2.6.1.1 Features for CH10 File Debugging

When you are developing CH10 recorders or software yourself, have to validate CH10 soft- or hardware deliveries from suppliers, have to deal with corrupted files or encounter errors in your measurement chain, these features may be useful for you:

- When you are loading a CH10 file you can enable the data checksum checking and control how FLIDAS handles massively corrupted files.
- After loading a CH10 file, look up the General CH10 Source Information Properties for an error summary and look up the Channel View for the general recording configuration, detected TMATS problems and packet counts per channel.
- Use the CH10 Verification View ("CH10 validation") to check the file format for problems and directly inspect the detected problems in detail on the GUI.

-
- Use the [TMATS View](#) to see the uninterpreted TMATS data and search in the content.
 - Look up a list of all corrupted packets with the [Packet View](#) in the corrupted packets mode. You can also use the Packet View in its regular mode and jump from error to error.
 - Look into packets to see its decoded content in the [Packet Entry View](#).
 - Look up the [General Packet Properties](#) to see all the packet header information in raw and interpreted form including error information.
 - Use the [Hex Packet View](#) to see the whole packet's content or a packet entry in a structured hex editor style.
 - Use some of the lower level [exporters](#) to extract recorded raw data and compare it with your input.
 - Create new CH10 files by [filtering Channels and time slices](#) to concentrate on the data you are interested in, share it with others or send to Data Bus Tools for debugging.
 - Analyze problems with your time source and RTC using the [Time Communication Summary](#).
 - Use [communication summaries](#) to compare the recording of a single channel with the results you expect from a reference data source like a METS 231 from [Scientific Data Systems LLC](#)
 - Check the bit rates created by your recorder or individual channels using the [Channel View](#).
 - You can analyze and detect protocol errors in IRGI 106 Chapter 7 streams using the [CH7 Communication Summary](#)
 - You can look for times in the recording when the recorder stopped the recording using the [Recording Gap Viewer](#).

2.6.1.2 Protocol Level Analysis

When you are looking into general problems with your avionics system like broken cables, spurious parity errors, unresponsive LRUs etc. or you want to verify the protocol level behavior, these features may be useful for you:

- When you are [loading a CH10 file](#) you can suppress data types in the file that you are not interested in.
- After loading a CH10 file, look up the [Channel View](#) for the general recording configuration, detected TMATS problems and packet counts per channel.
- Use [communication summaries](#) to look into the different types of messages circulating on your avionics system, their counts, frequencies and error conditions. Use this to locate protocol error and special event positions in the file as well.
- Look into interpreted protocol level data at specific points in time with the [Packet Entry View](#). Use its features to jump from error condition to error condition within a group of packets. Specific features allow seeing [PCM minor frames](#) and [analog curves](#).
- Use [various types of exporters](#) to export protocol data and analyze it in external applications. This includes exporting several interfaces of the same data type time interleaved.
- [Define error bits of MIL-STD 1553B status words as parameters](#) so you can see occurrences of these errors in the [Parameter Table Viewer](#)
- You can look for times where one or more data sources stopped transmitting data using the [Recording Gap Viewer](#).

2.6.1.3 Parameter Level Analysis

When you are evaluating the actual performance of an aircraft and validating its behavior by analyzing parameters transmitted on the avionics system or collected by FTI equipment, these features may be useful for you:

- After loading a CH10 file, look up the [Channel View](#) for the general recording configuration.
- [Define an ICD](#) to describe how your parameters are decoded from the recorded raw data.
- Define which ICD to use on which channel by creating an [ICD mapping](#).
- Control which ICDs are active and start most parameter-based actions from the [Parameter Plug-in Control View](#).
- [Organize your parameters in parameter sets](#) to reuse these sets or joined sets for future exports and other actions.
- [Export parameter values](#) over a time range for use in other applications or view them directly in FLIDAS as [tables](#) or [graphs](#) or print them as [graphs](#).
- Define an [ICD for recording events](#) to correlate recording events with other decoded parameters in the same display or export functionality.

2.6.1.4 Data Presentation

If you require visual or audio presentation of data contained in CH10 files, these features may be useful for you:

- Use the [Navigation Data View](#) to replay NMEA 0183 navigation and basic flight data on a map and some instruments. Use the [User Map Preferences](#) to integrate maps of your own test range.
- [Export MPEG video data](#) to immediately see them in external applications.
- Replay [analog](#) or [CVSD](#) audio data directly in FLIDAS or export [analog](#) or [CVSD](#) audio to WAV files.
- See [analog curve forms](#) from one or several Packets.
- [Export](#) parameter values over a time range for display in other applications or see and print them in the FLIDAS [Parameter Graph Viewer](#).
- Use [instrument panels](#) to design pages containing [gauges](#) to visualize values of parameters that you replay from a CH10 file.

2.6.1.5 Test Data Generation

When you want to create test data to verify the operation of your CH10 software or recorders, you can prepare specific test data to check if your software works correctly on them or if your recorder can replay the data properly. To do so you could use real files and manipulate them to include very specific error cases that you might not be able to easily produce otherwise.

- Use the [Hex Packet View](#) to modify positions inside a CH10 packet and set error flags or manipulate data.
- The [XML CH10 mapping](#) converts between CH10 and XML files. You can convert real files to XML for manipulation or write XML files directly.

- You can use the [CH10 Verification View](#) to verify that your generated test data conforms with IRIG 106 chapter 10 to some extent or deviates from it in the form you intended.

2.6.1.6 Working with Distributed CH10 Multiplexer Sources

When you need to decode CH10 files that merge data from several [distributed CH10 multiplexer sources](#) to a non-valid CH10 file you may need special features to work with the data.

- You can [split up such a CH10 file](#) to several files separated by their multiplexer source IDs (MSIDs) to get files that align better with the standard and may be easier to process with FLIDAS and possibly other software as well. Note that FLIDAS covers a wider range of these files out-of-the-box.
- You can align time offsets of the different sources using the [time channel offsets wizard page](#).
- You can control if your files loaded into FLIDAS can use different time channels for each MSID and make use of secondary header timing during [CH10 file loading](#).

2.6.2 Live Data Stream-Based Features

2.6.2.1 Features for CH10 Stream Debugging

When you are developing CH10 recorders or software yourself, have to validate CH10 soft- or hardware deliveries from suppliers, have to deal with corrupted data or encounter errors in your measurement chain, these features may be useful for you:

- When you are [creating a live data stream](#) you can enable the data checksum checking.
- While receiving a live data stream, look up the [General CH10 Source Information Properties](#) and [General Live Data Stream Information Properties](#) for an error summary and look up the [Channel View](#) for the general recording configuration, detected TMATS problems and packet counts per channel.
- Use the [TMATS View](#) to see the uninterpreted TMATS data and search in the content.
- Look up live statistics, bit rates and errors of incoming data per channel with the [Channel Activity Live Processor](#).
- Look up the status and statistics of the data stream tasks in the [Repository View](#).
- Use the [Data Streaming Preferences](#) to configure the data stream transmission and reception in case of problems.
- [Record a data stream](#) to use the CH10 file-based debugging features (see above) on the file.
- [Publish existing CH10 files as data stream](#) and receive it with FLIDAS or an external software to verify proper reception.
- You can analyze and detect protocol errors in IRGI 106 Chapter 7 streams using the [CH7 Live Processor](#)

2.6.2.2 Protocol Level Analysis

When you are looking into general problems with your avionics system like broken cables, spurious parity errors, unresponsive LRUs etc., want to configure or calibrate your FTI

equipment or you want to verify the protocol level behavior, these features may be useful for you:

- When you are creating a live data stream you can suppress CH10 data types in the live data stream that you are not interested in.
- Look up the Channel View for the general recording configuration, detected TMATS problems and packet counts per channel.
- Use live processors to look into live data in specific displays for some CH10 data types that can be used to calibrate your sensors for example.
- Record a data stream to use the CH10 file-based protocol analysis features on the file.
- View live raw data from a PCM stream and graphically plot selected PCM words.
- Use Raw Data Live Processor to show raw data and protocol information of selected CH10 data types.

2.6.2.3 Parameter Level Analysis

When you are evaluating the actual performance of an aircraft and validating its behavior by analyzing parameters transmitted on the avionics system or collected by FTI equipment, these features may be useful for you:

- While receiving a live data stream, look up the Channel View for the general recording configuration.
- Define an ICD to describe how your parameters are decoded from the recorded raw data.
- Define which ICD to use on which channel by creating an ICD mapping.
- Control which ICDs are active and start most parameter-based actions from the Parameter Plug-in Control View.
- Organize your parameters in parameter sets to reuse these sets or joined sets for parameter live processors.
- See decoded values of parameters live.
- Define an ICD for recording events to correlate recording events with other decoded parameters in the same display functionality.

2.6.2.4 Data Presentation

If you require visual or audio presentation of data contained in CH10 files, these features may be useful for you:

- Use the MPEG TS Replay Live Processor to immediately see an MPEG video in an external application.
- Live replay of CVSD or analog audio.
- Record a data stream to use the CH10 file-based data representation features on the file.
- Use raw data live processors to show live strip charts of raw data of selected CH10 data types.
- Use instrument panels to design pages containing gauges to visualize current values of parameters.

2.6.2.5 Data Publishing

If you want to make live data available to other destinations, these features may be useful for you:

- Publish CH10 data as UDP stream from an incoming IRIG 106 chapter 7 PCM stream.
- Transmit TmNS messages (Telemetry Network Standard, IRIG 106 Chapter 24) from an incoming IRIG 106 chapter 7 PCM stream.
- Transmit live raw PCM or UART data as simple UDP stream for use in external applications.

2.6.2.6 Working with Distributed CH10 Multiplexer Sources

When you need to receive live data streams that merge data from several distributed CH10 multiplexer sources you may need special features to work with the data.

- You can align time offsets of the different sources using the time channel offsets wizard page.
- You can control if the stream processing in FLIDAS can use different time channels for each MSID and make use of secondary header timing during the creation of the live data steam.

3 Concepts

3.1 CH10 Sources

A *CH10 source* is a source for data in the IRIG 106 chapter 10 format. This source can either be recorded data from a CH10 file or live data coming in from a live data stream.

All CH10 sources will be visible in the Repository View from where you can take actions on them.

When you close a CH10 source, all views connected with it will be closed too.

CH10 sources use system resources so you should consider closing them when you don't need them anymore.

3.1.1 CH10 Files

In FLIDAS you are loading IRIG 106 chapter 10 files to analyze them or export their content. In short, we will call them *CH10 files*. Almost every aspect of the system depends on a specific file that has been loaded.

A CH10 file is one type of a CH10 source.

When you are loading a file, the system will read some of its TMATS attributes and collect some information about the contained packets. This is always necessary before you can work with the file. Some data may be skipped when reading, which depends on your CH10 data type filters configuration.

You can cancel the loading to show what data has been processed so far.

You can load several files at a time. Even the same file can be loaded several times with different configuration. Each file will have an assigned color that will be reused in other views to allow you distinguishing between the different files the views are based on.

Only limited data is collected when loading the file. So, actions like inspecting a packet's content or exporting data from it will access the file again.

When the loading of a file encounters errors, FLIDAS gives you the choice to use the data loaded so far. This allows working with corrupted files.

Loading a CH10 file may come to one of the following results:

- **OK:** The file has properly loaded. Never the less it may contain errors but these have been handled properly.
- **User abort:** You have manually cancelled the loading.
- **Corrupted packets:** During the load a corrupted packet was encountered (i.e., invalid packet sync or packet header checksum). Your current configuration tells FLIDAS to stop loading under this condition. You can configure FLIDAS to search for the next valid packet and continue loading from there with the CH10 file loading wizard.

- **Unexpected error threshold:** FLIDAS tries to handle most kinds of errors. There may be errors that haven't been anticipated during development. In this case FLIDAS logs these errors and continues processing the next packet. After reaching the maximum allowed number of these errors, FLIDAS gives up loading the file. The maximum error number can be configured in the CH10 file loading wizard.
- **File error:** There was an error accessing the CH10 file dealing with disk or network (if remotely stored) access.
- **Unrecoverable error:** FLIDAS encountered an error and tried to find the next valid packet within the file to continue processing from a safe position in the file. It was not able to find such a safe position because of follow up errors.

3.1.2 Live Data Streams

A *live data stream* is a stream of data coming in on a CH10 UDP data stream and used for live processing. The source can be a recorder replaying or streaming the data or any other device capable of creating an IRIG 106 chapter 10 UDP data stream.

A live data stream is one type of a CH10 source.

FLIDAS needs TMATS attributes for a live data stream to gain basic knowledge about the type of data. They can either come from an external file (TMATS overriding) or be taken from the data stream.

For FLIDAS live data streams must include a time channel and all incoming data will be skipped until TMATS and time has been initialized.

A live data stream is created from the file menu. Once it is initialized it will be visible in the Repository View from where you can take actions on it. You may run several live data streams in parallel on different UDP ports although this is not recommended.

A live data stream may show several states in the Repository View

- **OK:** The live data stream is ready for use.
- **Network error:** This state is reached if there was an error with the network receiving the data. In this case the live data stream is kept alive so you can still inspect the last values and statistics before the network error.
- **Not connected:** This state is reached if the network handling has stopped but the live data stream still exists for inspection of the last values and statistics. One example is if you manually stop the data stream task for this live data stream but do not close the live data stream itself.
- **Unexpected error threshold:** FLIDAS tries to handle most kinds of errors. There may be errors that haven't been anticipated during development. In this case FLIDAS logs these errors and continues processing the next packet if possible. After reaching the maximum allowed number of these errors, FLIDAS stops the data stream task of the live data stream but keeps the live data stream alive in this state for problem analysis.

A live data stream will have some general statistics collected over the whole lifetime of the live data stream but most live processors will just take into account the data coming in from the time on you start the function and not have access to past data.

3.2 Packets

When this manual refers to a packet it refers to the IRIG 106 chapter 10 definition of a packet. This means it is a collection of data retrieved from a channel. Most of the time this is data coming from another system where it has been monitored.

Packets may collect data of up to 100 ms. FLIDAS expects that packets from the same channel are stored in time sequence while packets of different channels may be out of order in the file.

FLIDAS offers a Packet View to see a list of packets with some properties associated with it.

Most packets decode into smaller units of data called packet entries. This would be the level of a message for most CH10 data types.

FLIDAS calculates the timestamps of packets relative to the time packets in the CH10 source. Details can be seen in the time section.

When processing packets, FLIDAS will check the sequence numbers in the packets. If within one channel a sequence number is missing, FLIDAS will insert a lost packets marker into the stream of packets. This marker will show information on the number of missing packets on the channel. The timestamp of the marker will be 100 nanoseconds before the packet that revealed the missing sequence number and will indicate the type of this packet. This means that missing computer-generated packets (TMATS, user defined data, recording indices / events) may not indicate the correct type as all types of computer-generated packets share the same sequence counter (if they are on Channel 0) and therefore it is not possible to know which type is missing.

3.3 Packet Entries

Packet entries is the common term used here for CH10 data type dependent data structures that are contained in a packet. This is the real data transmitted on a physical interface. For Ethernet this is for example a frame, for MIL-STD 1553 a message.

FLIDAS decodes packets into packet entries only on request. Simple processing of the CH10 source does not decode it and this is done for example when the packet content is viewed or the data is exported.

FLIDAS offers packet entry viewers to see the packet entries within one packet. The packet entry viewers are individual for each CH10 data type in all commercial versions.

3.4 Channels and Subchannels

The term channel is used here in the same sense as in the IRIG 106 standard. It is therefore a stream of data of the same CH10 data type. Some channels can be subdivided in subchannels. For some CH10 data types IRIG 106 uses different terms for subchannels like "bus" on ARINC 429 and "Network" on Ethernet. We use the term subchannel for all of them here.

Depending on the CH10 data type, some packets contain packet entries from different subchannels. It is therefore only possible to distinguish data on the subchannel level, when packets are decoded into packet entries.

FLIDAS shows channels and subchannels in the Channels View.

Note: If all computer-generated packets use channel ID 0 (as allowed until IRIG 106-11) then this channel 0 will be split artificially by FLIDAS to separate TMATS, user defined, recording index and recording event channels.

Note: FLIDAS will use subchannel numbers that start with 1 for all CH10 data types independent if the information comes from the CH10 recorded data section or the CH9 TMATS definition. To achieve this the places where numbering normally starts with 0 are increased by 1. See also the Subchannel Number Preferences that help achieving this.

Note: In general using subchannels reduces the performance. FLIDAS has indexed the CH10 packet headers and can filter very effectively for channels. Subchannels are only visible in the packet data, so any filtering based on subchannels requires loading and decoding the whole packet. This is not kept in memory and must be done every time the data is processed, even if the file is already loaded. It seems that at least some recorder manufacturers see the problem and try avoiding subchanneling at all. The concept is part of CH10 and cannot be avoided totally but it is possible to have separate channels with only one subchannel each instead of having one channel with many subchannels. If you have the option, you should prefer the one subchannel per channel approach for better performance with FLIDAS. For analog data you cannot avoid the subchannels.

3.5 CH10 Data Types

The term CH10 data type is used in the same sense as *data type* in the IRIG 106 standard. It is used to distinguish physically and logically different sources of input data like MIL-STD 1553, Ethernet or ARINC 429. It is also used on data that is not recorded from externally but generated inside the CH10 recorder like recording indices.

CH10 data types have different attributes of interest that are described in the CH10 data type support section of the manual. Not all CH10 data types defined in the IRIG 106 standard are fully supported yet.

3.6 TMATS Attributes

The TMATS attributes are the configuration of the system that created the CH10 source. They contain important information about the interpretation of the data.

FLIDAS uses some of these attributes and combines this with information directly extracted from the CH10 source data.

Relevant parts where the TMATS attributes can be seen are:

- General attributes in: General TMATS Attributes Properties
- Recorder-Reproducer Attributes in: Channel View
- Recording Events Definitions: Recording Event Definitions View

- [Channel and subchannel properties](#)

The full but uninterpreted TMATS attributes can be seen in the [TMATS View](#)

Sometimes your TMATS settings are not sufficient for proper FLIDAS operation. This may be because of a wrong setup of the TMATS data, because you use a rather old TMATS version or because FLIDAS requires optional attributes sometimes. FLIDAS offers three strategies to overcome this:

- You can provide the missing data for each CH10 source separately with temporary [TMATS overriding](#)
- You can provide the missing data for each CH10 file separately with permanent [TMATS replacement](#).
- You can configure some missing TMATS attributes with the same value for each file using the [TMATS Default Preferences](#)

Note: Often the need for the most solutions above is rooted in the use of old TMATS versions. Newer versions often allow or even require the attributes. If you have the possibility, upgrade to the most recent TMATS version you can use. Especially during the development of FLIDAS we discovered many problems in TMATS and reported them to the standards body which has corrected some of these in recent standard releases.

3.7 Recording Events

The term recording events is used in the sense of the IRIG 106 standard. Their meaning is defined in the TMATS attributes and their occurrence is logged within the recording event packets of the CH10 source.

Recording events always belong to a [channel](#).

The recording event definitions can be seen in the [Recording Event Definition View](#).

Recording event occurrences can be seen in the recording events [packet entry viewer](#).

Recording events can also be used to [define time ranges for data processing](#).

You can also define recording events as [parameters](#) and use them together with parameters of other sources in exports or graphical displays. You need to define a [recording events ICD](#) for this.

Note: IRIG 106 chapter 10 allows recording event packets to be at any position in the CH10 source, even at a position that does not relate to their packet's timestamp. For example, a recorder manufacturer may choose to store recording event packets at the end of the file (although we haven't seen this so far). In [data streaming](#) FLIDAS will use the packet's timestamp for publishing a data stream (i.e., replay it at the time of the packet and not at the end if the packet is in the end of the file) but will not be able to process the recording events in time if a real hardware publishes the packet at the end of the data stream.

3.8 Time Markers

A time marker represents a point in time within a recording or live data stream that is of specific interest for the user. Each time marker can have a note on it.

Time markers can be managed in the Time Marker Properties when you right-click a CH10 Source in the Repository View and select **Properties > Time Markers**.

You can create time markers on most places where you see or use a timestamp. For live data streams you can usually apply an offset to the time stamp to e.g. mark the event you have seen two seconds earlier.

Time markers can be used for example to select time slices for exports or to jump to the point in time for example in some Packet Views or the Parameter Graph Viewer. You can also export them for use in other software.

For CH10 files, time markers can be saved in a property file. You need to enable this when you load the CH10 file or by explicitly saving it in the Repository View. If you enable it on loading the time markers you saved in an earlier session will be made available.

Time markers are associated with a CH10 source. If you close the source or close FLIDAS and have created time markers, FLIDAS will:

- for CH10 files save them with the file properties if enabled
- for live data streams offer to export them to a file. You can then use them for example in another application. The export is the same as when done from the Time Marker Properties

3.9 Exporters

An exporter is a FLIDAS feature that converts parts of the data contained in a CH10 file to another format. The other format may be a file or also a direct output (like replay of contained audio data).

Exporters most of the time focus on a specific CH10 data type and allow selecting a combination of channels or subchannels and a time slice.

In FLIDAS there are three types of exporters that differ on how the input data is defined:

- **CH10 data type exporters** just need one or more CH10 channels or subchannels
- **Encapsulated protocol exporters** need additional user input on how to unpack data from a protocol encapsulated within regular CH10 channel or subchannel data. An example would be an exporter that accesses UDP packets. These are encapsulated in regular CH10 Ethernet packets. To select the right UDP packets a user selection of MAC / IP addresses and port numbers is needed.
- **Parameter exporters** need a set of parameters as input. These already carry the information where to find the source data.

The source and target format of the exporter are sometimes not able to contain exactly the same information, so a loss of information might occur (for example, loss of timestamp precision, ambiguous mapping of error classes etc.). This is a result of the format specifications and not the conversion algorithm.

Exporting may be a time-consuming process since the data is not already in memory but retrieved from the source file while exporting.

FLIDAS can form the base for customer specific export formats. If you require a specific conversion, write an email to info@ databustools.de or go to www.databustools.de for further contact information

Depending on the type exporters are started from the Repository View by right-clicking the file and selecting **Export** or in a similar fashion from the Channel View or a communication summary. Parameter exporters are started from the Parameter Plug-in Control View or Parameter Set View.

3.10 Packet Entry Viewers

A packet entry viewer is used inside the Packet Entry View to display the content of packets in a specialized way for the CH10 data type.

The individual packet entry viewers are described in their reference chapter.

3.11 Protocol Viewers

A protocol viewer is a mechanism to display higher-level information about some protocol. The protocol may be one of the CH10 data types or even a higher-level protocol contained in these CH10 data types. They look at a wider scope than a single packet.

An example is interpreting an UART subchannel as NMEA 0183 protocol containing navigation data. Other examples are communication summaries for a channel or subchannel.

Protocol viewers are invoked on the Channel View or within communication summaries. They may need some time to process, since they are not focusing on the packet level and therefore must load the whole data in order to display. You can continue working while loading. Most protocol viewers will allow canceling the loading and will then show the data processed so far instead of the whole time span's data.

3.12 CH10 Data Type Filters

A CH10 data type filter is a mechanism to filter certain CH10 data types out of a CH10 source. CH10 data type filters are always applied when a CH10 source is opened and cannot be changed afterwards.

The configuration is done in the CH10 Data Type Filters Wizard Page.

3.13 Preferences

Preferences are globally active settings for FLIDAS that apply to all CH10 sources and the overall handling of the tool.

Preferences are subdivided in different categories.

To change preferences, open the preference dialog by clicking **Window > Preferences**.

Some preferences, like the table layout of most views are directly changed there.

Preferences can be ex- and imported as well in the preferences dialog by using the buttons on the lower left of the dialog. This will transfer the wizard settings as well.

3.14 Properties

Properties are attributes associated with a specific object.

Right now, properties are used for CH10 sources, channels and subchannels and packets.

Properties are subdivided in different categories.


Properties are accessed by right-clicking the object and selecting **Properties** from its context menu. This works for example on the CH10 sources within the Repository View, for channels and subchannels in the Channel View and for packets in the Packet View.

3.15 Views

View is the term that is used for a window with a specific behavior. The term is taken from the Eclipse framework that is used by FLIDAS.

For a detailed discussion of views, you can look up the Internet and read about Eclipse views. However, some basic ways of arranging the views on the screen should be explained here:

- Double-click the views title to resize it to fill the whole window. Double-click it again to size it normal
- Drag the title bar next to the title bar of another view (like the Repository View). It will then be stacked with the other view.
- Drag the title bar to another position inside the main window. The area will be split between this view and other existing views. Some indicators will show where it will be placed
- Drag the line between two views to resize them

Also notice that some views have icons to their top right to execute further standard or individual actions and even might have a view pull-down menu for further options invoked by clicking at the three dots icon  at the top right of the view.

Views that relate to a specific CH10 source have a colored border around it that can be configured individually for each CH10 source in the General CH10 Source Information Properties. The colored border for CH10 files completely surrounds the view while it is broken in the middle of each side for live data streams.

To make better use of your screen in a multi-monitor configuration, you can also detach views from the main window. To do so, simply drag them outside. You can re-attach them when you drag them inside the main window and place them in a stack with other views. Note that you need to drag the **Tab** of the view and not the detached window's title bar.

3.16 Wizards

A wizard is a set of configuration pages that you need to set up to execute an action. The term is taken from the Eclipse framework that is used by FLIDAS.

Wizards appear for example when you start an exporter, protocol viewer or live processor. All of them offer similar processing and share some of their configuration pages.

The reference section for Wizards describes the available Wizard Pages.

3.17 Error Logging

Whenever an error occurs it will be written to the error log. Some of them will also pop up to inform you. "Errors" are categorized in errors, warnings and plain information.

To show the Error Log View select **Window > Show View > Error Log**. You can double-click on an entry to get more information.

If you need the error log as a file (e.g., to send it in with an error report), you will find it below your installation directory in the subdirectory **workspace/.metadata**. The file name is **.log**. The files and subdirectories may appear hidden on some operating systems.

3.18 ICDs

ICD stands for *interface control document*. This type of document describes information going into and coming from a system. Most avionic systems have an ICD that describes the parameters circulating on their digital data buses.

In FLIDAS you need to define an ICD to be able to interpret recorded raw data as engineering units. These ICDs may have different formats depending on the type of data. It is possible to request support for company internal ICD formats which allows for unique handling and saves you from entering and maintaining additional ICDs.

Currently there are file-based and hard coded ICDs. Other types like data base related are possible for the future.

ICDs are intended to be created outside of FLIDAS (like in Excel or with your already existing company internal tools).

ICDs don't need to be complete. You can also just enter a few parameters that you need and extend them over time.

ICDs are used by FLIDAS with the ICD mapping.

Parameters within ICDs can be organized in hierarchical groups to keep a better overview. For example, MIL-STD 1553 parameters could be grouped by their source RT address and then finer grouped by the messages within this RT. This hierarchy is completely for organizational purposes and carries no further information used by FLIDAS other than a name and a parent-child relationship between the container and the contained parameters.

Therefore, you can also organize by other means like engine and navigation parameters or completely leave out this type of organization. However, this hierarchy will help you keeping an overview when selecting parameters within the [Parameter Set View](#). The hierarchy of the names of the groups will form a path to the parameter within this ICD. This path will be stored within [parameter sets](#). Therefore, changing the names in the hierarchy will corrupt parameter sets.

The reference chapter for [ICD formats](#) lists the available ICD types.

ICDs are only supported in the [Parameter or Parameter Streaming Version](#).

3.19 Parameters

A parameter is a single signal that can take different values over time. Parameters are decoded from the CH10 raw data to an engineering unit level.

Parameters need some information to be defined:

1. General information: Name and engineering unit
2. Addressing: Where can the raw data be found within a communication line? These can be things like word and bit positions and length, ARINC 429 label and SDI, MIL-STD 1553 RT and subaddress etc.
3. Decoding: How is the raw data converted to engineering units? These are things like data type (BCD, two's complement ...), scaling, enumeration definition etc.
4. Channel / subchannel mapping: Which electrical line the parameter is found on?

In FLIDAS the first three groups of information are defined in [ICDs](#). The channel / subchannel mappings are defined by the [ICD mapping](#) to reuse ICDs with CH10 sources with different channel / subchannel setup.

Parameters from an ICD cannot be used directly. Parameters are grouped to [parameter sets](#) for use.

Parameters are only supported in the [Parameter or Parameter Streaming Version](#).

Numeric parameter values are internally stored as 64-bit floating point values (IEEE 754). This means for integral values using more than 53 bits the value will not be exact but only correct in magnitude.

3.20 Parameter Sets

Parameter sets are groups of [parameters](#) for a specific test / analysis situation. You can for example create sets for primary flight information and status information on the status of the electrical system.

Other functions of FLIDAS allow using these parameter sets for example as a selection of parameters to display. Joining of parameter sets and temporary modification are supported.

Parameter sets depend on an active [ICD mapping](#) to provide the parameter definitions through the connected [ICDs](#). The parameter sets store the parameter name, path and a

"mapping ID" defined in the ICD mapping and use these to identify the parameter definition. The use of the mapping ID allows reusing stored parameter sets over changing ICD mappings. If any parameter of a parameter set cannot be found through the ICD mapping it is considered an "unlinked parameter". These are indicated as a warning.

Parameter sets are edited in the [Parameter Set View](#).

Obviously working with larger parameter sets will impact the performance when decoding the data from the CH10 source. You should therefore consider making them only as big as needed. The separation into separate parameter sets that can be joined on request should help you with the management of smaller sets. In general, adding a parameter of a previously unused channel will decrease the performance more than adding an additional parameter from an already used channel (subchannels don't matter). All used parameters increase the memory footprint equally if you are working with [parameter viewers](#) (not [parameter exporters](#)). If you run into memory problems the [FAQ](#) offers a solution.

Parameter sets are only supported in the [Parameter or Parameter Streaming Version](#).

3.21 ICD Mapping

An ICD mapping associates [ICDs](#) with [channel and subchannel](#) numbers. In effect this defines which [parameters](#) are available and where to find them in a CH10 source.

ICD mappings are intended to be used over a broader range of CH10 sources. This works as long as they record the same physical communication lines on the same channels and subchannels. ICD mappings need to reference to ICDs with the data you want to decode to engineering values, so a variation in channel and subchannel numbers on unused data is acceptable.

An ICD mapping is therefore independent of a CH10 source but can be created using a concrete CH10 source as template.

You can only select one ICD mapping at any time and this can also be auto selected via the [Default Directories and Files Preferences](#). If you have several CH10 sources that use the same mapping you can still work with them in parallel.

An ICD mapping can have an active or inactive state. Only the active state loads the parameters from the ICDs. Therefore, most functions are only available with an active ICD mapping. Especially [parameter sets](#) rely on an active ICD mapping.

ICD mappings also store links to parameter sets ("associated parameter set") that can be used with this ICD mapping. These are just shortcuts and unassociated parameter sets can still be used.

ICD mappings are only supported in the [Parameter or Parameter Streaming Version](#).

You can only map one ICD on a channel ID (but you can map different ones on different subchannel IDs of the same channel). This means especially:

- For channel 0 you must decide if you want to decode user defined computer-generated data or recording events if both are available. Note that this can only happen in IRIG 106-11 or older. Since 2013 both need separate channels anyway.
- On a PCM channel you cannot map two different ICDs e.g. a CSV and a TMATS ICD.

3.22 Parameter Viewers

A parameter viewer is a mechanism to display engineering values of parameters over time.

Parameter viewers are invoked from the Parameter Plug-in Control View or Parameter Set View. They may need some time to process but you can continue working while loading. Most parameter viewers will allow canceling the loading and will then show the data processed so far instead of the whole time span's data.

Parameter viewers are only supported in the Parameter or Parameter Streaming Version.

3.23 Parameter Sampling

When parameter's values are collected over a period of time, this is called sampling.

3.23.1 Sampling Modes

FLIDAS supports seven modes of sampling that can be used to generate data for parameter viewers and Exporters. Not all are available everywhere.

Time discrete sampling will collect a parameter's value at equidistant points in time (e.g., continuously take a value every second).

Asynchronous sampling will use the points in time where parameters were really recorded for sampling. If several parameters are sampled together, all parameters will be sampled at the time when any of the parameters was recorded. This mode has three sub modes that can be used to reduce the amount of data:

- **Asynchronous full:** This mode collects all the parameter values that are stored in the CH10 source.
- **Asynchronous change:** This mode limits the asynchronous full mode to the points in time where any of the parameters in the parameter set changed its value. I.e., static values that are just recorded several times will not be sampled again.
- **Asynchronous valid change:** This mode further limits the asynchronous change mode by only looking at the points in time where a valid value changed. E.g., if one value changes from the invalid value "not measured" to "BCD coding error" it will not be sampled again. This also indicates that the user is not interested in error details so he will simply be shown that the value is invalid.

Note: Asynchronous sampling can lose data in special rare situations:

- *If your packet entries do not carry time stamps. UART and TSPI data has the option to not carry time stamps (you might be able to enable time stamping in your hardware*

though). If they don't all packet entries within one CH10 packet (up to 100 ms) will use the same time stamp.

- If a packet entry may contain data from multiple "messages". The only supported case here would be UART. For example a UART packet entry does not know what a "message" is and may record multiple "messages" in the same packet entry. If an ICD splits it into "messages" for parameter decoding all will share the time stamp of the packet entry. A case for this can be NMEA0183. Your hardware might allow to start a new packet entry on any transmission gap. This together with enabled time stamps will help prevent the problem.

All parameters decoded in the above two situations will have the same time stamp. But at this time stamp the asynchronous sampling will process all data from this time stamp. If all parameters are only contained in one of the packet entries or "messages" there is no problem. Any parameter contained in multiple of the packet entries or "messages" at the same time stamp will only take the value of the final state.

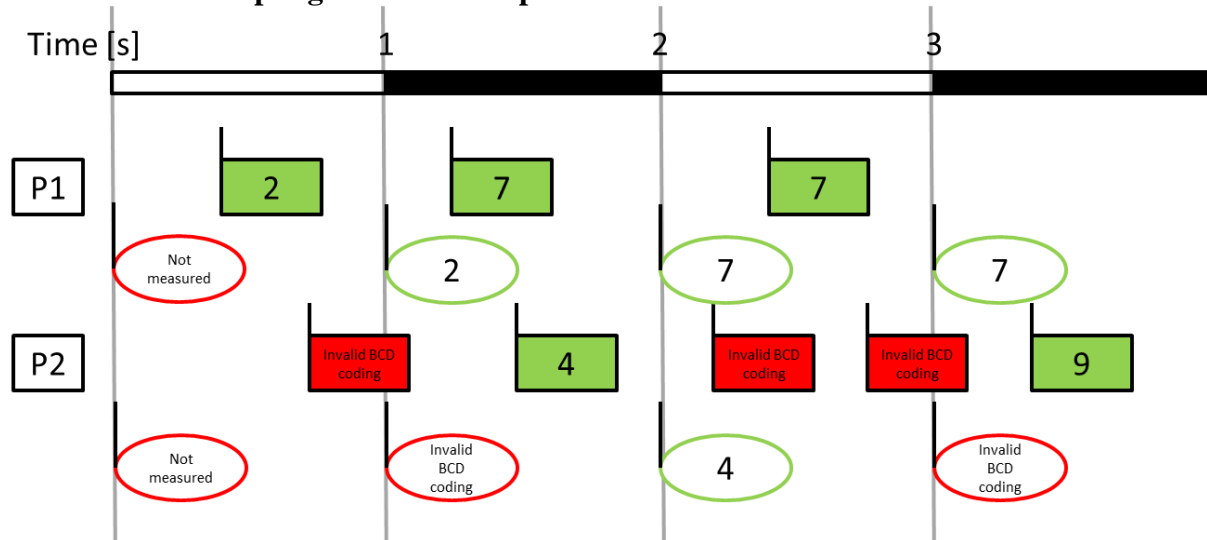
Separate timestamp asynchronous sampling is identical to asynchronous sampling except that each parameter will sample independently and will not share its timestamps with the others. I.e., if parameter A is sampled, no sample for parameter B is automatically taken at the same time. This mode has the same three sub modes as asynchronous sampling only that sampling occurs independently on each parameter:

- **Separate timestamps full**
- **Separate timestamps change**
- **Separate timestamps valid change**

The potential data loss for asynchronous sampling described above will not happen here. Instead multiple samples at the same time stamp will occur.

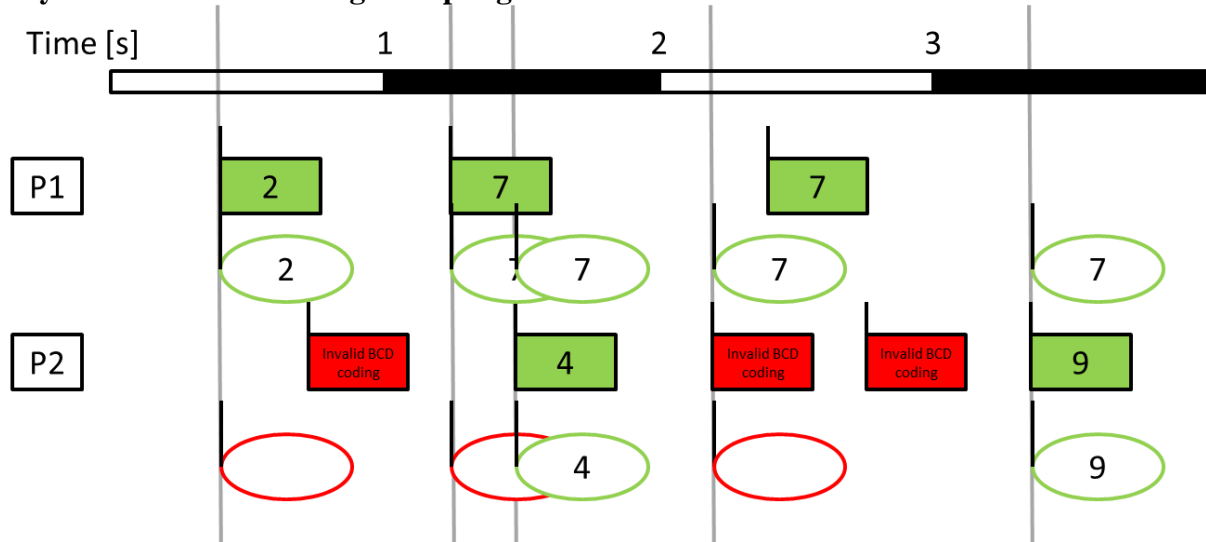
The following pictures visualize the sampling modes with two parameters P1 and P2. The boxes indicate recorded measurements while the ovals show samples taken by FLIDAS at the time of the vertical grey line.

Time discrete sampling with 1000 ms period

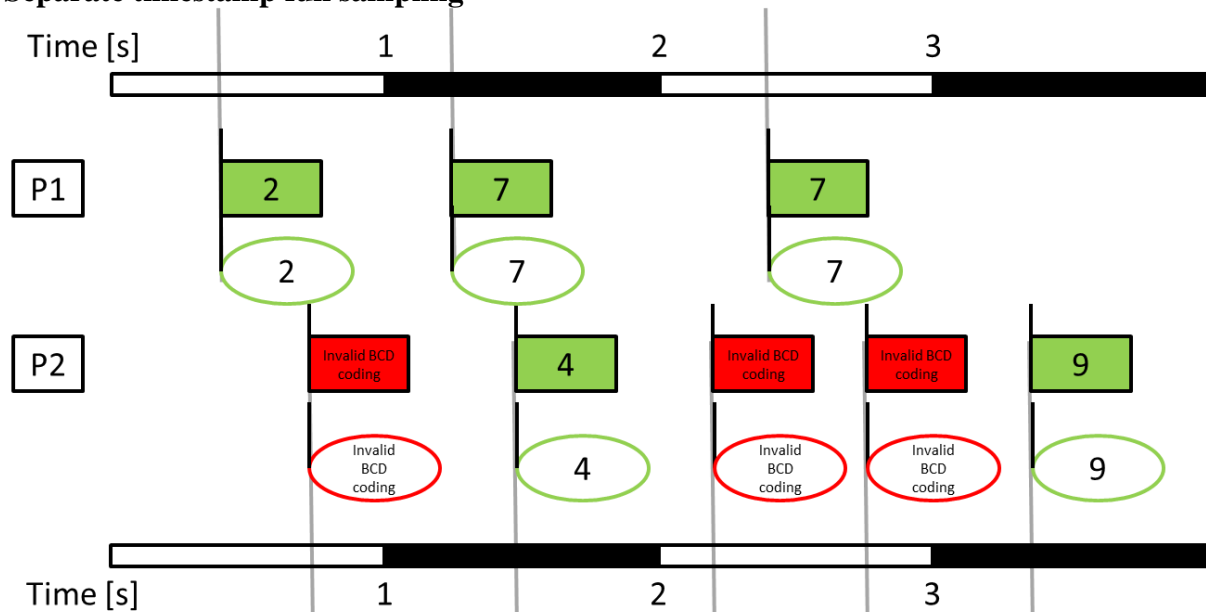


Asynchronous full sampling**Asynchronous change sampling**

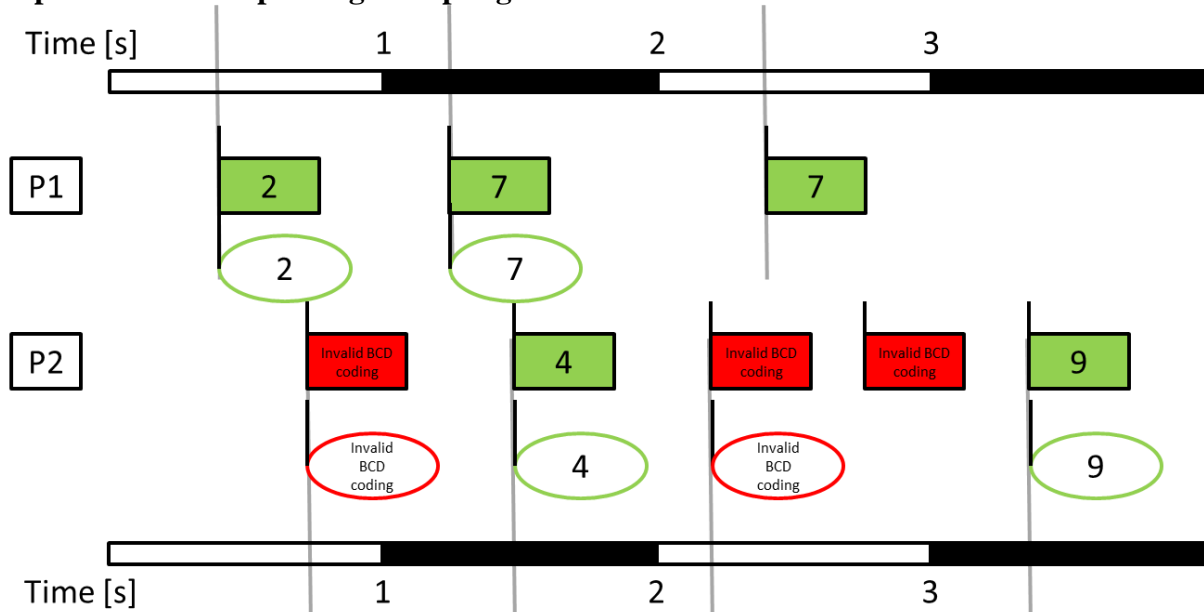
Compared to asynchronous full sampling two sampling points in time are missing at 2.4 and 2.7 seconds. The first one is missing because P1 is not changing but stays at 7 and the second one is missing because P2 doesn't change and stays at an invalid BCD coding.

Asynchronous valid change sampling

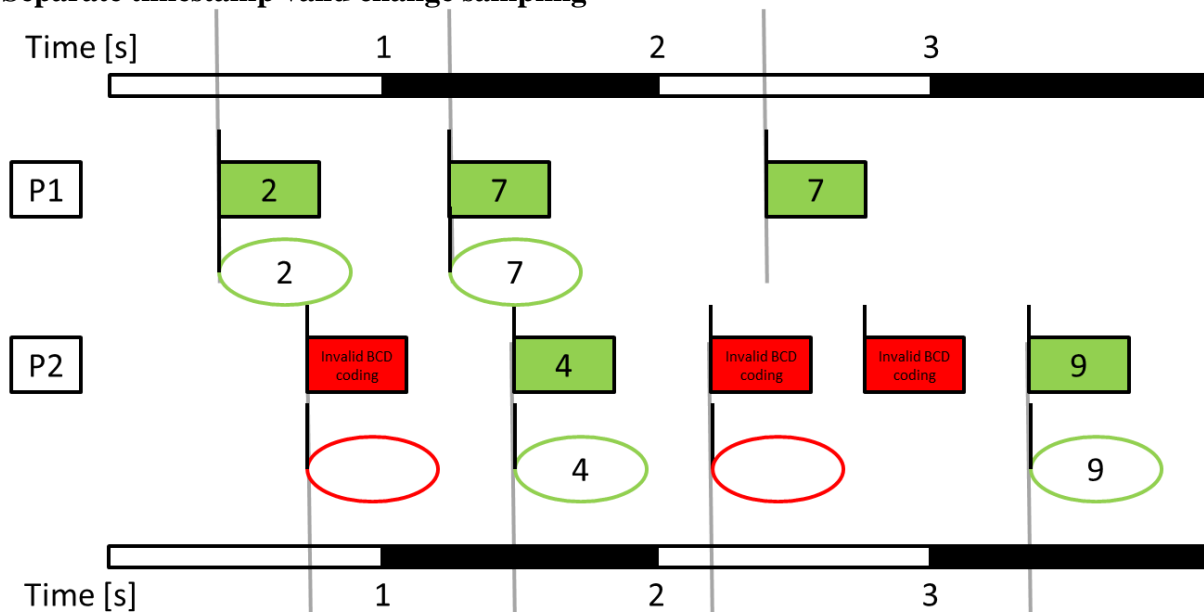
Compared to asynchronous change sampling one sampling point in time is missing at 0.7 seconds because P2 is changing from "not measured" to "invalid BCD coding" but all invalid values are considered equal no matter why they are invalid in this mode.

Separate timestamp full sampling

Compared to asynchronous full sampling you see that a second independent time scale appears on the bottom. Sampling of the P1 parameter doesn't cause P2 to be sampled at the same time anymore.

Separate timestamp change sampling

Compared to separate timestamp full sampling the sampling of P1 at 2.4 seconds and P2 at 2.7 seconds is missing because the value stayed the same.

Separate timestamp valid change sampling

Compared to separate timestamp change sampling only the error descriptions of the invalid values on P2 are missing because you are not interested in them here. The third invalid value at 2.7 seconds has already been suppressed during the separate timestamp change sampling mode because it was the same error as directly before. If it would have been different it would have persisted in the previous sampling mode but would have been removed by this sampling mode.

3.23.2 Initialization Values

When starting parameter sampling the first set of values (initialization values) is found by looking back up to one second. Values that do not appear during this time (because they are probably only transmitted at a low frequency or asynchronously) will be indicated as "not measured".

Collecting initial values will behave differently depending on the sampling mode and if you are constraining the start time of the sampling or sample from the beginning of the CH10 source.

Sampling mode	Start time constrained	Start time is begin of recording
Time discrete	Collect initial values from one second before start until start.	Collect initial values from start for a time span smaller than the sampling period up to a first "nice" timestamp (e.g., when starting at a time of 0,123456789 seconds and a sampling period of 1 second, it will collect initial values until second 1).
Asynchronous		No initial value collection. The first result of the sampling process will be from the first timestamp where any of the parameters was first recorded.
Separate timestamp sampling		No initial value collection. The first result of the sampling process will be from the first timestamp where each parameter was first recorded.

3.24 Data Streaming

Data Streaming refers to the reception or transmission of IRIG 106 chapter 10 data over a network. In FLIDAS only UDP data streaming with IP4 is supported.

There are two reception and one transmission mode for data streaming supported:

- **Publish data stream**: Publishing a data stream with FLIDAS means transmitting an existing CH10 file close to real-time on a network.
- **Record data stream**: Recording a data stream with FLIDAS will create a file with the recorded CH10 packets.
- **Live data stream**: A live data stream provides the received data for live processing and display.

Data streaming offers a different set of features compared to working with CH10 files. Data must be processed while it is coming in and will not be available later. In consequence features like protocol viewers, parameter viewers and exporters are not available and live processors are used instead. Also, all features to inspect recorded packets and packet entries are not available as the packets only exist for a short amount of time.

3.24.1 Data Stream Tasks

A data stream task is a process that works in the background of FLIDAS in parallel to your regular work. All data stream reception and transmission modes described above are executed by a data stream task. In addition, live processors process the incoming data for your actual

testing business and each use a data stream task for this. Since work done in the background without your actual attention may overload your system FLIDAS will make these data stream tasks visible in the Repository View. You can use this to see if anything unnecessary is left running taking up system resources and to explicitly stop these tasks. For example, this is the only place to stop a recording or to stop the publishing of a data stream before the published CH10 file is finished.

Most data stream tasks also define a way to check if they are overloaded. When detected this will lead to a red marker on the bottom of your FLIDAS window.

3.24.2 Real-time and Reliability

FLIDAS runs on non-real-time operating systems. It uses the Java programming language which also has no real-time capabilities. In effect real-time in the sense of "an action is guaranteed to take place within 50 milliseconds" is not achievable (this applies to all software running on regular Windows or Linux versions). However, it is still possible to display data and perceive it as "live" if you don't overload your system. Detected overloads will be indicated and you can also check statistics about the different data stream tasks to see where problems are.

Since FLIDAS runs separate data stream tasks in parallel a multi-core computer system is better suited to offer near-real-time performance.

The IRIG 106 chapter 10 data streaming with UDP in general is not reliable. The UDP protocol does not give any guarantees that UDP packets are delivered to the target so the receiver must always be aware that data may be lost. This applies to any software using UDP (also see IRIG 106 chapter 10.9.3.1 which explicitly states this). FLIDAS allows you to inspect packet loss and other statistics.

The IRIG 106 chapter 10 standard allows recorders to transmit data on the data stream with up to one second delay (and the delay may be different for each channel). FLIDAS doesn't know which Channel comes in with which delay (and it may vary) so FLIDAS buffers all incoming data for another second to see if some data arrives later that must be processed earlier. There is also an option to skip this reordering if you don't need the time correlation between different channels.

Transmitting data over a network like Ethernet also gives no guarantees on the delay caused and this delay may also be different for each unit of data. Also, UDP packets may arrive out of sequence and need reordering. This adds to the delay for the display of the data.

If you directly connect the data transmitter to a regular desktop PC running FLIDAS you should however be able to receive data streams of several hundred MBit and for example display a live video and process several 10 MBit PCM throughput streams. The delay when you reorder the packets to time sequence should be close to two seconds and if you skip reordering around one second. These are no guarantees though since it depends on your system, your network and the type of your data. Use the free FLIDAS version or request an evaluation to get more insight into this if you are in doubt.

3.24.3 IRIG 106 CH10 Data Streaming Format Support

There are three data streaming formats defined in IRIG 106 CH10. Formats 1 and 3 are supported while 2 is not (this is a proprietary format of a single recorder vendor that has only been included in IRIG 106 when this use became publicly known years later. If you need support for this, contact us).

The format 3 has been introduced in IRIG 106-2017 and there are no known devices producing this format at the time of the FLIDAS implementation. Tests have been committed with it internally in FLIDAS. There are some assumptions made for the implementation:

- At the receiving IP address only one format will occur while FLIDAS is receiving data
- If the receiving IP address receives a format 3 the "source ID length" will be constant while FLIDAS is receiving data
- Within one reception of data by FLIDAS the transmitting device either knows about the packet positions within the data stream (offset to packet start ≥ 8 or $= 0$) or it doesn't (offset to packet start $= 1$). The latter is like a PCM throughput mode. Devices will not switch between both.

The format 3 introduces the concept of a "source ID" without placing further limitations or descriptions on this. FLIDAS will only filter out one source when receiving or transmit as one source.

3.25 Live Processors

A live processor processes data from a live data stream and continuously updates the results on the screen.

An example would be to replay a video or to display a summary of incoming PCM streams.

Live processors are invoked on the Channel View. Once they are running, they use up system resources so close their views if they are not needed anymore or stop their data stream task in the Repository View.

The processed data is always limited to the time span since the live processor was started.

The chapter live processors contains the descriptions for all available live processors.

3.26 Parameter Live Processors

A parameter live processor processes data from a live data stream, decodes it to engineering values and continuously updates the results on the screen (or to the speaker). Parameter live processors require the Parameter Streaming Version.

An example parameter live processor would be a table that shows current values of selected decoded parameters.

The displayed values are always limited to the time span since the parameter has been selected for display or may contain the most recent value before this if the same parameter is already processed by another parameter live processor.

Parameter live processors are invoked on a parameter set from the Parameter Set View or the Parameter Plug-in Control View or from the Repository View on a live data stream. Once they are running, they use up system resources so close their views if they are not needed anymore or stop their data stream task in the Repository View.

Parameter live processors may allow the removal or adding of further parameters while they are running.

Parameter live processors share a common data stream task called "live parameter provider" that calculates the current values of the parameters used in any parameter live processor.

The chapter parameter live processors contains the descriptions for all available parameter live processors.

Note: Parameter live processors will not adapt properly if you switch to a different ICD mapping while they are running.

3.27 Instrument Panels

An instrument panel is a view that arranges user configurable gauges to display current parameter values of a live data stream or of a CH10 file being replayed for the instrument panels.

Instrument panels depend on an active ICD mapping to provide the parameter definitions through the connected ICDs. The instrument panels store the parameter name, path and a "mapping ID" defined in the ICD mapping and use these to identify the parameter definition. The use of the mapping ID allows reusing stored instrument panels over changing ICD mappings. If any parameter of an instrument panel cannot be found through the ICD mapping it is considered an "unlinked parameter". The problem is visible on the instrument panel.

The operation of instrument panels is explained in the Instrument Panel View.

Instrument panels can only be used in the Parameter or Parameter Streaming Version.

3.28 Distributed CH10 Multiplexer Sources

Sometimes there is a need to join data from different CH10 sources in a single CH10 file. Each source is already producing valid CH10 data (e.g., has TMATS information, a time channel and a free running RTC (real time counter)). When data from these different sources is merged in a single CH10 file, it becomes an invalid CH10 file (at least there is now data from multiple free-running RTC, there will be issues how to join the TMATS information, possibly duplicated channel IDs etc.).

Situations like this occur more and more frequently since CH10 systems become more available and are used in many locations in a single test. An example could be where a ground recording is done in parallel to PCM data coming from an airborne data acquisition unit (DAU). This DAU produces CH10 data itself and streams these CH10 packets to the ground recorder using the CH7 protocol on PCM. The ground recorder extracts the CH10 data from the PCM stream and inserts the complete CH10 packets into the recording file together with its own CH10 packets from local recording. The result is a CH10 file from two

different CH10 sources with independent RTC and time packets and data merged together. This is not a valid CH10 file (at least so far (2022 standard)).

Different FTI companies come up with different solutions to these problems besides the standard. Often the solution involves splitting the channels from the independent CH10 systems into groups. The groups are distinguished by the "Multiplexer Source ID" = MSID. This is taken from the upper bits of the channel ID. How many bits are used is defined in the TMATS attribute R-x\NSB (number of source bits). You can see this in the [General TMATS Attributes Properties](#). The channels with the same MSID use the same time channel and share the RTC. However, CH10 does not define what this MSID is and how it is used. This is just how it is applied by some FTI companies. There may also be channels without a valid RTC (possibly using secondary header time), more or all MSID groups may share an RTC without a mechanism to define the relation etc. Other issues come up with the TMATS information of the different systems and if they are merged into one R-Group, multiple R-groups exist or even multiple TMATS channels exist in the CH10 file.

Although the sections above talked about CH10 files, similar problems exist for CH10 streams.

Over time multiple different approaches to handle these distributed systems were individually invented. Feel free to contact the FLIDAS support to explain your usage and see if more support for this can be delivered. This includes users of the free version.

3.28.1 Currently Supported Distributed Architectures

FLIDAS supports working with this type of non CH10 conformant files or live data streams:

- Sources using RTC timing are distinguished by MSIDs and have their own time channel with valid RTC
- Sources with absolute secondary header timing are allowed on any channel and any MSID and don't require valid RTC or time channels. Acceptable absolute secondary header timing is IEEE 1588 or Chapter 4 (chapter 4 is not properly defined. Assumption is that time stamp contains time of year).
- For live data streams all data for an MSID using RTC timing will be dropped until a valid time packet arrived.
- For files there must be at least one valid time packet for each MSID using RTC timing
- TMATS information on Channel 0 describes the system setup for all sources. Alternatively, you could provide your own complete TMATS using [TMATS overriding](#)
- CH10 packets are committed to the stream or file within 1 second (measured in time rather than RTC counts)
- If the time channels use different time systems like UTC, TAI, GPS or local time, users must [provide information on time correction](#) when opening the CH10 file or live data stream. TMATS and CH10 does not provide enough information about the different time channels in all situations
- On live data streams all data is streamed to a single UDP port

Future FLIDAS versions will try to relax some of these requirements. If you have special requirements, you could send in sample files and discuss your use case.

Even if the conditions are not satisfied FLIDAS tries to work with the data but might run into problems:

- Files with missing time channels will still show their packets and have some kind of time stamp on them
- Data channels not declared in TMATS will be detected when packets appear
- If different time systems are used without configured time correction, files will show the right time stamps in each system. However, FLIDAS will not know they appear on the same point in time. Timing problems in various areas like the publishing of CH10 files or replay on instrument panels are to be expected. Live streams will not properly synchronize the data and probably "old" data will be processed immediately when it comes in while "new" data is synchronized. Live graphs which show e.g., the most recent 10 seconds of data will not show any data if it always comes in 18 seconds late.

3.28.2 Tips and Tricks

In practical situations like merging CH10 data from a CH7 stream into a file with another source, there might be a lot of corruption due to bit errors on the telemetry. Some parts of the packet header are more likely to be corrupted. If the data type of the packet changes for example, you could protect from this using the [CH10 data type filters](#). Often bit errors in the data type field lead to an "Unknown packet" type. If you filter these out when loading a file or creating a live data stream you prevent this problem. Of course, it gives more protection to filter out more data types that your system does not use. Bit errors in the RTC will likely cause an error that shows as an "RTC jump detected".

If you want to use RTC timing although secondary header time is available, you can disable secondary header timing in the [CH10 file loading](#) or [creation of a live data stream](#).

If you do not want to use MSIDs you can disable it in the [CH10 file loading](#) or [creation of a live data stream](#). This declares all channels to be part of MSID 0 and share one time channel. You might want to do this if you have multiple sources but they somehow synchronize the RTC between each other and just have one time channel.

The use of the "RTC sync error" in the CH10 packet header is not clearly defined in CH10. FLIDAS assumes it can also be used if no valid RTC is placed in the CH10 packet header since secondary header time is available. In these situations, the RTC sync error is not treated as an error that makes the packet invalid.

A simple way to improve operability with a non-conformant CH10 file is to split it to files with just one MSID each for better processing using the [MSID Channel Splitter Exporter](#).

3.28.3 Ongoing Discussions in IRIG 106 Standardization Committees

Currently the extension of CH10 for these use cases is under discussion in the RCC and TSICC. There is one working document on this which describes possible implementations. There are still open details. For those familiar with this document (version by end of April 2021) here the FLIDAS support for the issues in general:

- Time synchronization cases T1 to T4 (all). If different MSIDs share an RTC and time channel there is no way of knowing this so each MSID must repeat the time packets on a channel within its MSID.
- Channel ID overlapping cases I1 to I3 (all)
- Configuration merging cases C1 and C2 (C3 and C4 missing)

4 Reference

4.1 Help System

You can access the help system with **Help > Display Help**.

Many FLIDAS elements offer context sensitive help. To show it:

1. Activate the element you want help for (i.e., click it).
2. Push a key on the keyboard that brings up help. The key depends on the operating system and is:
 - Windows: **F1**
 - Linux: **Shift+F1**
 - Mac: **Help**

Alternatively, some dialogs have a question mark button in the lower left to activate help.

In some cases, the links inside the manual link you to a general topic and the information you are looking for might be in one of the subchapters. So, remember to look up the current section in the table of contents and look out for subchapters (the help system has an icon in the toolbar to the top right for this).

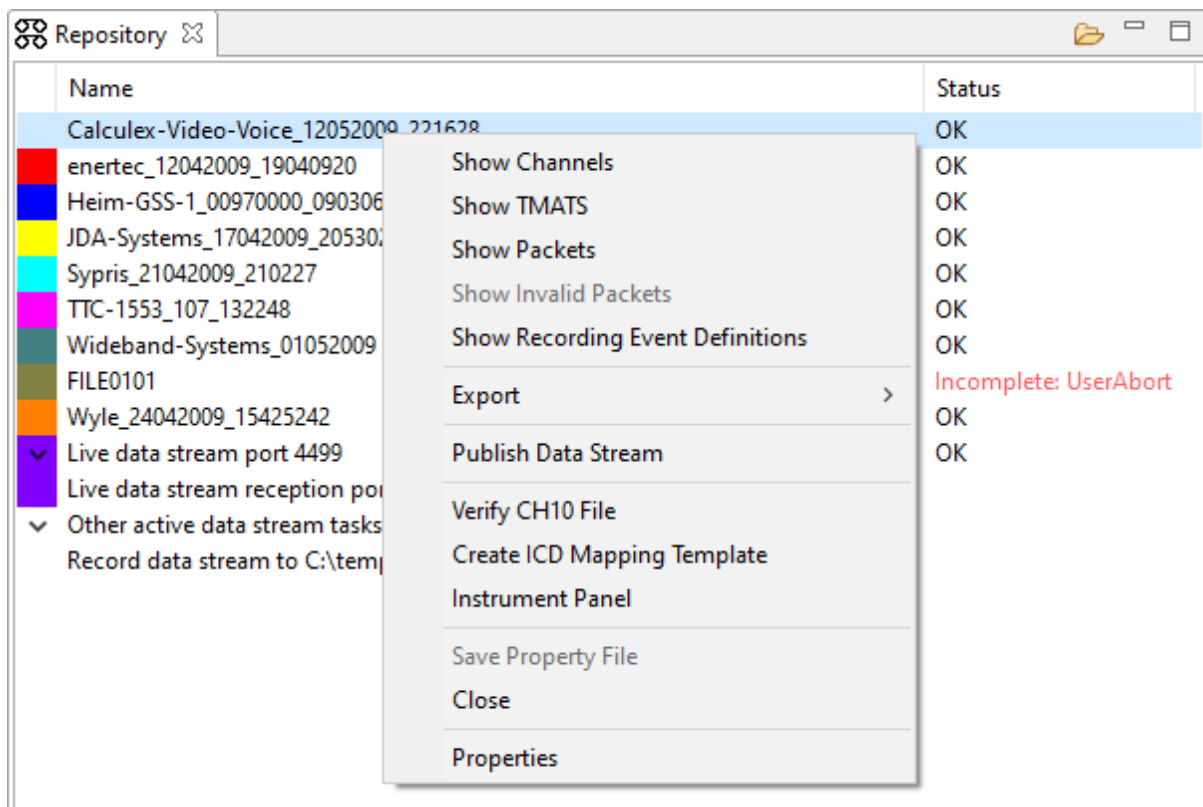
The PDF version of the manual is generated from the online help so both have the same contents.

4.2 Views

4.2.1 Repository View

The Repository View shows all CH10 files and live data streams that are currently available and their assigned color. It also shows the data stream tasks currently processed for one of these CH10 sources. It is the place where most actions on a CH10 source or data stream task can be invoked.

The Repository View opens automatically when you have finished loading a new CH10 file or initialized a live data stream. You can also bring it up by clicking **Window > Show View > Repository**.



Each CH10 source shows a color in the Repository View. This color is reused in other views that deal with data of this CH10 source.

The **Status** column shows the result of loading the CH10 file or the status of the live data stream. This should normally be **OK** but could also indicate an incomplete load in red color. The specific results are described in the [CH10 files](#) or [live data streams](#) section.

If you right-click a CH10 source, you will see several actions to invoke on the CH10 source. See the links below for a closer description:

- **Show Channels:** Shows the [channels and subchannels](#) of the CH10 source with some TMATS properties. A double-click on the CH10 source has the same effect.
- **Show TMATS:** Shows the raw TMATS of the CH10 source
- **Show Recording Event Definitions:** Shows the [recording event](#) definitions from TMATS
- **Create ICD Mapping Template:** This menu entry is only available in the [Parameter or Parameter Streaming Version](#). It creates an ICD mapping based on the channels and subchannels of the selected CH10 source.
- **Instrument Panel:** Opens an instrument panel that is already connected to this CH10 source. An [ICD mapping](#) must be active already (only available in the [Parameter or Parameter Streaming Version](#)).
- **Close:** Closes the CH10 source and all associated views.
- **Properties:** Shows the
 - [CH10 Data Type Filter Properties](#)
 - [General TMATS Attributes Properties](#)
 - [General CH10 Source Information Properties](#) (also allows choosing the color for the CH10 source)

- Time Marker Properties
- either General File Information Properties
- or General Live Data Stream Information Properties

For CH10 files you will find additional actions:

- **Show Packets**: Shows all packets from a CH10 file.
- **Show Invalid Packets**: Shows all packets from the CH10 file that had an error on packet level. This feature is not available in the free version.
- **Export**: Exports various data to other file formats.
- **Publish Data Stream**: Transmits the CH10 file as a data stream on the network.
- **Verify CH10 File**: This menu entry is only available in all commercial versions. It does some checks on the conformance of the CH10 file with the IRIG 106 standard.
- **Save Property File**: Saves the property file containing time markers. If you did not define a property file name when loading the CH10 file it will use the default filename and location (CH10 file name extended by ".flidas" in the same directory). The saving is only possible if there are changes to save. To use the properties file the next time you load the CH10 file you need to enable it on loading.

For live data streams you will find additional actions:

- **Channel Activity Live Processor**: Shows live statistics on activity on a live data stream for each channel.
- **Parameter Table Live Processor**: Shows current values of parameters received on a live data stream (only available in the Parameter Streaming Version).
- **Parameter Graph Live Processor**: Shows graphs of parameters received on a live data stream (only available in the Parameter Streaming Version).

The yellow icon in the Repository View's toolbar opens a new CH10 file.

4.2.1.1 Active Data Stream Tasks

Some CH10 sources may have an active data stream task connected to it. In this case you can expand the tasks by clicking at the symbol before the CH10 source.

You can request the status of one of the data stream tasks when you right-click it and select **Show Status**. This will give you some information about the configuration of this task like IP and port addresses and often information about the pending amount of work. This can be used to identify overloads. In general, every task will have some work pending from time to time. This is only a problem if the number of pending elements is constantly increasing. The information displayed shows the situation when you request the status. It will not be updated until you request a new status.

You can stop any of the data stream tasks from executing by right-clicking it and selecting **Stop**. For some of them, like the recording, this is the proper way to end them, for others there are other ways. However, this is a way to close data stream tasks in case they stop reacting properly although this situation has not been encountered during testing.

Whenever there is an active data stream task you will also see the text **Active streaming tasks** at the bottom of the FLIDAS main window. This is an **overload indicator**. When an

overload is detected, the background is red, otherwise green. The checking for an overload depends on the amount of pending data for processing or the delay of the data for processing to the stream itself. If a specific amount really indicates an overload heavily depends on your computer environment and the type of data stream you receive. If for example your data stream sends 10 CH10 packets within a millisecond it is acceptable for a task to have 100 packets waiting for processing. If you only receive 10 CH10 packets per second 100 waiting packets would be unacceptable. Within the [Data Streaming Preferences](#) there is an option to automatically stop overloaded data streaming tasks. Use this with care.

There are some data stream tasks that are not connected to a CH10 source. These are displayed after a common item named **Other active data stream tasks**.

4.2.2 Packet View

The Packet View shows the [packets](#) within a CH10 file in the order that they appeared there together with some properties.

To show the regular Packet View right-click the CH10 file in the [Repository View](#) and select **Show Packets**.

There is a second mode of the Packet View which just shows the packets of one [channel](#). This mode is accessed in one of the following ways:

- Right-click a packet in the Packet View and select **Show Channel's Packets**
- Double-click a channel in the [Channel View](#)

A third mode of the Packet View just shows corrupted packets. This mode is accessed from the [Repository View](#) with a right-click on the CH10 file by selecting **Show Invalid Packets** (not in free version).

The colored border around the view matches the color in the Repository View.

Time	Chan #	Type	Errors	Data
4/7/09 9:03:05 AM 961:262:900	87	MIL-STD-1553 Data, Fmt 1		Messages: 26 Time tag reference: First bit of the first word of the me...
4/7/09 9:03:05 AM 961:262:800	88	MIL-STD-1553 Data, Fmt 1		Messages: 26 Time tag reference: First bit of the first word of the me...
4/7/09 9:03:05 AM 961:262:900	89	MIL-STD-1553 Data, Fmt 1		Messages: 26 Time tag reference: First bit of the first word of the me...
4/7/09 9:03:05 AM 961:262:800	90	MIL-STD-1553 Data, Fmt 1		Messages: 26 Time tag reference: First bit of the first word of the me...
4/7/09 9:03:05 AM 961:262:800	91	MIL-STD-1553 Data, Fmt 1		Messages: 26 Time tag reference: First bit of the first word of the me...
4/7/09 9:03:05 AM 961:262:800	92	MIL-STD-1553 Data, Fmt 1		Messages: 26 Time tag re
4/7/09 9:03:05 AM 961:262:800	93	MIL-STD-1553 Data, Fmt 1		Messages: 26 Time tag re
4/7/09 9:03:05 AM 961:262:800	94	MIL-STD-1553 Data, Fmt 1		Messages: 26 Time tag re
4/7/09 9:03:05 AM 861:152:200	62	Analog Data, Fmt 1		#Subchannels: 1 Packing
4/7/09 9:03:05 AM 943:032:200	61	Analog Data, Fmt 1		#Subchannels: 1 Packing
4/7/09 9:03:05 AM 953:702:600	55	PCM Data, Fmt 1		Mode: Packed 16-bit align
4/7/09 9:03:05 AM 953:702:600	56	PCM Data, Fmt 1		Mode: Unpacked 16-bit a
4/7/09 9:03:05 AM 967:715:000	51	PCM Data, Fmt 1		Mode: Throughput 16-bi
4/7/09 9:03:05 AM 970:303:400	52	PCM Data, Fmt 1		Mode: Throughput 16-bi
4/7/09 9:03:05 AM 975:465:200	53	PCM Data, Fmt 1		Mode: Throughput 16-bi
4/7/09 9:03:05 AM 993:702:600	54	PCM Data, Fmt 1		Mode: Throughput 16-bi
4/7/09 9:03:06 AM 001:227:102	96	Ethernet Data, Fmt 0		Frames: 72 Format: Ether
4/7/09 9:03:05 AM 735:179:000	43	Video Data, Fmt 0 (MPEG-2/H...		Little endian Payload: MI
4/7/09 9:03:05 AM 977:418:700	95	Ethernet Data, Fmt 0		Frames: 44 Format: Ether
4/7/09 9:03:06 AM 019:982:833	87	MIL-STD-1553 Data, Fmt 1		Messages: 25 Time tag re
4/7/09 9:03:06 AM 019:982:733	88	MIL-STD-1553 Data, Fmt 1		Messages: 25 Time tag reference: First bit of the first word of the me...

Show Packet Entries
Show Channel's Packets
Show Within All Packets
Show in Hex
Export Table Content to CSV
Copy Table Cell to Clipboard
Use Monospace Font
Create Time Marker
Channel Properties
Packet Properties
Export Binary Packet Body

The table shows some data about each packet:

- **Time:** The timestamp of the packet up to nanoseconds
- **Chan#:** The number of the channel the packet belongs to
- **Type:** The type of the data in the packet (like ARINC 429, Ethernet etc.)
- **Errors:** Any errors that have been detected on the packet (packet sync, packet header checksum, secondary header checksum, data checksum, data overflow, RTC sync). If there was any error, the background color of the row will also be red.
- **Data:** Any further information interpreted from the packet. More information can be found in the [supported CH10 data types section](#) and depends on the packet's data type.

Double-clicking a packet shows its [packet entries](#) in the [Packet Entry View](#).

The red arrow buttons in the upper view menu allow navigating between packets with errors. They jump from the current selected packet to the previous or next packet with an error. If no packet is selected, the search will start from top or bottom. The search will not wrap. The function is not available in the free version.

To show the [properties](#) of the channel of the packet, right-click the packet and select **Channel Properties**.


To show the detailed [properties](#) of the packet, right-click the packet and select **Packet Properties**.

To show the [packet's raw content in hexadecimal](#), right-click the packet and select **Show in Hex**. This is not available in the free version.

To [export the body of a CH10 packet](#), right-click the packet and select **Export Binary Packet Body**. This is not available in the free version.

To show the selected packet in an unfiltered Packet View, right-click the packet and select **Show Within all Packets**. This can be used when you want to see which packets are found before / after your selected packet but you have filtered for the packets of a single channel or for corrupted packets. The feature is not available in the free version.

To create a [time marker](#) for a packet, right-click it and select **Create Time Marker**.

If the Packet View is limited to packets of a specific channel, you can jump to the packet at a [time marker or recording event](#). To do so use the  icon on the right top of the view. The view will locate the packet right before or at the time (or after if the time is before first packet).

To see how to export all the visible data of the table to a CSV text file read the section about [table exporting](#).

4.2.3 Packet Entry View

The Packet Entry View hosts a [packet entry viewer](#) to display the content of [packets](#).

You can open it with a double-click on a packet in the [Packet View](#).

The screenshot shows the FLIDAS Data Bus Tools interface. The top toolbar includes a filter icon, a link icon, and navigation arrows. The main window is divided into two sections: 'Packets' and 'Packet Entries (44)'. The 'Packets' section shows a summary of the data. The 'Packet Entries' section is a table with columns for Time, Bus, Msg Type, Rx Cmd, Tx Cmd, Rx St, Tx St, Rx RT, Rx SA, Rx WC, Tx RT, Tx SA, Tx WC, RxSt RT, RxSt Flags, TxSt RT, TxSt Flags, and a status column. A context menu is open over the table, showing options: 'Show in Hex', 'Export Table Content to CSV', 'Copy Table Cell to Clipboard', 'Use Monospace Font', and 'Create Time Marker'.


Time	Bus	Msg Type	Rx Cmd	Tx Cmd	Rx St	Tx St	Rx RT	Rx SA	Rx WC	Tx RT	Tx SA	Tx WC	RxSt RT	RxSt Flags	TxSt RT	TxSt Flags	N
4/17/09 1:22:48 PM 997:206:000	B	RT to BC transfer		5E54		5800				11	18	20			11		
4/17/09 1:22:48 PM 997:660:000	A	RT to RT transfer		36C0	6800	3001	13	7	32	6	22	32	13		6	Term	
4/17/09 1:22:48 PM 998:397:100	A	Mode command with...								21	0	2			21		Ti
4/17/09 1:22:48 PM 998:458:100	A	Mode command with...								7	31	18			7		Ti
4/17/09 1:22:48 PM 998:534:100	A	Mode command with...	78						17				15				S
4/17/09 1:22:49 PM 000:017:099	A	BC to RT transfer	09						31				1				
4/17/09 1:22:49 PM 000:681:095	B	RT to BC transfer								11	18	20			11		
4/17/09 1:22:49 PM 001:135:092	A	RT to RT transfer	68						32	6	22	32	13		6	Term	
4/17/09 1:22:49 PM 001:872:087	A	Mode command with...								21	0	2			21		Ti
4/17/09 1:22:49 PM 001:933:086	A	Mode command with...		3FF2		3800				7	31	18			7		Ti

The above screenshot shows a commercial version with MIL-STD 1553 data. It looks different depending on the CH10 data type and if the version is commercial or free.

The Packet Entry View consists of up to three areas:


- The packet area shows the data about the packets that's content is displayed here. You can also show the content of several packets by using the yellow arrow buttons on top of the view. The left arrow will increase the range of displayed packets content by one packet at the beginning, the right arrow will add a packet to the end.
- The packet entry area shows the packet entries of the packets in the above packet area. The heading line of this area also shows the number of packet entries displayed.
- The third area is optional and can contain additional information depending on the CH10 data type. Analog packets will for example display a graphical plot of the samples there.


By dragging the headings of the second and third area you can redistribute the available space.

The Packet Entry View can be configured to stay synchronized with the packet selection in the Packet View of the same CH10 source (i.e., same color code). It will then update when you select a new packet. You can control this linking between the packet selection and the Packet Entry View with the "link with selection"  icon in the view's toolbar.

When you are scrolling through packets using the keyboard, there is a small delay before synchronizing the Packet Entry View. This is to avoid overloading the system when scrolling fast. Packet selections with the mouse will be executed immediately.

The red arrow buttons in the upper view menu allow navigating between packet entries with errors. They jump from the current selected packet entry to the previous or next packet entry with an error. If no packet is selected, the search will start from top or bottom. The search will not wrap. The function is not available in the free version.

The filter icon  in the upper view menu allows filtering for packet entries of a specific subchannel within the channel. If the filter is active, the packet entries will use an italic font. Another click on the filter symbol disables the filter.

You can save the state of the "link with selection" option using the  icon on the top right of the view.

To see how to export all the visible data of the table to a CSV text file read the section about [table exporting](#).

To show the [properties of a packet](#) right-click a packet and select **Packet Properties**.

To show the [properties of the channel](#) of the packets right-click the heading line of the packet area and select **Channel Properties**.

To create a [time marker](#) for a packet or packet entry, right-click it and select **Create Time Marker**.

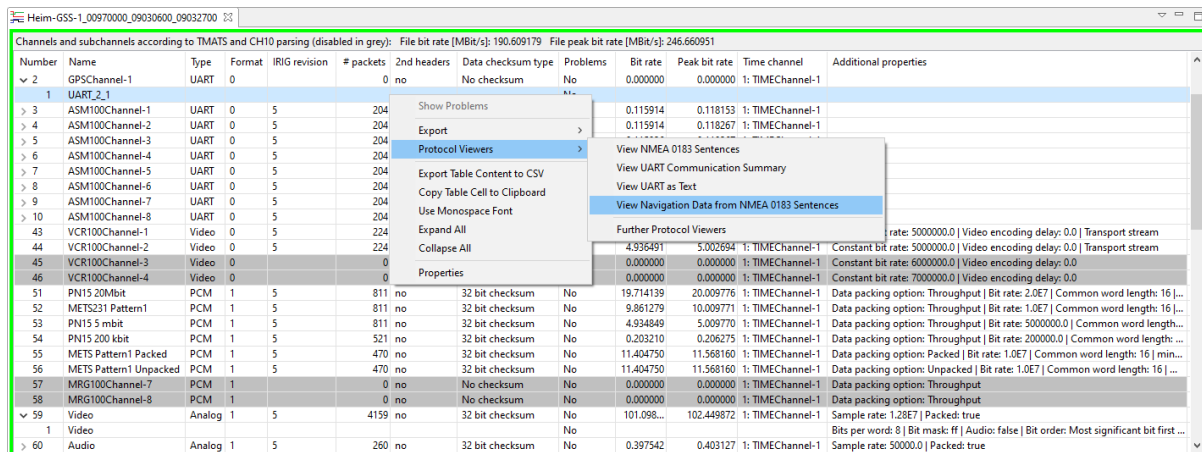
To show the [packet's raw content in hexadecimal](#), right-click the packet or packet entry and select **Show in Hex**. For packet entries the selected packet entry will be highlighted in the Hex Packet View if possible. The Hex Packet View is not available in the free version.

To [export the body of a CH10 packet](#), right-click the packet and select **Export Binary Packet Body**. This is not available in the free version.

4.2.4 Channel View

The Channel View shows [channels](#) and [subchannels](#) from one CH10 source together with some of their properties. It also allows some actions on these channels and subchannels.

To see the Channel View, double-click a CH10 source in the [Repository View](#).



Number	Name	Type	Format	IRIG revision	# packets	2nd headers	Data checksum type	Problems	Bit rate	Peak bit rate	Time channel	Additional properties
1	UART_2.1	UART	0	5	0	no	No checksum	No	0.000000	0.000000	1: TIMEChannel-1	
3	ASM100Channel-1	UART	0	5	204				0.115914	0.118153	1: TIMEChannel-1	
4	ASM100Channel-2	UART	0	5	204				0.115914	0.118267	1: TIMEChannel-1	
5	ASM100Channel-3	UART	0	5	204							
6	ASM100Channel-4	UART	0	5	204							
7	ASM100Channel-5	UART	0	5	204							
8	ASM100Channel-6	UART	0	5	204							
9	ASM100Channel-7	UART	0	5	204							
10	ASM100Channel-8	UART	0	5	204							
43	VCR100Channel-1	Video	0	5	224							
44	VCR100Channel-2	Video	0	5	224							
45	VCR100Channel-3	Video	0	5	0							
46	VCR100Channel-4	Video	0	5	0							
51	PN15 20Mbit	PCM	1	5	811	no	32 bit checksum	No	19.714139	20.009776	1: TIMEChannel-1	Data packing option: Throughput Bit rate: 2.0E7 Common word length: 16 ...
52	METS231 Pattern1	PCM	1	5	811	no	32 bit checksum	No	9.861279	10.009771	1: TIMEChannel-1	Data packing option: Throughput Bit rate: 1.0E7 Common word length: 16 ...
53	PN15 5 mbit	PCM	1	5	811	no	32 bit checksum	No	4.934849	5.009770	1: TIMEChannel-1	Data packing option: Throughput Bit rate: 5000000.0 Common word length: ...
54	PN15 200 kbit	PCM	1	5	521	no	32 bit checksum	No	0.203210	0.206275	1: TIMEChannel-1	Data packing option: Throughput Bit rate: 2000000.0 Common word length: ...
55	METS Pattern1 Packed	PCM	1	5	470	no	32 bit checksum	No	11.404750	11.568160	1: TIMEChannel-1	Data packing option: Packed Bit rate: 1.0E7 Common word length: 16 min...
56	METS Pattern1 Unpacked	PCM	1	5	470	no	32 bit checksum	No	11.404750	11.568160	1: TIMEChannel-1	Data packing option: Unpacked Bit rate: 1.0E7 Common word length: 16 ...
57	MRG100Channel-7	PCM	1	5	0	no	No checksum	No	0.000000	0.000000	1: TIMEChannel-1	Data packing option: Throughput
58	MRG100Channel-8	PCM	1	5	0	no	No checksum	No	0.000000	0.000000	1: TIMEChannel-1	Data packing option: Throughput
59	Video	Analog	1	5	4159	no	32 bit checksum	No	101.098...	102.449872	1: TIMEChannel-1	Sample rate: 1.28E7 Packed: true
60	Audio	Analog	1	5	260	no	32 bit checksum	No	0.397542	0.403127	1: TIMEChannel-1	Bits per word: 8 Bit mask: ff Audio: false Bit order: Most significant bit first ...

You will see a hierarchy with the channels being on top and the subchannels organized as children below. You need to click on the "+" symbol in front of a channel to see its subchannels.

The information has been extracted from TMATS and CH10 source parsing.

Disabled channels will be indicated with a grey background.

The table shows these columns:

- Number: The number of the channel or subchannel.
- Name: The name of the channel or subchannel.
- Type: The CH10 data type of the channel
- Format: The format of the packets as used in IRIG 106 chapter 10
- IRIG revision: The IRIG 106 chapter 10 document revision that the packets on this channel stick to
- # packets: The number of packets of this channel found while parsing this CH10 source (taking into account the CH10 data type filter)
- 2nd headers: If secondary headers are used
- Data checksum type: If checksums are used for the data and which type
- Problems: FLIDAS does a quick check of the availability of some TMATS attributes of channels and subchannels that are needed for some FLIDAS functionality. If problems are detected, an indication is shown in red (or yellow if the problems are unlikely to cause any problems). Details can then be found with a right-click and selecting **Show Problems**. For CH10 files there will be an additional warning after loading (only for problems marked in red). For live data streams this may only be detected during initialization so there is no additional warning except the indication here. Problems detected here do not always indicate an incorrect TMATS file. CH10 sources may be correct and perfectly valid and still do not contain enough information for some FLIDAS functionality. You can try to provide additional TMATS attributes by TMATS overriding or TMATS Default Preferences.
- Bit Rate [Mbit/s]: This is the average bit rate caused by complete CH10 packets of this channel. The average is calculated from recording start to recording end and will therefore most likely include some setup times without actual data recording. For a definition of start and end time see General CH10 Source Information Properties. This property and the peak bit rate in the next column are not initially available and only calculated on request as it may take some time. To calculate both properties select **Calculate Bit Rates** from the view's pull-down menu. This will also calculate the average and peak bit rate for the CH10 file itself which is then displayed above the table. The feature is not available for live data streams.
- Peak Bit Rate [Mbit/s]: This is the maximum bit rate caused by complete CH10 packets of this channel within any period of about a second. There might be strange results in some occasions:
 - If the channel has less than a second of data, the end of the recording dominates the result. Since for the last packet the duration is unknown (the duration is calculated as the span to the next packet) the maximum allowed duration for a packet of 100 ms will be used for calculation. The effect is a reduction in the bit rate. If you have more than a second of data usually the reduction on the end is not relevant most of the time as the data before already calculated a reasonable peak bit rate.
 - Time packets do not cover a time range of data but just indicate an instant moment in time. In FLIDAS the time packet will account for a duration of 100 ms as this is the maximum packet duration. So, time packets occurring with 1 Hz will cause a "peak" of 100 ms and 900 ms of no data. In result the peak bit rate will be about 10 times the average bit rate. Similar effects will occur on other data channels when the data is not continuously transmitted but has larger gaps.

- Time Channel: Shows which time channel is used for resolving RTC timing on each channel. See [distributed CH10 multiplexer sources](#) on this. Note that this time channel will not be used if absolute secondary header timing is available.
- Additional properties: More properties that are specific to the CH10 data type. This is the place where the free version lists these properties. For all commercial versions you can see them also in the [Channel and Subchannel Properties](#) (right-click > Properties).

To see more detailed [properties on the channels and subchannels](#), right-click the channel or subchannel and select **Properties**.

To see how to export all the visible data of the table to a CSV text file read the section about [table exporting](#).

For CH10 files there are additional features:

- To see the packets of one channel in sequence, double-click it. You can also right-click it and select **Show Packets**. The [Packet View](#) will open for this channel. This is not possible for subchannels since each packet may contain data of several subchannels.
- You can export data from some kind of channels or subchannels by selecting one or more, right-clicking them and selecting **Export** and the appropriate type from the menu. For more information see the [exporters](#) section.
- There are specific options on some elements that start a [protocol viewer](#).

For live data streams there are additional features:

- There are specific options on some elements that start a [live processors](#).

4.2.5 TMATS View

The TMATS View shows the TMATS setup of a CH10 source.

To see the TMATS View, right-click the CH10 source in the [Repository View](#) and select **Show TMATS**.

You will see the uninterpreted TMATS data. For interpretations see for example the [Channel View](#) and the [General TMATS Attributes Properties](#).

The TMATS view allows searching inside the TMATS data by entering your keyword in the **search term** field on top and clicking **Next** or **Previous** for forward or reverse search from the cursor position. The search will look for the exact, case sensitive search term entered.

The keyboard shortcut **Ctrl+F** will position the cursor in the search term field and hitting return there will automatically search for the next match.

If you are using [TMATS overriding](#) this doesn't show the original TMATS of the CH10 source but the external file. You will see a warning in this case. [TMATS Default Preferences](#) can also invisibly modify the TMATS setup shown here.

4.2.6 Recording Event Definition View

The Recording Event Definitions View shows recording event definitions from the TMATS section of the CH10 source.

To show it right-click the CH10 source in the Repository View and select **Show Recording Event Definitions**.

Number	Name	Description	Priority	Limit count	Type	Capture mode	Initial capture
0	D 00000004	Discrete Event Mark	1	0	External	1	false
1	CLX_D 00000000	Discrete Record Stop	1	0	External	1	false
2	CLX_D 00000000	Discrete Record Start	1	0	External	1	false
3	CLX_D 00000000	Discrete BIT	1	0	External	1	false
4	CLX_S 00000000	Serial Record Stop	1	0	External	1	false
5	CLX_S 00000000	Serial Record Start	1	0	External	1	false
6	CLX_S 00000000	Serial BIT	1	0	External	1	false

The table shows one line for each recording event definition and the following decoded properties (for reference the TMATS-ASCII codes are given in italics):

- Event number (counted up from 0 for each event)
- Event ID (*R-x\EV\ID-n*)
- Event description (*R-x\EV\D-n*)
- Event priority (*R-x\EV\P-n*)
- Recording event limit count (*R-x\EV\LC-n*)
- Event type (*R-x\EV\T-n*)
- Event capture mode (*R-x\EV\CM-n*)
- Event initial capture (*R-x\EV\IC-n*)

To see how to export all the visible data of the table to a CSV text file read the section about table exporting.

4.2.7 Hex Packet View

The Hex Packet View allows analyzing a packet's raw bytes from the CH10 file. It can be used to debug corrupted files or prepare test data by manipulating existing files. It is available in all commercial versions.

To open the Hex Packet View, right-click a packet or packet entry in the Packet View or Packet Entry View and select **Show in Hex**. Most detected errors in the CH10 Verification View also allow this.

HEX MIL-STD-1553 Data, Fmt 1 @ 4/17/09 1:22:49 PM 113:981:124																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000	25	EB	00	0A	64	08	00	00	4A	08	00	00	01	06	00	19
00010	DF	70	58	D6	00	00	0B	6C	2C	00	00	80	DF	70	58	D6
00020	00	00	00	00	00	20	30	00	2C	00	54	5E	00	58	01	00
00030	FE	9E	D9	07	6B	00	00	00	29	BC	01	00	2C	BD	8B	00
00040	C5	1D	00	00	00	00	FE	9E	FE	9E	FE	9E	FE	9E	00	00
00050	33	02	FE	9E	FE	9E	9B	82	58	D6	00	00	00	00	00	08
00060	3B	45	48	00	E0	68	C0	36	01	30	01	00	FF	9E	D9	07
00070	6B	00	00	00	29	BC	01	00	F2	BE	8B	00	A5	AB	00	00
00080	00	00	FF	9E	FF	9E	FF	9E	FF	9E	FF	9E	FF	9E	FF	9E
00090	FF	9E	FF	9E	FF	9E	FF	9E	FF	9E	FF	9E	FF	9E	FF	9E
000A0	00	00	33	02	FF	9E	FF	9E	FF	9E	00	68	65	9F	58	D6
000B0	00	00	00	00	00	00	4F	00	04	00	02	AC	00	A8	C7	A1
000C0	58	D6	00	00	00	00	00	00	59	00	06	00	F2	3F	00	38
000D0	2D	38	BF	A4	58	D6	00	00	00	00	00	00	63	00	06	00
000E0	11	78	EE	87	00	78	85	BB	58	D6	00	00	00	00	00	00
000F0	27	00	42	00	7F	09	01	00	03	9F	D9	07	6B	00	00	00
00100	29	BC	01	00	A3	C4	8D	00	F5	72	00	00	00	00	03	9F
00110	03	9F	03	9F	03	9F	03	9F	03	9F	03	9F	03	9F	03	9F
00120	03	9F	03	9F	03	9F	03	9F	03	9F	03	9F	90	01	33	02
00130	03	9F	03	9F	00	08	75	D5	58	D6	00	00	00	00	00	20
00140	30	00	2C	00	54	5E	00	58	01	00	04	9F	D9	07	6B	00

Position: 18 (hex) 24 (dec)
Section: Packet body: CSDW
Position within section: 0 (hex) 0 (dec)
Endianness: Little

Base ty...	Bits	Value
unsigned	8	44
signed	8	44
ASCII	8	,
binary	8	0010 1100
hex	8	2C
unsigned	16	44
signed	16	44
binary	16	0000 0000 0010 1100
hex	16	00 2C
unsigned	32	2147483692
signed	32	-2147483604
binary	32	1000 0000 0000 0000 0000 0000 0010 1100
hex	32	80 00 00 2C
unsigned	64	15445219027294617644
signed	64	-3001525046414933972
binary	64	1101 0110 0101 1000 0111 0000 1101 1111 1000 0000 0...
hex	64	D6 58 70 DF 80 00 00 2C

You will see the content of the packet byte by byte on the left. The displayed content will be limited to one megabyte, which is enough to see all regular packets even if they are corrupted and exceed their size by 100%. Only TMATS packets which may be up to 128 MB are affected.

The sections of the file will be marked in color if the packet is valid and the sections could therefore be safely recognized. The sections are:

- Packet header (yellow)
- Optional packet secondary header (green)
- Packet body (cyan)
- Optional filler in packet trailer (magenta)
- Optional data checksum in packet trailer (light green)

For some CH10 data types the packet body will be further decoded to:

- Channel specific data word (orange)
- Packet entries (alternating in two different blue colors to show the different entries). This will also include fillers at the end of the packet entry unless they are part of the packet trailer. If you opened the Hex Packet View from a packet entry, then this packet entry will be displayed in a grey color instead.

The CH10 data types supported for this are:

- ARINC 429
- CAN bus
- Ethernet
- AFDX
- Discrete
- MIL-STD 1553
- PCM
- UART
- MPEG video
- Fibre channel FS layer

- Recording index
- Recording events
- TMATS setup record
- Time format 1
- Network time
- Streaming Configuration

To the right there is a table which interprets the data that is currently selected. You can select a byte with the mouse or navigate with the keyboard to select something. The table will show the meaning of this byte or a group of bytes to the right in different formats like unsigned, signed (two's complement), binary or hexadecimal. The selection above the table controls if the interpretation of several bytes uses little or big-endian.

If you select a line in the interpretation table, the data bytes that apply to this line will be printed in bold font in the packet content to the left.

To the top right there is information on the current position of your selection within the packet:

- **Position** shows the offset of the packet start to the beginning of your selection. 0 means you selected the first byte.
- **Section** describes which part of the packet you are in. An example would be *Packet header: RTC* or *Packet body: Packet entry 3*
- **Position within section** shows the offset of the beginning of the section given in the line above to the beginning of your selection.

To see the packet properties, right-click inside the data and select **Packet Properties**.

4.2.7.1 Data Manipulation

It is also possible to edit the data in the CH10 file. In many environments this would be strictly forbidden and therefore FLIDAS must be started with a specific command line option to enable this. There are also safety questions so it is not possible to manipulate the data accidentally.

The function is intended for recorder or software developers or people who verify their new CH10 equipment. In this case you might want to use a real recording and inject a specific error at one place (like a parity error or wrong timestamp). You can then replay the data and see how your equipment behaves.

The changes are directly written back to the source file so you need to make a copy in advance if you need one.

The feature is intended for small, non-structural changes. It is not possible to add packet entries, packets etc. It should only be used by people who are very familiar with the CH10 file format.

FLIDAS loads packet header data and channel specific data words in the initial loading of the file. Changes here would only take effect during the next loading. This means you might need to reload the file after one change before applying the next change. If you change the

checksum type from 8 to 16 bits for example, changes in the packet body would still generate the 8-bit checksum until you reload the file.

To enable manipulation, you must start FLIDAS with a command line option. There are two ways to do this:

- The directory that contains the FLIDAS executable also contains a file named FLIDAS.ini. This file contains default command line options. It contains two lines reading:
 - -ch10Modification
deny

You need to change the text "deny" to "allow". Do not add any whitespaces here. If the two lines have been removed for some reason you can add them before the line "-vmargs".

- A second option is to remove the two lines mentioned in the previous option from the FLIDAS.ini. You can then start the FLIDAS executable from the command line and pass the option "-ch10Modification allow". This way you can decide each time you start if you need modification or not. Of course, you can also create two Windows batch files to start FLIDAS with either of the options.

The command line options will be case insensitive and will default to no modification in case of spelling errors.

To modify a position in the packet, locate the position with the cursor and double-click in the interpretation table. You will be able to edit the data in the format of the line you have clicked at (i.e., endianness, byte size, hex / bin / unsigned / signed / ASCII applies here).

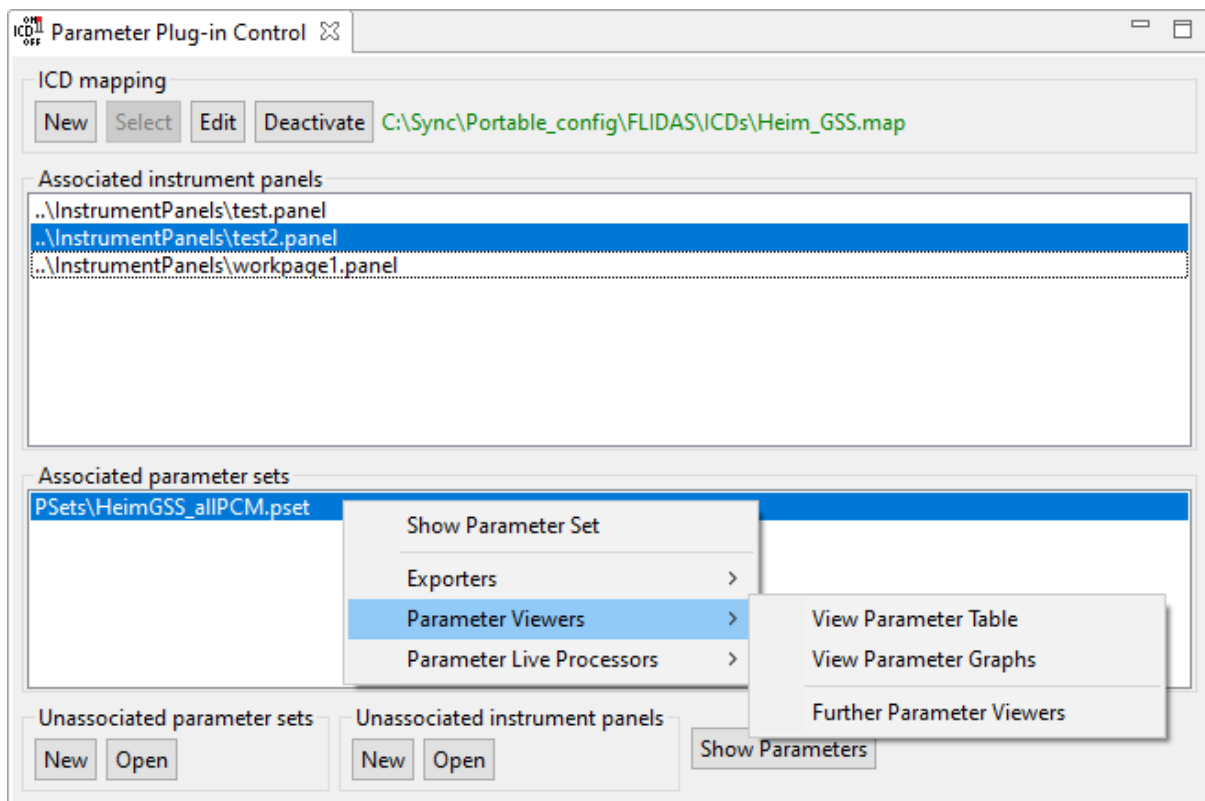
If the changed bytes are covered by any checksum, you will be asked if you want to update the checksums. If the packet is corrupted, the positions of the checksums will not be known and can therefore not be updated.

It is not possible to insert or delete bytes, only overwriting. This is mainly useful for setting error flags.

4.2.8 Parameter Plug-in Control View

The Parameter Plug-in Control View controls the main actions available in the Parameter or Parameter Streaming Version. It controls the life cycle of ICD mappings, manages instrument panels and parameter sets and allows parameter based export and display functions.

The Parameter Plug-in Control View is opened by clicking **Window > Show View > Parameter Plug-in Controls** or using the keyboard shortcut **Alt+P**.



4.2.8.1 ICD Mapping Management

The upper area shows some buttons for managing the current ICD mapping.

To create a new ICD mapping, click the **New** button. This also opens the ICD Mapping View to edit the new ICD mapping. Note that you can also create a template of an ICD mapping by right-clicking a CH10 source from the Repository View and selecting **Create ICD Mapping template**.

To select an ICD mapping click the **Select** button. A selected ICD mapping just serves as a preselection for other actions like editing and activation. The selected file will be shown to the right of the buttons.

To edit an ICD mapping click the **Edit** button. The ICD Mapping View will open.

To activate an ICD mapping click the **Activate** button. This may take some time if you have big ICDs but the progress will be reported and the activation can be cancelled. When activating has finished the displayed file name of the ICD mapping will be shown in green color and the button text will switch to **Deactivate** to deactivate the ICD mapping again. Deactivation requires that you close all open parameter sets before because they depend on an active ICD mapping.

As soon as the ICD mapping is activated, the parameter set based function below become active.

4.2.8.2 Instrument Panel Management

Below the ICD mapping area, there are two areas for managing instrument panels.

The associated instrument panels area shows instrument panels that are listed within the ICD mapping file as applicable to the ICD mapping. These are shortcuts to work with them. You can open these if you right-click a selection of one or several instrument panels and select **Open Instrument Panel**. You can also double-click on an instrument panel to open it.

The unassociated instrument panels area allows working with instrument panels that are not listed within the ICD mapping file as applicable to the ICD mapping.

- To open one or several instrument panels click **Open** and choose the files.
- To create a new instrument panel click **New**.

4.2.8.3 Parameter Set Management

Below the ICD mapping area, there are two areas for managing parameter sets.

The associated parameter sets area shows parameter sets that are listed within the ICD mapping file as applicable to the ICD mapping. These are shortcuts to work with them. A right-click brings up several actions on the listed parameter sets. Note that you can select more than one set at a time to join the selected parameter sets to one without duplicates.

- To open a parameter set click **Show Parameter Set**. You can also simply double-click it as a shortcut. This shows the Parameter Set View which allows modification of the parameter set and export and view actions on the modified set. If you select several parameter sets you can store the joined parameter set to a new file or just use it for a one-time purpose.
- To export parameter values click **Exporters** and select the appropriate one. See the exporters section of the manual to see a list of all exporters.
- To view parameter values on the GUI click **Parameter Viewers** and select the appropriate one. See the parameter viewer section of the manual to see a list of all parameter viewers.
- To view live parameter values from a live data stream on the GUI click **Parameter Live Processors** and the appropriate parameter live processor. See the parameter live processors section of the manual to see a list of all parameter live processors.

The unassociated parameter sets area allows working with parameter sets that are not listed within the ICD mapping file as applicable to the ICD mapping.

- To open a parameter set click **Open** and choose the files. This opens the Parameter Set View which allows modification of the parameter set and export and view actions on the modified set. If you select several parameter sets you can store the joined parameter set to a new file or just use it for a one-time purpose.
- To create a new parameter set click **New**. This opens the Parameter Set View on the empty set. You can also use the new set without saving for a one-time selection of parameters to run actions on it.

4.2.8.4 Parameter Details

To see details on the parameters from the activated ICD mapping, you can click the **Show Parameters** button on the bottom. It will open the Parameter List View.

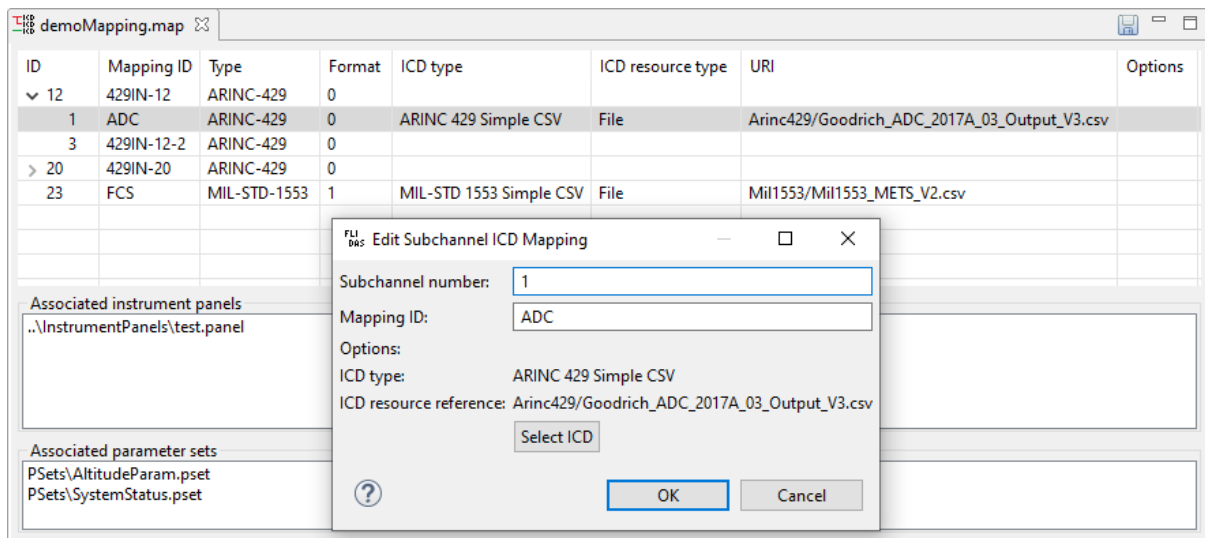
4.2.9 ICD Mapping View


The ICD Mapping View is used to display and edit the ICD mapping.

This view is only available in the Parameter or Parameter Streaming Version.

There are several ways to open an ICD Mapping View:

- You can open it with an empty ICD mapping with the **New** button in the Parameter Plug-in Control View
- You can open an existing one when you select it and use the **Edit** button in the Parameter Plug-in Control View
- You can create a new ICD mapping from a CH10 source as a template and immediately open it by right-clicking the CH10 source in the Repository View and selecting **Create ICD Mapping Template**



Any modification you do to the ICD mapping can be saved using the  symbol in the upper right of the view. If there are unsaved changes the title of the view will start with a "*". If you close the view without saving, you will get the chance to save before. If you have edited the active ICD mapping this will not apply the changes until you deactivate it and activate it again using the Parameter Plug-in Control View.

The three other areas of the view cover the channel / subchannel ICD mapping management and associated instrument panels and parameter sets.

4.2.9.1 Channel / Subchannel ICD Mapping

The upper area of the view is occupied by a table showing a hierarchy of all channels and subchannels defined for the ICD mapping. This can be less than the channels and subchannels of the CH10 sources you apply this mapping to but should not be more. The table has these columns:

- **ID:** The Channel or Subchannel number where 1 is the lowest number. Note that some TMATS setups use a different numbering system than the CH10 Sources. See also the [Subchannel number Preferences](#) for this.
- **Mapping ID:** This is a user definable mnemonic for the channel or subchannel. For example, if it records your air data computer it could be ADC. This mapping ID is stored in [parameter sets](#) to identify which [ICD](#) the parameter definition should be taken from. This decouples the parameter sets from the ICD mapping, so you can use different channel assignments and ICD mapping files but will still be able to reuse parameter sets as long as you keep the mapping ID consistent. The mapping ID may be any text but not contain "#".
- **Type:** The [CH10 data type](#)
- **Format:** The data type format as defined in IRIG 106 chapter 10
- **ICD type:** The type of the [ICD](#). This may be a standard one provided with FLIDAS or can also be a proprietary extension to support your company internal definitions formats.
- **ICD resource type:** The type of the resource that contains the ICD definition. Currently this can be a file or hard coded within a FLIDAS plug-in. Future extensions may be other types like databases.
- **URI:** A link to the resource that defines the ICD if needed. This may currently only refer to a file.
- **Options:** Any further option on the ICD if required by the specific ICD type. The format will be given as "Option1=Value1;Option2=Value2;"

You have these options to modify the channel and subchannel mapping definition:

- To add a channel ICD mapping right-click another channel (or select nothing and right-click) and select **Add Channel**. An edit dialog pops up that is described later.
- To add a subchannel ICD mapping right-click the enclosing channel and select **Add Subchannel**. An edit dialog pops up that is described later.
- To remove a channel or subchannel right-click it and select **Remove**. If you remove a channel its subchannels will also be removed. You can use the keyboard shortcut **Del** to remove the selected element too.
- To modify a channel or subchannel right-click it and select **Edit**. You can also double-click it instead. An edit dialog pops up that is described below.

The edit dialog allows modifying the ID, mapping ID and mapped ICD as defined above. To map another ICD, click at **Select ICD** and select the type from the **ICD type** drop-down list in the upcoming dialog. You can select **None** to clear the ICD mapping. If you select a file, you can store an absolute or relative path. Relative paths will be relative to the location of the ICD mapping file. Some ICD types may have additional options to configure.

Editing a channel also allows modifying the CH10 data type and format. These also affect the subchannels and may invalidate subchannel ICD mappings. In this case you get a warning.

4.2.9.2 Associated Instrument Panels

This area of the view shows instrument panels that are intended to be used with this ICD mapping. They are shown as shortcuts in the Parameter Plug-in Control View when the ICD mapping is active. It is still possible to use other instrument panels.

To associate instrument panels right-click the list and select **Add**. You can select one or more instrument panel files in the following file dialog. You can choose to store absolute or relative paths to the files. Relative paths will be relative to the ICD mapping file.

To remove associations to instrument panels select one or more from the list and select **Remove** or use the keyboard shortcut **Del**.

4.2.9.3 Associated Parameter Sets

This area of the view shows parameter sets that are intended to be used with this ICD mapping. They are shown as shortcuts in the Parameter Plug-in Control View when the ICD mapping is active. It is still possible to use other parameter sets.

To associate parameter sets right-click the list and select **Add**. You can select one or more parameter set files in the following file dialog. You can choose to store absolute or relative paths to the files. Relative paths will be relative to the ICD mapping file.

To remove associations to parameter sets select one or more from the list and select **Remove** or use the keyboard shortcut **Del**.

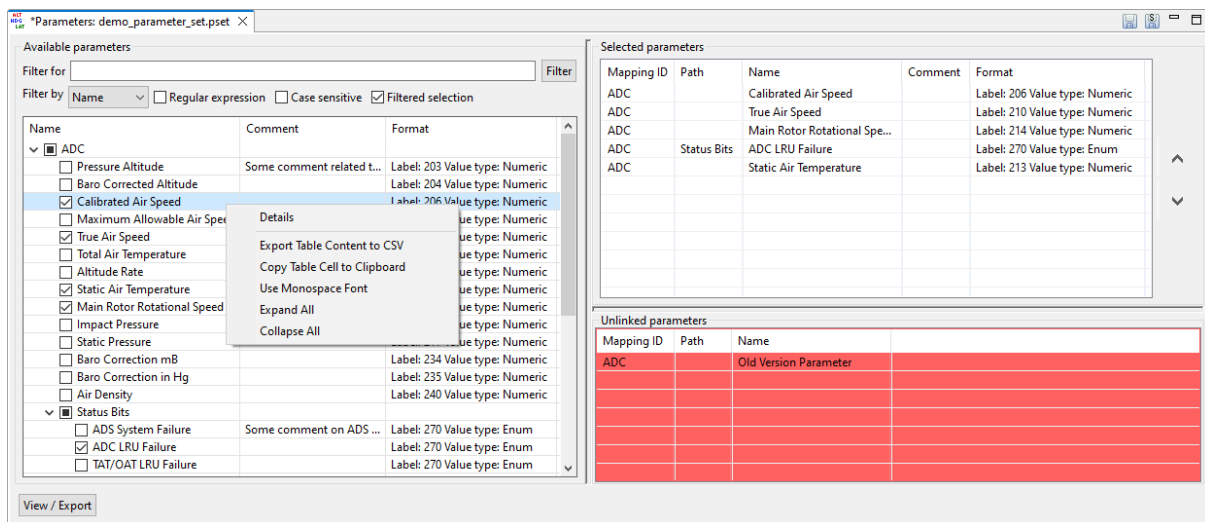
4.2.10 Parameter Set View


The Parameter Set View is used to display and edit parameter sets. It also allows running export and display actions on the selected parameter values.

This view is only available in the Parameter or Parameter Streaming Version.

There are several ways to open a Parameter Set View:

- You can open it with an empty parameter set with the **New** button in the Parameter Plug-in Control View in the area for unassociated parameter sets.
- You can open an existing one with the **Open** button in the Parameter Plug-in Control View in the area for unassociated parameter sets. If you select several parameter sets at once, you get a new, unsaved parameter set with the joined content of the selected ones.
- You can open an existing one that is associated with the ICD mapping when you right-click it in the Parameter Plug-in Control View in the area for associated parameter sets and select **Show Parameter Set**. If you select several parameter sets at once, you get a new, unsaved parameter set with the joined content of the selected ones.



Any modification you do to the parameter set can be saved using the  symbol in the upper right of the view. If there are unsaved changes the title of the view will start with a "*". If you close the view without saving, you will get the chance to save before. However, often you will just use this view for a temporary selection of parameters for a one-time action.

You can redistribute the space between the tables by dragging the lines between the tables.

The table **Available parameters** shows a hierarchy of all parameters available through the active **ICD mapping**. They are grouped by the mapping ID associated with the channel or subchannel and further hierarchical grouping defined by the ICD. They appear in the order of the ICD definition and show some basic information about the parameter format and the associated comment.

You can select and deselect parameters for the parameter set by using the check boxes in front of the parameter name. You can also select / deselect all parameters in a hierarchy segment by using the check box at a higher level in the hierarchy.

You can filter parameters like described in the [Parameter List View](#).

When filtering is active and you select/deselect a higher hierarchy level to select/deselect the parameters below, there are two modes: The selection/deselection will either only apply to the parameters below that are visible or to all, also the ones hidden by the filter. This depends on your choice on the check box labeled **Filtered selection** in the filter configuration.

Be careful as performance of working with parameter sets degrades with the number of selected parameters.

The table **Selected parameters** shows all parameters from the parameter set that are valid in the active ICD mapping with their mapping ID, path, name, comment and some information about the format. The order within the table is used by actions run on this parameter set like exporting or displaying. You can change the order by selecting one or more and using the up / down arrow buttons to the right. You can also deselect one or more parameters by selecting them, right-clicking and selecting **Remove**.

You can get information on available or selected parameters by right-clicking a single parameter and selecting **Details**. You can get information on ICDs by right-clicking an ICD in the available parameters table and selecting **Details** as well.

The table **Unlinked parameters** shows all parameters from the parameter set that are invalid in the active ICD mapping and is only visible when such parameters exist. Invalid means the combination of mapping ID and parameter path and name is not found. These parameters will be unavailable when you use this parameter set and will trigger additional warnings. The unlinked parameters show their mapping ID and name to help you searching the problem. If you want to remove unlinked parameters from the parameter set, select one or more, right-click them and select **Remove**.

The **View / Export** button allows running export and display actions related to the current selection of parameters:

- To export parameter values click **View / Export** and select **Export** and the appropriate exporter. See the [exporters](#) section of the manual to see a list of all exporters.
- To view parameter values on the GUI click **View / Export** and select **Parameter Viewer** and the appropriate parameter viewer. See the [parameter viewer](#) section of the manual to see a list of all parameter viewers.
- To view live parameter values from a live data stream on the GUI click **View / Export** and select **Parameter Live Processors** and the appropriate parameter live processor. See the [parameter live processors](#) section of the manual to see a list of all parameter live processors.

When working with a [parameter live processors](#) you can add new parameters to it while it is running by dragging a parameter from the available parameter section and dropping it to a parameter live processor. The exact use depends on the type of parameter live processor. When you drag a hierarchy segment of the available parameters and it is not expanded, it will drag either the whole elements below or just the ones that pass the filter depending on the **Filtered selection** check box. If the hierarchy is expanded you can only manually select a sub element. You can also drag and drop one or more parameters from the selected parameters table. If you just want to drag and drop parameters, the [Parameter List View](#) is a more flexible option.

4.2.11 Parameter List View

The Parameter List View shows all parameters defined by the active [ICD mapping](#). It can be used as a source to drag and drop parameters to [parameter live processors](#).

This view is only available in the [Parameter or Parameter Streaming Version](#).

To open the Parameter List View click at **Show Parameters** in the [Parameter Plug-in Control View](#).

The view shows a hierarchy of all parameters available through the active [ICD mapping](#). They are grouped by the mapping ID associated with the channel or subchannel and further hierarchical grouping defined by the ICD. They appear in the order of the ICD definition.

You can get information on parameters or ICDs by right-clicking a single one and selecting **Details**.

When working with a parameter live processor you can add new parameters to it while it is running by dragging a parameter and dropping it to a parameter live processor. The exact use depends on the type of parameter live processor. When you drag a hierarchy segment of the parameters and it is not expanded, it will drag either the whole elements below or just the ones that pass the filter (see below) depending on the **Filtered selection** check box. If the hierarchy is expanded you can only manually select a sub element. You can also select multiple parameters at once.

4.2.11.1 Filtering

You can filter the parameters you see by entering a part of the name or comment in the filter text and hitting return or clicking at **Filter**. You can choose to filter either for name or comment. Disable the filter by removing the filter text and applying.

A yellow background on one of the filter configurations means that the currently active filter doesn't use this configuration yet. So for example if you apply a filter and then modify the filter text field, it will be yellow until you apply this filter.

Filtering is case insensitive by default unless you enable the option **Case sensitive**.


Filtering also applies to the parent elements of the parameters (hierarchical grouping or mapping ID). Since they have no comment attached, only the name filter works.

Filtering uses regular expressions when the option **Regular expression** is enabled. By regular expressions you can define patterns for a part of the name or comment you are searching for. Examples:

Regular expression	Description	Matches	Doesn't match
[1-9][0-9]	A number 10 ... 99	Tank 13	Tank 3 Tank 03
[0-9]\$	Ending with a digit	Tank 13 Tank 3	Tank 3 left side
.{10,}	10 or more characters (i.e. 10 of anything, character not meaning letter)	Tank 13 left side	Tank 3 LH

For a more detailed explanation of regular expressions look up the external website www.regular-expressions.info for example. Note that regular expressions are used in many programming languages and tools with slightly different syntax although the basic syntax and features are usually the same. FLIDAS uses the Java syntax (and Java is not JavaScript).

If you use an invalid regular expression, the filter text will show a red background and no filter will be applied.

You can save the filtering options using the  icon on the top right of the view.

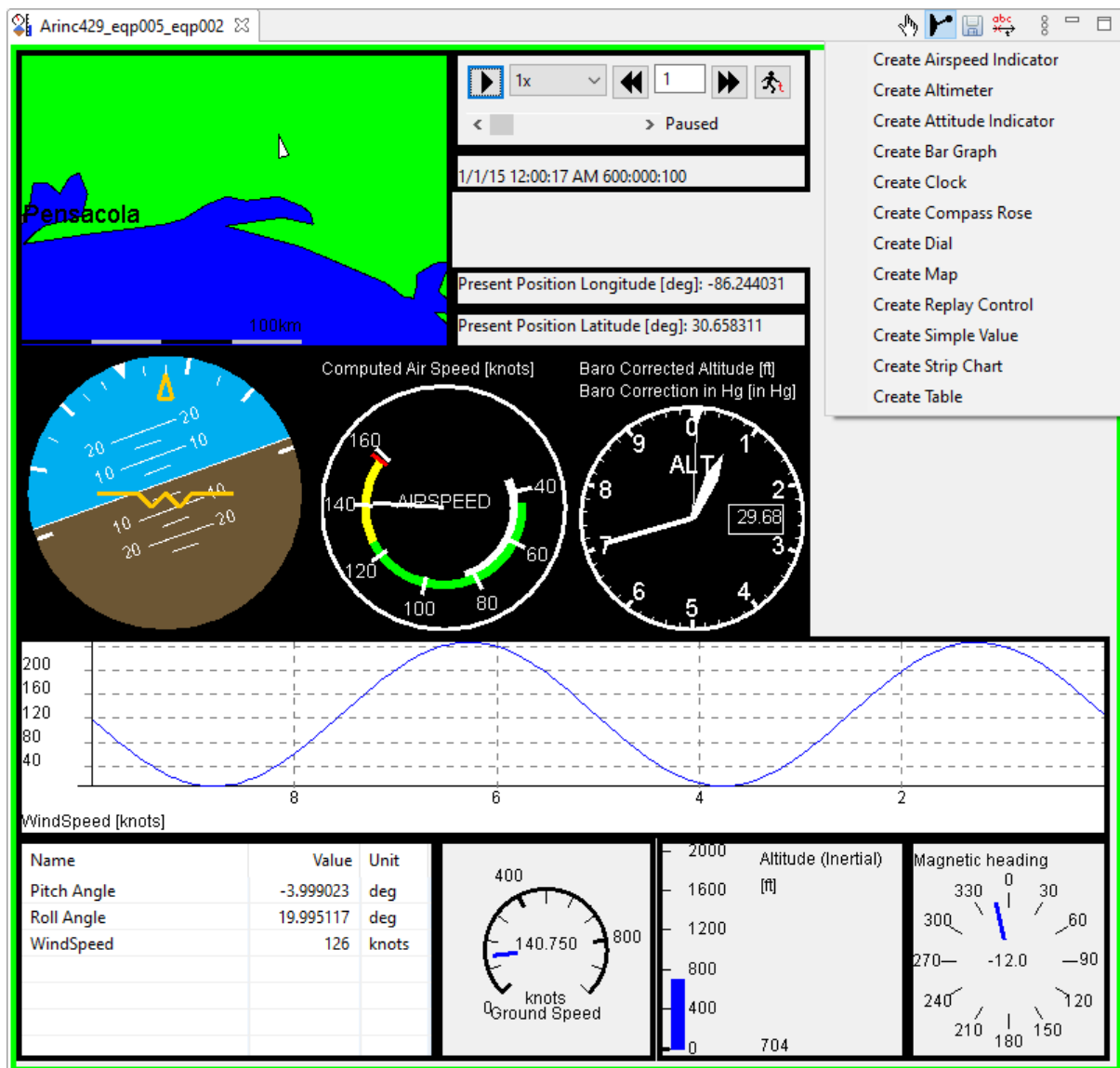
4.2.12 Instrument Panel View


The Instrument Panel View arranges user configurable gauges to display current parameter values of a live data stream or a CH10 file being replayed for the instrument panels.


This view is only usable in the Parameter or Parameter Streaming Version.

There are several ways to open an Instrument Panel View:


- You can open an empty instrument panel already connected to a CH10 source when you right-click the CH10 source in the FLIDAS Repository and select **Instrument Panel**. You must have activated an ICD mapping before.
- You can open an empty instrument panel with the **New** button in the Parameter Plug-in Control View in the area for unassociated instrument panels.
- You can open one or several existing instrument panels with the **Open** button in the Parameter Plug-in Control View in the area for unassociated instrument panels.
- You can open one or several existing instrument panels that are associated with the ICD mapping when you right-click it in the Parameter Plug-in Control View in the area for associated instrument panels and select **Open Instrument Panel**.




Any modification you do to the instrument panel can be saved using the  symbol in the upper right of the view. If there are unsaved changes the title of the view will start with a "***".


An instrument panel can be connected to a CH10 source and therefore show its current parameter values. This connection is controlled using the  button on the upper right to activate the instrument panel. An active instrument panel will show the color of the CH10 source on the outer border. If a lot of data comes in and graphical updates are required, this mode will influence the performance of your system. An inactive instrument panel does not impact performance.

While an instrument panel connected to a live data stream automatically shows the current values of parameters, an instrument panel connected to a CH10 file needs user interaction to control the replay. See the [gauge](#) chapter for the Replay Control.

While active you can create time markers if the source is at a time stamped position by pushing the  button on the upper right.

4.2.12.1 Edit Mode

An instrument panel can be in an edit mode that allows changing the layout and some properties of the gauges. Without the edit mode nothing can be changed and only the current values will be displayed (if active). The edit mode is controlled using the  button on the upper right.

You can create a gauge by using the three dots icon  at the top right of the view to drop down a menu. The different gauges that you can select are described in the [gauge's section of the manual](#).

You can drag the borders and corners of each gauge to resize it in that direction. You can also drag the green bar on top of each gauge to move it.

If you click at the green bar or the gauge borders, the gauge will move on top in case it was covered partly by other gauges.

You can right-click the green bar on top of the gauge to open the gauge menu. Some gauges have special options but the most common are:

- **Delete Gauge** deletes the gauge
- **Set Min/Max** configures minimum and maximum values to be displayed on numeric gauges. If a parameter is assigned to the gauge and it defines minimum and maximum values in the ICD you can also select these in the upcoming dialog. The configuration is then like if it was manually entered, i.e., it will not change when you assign another parameter or update the ICD.
- **Assign Parameters**

opens a dialog box to change parameter assignment to variables used in the gauge. The dialog is described below.

- **Show Details** gives some information about the gauge's configuration like the connected parameter, the configured minimum and maximum value and the gauge type

4.2.12.1.1 Parameter assignment

Gauges may accept none, one or also several parameters for visualization.

To assign parameters, drag and drop them on the gauge. You can drag them from the [Parameter List View](#) or the [Parameter Set View](#) for example.

When a gauge accepts several parameters, a dialog will open that allows assigning the parameters to the variables (like latitude and longitude for a map) of the gauge. For each variable you may assign one parameter or none at all. A drop-down menu allows choosing the parameter you want to connect. Only the parameters already assigned to the gauge and the newly dropped are available. The drop-down menus are also filtered for the appropriate parameter type (like numeric, enumeration ...). The drop-down menus show the parameter name followed by the mapping ID in brackets. A variable may be highlighted in red if no parameter is assigned but an assignment is required. It will also be red if the parameter has

not been found (like an "unlinked parameter" in a parameter set). This means the instrument panel contains the name of the parameter but your current ICD mapping does not define this parameter. It may have been renamed and you should replace it with a proper new parameter. These unlinked parameters are shown in brackets in the drop-down box. A variable may be highlighted in yellow if no parameter is assigned and the assignment is optional (for example, a map may optionally show the direction of the object in addition to the required position).

If a gauge misses required parameters or uses unlinked parameters, it will have a red border. If it only misses optional parameters, it has a yellow border.

4.2.12.2 Further notes

For live data streams the displayed values are always limited to the time span since the parameter has been selected for display or may contain the most recent value before this if the same parameter is already processed by another parameter live processor. For CH10 files the values are limited to the point in time since the replay started. If you jump in the CH10 file to a different point in time, all Parameters turn to an invalid value and are then updated from that point in time when replaying.

Activated instrument panels use up system resources so close their views or deactivate them if they are not needed anymore.

An instrument panel connected to a live data stream shares a common data stream task called "live parameter provider" with all Parameter Live Processors. It calculates the current values of the parameters used in any parameter live processor or instrument panel. Instrument panels connected to a file also use a live parameter provider that is shared with all instrument panels on the same CH10 file.

Note: instrument panels will not adapt properly if you switch to a different ICD mapping while they are running.

4.2.13 CH10 Verification View

The CH10 Verification View shows the result of the check of a CH10 file against the IRIG 106 chapter 10 specification up to the 2022 revision. It is available in all commercial versions. It does the same type of work as other COTS "CH10 validators" but gives a GUI to directly link to and inspect the detected errors with the existing FLIDAS features. FLIDAS uses the term verification intentionally as verification is the test against a specification (which can be automated here) while validation is the check if something satisfies the purpose it is built for (which can usually not be automated but needs human checking).

To show the CH10 Verification View, right-click a CH10 file in the Repository View and select **Verify CH10 File**.

A wizard will come up where you can select which revision of IRIG 106 you want to check against. This is initialized from the maximum of the TMATS attribute *G106* and the *data type version* of all CH10 channels detected from the first packet headers. Whenever a check explicitly checks the conformance of a CH10 packet with the standard, it is always checked against the *data type version* of the packet header, so the value here only applies to global checks.

In the wizard you can also limit the number of errors per check you want to see. You can also leave it empty to show all errors.

The checking will require some time and can be canceled. In this case the problems found so far will be displayed.

After the check the CH10 Verification View will come up. It has two areas.

4.2.13.1 General Error Indications

Independent of this checking FLIDAS already did some error checking when first loading the CH10 file. These errors are not re-checked here. Instead, indicators will show up on the top of the view if needed.

The button **Show Channel View** takes you to the Channel View which already shows some problems highlighted in red on the according channel, mostly related to TMATS. A right-click on the channel and a click on **Show Problems** shows the details. Note that the subchannels list problems of their own if they have some. So, on channels with subchannels also look up these problems. Note that some problems in TMATS may not show if the missing information could already be reconstructed from the CH10 file's data section. Also, problems here can indicate an error in the recorder but don't need to. It only tells you that FLIDAS is missing information to properly process the data. It is possible that the data is optional though and the recorder is correct.

A second button **Show Invalid Packets** takes you to the Packet View, limited to all CH10 packets that had any problems detected during loading. You can use regular FLIDAS features to inspect more there (e.g., look up packet properties, or packet entries or inspect the packet in the structured Hex Packet View).

When first loading the CH10 file you can select to suppress specific CH10 data types using the CH10 data type filters or suppress data checksum checks. In both cases you might see an incomplete picture of your CH10 file verification as these two settings affect the verification. Therefore, you will see warning indicators for this on top.

4.2.13.2 Results from Error Check

The bigger lower area of the CH10 Verification View shows a table with the errors grouped by checks. Checks that detected an error will be highlighted in red. Each group will show in general a detected error condition with these columns:

- **Count:** Shows the number of errors in this category. If you limited the maximum number of errors per category and the limit suppressed errors then the number will be followed by "(more)".
- **Error:** A general error description
- **Class:** A classification of the error like "time" or "file structure"

Each error group can be expanded to show the individual occurrences of this error. The individual lines will show these columns:

- **Time:** The timestamp of the CH10 packet that this error was detected for

- **Channel:** The CH10 channel number of the CH10 packet
- **Type:** The CH10 data type of the CH10 packet
- **Error:** More information on this individual error

You can right-click each individual error. The upcoming menu shows you more options for debugging:

- **Show in Packet View:** Opens the [Packet View](#) and shows the packet with the error. Here you can inspect it in its context in the CH10 file with the CH10 packets coming before and after. You can use regular FLIDAS features to inspect more there.
- **Show in Packet View of this Channel:** Opens the [Packet View](#) and shows the packet with the error. Only the packets of the same CH10 channel will show so you can inspect in this context. You can use regular FLIDAS features to inspect more there.
- **Show Packet in Hex:** Show the CH10 packet in the structured [Hex Packet View](#).
- **Show Packet Entries:** Show the CH10 packet content in the [Packet Entry View](#).
- **Packet Properties:** Show the [CH10 packet properties](#)

As on most tables in FLIDAS you can [export this table to a CSV file](#) to get some kind of report. It will only include the data of the table, not the problems indicated above the table.

4.2.13.3 Further Problem Analysis and Solution Prototyping

Should you be interested to analyze the error further or even try to prototype a fix you intend to implement in the hardware or use to repair a corrupted file, a general idea is to export the CH10 file to XML, repair it in a text editor, convert back to CH10 and verify again. You can do this using the [XML CH10 mapping](#). If the files become too big consider time slicing or suppressing CH10 channels before using the [Filtered CH10 Exporter](#). To locate the error in the XML file use the CH10 to XML conversion option to add timestamps as comments. This way you can directly search for it.

4.2.13.4 Checks Executed

Explaining all the checks executed in detail would require too much documentation here. You can see them in general when looking at the error groups listed in the table. If details are needed ask for it and there may be more information in the future. Right now:

- There are some checks on the general file structure (commit time, 100ms per packet rule for CH10 data types with specific support in FLIDAS, TMATS on start, time packet is first dynamic, RTC increasing per channel, same CH10 data type on channel, channel 0 use ...)
- There are some checks on the general structure of a CH10 packet (reserved bits, valid options, packet size limitation, filler use ...)
- For almost all data types the CSDW bits are checked for valid states and sometimes some level of content verification is provided.

It seems that existing COTS CH10 validators do not check the 100 ms limit for each packet for example or only for a very limited subgroup of CH10 data types. FLIDAS does this for all CH10 data types that are decoded into packet entries.

Although right now FLIDAS commits less checks than other COTS validators it seems to do some checks others do not do and checks up to the 2022 revision and provides a GUI to directly inspect the problems in detail.

FLIDAS handles some aspects of distributed CH10 multiplexer sources. Still many of these aspects break the IRIG 106 specification. The validation feature does not treat these additions as valid until they become part of the standard.

4.2.13.5 Disclaimer

When implementing the checks already some files from widely used recorders came up with issues that seem like they are not following the specification. However, in some of these cases it seemed reasonable to break the specification. So, in some cases the checks have been adapted to reflect this. But keep in mind if something pops up as an error it is an indication to investigate it. It can be valid or at least acceptable anyway.

4.3 Packet Entry Viewers

The visualization of packet entries within packets as shown in the Packet Entry View will in general be according to the following standard. Some packet entry viewers deviate from this and are described in the subchapters.

In the free version of FLIDAS the packet entries will always be shown in a table with a timestamp and a second column containing all further data in an unformatted text string.

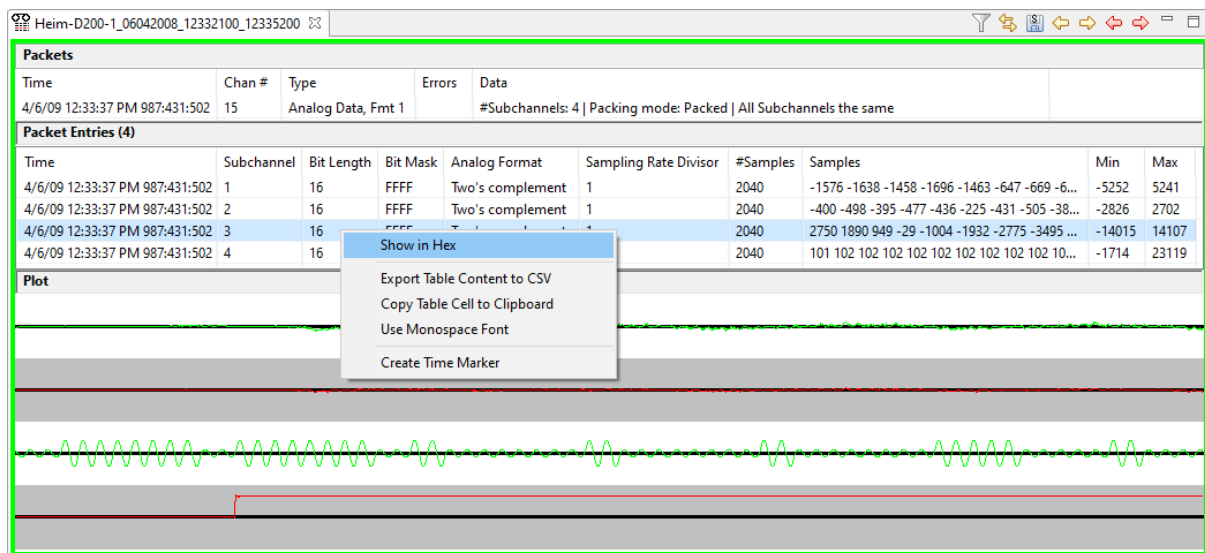
All commercial versions of FLIDAS contain more specific tables for the data that spread it out in separate columns and possibly offer other visualizations (like graphs) and further attributes. They also show packet entries with errors with a red background color (only errors directly indicated in IRIG 106 chapter 10 specification).

The packet entry viewers generally show the properties as indicated in the CH10 data type support section. Data payload will be limited to a reasonable amount (first 32 in most cases).

The property format (hexadecimal, binary etc.) is given in the column headers tooltips (text shown when hovering with the mouse over the header) if not decimal.

4.3.1 Analog Packet Entry Viewer

The packet entry viewer for the analog CH10 data type shows not only the decoded properties in a table but also shows a graph of each analog subchannel for visual recognition of the nature of the signal and possible anomalies.



If the packet entry view shows several packets, the packet entry table will list one packet entry for each subchannel of the first packet before continuing with the next packet and so on.

Each subchannel will get an own horizontal stripe for its graph including a horizontal axis for the 0 value.

The stripes will appear in the same sequence as for the first packet in the packet entry table above. They will contain the joined samples of all displayed packets of its subchannel.

The horizontal axis will spread the whole packets range over the whole width of the view.

The vertical axis will be scaled to the full bit range for integer samples (e.g., 0 ... 255 for unsigned 8 bit, -128 ... 128 for signed 8 bit (not -128 ... 127 to have symmetric range)). Floating-point samples will be scaled to a symmetric range around 0 so that the maximum/minimum value of the packet entries of the subchannel fits just in the scale (i.e., if the samples range from -3.2 ... 5.7 the scale will go from -5.7 ... 5.7).

Neither of the axes will show a scale.

If the graphs cause performance issues especially for very high sampling frequencies or the display of many packets at the same time use the [Packet Entry Viewers Preferences](#).

4.3.2 PCM Packet Entry Viewer

The packet entry viewer for PCM CH10 data type shows not only the decoded properties in a table but also the separation of the PCM data of the selected packets into PCM minor frames.

The screenshot shows the FLIDAS Packet Entry View interface. At the top, the title bar reads 'Heim-GSS-1_00970000_09030600_09032700'. Below the title bar, there are three main sections:

- Packets:** A table with columns: Time, Chan #, Type, Errors, Data. It contains two rows of PCM Data.
- Packet Entries (2):** A table with columns: Time, Data. It contains two rows of raw content in hexadecimal.
- Minor Frames:** A table with columns: Time, Minor frame sync, Major frame sync, FS, W1, W2, and several data columns. It contains multiple rows of frame synchronization data.

A context menu is open over the 'Packet Entries (2)' table, showing the following options:

- Show in Hex
- Export Table Content to CSV
- Copy Table Cell to Clipboard
- Use Monospace Font
- Create Time Marker

The table in the third area of the Packet Entry View will list the PCM minor frames with the following columns:

- A timestamp
- The state of the minor and major frame synchronization
- The hexadecimal frame synchronization pattern
- The decimal subframe word positions decoded from all subframe ID counters
- The hexadecimal data words. Each shows just enough hex digits to contain the individual data word's maximum bit length. You can limit the number of columns for data words in case of performance issues or operating system problems with the [General Preferences](#).

In case of a detected minor frame synchronization loss (PCM throughput mode only) the table will contain a red line with the timestamp where the synchronization loss was detected.

To create a [time marker](#) for PCM minor frame, right-click the line and select **Create Time Marker**.

4.3.3 UART Packet Entry Viewer

The packet entry viewer for the UART CH10 data type shows not only the decoded properties in a table but also shows text interpretation of the data for each subchannel.

The third area on the bottom labeled **Text Display** shows separate text areas for each subchannel. Each can be collapsed. The content will be the interpretation of the UART subchannel data according to a specific character encoding. This **Character encoding** can be selected on the top of the area. The available character encodings depend on your local computer but US-ASCII (7 bit) will always be available. This is also preselected and will interpret 8-bit wise (i.e., if the MSB is set it will show an unknown character symbol).

FLIDAS will convert any kind of recognized line break to one that will be visible in the display (this especially means that Windows (CR LF), Unix (LF) or Macintosh (CR) style line breaks should be visible on all systems).

The displayed text will also include text taken from packet entries with parity errors. When interpreting this data as text it is usually still human readable if there are single parity errors.

Note that chapter 10 only defines a storage format for 8-bit UART. All other (if stored) is undefined.

4.3.4 TSPI / CTS Format 0 and 2 Packet Entry Viewer

The packet entry viewer for the TSPI / CTS format 0 and 2 data type allows displaying the data of a packet entry either as hexadecimal raw data or as ASCII text.

The third area on the bottom labeled **Data Display Mode** allows choosing between ASCII or hex display.

4.3.5 TSPI / CTS Format 1 Packet Entry Viewer

The packet entry viewer for the TSPI / CTS format 1 data type allows displaying the data of a packet entry as hexadecimal raw data.

The third area on the bottom displays the joined data in a multi-line text display.

Note that the display should be improved as soon as the specification problems with format 1 are fixed. If you have sample data, you can help improving support for format 1 in FLIDAS.

4.4 Preferences

4.4.1 General Preferences

The General Preferences collect some settings that are globally active in the application that do not fit in somewhere else.

You can access these preferences by clicking **Window > Preferences > General**.

General

CSV column delimiter: ;

Locale: English (United States) - en_US

☐ Suppress scientific notation for numbers

☒ Use Unicode UTF-8 character encoding on text files

Scale for default column widths: 1.3

Printing line width [dots]: 2

☒ Limit dynamic table columns to no more than: 256

☒ Limit data word display to no more than: 32

The preferences are:

-
- **CSV column delimiter** defines a default to use to separate table columns when exporting data to a CSV text file.
 - **Locale** configures if FLIDAS uses a different localization than configured for your computer system. Mainly this controls how timestamps look and which decimal separator is used when displaying or exporting data or loading ICDs. If you change it, you must restart FLIDAS to take effect. The setting **(System default)** on top of the list removes FLIDAS specific localization and uses whatever is configured in your operating system.
 - **Suppress scientific notation for numbers:** In some situations FLIDAS uses a specific way of displaying numbers. If in these cases the numbers are very big or of a very small absolute value the normal display might not be very convenient. In these cases FLIDAS will use a scientific notation for the number (like 123.456E2). You can enable this option to leave the normal display instead. The intention of this option is mainly to allow staying compatible with the old number representation in all cases in case users created external software for parsing this output. In doubt, leave it off.
 - **Use Unicode UTF-8 character encoding on text files:** Makes sure that all text-based files will be treated as UTF-8 encoded Unicode files. This applies for example to ICDs, parameter sets, ICD mappings and exported CSV files. In result you can use Japanese, Chinese or Cyrillic letters for parameter names and engineering units for example or use additional mathematical symbols. If not enabled, your default system character encoding will be used. Some functions especially export file formats (e.g., MATLAB) might restrict the use of Unicode so FLIDAS must make adaptations there.
 - **Scale for default column width** allows scaling column widths on tables with a single preference. This applies only to tables where the column widths have not been individually configured. You can use this setting if you are using an operating system setup with bigger fonts. The Linux default for example is different from Windows. Increasing the value increases the column width proportionally.
 - **Printing line width [dots]:** Default line width used when printing on paper. The width is the number of dots in the selected printer resolution.
 - **Limit dynamic table columns to no more than:** Sometimes there are tables on the GUI that have many columns depending on the content (like a PCM minor frame with several hundred data words displayed in separate columns). Some operating systems or configurations have problems to display that many columns. Several hundred columns are usually no problem though. You can activate the check box and enter a number in the field to limit the tables to no more than this number of columns. The limitation will only apply to tables which have a dynamic number of columns i.e., depend on some external factor like the number of PCM words in a minor frame. If a table is limited, the last column will not contain data but an indication that further columns are missing.
 - **Limit data word display to no more than:** In some places of FLIDAS raw data words of things like messages are displayed (especially in Packet Entry Viewers). If you enable the check box and enter a number in the field, the data word output is limited to this number of words to avoid performance or resource problems. Operating systems might further limit the output themselves (Windows to 259 characters (not data words)). If data words have been suppressed due to the configured limit there will be a display of "..." in the end. Often there is a specific exporter (not the table exporter) available to export the data to a CSV file which will contain all raw data then.
-

4.4.2 Packet Entry Viewers Preferences

These preferences allow a global configuration of the packet entry viewers. They are available in all commercial versions.

To access the preferences click **Windows > Preferences > Packet Entry Viewers**.

The preferences are:

- **Suppress analog graphs:** The Analog Packet Entry Viewer plots a graph of the analog samples. This may cause performance issues if many samples are contained in the displayed packets. This option can suppress plotting the graph. If you activate it, you must define an upper limit on the number of samples inside the displayed packets that you want to plot. If the samples exceed this amount, no plotting will occur at all. The number of samples takes into account the samples of all subchannels inside all displayed packets.

4.4.3 Default Directories and Files Preferences

These preferences set up in which directories you start when selecting a file in a specific situation. You can still select files of other directories but you need to navigate to this directory yourself then.

The preferences also control which ICD mapping is preselected on start-up.

You can access these preferences by clicking **Window > Preferences > Default Directories and Files**.

The first setting controls the **Directory strategy**. This defines the mode by which you want to manage the default directories. FLIDAS distinguishes these directories by situations in which they are used in (like opening a CH10 file or exporting data). It keeps track of the directories you used the last time in each of these situations. You can define default directories for most of these situations. The directory strategy controls if you prefer to use the configured defaults or the history and it will fall back to the other type if no default or history is available for a specific situation.

The options for the directory strategy are:

- Prefer configured defaults over history
- Prefer history over configured defaults
- Use system default

You can configure default directories for these situations:

- Directory where CH10 files are loaded from.
- Directory where files are exported to
- Directory where ICD mappings are loaded from
- Directory where ICDs are loaded from
- Directory where parameter sets are loaded from

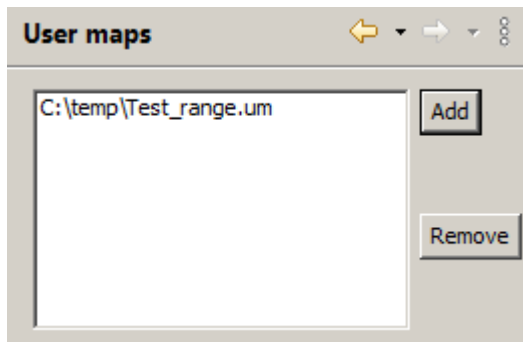
TMATS override files will be selected from the same directory as the selected CH10 file unless you select the "prefer history over configured defaults" directory strategy.

4.4.4 User Map Preferences

The User Map Preferences make maps available for features like the Navigation Data View or instrument panels with a map gauge. You can use this to replay your flight routes on a local map of your test range.

You can access these preferences by clicking **Window > Preferences > User Maps**.

These preferences are only available in the Parameter or Parameter Streaming Version



The preferences show a list of files that define your user maps. You can add files by clicking at the **Add** button and remove them by selecting a file and clicking **Remove**.

User map definitions are simple text files with a file extension of your choice. They define four values in the order given below. Each value must have its own line and have no additional whitespaces. The values must accord to your configured locale.

1. The latitude of the center of the map in degrees. + means north, - means south. Fractional values will be interpreted as degrees not minutes. Therefore, 3.50 means 3 degrees, 30 minutes.
2. The longitude of the center of the map in degrees. + means east, - means west.
3. The span of the latitude from bottom to top in degrees. E.g., if the top of your map has a latitude of 51 and the bottom of 50, then the span would be 1 degree. The value must be positive.
4. The filename and path of an image of the map. If the path is relative (or no path at all) then it is relative to the directory containing the user map definition file

The image of the map must face north on the upper border.

The projection of the map expects that one pixel in horizontal and vertical direction equals the same number of degrees in latitude as in longitude. For example, if 1 pixel equals 0.001 degrees in latitude, it must also equal 0.001 degrees in longitude. This is normally not true for maps covering a wider area and therefore the mapping of positions to the map image will get worse the larger the covered area is. Normally this is good enough to cover a test range. For wider areas you can use the default maps that offer a precise map projection.

The image formats PNG and JPG are supported.

CH10 sources so you can safely provide a default value if your CH10 sources do not include the TMATS attribute.

FLIDAS doesn't care if the attributes are newer than the IRIG 106 revision used for the rest of the TMATS setup. You can provide newer attributes.

You can access these preferences by clicking **Window > Preferences > TMATS Defaults**.

To set a value double-click it and enter the value. Setting an empty value means that the value will not be configured at all. The value you enter is a free text. It must conform to ASCII TMATS as defined in the Chapter 9 of IRIG 106.

Currently you can set default values for these attributes. The list might be extended when needed:

- R-x\AF-n-m: This is the analog format used to decode analog values. It is essential to decode analog data and usually the same for recorders of one manufacturer. Most of the time it is even the same for different manufacturers. The attribute causes problems often because it was removed by accident from the IRIG 106-2009 standard and then reintroduced in 2011. Recorders strictly conforming to TMATS 2009 therefore do not provide this attribute.

All displays and behavior in FLIDAS will be as if the values have been set directly in the TMATS files (only for CH10 sources processed after the setting and only if they do not contain the TMATS attribute themselves).

To completely exchange the TMATS setup of a file see also [TMATS overriding](#).

4.4.7 Data Streaming Preferences

The Data Streaming Preferences contain some settings for influencing the performance of [data streaming](#).

You can access these preferences by clicking **Window > Preferences > Data Streaming**.

These preferences should normally not be modified especially if data streaming is running well. If you encounter problems and you have a good understanding of the topic or if you trying to solve a problem in cooperation with the FLIDAS development team you can change them.

The preferences are:

- Receive preferences:
 - **Maximum UDP receive payload to expect:** When FLIDAS is receiving UDP packets from data streaming it will be able to receive a payload of not more than the given number of bytes. Larger numbers create more CPU and memory load. IRIG 106 chapter 10 defines a maximum of 32726 until the 2015 standard and 65507 from 2017 on so the latter would be a safe setting but probably causing unnecessary performance overhead. If you know your devices are using lower payload sizes you can reduce it.

- **Desired UDP receive buffer size:** FLIDAS is not running on a real-time operating system. When an UDP packet is received by the computer FLIDAS will most of the time not immediately be able to process it. The operating system will buffer it until FLIDAS has time to pick it up. By default the operating system only buffers a small amount of data and then starts throwing data away (this is permitted for UDP). With this setting you can request a bigger buffer on the operating system. However, some operating systems have a maximum size that they are going to allow an application to use. So, your configured number might not be what you get. The actual number used can be seen for example in the [General Live Data Stream Information Properties](#). The default value is 4 megabytes so you must configure the value 4194304. Higher values are always desirable if you can afford the additional memory and it will likely further reduce UDP packet drops if this occurs at all.

*Note: To get a feeling about the required size you can use this rule of thumb: Assume you have a 100 MBit/s peak data rate coming in. This amounts to 12.5 MBytes/s. There will be some protocol overhead so let's assume 15 MBytes/s. Neither your operating system is real-time nor is Java real-time. Assume there will be complete durations like 100 ms where FLIDAS is not able to service the incoming UDP traffic. This may be due to operations of your operating system or due to issues within Java like regularly occurring "garbage collection" or single occurrences of "just in time compilation". Within these 100 ms 1.5 MBytes of UDP packets would accumulate. If the receive buffer is less the data will be lost. **So a rule of thumb could be to divide the peak data rate in MBit/s by 50 and use this as MBytes for receive buffer size.** This would provide for 133 ms of FLIDAS not processing incoming UDP packets without loss. Depending on your system you might need to be prepared for longer periods without servicing incoming traffic.*

Note: On Linux your maximum value is normally quite limited. It is sufficient for normal PC work but not to receive several hundred MBits of incoming UDP traffic. You can increase the operating system maximum by running this command as root `sysctl -w net.core.rmem_max=4194304`. Change the number to the number of bytes you want. The command just changes the buffers for the current session. You can do permanent changes by adding `net.core.rmem_max=4194304` to the `/etc/sysctl.conf` file. Do this on your own risk and read about the setting `net.core.rmem_max` in the Internet before. You should especially not reduce it in case your system admin already used a higher value.

- **UDP packet timeout [ms]:** If a UDP packet is missing in a sequence it is possible that it will be received out of sequence later. This is allowed in the UDP protocol. FLIDAS will therefore wait and reorder the UDP packets to a proper sequence. Since it is allowed that UDP packets do not arrive at all it is not possible to wait too long for it. You can configure the time after which FLIDAS will decide that the packet is lost and works without it. If you configure this time too low, you will drop more packets than necessary because they might still arrive after this time. If you choose a too high value, you will create additional delay on the processing and additional memory and CPU load for buffering more incoming data. The default value is 100 ms. If

your network will never deliver UDP packets out of order you can also set this value to 0. A possible situation for this is if you publish and receive the data stream on the same computer. In theory the packets might still be out of order but it is so unlikely that you could accept a lost UDP packet if it occurs.

- **UDP sequence timeout [ms]:** FLIDAS processes incoming UDP packets based on sequence numbers. If the sequence changes suddenly and drastically, FLIDAS will not process the packets. In some cases you might want to still process the data. An example may be that you have a device that continuously transmits data but stops transmission for a short time during a power loss. After the power comes back it starts transmission again but the sequence numbers will not be consistent with the previous stream. You can now define for example that a transmission gap of 2000 milliseconds or more causes FLIDAS to accept new sequence numbers. This does not harm also in normal situations as FLIDAS of course also continues to work if sequence numbers consistent with the previous ones appear. Also most normal streams have time packets at least transmitted once a second so the situation will not occur. This is also the reason why you cannot configure the timeout value to less than 1200 ms. Be aware that FLIDAS just accepts the stream for further UDP reception. It does not mean it is valid in any other way. In the example a power loss of the device would likely also cause the RTC to change drastically which corrupts a lot of FLIDAS functionality. However, if you run a CH10 recording you will not miss packets but record the whole situation also at the time of power loss. You may be able to use this recording for analysis. The default timeout value will be 2000 ms. Whenever a timeout occurs a gap is counted in the General Live Data Stream Information Properties.
- **Drop out of sequence UDP packets:** As described with the last option, UDP packets are allowed to arrive out of sequence. FLIDAS is reordering them to the proper sequence which causes additional processing and memory use. If on your network out of sequence delivery is at a tolerable level for your application you can decide to drop the UDP packets out of sequence immediately and skip the additional processing. This reduces memory and CPU load and results in better latency at the cost of more packet loss. The General Live Data Stream Information Properties show how many times out of sequence UDP packets arrived on a running stream. Consult this to estimate the impact of enabling this option.
- **Transmit preferences:**
 - **Maximum UDP transmit payload:** This configures the maximum payload size of the UDP packets used when publishing a data stream. Larger numbers may cause IP fragmentation depending on your network environment but offer better performance with FLIDAS since parts of the packet handling is done on low level in the operating system or network hardware. Low values will create a lot of overhead on the transmission and additional work reconstructing the data. The maximum number allowed by IRIG 106 chapter 10 is 32726 until the 2015 standard and 65507 from 2017 on. The first value is also the default value as this works in both standards and makes good use of the network.
 - **UDP packet transmit queue size:** When publishing a data stream FLIDAS will load some data in advance and keep it ready for transmission at the correct time. This value defines how many UDP packets shall be kept ready at maximum. Higher values can help transmitting timely despite concurrent system activity but need more memory. FLIDAS uses 1000 by default.

- Data stream task preferences:
 - **Stop data stream tasks on overload:** FLIDAS makes some internal evaluation if a specific data stream task causes an overload. See also the overload indicator described in the Repository View. You can use this option to automatically stop a data stream task that is detected to be overloaded. Use this with care as the definition of overloading is not precise and what is an overload in one environment may be OK in another.

4.4.8 User Defined Time Representations Preferences

The User Defined Time Representations Preferences allow customizing time stamps for time representations mainly used in exports. It is only possible to customize time representations with text (i.e., String) output.

You can access these preferences by clicking **Window > Preferences > User Defined Time Representations**.

These preferences are available in all commercial versions.

The preferences show a table of user defined time representations with the following columns:

- **Name:** The name for this time representation as it will appear for selection in the time representations wizard page and used as description for the time stamp in export files.
- **Format pattern:** This describes how the time stamps look like. The first part defines the output of date and time up to milliseconds. The optional second part can attach micro and nanoseconds at the end. If you use the optional second part, you must start it with an '@'.

Here is a list of letters you can use in the definition of the first part (case sensitive!). In general, for numbers a single letter means just the number while a repetition of the letter means that as many digits as letters will be used, adding leading 0 in front if necessary. An example follows further down:

Letter	Meaning	Comment
y	Year	Two letters mean last two digits of the year, all other numbers will use 4 digits year
M	Month in year	One letter means just the number of the month. Two letters will add a leading 0 for the months 1-9. Three letters or more will use the month's name
D	Day of year	
d	Day in month	
E	Day name (like Monday)	
a	AM/PM marker	
H	Hour in day (0-23)	

k	Hour in day (1-24)	
K	Hour in am/pm (0-11)	
h	Hour in am/pm (1-12)	
m	Minute in hour	
s	Second in minute	
S	Millisecond	

The optional second part after the '@' can be any text but the letter 'u' is replaced by the microseconds (of the millisecond, i.e., values will range from 000 to 999) and the letter 'n' by the nanoseconds (of the microsecond i.e., values will range from 000 to 999). A single 'u' or 'n' will be replaced just by the number with as many digits as it needs. If you type several 'u' or 'n' in a row it will always use at least that many digits, padding with 0 on front if necessary. I.e., for 5 nanoseconds the definition 'n' outputs 5, but for 'nnn' the output is 005.

Examples:

Format pattern	Output
HH:mm:ss.SSS	23:59:47.123
DDD-HH:mm:ss.SSS@uuunnn	031-23:59:47.123456789
HH:mm:ss SSS@:uuu:nnn	23:59:47 123:456:789
d.M.y HH:mm:ss SSS@:uuu:nnn	31.1.2019 23:59:47 123:456:789
EEEE hh:mm:ss.SSS a	Thursday 11:59:47.123 PM

- **ID:** A unique identifier to identify this time representation. May be any text if it is unique within your FLIDAS installation. Example: *timeOnlyMillisecondsPrecision*
- **Example:** Shows an example time stamp displayed with the configured format.

The buttons below the table allow you to **Add**, **Remove** or **Edit** a time representation. To edit or remove you must select one in the table before. When adding or editing three dialog boxes will open one after the other to enter the data. Only if you confirm all three with OK the adding or editing will take place.

None of the entered data may contain the characters ';' or '#'.

4.4.9 Predefined Time Channel Offsets

These preferences allow predefining offsets between time channels in a CH10 source. You use it to predefine different setups for the [time channel offsets wizard page](#). This is also where you can see the format description. This helps you with reoccurring setups like that aircraft type A always uses UTC time airborne but GPS time on ground recording.

There are several buttons below the table:

- **Add** adds a new predefined setup. You have to enter the name, seconds and hour/minute offset in three different windows.
- **Edit** modifies the predefined setup that is currently selected in the table. You can also double click the entry to edit it. You can modify the name, seconds and hour/minute offset in three different windows.
- **Remove** removes the selected predefined setup.
- **Up** and **Down** moves the selected predefined setup up and down in the list to organize it to your preference.

The name of the setup must be unique. If you modify the name your preference in the CH10 file loading or creation of live data stream configuration might be broken.

4.4.10 Parameter Plug-in Preferences

The Parameter Plug-in Preferences collect global settings about functionality of the Parameter Plug-in.

You can access these preferences by clicking **Window > Preferences > Parameter Plug-in**.

The preferences are:

- **Prefer ICD min / max over autoscaling:** Most ICDs allow users defining minimum and maximum values for parameters. When parameter values are plotted in a graph for example, the graph shows the values over time in an initial value range from the minimum to the maximum value of the ICD if this option is enabled. Otherwise mostly the autoscaling would be used (minimum and maximum observed value). The graphs allow manual switching between different options as well, so this is just the initial setting when opening the graphs.
- Preferences on performance warnings for the Parameter Graph Viewer or Parameter Table Viewer. Displaying a lot of samples there can create memory problems or need a lot of time for displaying the graph/table. Therefore you can enable warnings before you execute it and configure limits for the warnings:
 - With **Warn on parameter graph/table with more parameters than:** you can enable a warning based on the number of parameters selected for display. This warning will be issued before collecting the samples and allows canceling.
 - With **Warn on parameter graph/table with more total samples than:** you can enable a warning based on the total number of samples collected for all selected parameters. This warning will be issued after collecting the samples and already spending some time. It allows canceling before starting the display. The number of samples might be lower than what you expect when looking at the sample rates as the parameter viewers may remove repeated samples in some sampling modes.
- **Maximum number of curve marks to display** limits for the Parameter Graph Viewer up to how many samples on each curve can be marked separately. This is to avoid performance and memory problems. Remember that marking multiple thousand samples on a two thousand pixel wide screen makes no sense.

4.4.11 GUI Scaling (Troubleshooting) Preferences

When you set up Windows to scale applications to other scales than 100% it will try to modify the graphics output. Although FLIDAS 4.10 should fix the problems it should be prepared for irregular behavior depending on system settings like Windows version, display types, screen resolution, multiple display configurations, configured windows scaling and Java version. This preference page should help you if you still have problems. In other cases, just leave this page alone.

Any changes made on this page will only properly apply if you restart FLIDAS. Also, if you change the scaling within your windows setup while FLIDAS is running you should restart it.

If you have a bad output, you can provide the data from the **System Information** section of the preferences to Data Bus Tools for help. Also, send a screenshot and description of where you experience problems and your configured windows scaling, Windows Version, Java Version and screen resolution. In return you might get some values to put in the scaling input fields on top of the preference page.

There is one additional function here: The scaling of your windows system will influence your individual column setup of tables. So, if you are unhappy with a setup you made before FLIDAS 4.10 you can reset all individually configured column widths (not order) to the default value by clicking at the **Reset all configured table column widths** Button.

4.5 Properties

4.5.1 General CH10 Source Information Properties

These properties show information about a CH10 source.

The properties can be seen when you right-click a CH10 source in the Repository View and select **Properties > General CH10 Source Information**.

General CH10 Source Information

Filtered packets:

5,591

Start time:

4/17/09 1:22:30 PM 997:206:000

End time:

4/17/09 1:22:44 PM 551:309:100

Duration:

0:00:13

MSIDs were ignored:

No

Secondary header times were ignored:

No

Used time channel IDs:

640

Time offsets:

Domain	Seconds added	Hours / minutes added
PTP time (secondary header)	19	+00:00
MSID 0 (time channel)	-18	+00:00
MSID 1 (time channel)	0	-02:30

Problems with channel definitions:

No

Packet sync errors:

0

Packet header checksum errors:

0

Secondary header checksum errors:

0

Data checksum errors:

0

RTC sync errors:

0

Data overflow errors:

0

Lost packets errors:

0

Any packet errors:

0

Data checksums were checked:

Yes

Erratic RTC:

No

Color

Pick Color

The properties are:

- The amount of packets that have been processed after applying the CH10 data type filters
- The start and end time of the CH10 source will be derived from the minimum and maximum time of all valid CH10 packet headers excluding computer-generated packets (TMATS, recording indices / events, user defined). Since one packet may contain up to 100 ms of data, the end time will be one million RTC ticks behind the maximum RTC.
- The duration of the recording (based on previous definition of start/end time)
- If MSIDs have been ignored when opening the CH10 file or creating the live data stream
- If secondary header times have been ignored when opening the CH10 file or creating the live data stream
- The time channel IDs that have been used. This is up to one time channel for each MSID (see distributed CH10 multiplexer sources) used as reference for RTC time decoding.
- Which time offsets were configured to be added to time packets of each MSID or secondary header PTP or chapter 4 time. The offsets are broken down to a multiple of

15 minutes offset (likely used for time zone correction) and an additional seconds offset (likely used for UTC / TAI / GPS time offsets).

- If problems have been detected in the TMATS setup of channels or subchannels. See the [Channel View](#) for details.
- The numbers of packet errors detected of the following types:
 - Packet sync
 - Packet header checksum
 - Secondary header checksum
 - Data checksum (if check is configured))
 - RTC sync (if you use secondary header timing and your system uses RTC sync errors in this case there is nothing to worry about)
 - Data overflow
 - Lost packets (i.e., the number of times a sequence number was not OK, not the actual number of missing sequence numbers)
 - Any of the above errors detected in a packet
- If the data checksums of packets have been checked (see [CH10 file loading wizard](#) or [creating a live data stream](#))
- If processing the CH10 source detected an erratic RTC, i.e., if the RTC made an unrealistic jump forward or backward (more than 2 seconds back or more than 1 hour forward)
- If the [TMATS file has been overridden](#) the file is shown.
- A button to select a new color to associate with this CH10 source (will be used in [Views](#) related to this CH10 source).

4.5.2 General File Information Properties

These [properties](#) show information about a loaded CH10 file. The properties common between CH10 files and live data streams are found in the [General CH10 Source Information Properties](#)

The properties can be seen when you right-click a file in the [Repository View](#) and select **Properties > General File Information**.

The properties are:

- The file name and path
- The loading result (described in the [CH10 files](#) section)
- The file size and information if you used [partial reading](#)
- If the loading was configured to search for the start of a new packet after a wrong packet has been found.
- The location of the property file containing the [time markers](#) if enabled.
- For files using [Safran CH10 file segmentation](#) an information if the segments have been limited to a certain count and how many segments have been tried to load (if you aborted the import before all was loaded it is still the full value). There is also a table showing information on each segment:
 - The name of the file
 - The time stamp of the first packet that starts in the segment (remember, packets are not totally ordered in time within a CH10 file. They might be out of order for up to one second. Also computer generated data packet time stamps are sometimes not in sync)

- The time stamp of the last packet that starts in the segment (see comment before as well)
- The segment size in MBytes

4.5.3 General Live Data Stream Information Properties

These properties show information about a live data stream.

The properties can be seen when you right-click a live data stream in the Repository View and select **Properties > General Live Data Stream Information**.

The properties are:

- The live data stream status
- If channels have been synchronized in time or if they are processed in the order they arrive
- The network interface if selected
- The IP address if this is a multicast stream
- The UDP port
- The data streaming format that the reception locked to (if it locked already) (see CH10 Data Stream Format Wizard Page)
- The source ID that the reception locked to (if it locked already) or it is configured to (see CH10 Data Stream Format Wizard Page)
- The UDP receive buffer size (see the Data Streaming Preferences for an explanation)
- The number of UDP packets that have been processed
- The number of UDP packets that have been dropped because they did not arrive in time or did not arrive at all. *Note that packet losses are allowed in the UDP protocol. However, if the number is unreasonably high you should consider increasing the UDP receive buffer size (see above).*
- The number of times a UDP packet was received that was not in sequence order with the previous packet.
- The number of times a longer gap was detected in the stream (see UDP sequence timeout in the Data Streaming Preferences)
- The number of CH10 packets that have been reconstructed from the UDP packets but have not yet been processed. A high and increasing number here shows an overload situation.
- The size of all CH10 packets that have been reconstructed from the data stream

4.5.4 General TMATS Attributes Properties

These properties show general TMATS attributes decoded from a CH10 source.

The properties can be seen when you right-click a CH10 source in the Repository View and select **Properties > General TMATS attributes**.

General TMATS Attributes

Program name:troublesome mux

Test item:None

Information

IRIG 106 revision: 5

Origination date: 09-02-2008

Revision number: 21

Revision date: 04-17-2009

Test number: 1

Data source identification

Active	ID	Type	MSID bits	Manufacturer	Model	Security classification	Original recording date	Original reco
Yes	DS_troublesome_mux	Other	3	TTC	Other			No

Test information

Security classification:Unclassified

Generated by TTCWare 3.25.11 Beta on 04-17-2009 at 09:15:20

Collected comments:

The properties are only shown when the TMATS attribute was defined. All TMATS attributes from the *General Information Group* of IRIG 106 Chapter 9 except the checksum are decoded. In addition the following TMATS attributes are displayed for each data source:

- Recorder-reproducer manufacturer (*R-x\RI1*)
- Recorder-reproducer model (*R-x\RI2*)
- Original recording (or modified recording) (*R-x\RI3*)
- Original recording date and time (*R-x\RI4*)
- Number of source bits (*R-x\NSB*) (also known as number of MSID (multiplexer source ID) bits). It is also displayed as 0 if it is not defined in TMATS.
- If it is the active data source. TMATS files can describe several data sources and when loading the CH10 file or creating a live data stream you defined which one to use by identifying it with the data source ID.

All comments (*G\COM*) are displayed in the order as they appear but are not displayed in their context where they appeared.

4.5.5 CH10 Data Type Filter Properties

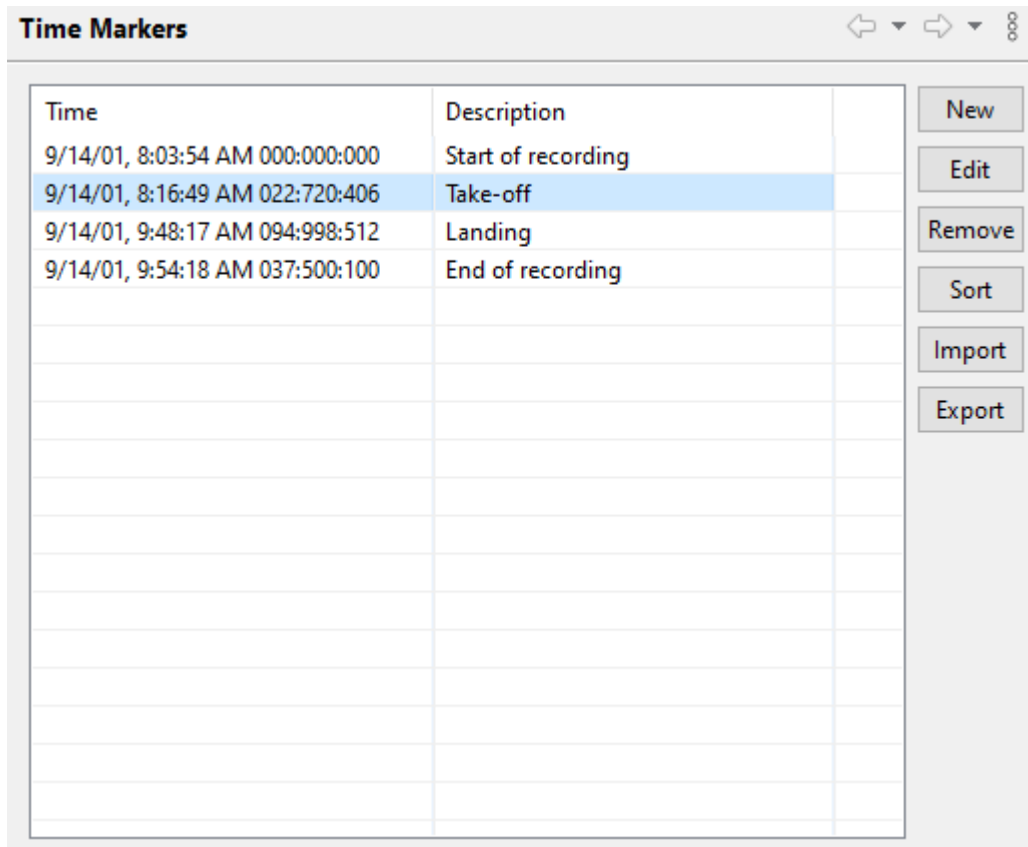
These properties show which CH10 data type filters were active when processing a CH10 source.

The properties can be seen when you right-click a CH10 source in the Repository View and select **Properties > CH10 Data Type Filters**.

4.5.6 Time Marker Properties

These properties allow managing the time markers associated with the CH10 file.

The properties can be seen when you right-click a CH10 source in the Repository View and select **Properties > Time Markers**.



You can edit a time marker by double-clicking or selecting it and clicking on **Edit**.

You can create a new time marker by selecting it and clicking on **New**. The default time will be the start time of the CH10 source. This is to show the time stamp format so you can easily edit it.

You can remove a time marker by selecting it and clicking on **Remove**.

You can sort the time markers by clicking on **Sort**. This only has impact on what you see. After you apply your changes, they will be stored in an ordered fashion anyway.

You can export the time markers to a CSV file by clicking on **Export**. You have to choose a file name and CSV column delimiter. You can also apply hour and second offsets to the time markers that are being exported. You can use this to correct time zone offsets and/or leap seconds differences. The export will contain two columns:

1. The time stamp as an ISO date and time format (ISO 8601) without time zone. An example would be: 2001-09-14T09:54:18.037500123
2. The description

You can import time markers from a CSV file by clicking on **Import**. You have to choose a file name and CSV column delimiter. You can also apply hour and second offsets to the time

markers that are being imported. You can use this to correct time zone offsets and/or leap seconds differences. The format must be like on the export above.

Duplicated time stamp / description combinations are not allowed and will be silently removed.

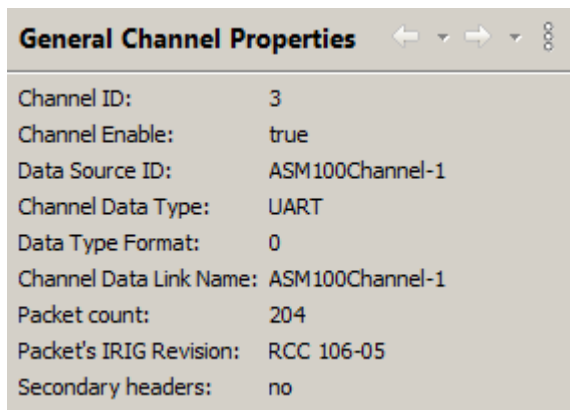
Note that time stamps can be part of the file properties and can be saved automatically if you enable this in the CH10 file loading wizard. The intention of export / import is to allow sharing data with external applications.

4.5.7 General Channel and Subchannel Properties

These properties show TMATS attributes and other information about a single channel or subchannel.

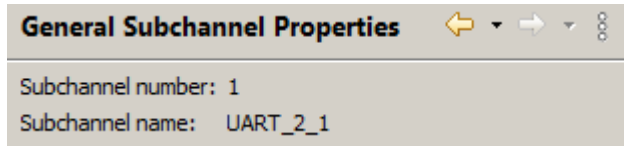
The properties can be seen when you right-click a channel or subchannel in the Channel View and select **Channel Properties > General Channel Properties** or **Properties > General Subchannel Properties**.

Channel Properties are also available with a right-click on a packet of the Packet View or a right-click on the packet area heading line in the Packet Entry View and selecting **Channel Properties**.



The general Channel Properties are (for reference the TMATS-ASCII codes are given in italics):

- Track number / channel ID (*R-x\TKI-n*)
- Channel enable (*R-x\CHE-n*)
- Data source ID (*R-x\DSI-n*)
- Channel data type (*R-x\CDT-n*)
- Data type format (these are individual attributes of each CH10 data type like: *R-x\ATF-n*, *R-x\ABTF-n*, *R-x\VTF-n* ...)
- Channel data link name (*R-x\CDLN-n*)
- Channel data type (*R-x\CDT-n*)
- Packet count: The number of packets that have been read (taking the CH10 data type filters into account)
- Packet's IRIG revision: The IRIG 106 revision that the packets stick to.



The general Subchannel Properties are:

- The subchannel name
- The subchannel number. Note that some subchannels start their numbering with 1 while others start counting by 0. FLIDAS sticks to the numbering system of IRIG 106 but some recorders seem to produce data that uses one numbering system in TMATS and one in the data which will lead to an invalid mapping of TMATS and CH10 data.

In all commercial versions more pages show individual properties for some [CH10 data types](#). These are described in the [CH10 data type support](#) chapter. The free and commercial versions also show some of these properties directly in the [Channel View](#) in the **Additional properties** column.

4.5.8 General Packet Properties

These [properties](#) show information that is common to all types of packets. They are available in all commercial versions.

The properties can be seen when you right-click a packet in the [Packet View](#) and select **Packet Properties > General Packet Properties**.

Packet properties are also available with a right-click in the [Hex Packet View](#) and selecting **Packet Properties**.

General Packet Properties

Packet header

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	25	E..	41	00	10	08	00	00	F4	07	00	00	02	1F	03	38
0010	34	D..	8..	11	07	00	34	38								

Packet sync [hex]: EB25
Channel [dec]: 65
Packet length [dec]: 2064
Data length [dec]: 2036
Data type version [hex]: 02
Data type version: RCC 106-05
Sequence number [dec]: 31
Packet flags [hex]: 03
Secondary header: Absent
IPTS time source: Packet header 48-bit RTC
RTC sync error: No
Data overflow error: No
Secondary header time format: Chapter 4 binary weighted 48-bit time format
Data checksum type: 32 bit checksum
Data type [hex]: 38
Data type: ARINC 429 Data, Fmt 0
RTC [dec]: 30359082036
Header checksum [hex]: 3834
Header checksum (calculated) [hex]: 3834

Packet trailer

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	B..	D..	22	F3												

Data checksum [hex]: F322D5B5
Data checksum (calculated) [hex]: F322D5B5

Other properties

File offset: 20560712
Effective time: 4/7/09 9:02:48 AM 766:116:102
Effective time from RTC: 4/7/09 9:02:48 AM 766:116:102
Original time from RTC: 4/7/09 9:03:06 AM 766:116:102
Delta RTC to previous packet on channel: 948945
Delta time to previous packet on channel [ns]: 94894661
Delta RTC to next packet on channel: 950100
Delta time to next packet on channel [ns]: 95010161
Delta sequence number to previous packet: 1
Delta sequence number to next packet: 1

The General Packet Properties are organized by the sections inside the CH10 data structures of a packet. They decode all data defined in the chapter 10 standard for these sections and also some calculated properties:

-
- Packet header:
 - A hexadecimal byte by byte representation of the 24 header bytes
 - All properties defined in the chapter 10 document
 - The calculated checksum
 - When the checksum or packet sync was wrong or any packet flag indicated an error, the property will be highlighted in red.
 - Optional secondary header:
 - A hexadecimal byte by byte representation of the 12 secondary header bytes
 - All properties defined in the chapter 10 document
 - The calculated checksum
 - When the checksum was wrong, the property will be highlighted in red.
 - There may be different time stamps:
 - Effective secondary header time: This is the time stamp used in FLIDAS for the secondary header. It is the time stamp read from the secondary header possibly manipulated by the time channel offsets.
 - Original secondary header time: This will only be shown if the time stamp above is a manipulated one. It will then show the original one.
 - Optional packet trailer:
 - A hexadecimal byte by byte representation of the trailer bytes
 - All properties defined in the chapter 10 document
 - The calculated checksum
 - When the checksum was wrong, the property will be highlighted in red.
 - Other properties:
 - The position of the first packet byte within the file (only for CH10 files)
 - Some absolute time stamps:
 - Effective time: This is the time stamp used in FLIDAS for the packet. If the packet has secondary headers and FLIDAS is configured to use the secondary header times, this is the effective secondary header time. Otherwise it is the effective time from RTC (see below). In situations with corruption (secondary header checksum error, RTC sync error) it will be other artificial time stamps. For valid time packets the displayed time is the time stamp from the time packets data section after manipulation for time channel offsets.
 - Effective time from RTC: This is the time stamp used in FLIDAS for the RTC time. It is the original RTC time (see below) possibly manipulated by the time channel offsets.
 - Original time from RTC: This is the time calculated from RTC based interpolation on the time channel for this channel's MSID (see distributed CH10 multiplexer sources) based on unmanipulated time packets.
 - The RTC difference to the previous and following packet on the same channel (positive values). Note that the previous and following packets are not available for the computer-generated data packets.
 - The time difference in nanoseconds like above
 - The sequence number difference like above
 - Negative RTC differences or sequence number differences unequal to one will be marked red.

More pages show individual properties for all packet types in decoded form and the raw channel specific data word. These individual properties are described in the CH10 data type

support chapter. The free and commercial versions also show some of these properties directly in the Packet View in the **Data** column.

4.6 CH10 Data Type Support

All CH10 data types will be recognized and their packets will show up with a timestamp and the packet level properties in the Packet View. However, only some CH10 data types are supported further so that they are decoded into packet entries. The following section will describe which properties are decoded and are therefore available for display or exporting.

4.6.1 ARINC 429

ARINC 429 packets have these properties:

- The timestamp
- The number of contained messages

ARINC 429 packets will be decoded to single ARINC 429 messages with these properties:

- The timestamp
- The bus speed (low or high speed (LS/HS))
- The number of the subchannel (1-based) within this IRIG 106 channel
- The parity bit
- The SSM
- The SDI
- The label number
- The 19 bit data
- An indication when a parity error was detected
- An indication when a format error was detected

4.6.2 MIL-STD 1553

The MIL-STD 1553 packets have these properties:

- The timestamp
- The number of contained messages
- A definition of the bit within the message that the timestamp refers to

MIL-STD 1553 packets will be decoded to single MIL-STD 1553 messages with these properties:

- The timestamp
- Bus A or B of the dual redundant buses
- The message type
- The receive and transmit command and status words if present. *Note: The mode commands will be classified as transmit or receive commands depending on the transmit/receive bit. This means that for example transmit command words for the broadcast RT can be displayed.*
- The command words additionally decoded to RT address, subaddress and word count
- The status words additionally decoded to the RT address and flags

- The type of the mode code if it was a mode code
- The number of data words that have been transmitted
- The data words
- The detected errors (format, invalid word, message, response timeout (Rx), response timeout (Tx), sync type, word count)
- The duration of the first and second gap within the message (if any) in 100 nanoseconds (time between data / command word and 1st and 2nd status word)

If a format, sync type or invalid word error occurred, the words on the bus cannot properly be decoded into command, status and data words. Instead all words on the bus will be displayed as data words.

MIL-STD 1553 will always be interpreted as MIL-STD 1553B. The MIL-STD 1553A variant is not explicitly supported. In some areas the features may be usable for the A variant. In doubt contact us for further discussion.

4.6.2.1 MIL-STD 1553 16PP194

The MIL-STD 1553 16PP194 packets have these properties:

- The timestamp
- The number of recorded transactions

MIL-STD 1553 16PP194 packets will be decoded to single MIL-STD 1553 16PP194 transactions with these properties:

- The timestamp
- The detected errors on transaction level (transaction, message timeout, status, echo)
- An indication if a 16PP194 bus master reset assertion was detected
- Then all six 16PP194 words of the transaction will be decoded with these properties:
 - The bus(es) the word was on. CH10 may also encode "Incomplete transaction" here. In this case FLIDAS treats this word as not existing.
 - The gap duration
 - The RIU address and subaddress as decoded from the 16PP194 word
 - The 16-bit data field
 - The detected errors on the word (Manchester coding, parity)

The transaction will be considered invalid if any of the transaction or word level errors occurred or any of the six words were missing.

4.6.3 Ethernet

Ethernet packets have these properties:

- The timestamp
- The number of contained frames
- The Ethernet packet type
- A definition of the bit within the message that the timestamp refers to

Ethernet packets will be decoded to single Ethernet frames with these properties:

- The timestamp
- The physical network ID (1-based)
- The speed
- An indication for the following errors in the frame:
 - Frame
 - Frame checksum
 - Data length
 - Data checksum
- The definition of what has been recorded (full frame vs. payload only). If only the payload is recorded, some analysis features of FLIDAS are limited on the data.
- The number of bytes in the frame
- The bytes of the frame

4.6.4 ARINC 664 (AFDX) / UDP Payload

The IRIG 106 standard supports AFDX packets and an unspecified Ethernet UDP payload within the same data type. Most likely AFDX will be the most common in the future.

AFDX packets have these properties:

- The timestamp
- The number of contained messages

AFDX packets will be decoded to single AFDX messages with these properties:

- The timestamp
- The source and destination IP addresses and ports
- The virtual link
- An indication when an error occurred in the frame
- An indication when the data was simulated data
- The number of bytes in the frame
- The bytes of the frame

4.6.5 MPEG Video

MPEG video exists in three format variants in IRIG 106 chapter 10. All of them are handled quite similar in FLIDAS.

Note: The formats 1 and 2 offer the option to have an MPEG program stream or a transport stream. No sample files were available to us to test with program streams. Some limited tests have been done with regular MPEG files. So, the program stream decoding is considered experimental and delivered for free. However, there is no known recorder producing program streams in CH10 files and experts argue it will never occur.

Note: MJPEG and MJPEG 2000 videos are handled differently in FLIDAS.

Note: In 2022 format 1 has been deprecated and some properties on format 0 and 2 have been removed from the CH10 data. FLIDAS treats the deprecated format 1 the same as in the previous IRIG106 releases except that it indicates an error on CH10 file verification. Format

0 and 2 will not decode the removed properties but the packets will still show the same type name as on the old versions.

Video channels show these properties if available (TMATS-ASCII codes given in *italics*):

- An indication if the stream is a program or transport stream
- Constant bit rate (*R-x\CBR-n*)
- Variable peak bit rate (*R-x\VBR-n*)
- The video encoding delay in milliseconds (*R-x\VED-n*)

All video packets have these properties:

- The timestamp
- An indication when a timestamp is encoded within the IRIG 106 packet entries
- An indication if the MPEG system clock reference is synchronized with the real-time clock of the recorder
- An indication when a timestamp is encoded within the MPEG data (removed in 2022)
- An indication when KLV metadata is present in the MPEG-2 data (removed in 2022)

Video packets of format 0 (MPEG-2/H.264, since 2022 named "Legacy" instead) have these additional properties:

- An indication if the video stream uses big or little-endian 16-bit byte alignment
- The MPEG-2 payload type (removed in 2022)

Video packets of format 1 (ISO 13818-1 MPEG-2) is deprecated since 2022. Still it is decoded by FLIDAS with these additional properties:

- An indication if variable or constant bit rate has been used
- An indication if the stream is a program or transport stream
- The MPEG-2 profile and level of the encoded bit stream
- The number of MPEG-2 packets (only if provided from the recorder)

Video packets of format 2 ((ISO 14496 MPEG-4 part 10) AVC/H.264, since 2022 named "MPEG-2 Stream" instead) have these additional properties:

- An indication if the stream is a program or transport stream
- An indication if variable or constant bit rate has been used (removed in 2022)
- The MPEG encoding level of the encoded bit stream (removed in 2022)
- The MPEG encoding profile of the encoded bit stream (removed in 2022)
- The type of the contained audio (removed in 2022)
- The number of MPEG-2 packets (only if provided from the recorder) (removed in 2022)

The packets are decoded into single MPEG packets with these properties:

- The timestamp (packet time stamp if no intra-packet timestamp)
- The bytes of the data. Different byte alignment has already been handled.

4.6.6 UART

UART subchannels show these properties if available (TMATS-ASCII codes given in *italics*):

- Baud rate (*R-x\UCR-n-m*)
- Bits per word (*R-x\UCB-n-m*)
- Parity (*R-x\UCP-n-m*)
- Stop bit (*R-x\UCS-n-m*)
- Interface (*R-x\UCIN-n-m*)
- Block size (*R-x\UCBS-n-m*)
- Sync word length (*R-x\UCSL-n-m*)
- Block sync value (*R-x\UCSV-n-m*)
- Block rate (*R-x\UCBR-n-m*)

The UART packets have these properties:

- The timestamp
- An indication when the individual UART packet entries have separate timestamps

The packets are decoded into UART messages with these properties:

- The timestamp (if there is no timestamp for the packet entries, they will share the timestamp of the packet)
- The UART subchannel number (1-based)
- The number of data bytes
- The data bytes
- An indication when a parity error was detected

4.6.7 Analog

The support is for analog format 1 and format 3 (I/Q Recording of intermediate frequency) data.

FLIDAS features for analog format 1 can be used for format 3 as well. Still you have to consider the high sampling rate which will not make everything a reasonable feature for format 3 (e.g., audio replay with a sampling rate of 150 million samples per second).

Analog channels show these properties if available (TMATS-ASCII codes given in *italics*):

- Sample rate (*R-x\ASR-n*)
- Data packing option (*R-x\ADP-n*)

Analog subchannels show these properties if available (TMATS-ASCII codes given in *italics*):

- Measurement name (*R-x\AMN-n-m*)
- Sub channel enabled (*R-x\AMCE-n-m*)
- Data length (*R-x\ADL-n-m*)
- Bit mask (*R-x\AMSK-n-m*)
- Measurement transfer order (*R-x\AMTO-n-m*)
- Recorded analog format (*R-x\AF-n-m*)

- Sample rate (derived from the channel's sample rate and the subchannel's sample factor ($R-x\backslash ASF-n-m$))
- If this is an audio recording ($R-x\backslash AV-n-m$)
- Audio format ($R-x\backslash AVF-n-m$)

The analog packets decode these properties:

- The timestamp
- The number of subchannels (only format 1. Format 3 is fixed to two subchannels)
- An indication if the samples are packed
- An indication when all subchannels have the same definition (only format 1. Format 3 always has two subchannels with same definition)

The packets are decoded into one packet entry for each subchannel (in contrary to one entry for each sample). They have these properties:

- The timestamp (the same as the packet)
- The analog subchannel number
- The bit length of each sample
- The bit mask of each sample (taken from TMATS)
- The analog format, e.g., two's complement or unsigned binary (taken from TMATS)
- The sampling rate divisor, i.e., the number the sampling rate of the channel must be divided by to get this subchannel's sampling rate. Note that this is derived from the "sample factor" but is not the same.
- The number of samples for this subchannel within this packet
- The samples. These have already been masked, probably reversed (only if TMATS indicates LSB first) and the analog format has been applied.
- The maximum and minimum sample in this packet

Decoding is heavily depending on TMATS attributes. If some are missing, you can probably provide them with TMATS overriding or TMATS defaults. Also, the Analog to CSV Exporter can export the data while correcting some settings.

In many areas of FLIDAS analog values are decoded and then stored internally as 64-bit floating point values. This means analog data encoded as integer with more than 52 bits may suffer from a loss of precision but the values will still be correct in magnitude.

4.6.8 Discrete

Discrete channels show these properties if available (TMATS-ASCII codes given in *italics*):

- Discrete mode ($R-x\backslash DMOD-n$)
- Sample rate ($R-x\backslash DSR-n$)
- Number of bits (discovered from the first packet)

The discrete packets have these properties:

- The timestamp
- The number of bits in each discrete packet entry
- An indication of LSB or MSB alignment of the packet entries

- An indication if the recording was done on a state change or on equidistant points in time

The packets are decoded into packet entries with a sample of the discrete bits at the same point in time. The packet entries have these properties:

- The timestamp
- The discrete bits in LSB alignment

4.6.9 PCM

This section covers the PCM data type format 1, not the PCM DQE format 2 introduced in 2019. In general, all FLIDAS features that refer to PCM are not for PCM DQE unless explicitly written.

PCM channels show these properties if available (TMATS-ASCII codes given in *italics*):

- Data packing option (*R-x\PDP-n*)
- PCM code (*P-d\DI*)
- Data randomized (*P-d\D7*)
- Input stream derandomization (*R-x\IDDR-n*)
- Bit rate (*P-d\D2*)
- Common word length (*P-d\F1*)
- Word transfer order (*P-d\F2*). If this is undefined FLIDAS will default to most significant bit first.
- Words per minor frame (*P-d\MF1*)
- Bits per minor frame (*P-d\MF2*)
- Minor frame sync type (*P-d\MF3*). "Fixed pattern" and "alternating code complement" is supported for decommutation and "fixed pattern" will be assumed by default. The value "other" leads to a warning for the channel.
- Sync pattern length (*P-d\MF4*)
- Sync pattern (*P-d\MF5*)
- In-sync and out-of-sync criteria (*P-d\SYNC1*) to (*P-d\SYNC4*)
- Words with different length than common word length (*P-d\MFW1-n* and *P-d\MFW2-n*)
- Minor frames per major frame (*P-d\MF\N*)
- All subframe ID counters with these properties:
 - Name (SFIDx with x a unique number if the name is undefined) (*P-d\ISF1-n*)
 - Word position (*P-d\IDC1-n*)
 - MSB position (*P-d\IDC3-n*)
 - Bit length (*P-d\IDC4-n*)
 - Bit transfer order (*P-d\IDC5-n*)
 - Initial counter value (*P-d\IDC6-n*)
 - Counting direction (*P-d\IDC10-n*)
 - Number of values (calculated from *P-d\IDC6-n*, *P-d\IDC8-n* and/or *P-d\IDC4-n*)
- All chapter 7 segments with these properties:
 - First word (*P-d\C7FW-n*)
 - Number of words (*P-d\C7NW-n*)

The PCM packets have these properties:

- The timestamp
- The mode (packed, unpacked, throughput)
- An indication if the individual PCM packet entries have intra packet headers (only in packed / unpacked mode)
- Whether 16- or 32-bit alignment was used
- The sync offset (word offset into the major frame for the first data word in the packet entry) (only in unpacked mode)
- The major and minor frame lock status (only in packed / unpacked mode)
- An indication if the packet entry starts with a minor or major frame (only in packed / unpacked mode)

Packed or unpacked PCM packets will be decoded to packet entries (PCM minor frames) with these properties:

- The timestamp
- If a minor and/or major frame lock exists
- The subframe word positions decoded from all subframe ID counters
- The minor frame sync pattern and the data words. The words are already adapted to the individual word's bit length and least significant bit first entries have been reversed (See also [open IRIG 106 issues](#) as this is not properly specified in IRIG 106). As many hex digits are shown as required to fit the bit length inside.

Each throughput PCM packet will be decoded to one packet entry (as these do not directly map to PCM minor frames) with these properties:

- The timestamp
- The raw data byte wise in transmission order.

The PCM minor frame level is supported for all modes of PCM packets with further features like the [PCM Packet Entry Viewer](#).

PCM minor frame decoding for PCM throughput mode works like this:

- The TMATS attributes for in and out-of-sync criteria are used to synchronize into the bit stream. This is done in the scope of the selected packets. Therefore, when looking at a subset of PCM packets, the beginning section and the end section will be lost because they are no complete minor frame or the synchronization process has not yet locked in.
- In case of a minor frame sync loss an empty minor frame will be generated which indicates the sync loss with its time. *Note that this is not provided in PCM packed / unpacked mode.*
- If the PCM stream indicates that it is randomized (through P-d\D1 or P-d\D7) and that the recorder did not derandomize (indicated by R-x\IDDR-n) FLIDAS will do a software derandomization using the 15-bit derandomization from IRIG 106 appendix C. You can use [TMATS overriding](#) if your recorder doesn't provide the attributes.

CVSD audio / voice embedded in a PCM stream is handled on the parameter level by defining a parameter with the containing PCM words and using specific functionality on this parameter.

The following features are unsupported so far. Contact us if you need some of them:

- Parity (currently parity will be considered part of the raw data of a data word), except measurement parity which is supported
- Some IRIG 106 PCM class II features (except data word length up to 64 bit, unlimited bit rate, unlimited bits / words per minor frame, fragmented words, most binary formats)
- IRIG 106 chapter 8
- Word lengths exceeding 16 bits in **unpacked** mode are not properly defined in IRIG 106 chapter 10. See [open IRIG 106 issues](#).

4.6.10 IRIG 106 Chapter 7 PCM

Some FLIDAS features allow processing IRIG 106 chapter 7 PCM data.

If the PCM channel defines chapter 7 segments of data within the minor frames, then FLIDAS uses them. Otherwise FLIDAS will use all PCM words after the minor frame sync pattern.

If not all data for a complete source packet is decoded, the source packet will not be used for further processing. However, in complete source packets there may still be bit errors.

IRIG 106 Chapter 7 had some major renaming of concepts in 2019 release. This manual uses the new wording:

New concept	New concept abbreviation	Old concept	Old concept abbreviation
transport packet	TP	packet telemetry data frame	PTFR
encapsulation packet	EP	packet telemetry data packet	PTDP
low-latency encapsulation packet	LLEP	low-latency packet telemetry data packet	LLP
source packet	SP	packet telemetry packet / PT packet	-

The optional Reed-Solomon error correction is not supported. If you need it, tell us.

For chapter 7 processing from a live data stream you should remember that the processing happens when the PCM data comes in on the live data stream and will therefore mainly depend on the delay of the live data stream. *Note that if your live data stream's purpose is only to carry the CH7 data, you can improve the latency by disabling the synchronize channels option in the [live data stream creation](#).*

4.6.11 CAN Bus

CAN bus subchannels show these properties if available (TMATS-ASCII codes given in *italics*):

- Bit rate (*R-x\CBBS-n-m*)

CAN bus packets have these properties:

- The timestamp
- The number of contained messages

CAN bus packets will be decoded to single CAN bus messages with these properties:

- The timestamp
- The number of the subchannel (1-based) within this IRIG 106 channel
- The identifier (8 digits hexadecimal if extended identifier, 3 digits hexadecimal if standard identifier)
- An indication if the RTR (remote transfer request) bit was set
- Indications for data or format errors if occurred
- The data bytes in hexadecimal

4.6.11.1 DATaRec4 CAN Bus Support

There is a separate commercial plug-in that integrates CAN buses in the DATaRec4 format (from Zodiac Data Systems) into FLIDAS. If the plug-in is installed DATaRec4 CAN buses can be used like regular CH10 CAN buses. All CAN specific features work here too.

Only ASCII TMATS is supported and no XML. However, Zodiac Data Systems used no XML TMATS at the time of the release of the plug-in

4.6.12 Fibre Channel FS Layer

The fibre channel FS layer (format 1) packets have these properties:

- The timestamp
- The number of frames in the packet

Fibre channel FS layer packets will be decoded to single fibre channel frames with these properties:

- The timestamp
- The start of frame type
- The end of frame type (EOFa, EOFni, EOFrti and EOFdti are considered invalid data)
- An indication for the following errors in the frame:
 - Frame
 - CRC
 - Overrun
- The number of bytes in the frame (header + data)
- The bytes of the frame

4.6.13 Fibre Channel PH Layer

The fibre channel PH layer (format 0) packets have these properties:

- The timestamp
- The format of the stored packet entries
- The number of frames in the packet

Fibre channel PH layer packets will be decoded to single fibre channel frames with these properties:

- The timestamp
- An indication if the stripped mode was used to record (data payload only)
- The type of frame:
 - Complete data frame
 - Complete link control frame
- The topology: only "passive" is defined.
- The start of frame type
- The end of frame type (EOFa, EOFni, EOFrti and EOFdti are considered invalid data)
- An indication for the following errors in the frame:
 - Frame
 - CRC
 - Overrun
- The number of bytes in the frame (header + data + CRC for non-stripped mode or data payload only for stripped mode)
- The bytes of the frame (see above)

4.6.14 Message Data

The message data packets have these properties:

- The timestamp
- An indication which segment of the message was recorded or if complete messages were recorded
- The number of recorded messages if complete messages were recorded
- The message segment number if segments were recorded

The packets are decoded into messages or segments of messages with these properties:

- The timestamp
- The subchannel number (1-based)
- If it is a complete message or first, middle or last segment
- The message segment number if segments were recorded
- An indication when a data of format error was detected
- The number of data bytes
- The data bytes

4.6.15 TSPI / CTS (GPS NMEA-RTCM)

The TSPI / CTS format 0 packets have these properties:

- The timestamp
- The type of GPS data contained
- An indication if additional timestamps are available for each entry

TSPI / CTS format 0 packets will be decoded to single packet entries with these properties:

- The timestamp (if there is no timestamp for the packet entries, they will share the timestamp of the packet)
- The number of data bytes
- The data bytes

For further processing of NMEA 0183 and NMEA 0183 HS data FLIDAS makes the following assumption which is highly probable but not 100 % covered by the specification: A single packet entry covers exactly one NMEA 0183 sentence including the start delimiter '\$' and the end-of-line marker carriage return and line feed. This means there are no fragments of sentences and no multiple messages. In general, FLIDAS will try to handle a missing end-of-line marker gently.

NMEA 0183 HS can be used at places where NMEA 0183 can be used in FLIDAS as well.

4.6.16 TSPI / CTS (EAG ACMI)

The TSPI / CTS format 1 packets have these properties:

- The timestamp
- The type of ACMI data contained
- An indication if additional timestamps are available for each entry

TSPI / CTS format 1 packets will be decoded to single packet entries with these properties:

- The timestamp (if there is no timestamp for the packet entries, they will share the timestamp of the packet)
- The number of data bytes
- The data bytes

4.6.17 TSPI / CTS (ACTTS)

The TSPI / CTS format 2 packets have these properties:

- The timestamp
- The ACTTS recording format
- An indication if additional timestamps are available for each entry

TSPI / CTS format 2 packets will be decoded to single packet entries with these properties:

- The timestamp (if there is no timestamp for the packet entries, they will share the timestamp of the packet)
- The number of data bytes
- The data bytes

4.6.18 Time

Time channels show these properties if available (TMATS-ASCII codes given in *italics*):

- Time format (*R-x\TFMT-n*)
- Time source (*R-x\TSRC-n*)
- The time offset added to the time stamps stored in the packets (configured in the time channel offset wizard page during live data stream creation or file loading)

There are two types of time packets. The regular time packet format 1 and network time packets.

The format 1 time packets have these properties:

- The timestamp
- The time format (IRIG A, IRIG B, IRIG G, native GPS, real-time clock, UTC from GPS, none (invalid time))
- The time source (external, internal, internal from RMM)
- If the time has a month and year or just days
- If the year is a leap year
- The IRIG time source (only on IRIG format and IRIG 106-17 or later)

The network time packets have these properties:

- The timestamp
- The time format (network time protocol version 3, IEEE-1588-2002, IEEE-1588-2008)
- The time status (valid or invalid)

FLIDAS will convert all timestamps to a full date, even if they do not contain year and month. In this case FLIDAS uses the year of the start of recording provided in the CH10 file loading wizard or when creating a live data stream and correct this by the "original recording date and time" (*R-x\RI4*) if this is available in TMATS and you have selected the option. Wraps of the day of year during the recording will be detected.

Time packets are used as a reference in the timestamp calculation of all other packets. FLIDAS will lock to the time packets of the channel of the first time packet it decodes. All other time packets will be read and decoded, but not serve as a reference.

If no time packets are available, then FLIDAS will treat all timestamps by using the RTC value as an offset to the beginning of the year defined for the recording start ("zero-time"). This means an RTC of 10 million will lead to a time of "1st January of recording start year, 00:00:01". Also, other rare situations in a corrupted CH10 source may create timestamps related to zero-time.

There is a difference in calculating packet times from the time reference between CH10 files and live data streams:

- **CH10 files:** To calculate the timestamp for a packet **C** FLIDAS uses the time packets before (**A**) and after (**B**) **C**. Time packets with detected packet header or data

checksum errors, invalid packet sync or RTC sync errors are ignored. **A** and **B** are determined by choosing the packets with the RTC before / after **C** not necessarily with the file position before / after **C**. From **A** and **B** the difference in the RTC can be related to a specific time difference in nanoseconds. This way the time difference per RTC tick is known and can be used to compensate drifts in the RTC. If **C** is before the first or after the last time packet in the file, the drift is not compensated.

In the above system the time would be calculated like this:

$$\text{time(C)} = \text{time(A)} + (\text{RTC(C)} - \text{RTC(A)}) / (\text{RTC(B)} - \text{RTC(A)}) * (\text{Time(B)} - \text{Time(A)})$$

- **Live data streams:** To calculate the timestamp for a just received packet **C** FLIDAS uses the most recently received time packet (**A**). Time packets with detected packet header or data checksum errors, invalid packet sync or RTC sync errors or invalid time are ignored.

In the above system the time would be calculated like this:

$$\text{time(C)} = \text{time(A)} + (\text{RTC(C)} - \text{RTC(A)}) * 100 \text{ Nanoseconds}$$

In effect this does not compensate the drift in the RTC. There may be slight time discontinuities when a new reference time packet arrives. In older types of recorders RTC drifts of up to 18µs per second were observed while in more recent hardware it is not uncommon to observe no drift at all.

4.6.19 TMATS

There are two types of TMATS packets. Both have these properties in common:

- The timestamp
- The CH10 revision

The format 1 setup record packets have these additional properties:

- The format (XML or ASCII)
- An indication when the TMATS record is a configuration change

The format 4 streaming configuration packets have these additional properties:

- The streaming configuration format which may either be a SHA2-256 checksum, a list of activated channels or a complete or segmented TMATS record in ASCII or XML format
- The active channels if applicable
- The SHA2-256 checksum if applicable
- An indication if this is the last segment of a segmented TMATS record

The TMATS packets are decoded into TMATS attributes. These are used throughout FLIDAS. Only the first TMATS packets are used to decode the attributes. This means FLIDAS locks to the first TMATS packet containing TMATS attributes and then possibly

accepts following packets of the same format until the end of a series is reached (for format 1 this is declared if a time packet is found after the TMATS packets).

In live data streams it may happen that you start receiving in the middle of a segmented TMATS record of format 4. Since only the end but not the start is flagged FLIDAS cannot recognize this and will work with the incomplete TMATS. For format 1 not even the end is flagged.

4.6.20 Recording Events

The recording event packets have these properties:

- The timestamp
- The number of contained events
- An indication when the packet defines additional timestamps

The packets are decoded into event occurrences with these properties.

- The timestamp
- The event number
- The event count
- The additional timestamp if it was defined in the packet
- If the event occurred in the period before the recording

4.6.21 Recording Indices

The recording index packets have these properties:

- The timestamp
- The index type (root or node index)
- The file size at the time the packet was created (if available)
- An indication when the packet defines additional timestamps
- The number of entries
- For root indices also the offset to the previous index and an indication if this is the first root index packet

The packets are decoded into index entries with these properties.

- The timestamp (the same as the packet)
- An offset which is either the offset to the data packet that this node index entry refers to or the offset to the node index that this root index entry refers to
- For node indices also the channel number and data type (hex) of the packet that the node index entry refers to
- The additional timestamp if it was defined in the packet

4.6.22 User Defined Computer-Generated Data

The user defined computer-generated data packets have these properties:

- The timestamp

This type of packet is a little bit different from others and has no packet entries by definition and no defined structure for its data as well. FLIDAS takes the whole data after the channel specific data word as a block of binary data and creates a single packet entry for it using the time stamp of the packet. This makes the data accessible for parameter decoding using the User Defined Computer-Generated Data Binary Simple CSV ICD Format.

4.6.23 Partly Supported CH10 Data Types

There is no full support for the CH10 data types in the following subchapters. The level of support is described but for most of them it means that the properties on packet level are decoded but no packet entries. In result they are for example not useful for exporting.

4.6.23.1 Image Data

The image data packets have these properties:

- The timestamp
- An indication if intra packet headers are used
- Sum: An indication if the packet contains fragments of images or integral images
- Parts: An indication which fragment of the image is contained in the packet
- Segment length: The byte length of the single segment

4.6.23.2 Still Imagery

The still imagery packets have these properties:

- The timestamp
- An indication if intra packet headers are used
- The image format
- Sum: An indication if the packet contains fragments of images or integral images
- Parts: An indication which fragment of the image is contained in the packet

4.6.23.3 Dynamic Imagery

The dynamic imagery packets have these properties:

- The timestamp
- An indication if intra packet headers are used
- The image format
- Sum: An indication if the packet contains fragments of images or integral images
- Parts: An indication which fragment of the image is contained in the packet if it does not contain integral images

4.6.23.4 IEEE 1394 Physical Layer

The IEEE 1394 physical layer packets have these properties:

- The timestamp
- The number of intra packets (i.e., entries contained in the CH10 packet)

4.6.23.5 IEEE 1394 Transaction

The IEEE 1394 transaction packets have these properties:

- The timestamp
- The packet body type
- The number of transactions
- The synchronization code for packets with packet body type 1

4.6.23.6 Parallel

The parallel data packets have these properties:

- The timestamp
- An indication if the DCRsi scan format has been used
- The scan number if the DCRsi scan format has been used
- The data word width if the DCRsi scan format has not been used

4.6.23.7 MJPEG

The MJPEG packets have these properties:

- The timestamp
- An indication if intra packet headers are used
- Sum: An indication if the packet contains fragments of frames or integral frames
- Parts: An indication which fragment of the frame is contained in the packet if it does not contain integral frames

4.6.23.8 MJPEG 2000

The MJPEG 2000 packets have these properties:

- The timestamp
- An indication if intra packet headers are used
- Sum: An indication if the packet contains fragments of frames or integral frames
- Parts: An indication which fragment of the frame is contained in the packet if it does not contain integral frames

4.6.23.9 PCM DQE

This section covers the PCM DQE (data quality encapsulation) data type format 2 introduced in 2019, not the regular PCM data type format 1. In general, all FLIDAS features that refer to PCM are not for PCM DQE unless explicitly written.

PCM DQE channels have the same properties decoded as regular PCM

The PCM DQE packets have these properties:

- The timestamp
- An indication if the space time code is enabled

- The low-density parity check (LDPC) code rate
- The low-density parity check (LDPC) block size
- The DQE payload size
- The alignment mode

4.7 Wizards

A wizard may consist of several pages and you can toggle through the pages with the **Next** and **Back** buttons as soon as you have provided enough information to leave one page.

The **Finish** button starts the action you are configuring with the current configuration. The Finish button may even be available before you have toggled through all pages when the other configuration is optional or already at good default values. It will then show the number of pages you are going to skip like "Finish (3)".

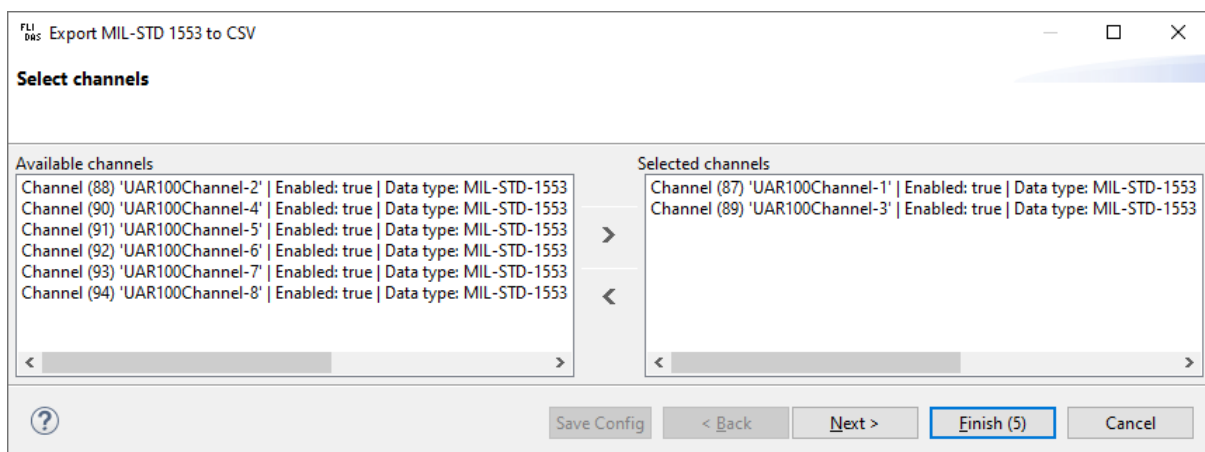
The **Cancel** button aborts the action.

Some pages will allow saving the configuration for future actions using the **Save Config** button. This means the next time you will see this page in this context (e.g., on the same exporter or protocol viewer), the same configuration will appear. Some pages might save independent of the context if the configuration likely applies to all situations. Only those values are saved that are likely to not be individually configured for each action and that are not already configured through the preferences. Once you save, the configuration saved will be displayed. Each page is saved individually but some might not allow saving at all. The saved configuration can be ex- and imported with the preferences.

The following chapters will describe the available wizard pages.

4.7.1 Channel Selection

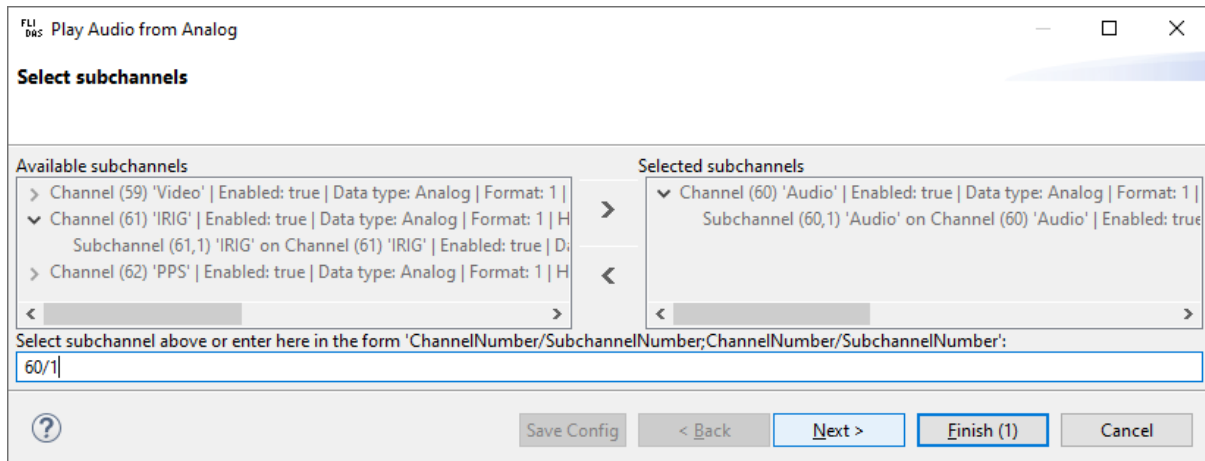
In some wizards you need to select one or several channels on the first page. The available channels are displayed on the left side and you can select them and move them to the right side with the right arrow button. All channels on the right side will be part of the configuration. You can remove channels from the configuration by selecting them and clicking the left arrow button. Channels can also be moved between the sides with a double-click. Multi selection of channels to move them with the arrow buttons is possible.



Some filtering will suppress channels that are of wrong type, disabled or detected as not suitable for other reasons.

4.7.2 Subchannel Selection

In some wizards a channel might not be the smallest unit to select. For example, ARINC 429 channels are divided into up to 256 subchannels. So you will be able to select the subchannels to be part of the configuration in a similar way as on the [Channel Selection Wizard Page](#). The difference is that the page shows a hierarchy of channels and subchannels and you can move the subchannels between both sides of the page.



The selection of a subchannel depends on the TMATS attributes of your CH10 source. Older TMATS versions did not report the available subchannels. In this case you can use the lower line of the page to manually enter the subchannels. You need to enter groups of channel and subchannel numbers separated by a '/'. If you select several subchannels, you need to separate the groups with a ';'. For example, '60/1;26/5' would select subchannel 1 on channel 60 and subchannel 5 on channel 26. Note that FLIDAS harmonizes all subchannel numbering to be 1-based. See also the [Subchannel Number Preferences](#) that help achieving this.

Note: If you need to know the available subchannels and they are not declared in TMATS, you can use the [communication summaries](#) in any commercial version to discover the subchannels and start actions from there.

4.7.3 Exporter Target File Selection

On this wizard page you can specify the file you want to export into. Click on the **Browse...** button to show a dialog and select the file.

If you have configured a [default export directory](#), there will be a suggested file name. If you haven't, you must use the **Browse...** button to at least select a directory (for the file name itself there will still be a proposal).

If the file name identifies an existing file, you can confirm that you want to overwrite it with the appearing checkbox.

4.7.4 Exporter Target Directory Selection

On this wizard page you can specify the directory you want to export into. The exporter will choose the file names automatically but inside this directory. This is used if exporters may create multiple export files.

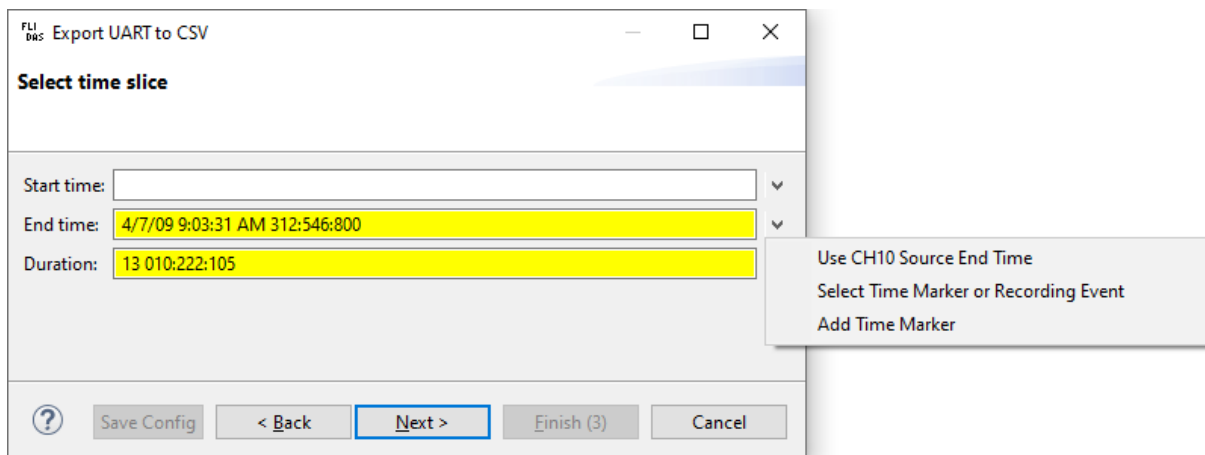
You must choose a parent directory. To do so click the **Browse...** button and select the directory. You can then optionally enter a subdirectory to be created or used inside the parent directory. It will be created if it doesn't exist. If you leave this out the export files are put in the parent directory.

If you have configured a default export directory, it will be predefined for the parent directory.

If the directory identifies an existing directory, you can confirm that you want to write into it (possibly overwriting existing data) with the appearing checkbox.

4.7.5 Time Slice Selection

Most exporters, protocol Viewers and parameter viewers allow limiting the time slice of used data.



You may enter the timestamps for the beginning and end of the data to use. If you omit the start or end timestamp, all the data from the beginning or all the data to the end will be used.

There are some options to use predefined timestamps here. You see them if you click on the downwards arrow to the right of the start or end time field:

- **Use CH10 Source Start Time** or **Use CH10 Source End Time**: Use the start or end time of the CH10 source. For a definition of start and end time see General CH10 Source Information Properties. If you enter nothing in the text field you will already use this, but setting it explicitly allows you modifying the timestamp manually.
- **Select Time Marker or Recording Event**
- **Add Time Marker**: Remembers the current timestamp defined in the field for later use as a time marker.

You can also use predefined timestamps as a template for the required timestamp format if you have trouble entering a proper timestamp. However, you can leave out the whole or parts of the second's fraction part (i.e., the `milliseconds:microseconds:nanoseconds` part)

If you enter a timestamp as start or end, this exact timestamp will be included (i.e., "data < start time" and "data > end time" will be suppressed).

The **Duration** field allows you to enter a duration for the time span. The duration will be either applied from the start or the end depending on which field contains a timestamp. The fields used for calculation will have a yellow background. If an empty field has a yellow background, it means the default is used here (recording start for start field, recording end for end field).

The format for the duration will be `hours:minutes:seconds` or `milliseconds:microseconds:nanoseconds`. You can leave out any number of the fractional second part and anything from the first part except the seconds. In the second part every value must be between 0 and 999. In the first part there is no limitation so you can choose to represent 120 seconds as 2 minutes and 0 seconds or as 120 seconds. Every single value must be integer.

4.7.6 Time Representation

This wizard page is used to configure the timestamps created by exporters.

Some exporters export timestamps. If they do, you may configure the type of timestamp and possibly also if more than one or even no timestamp is exported.

FLI DAS Export ARINC 429 to CSV

Configure timestamp properties

Time representation 1

Timestamp data type: String

Timestamp format: Day of year time

Reference time:

Time representation 2

Timestamp data type: Long

Timestamp format: Nanoseconds since reference time

Reference time: 4/7/09 9:03:05 AM 735:179:000

Time representation 3

Timestamp data type: None

Timestamp format:

Reference time:

? Save Config < Back Next > **Finish (1)** Cancel

You may select a **timestamp data type**. There are three general types:

- **Numeric:**
 - **Long** for integral or
 - **Double** for floating-point timestamp representations

These types are generally used in binary files but can also be exported to text files. Some formats may be limited in their precision. E.g., a Long timestamp representing seconds will only be precise to seconds while the same timestamp as Double will also indicate nanoseconds as fractions of the seconds as far as possible.

- **String:** This can only be used in text files
- **None:** Can be used to suppress timestamps

If several timestamps are allowed for one export an additional option will appear as soon as you configured the first timestamp. Leave the second configured to "none" if you don't need more timestamps.

In addition to the timestamp data type a **Timestamp format** must be selected. The general categories here are:

- Absolute time
- A timestamp relative to New Year of the applicable date but not including the year itself. It will refer to the year of the first timestamp converted and timestamps will in general not wrap at the end of the year but keep on counting (i.e., after the day 365 not day 1 but day 366 is following).
- A timestamp relative to a user defined point in time (like time after the beginning of the recording). The time can be manually entered or taken from a time marker or recording event or begin of recording using the drop-down menu right of the time field. To be relative to the start of the time slice copy / paste the timestamp from the Time Slice Selection Wizard Page.

Sometimes several precision variants of a type exist (like seconds or nanoseconds after New Year 1970) or segments of a timestamp are provided as separate timestamp (like only the time or only the date of a timestamp). Both primarily exist to remain compatible to older FLIDAS version that used these timestamp types.

See the table below for available timestamp formats. The table also includes samples in *italics* where you can also compare precision loss between Long and Double. The samples use a US localization and may look different on your computer. The primarily supported timestamps are printed in **bold** while the variants only supported for backwards compatibility or completeness use regular font.

	Absolute time	Relative to New Year	Relative to user defined reference
Numeric (Long / Double)	<ul style="list-style-type: none"> • Nanoseconds since New Year 1970: <i>1239094985961262900</i> <i>or</i> <i>1239094985961262850</i> 	<ul style="list-style-type: none"> • Nanoseconds since New Year: <i>8326985961262900</i> <i>or</i> <i>8326985961262900</i> 	<ul style="list-style-type: none"> • Nanoseconds since reference time:

	<ul style="list-style-type: none"> Seconds since New Year 1970: <i>1239094985 or 1239094985.961263</i> 	<ul style="list-style-type: none"> Seconds since New Year: <i>8326985 or 8326985.961262900</i> 	<i>226083900 or 226083900</i> <ul style="list-style-type: none"> Seconds since reference time: <i>0 or 0.226084</i>
	<i>Note: No matter which type of time source your recorder uses, the nanoseconds or seconds since anything will always respect leap years but never leap seconds. Therefore, FLIDAS will use the time from your recorded time source (with or without leap seconds) but represent it internally with days with 86400 seconds and Februaries with 28 or 29 days.</i>		
String	<ul style="list-style-type: none"> Date and time: <i>4/7/09 9:03:05 AM 961:262:900</i> Date: <i>4/7/09</i> Time to seconds: <i>9:03:05 AM</i> Time (second's fraction in ns): <i>961262900</i> 	<ul style="list-style-type: none"> Day of year time: <i>097-09:03:05.961262900</i> 	<ul style="list-style-type: none"> Time since reference time: <i>0:00:00 226:083:900</i>

4.7.7 CH10 File Selection

This wizard page is only used for parameter viewers or parameter exporters.

You can select the CH10 file that should be used as source for the parameters. Only CH10 files from the Repository View are available. If only one CH10 file is loaded, the page does not appear at all.

Files may appear in italics, which means that the file is not compatible with the active ICD mapping. It is still listed here, so you can click it and see the reason why.

4.7.8 Live Data Stream Selection

This wizard page is only used for parameter live processors.

You can select the live data stream that should be used as source for the parameters. Only live data streams from the Repository View are available. If only one live data stream is available, the page does not appear at all.

Live data streams may appear in italics, which means that it is not compatible with the active ICD mapping. It is still listed here, so you can click it and see the reason why.

4.7.9 Sampling Mode Selection

This wizard page is used by a few parameter exporters or viewers that sample parameter values over time. A time discrete sampling period can be defined in milliseconds or you can sample asynchronously. The sampling modes are described in the [concepts section](#).

*Note: Be careful with the asynchronous modes as they can create a huge amount of data when parameters change at a high frequency. For example, using a parameter set with a 20MHz analog parameter and 20 ARINC 429 parameters will not only create 20 million analog samples per second but also 400 (20*20) million ARINC 429 samples. This will happen at least if the analog parameter has some level of noise and is not perfectly stable at the same value.*

In general, asynchronous full sampling creates more data than asynchronous change sampling which creates more data than asynchronous valid change sampling.

4.7.10 CSV Properties

This wizard page configures properties for CSV files.

The **CSV column delimiter** defines the character that is used to delimit two columns of a CSV file. The global default is defined in the [General Preferences](#) but can be overridden here for a single time.

4.7.11 CH10 Data Type Filters

The CH10 Data Type Filters Wizard Page appears when you open a file within the [CH10 file loading wizard](#) or when [creating a live data stream](#). It controls which [CH10 data types](#) will be processed from the CH10 source at all. For example, you can just focus on ARINC 429 by deselecting all other CH10 data types.

The CH10 data types are listed in two categories:

- **Supported CH10 data types** lists CH10 data types which are supported by FLIDAS to a finer grained level. I.e., these will have details on the data.
- **Only packet level** lists CH10 data types which have no further support. These [packets](#) will be available but have no details about their contained [packet entries](#).

When the box in front of a CH10 data type filter is checked it means that the data of this CH10 data type will be available.

The CH10 data type filter for **Invalid packets** will control if packets that have an invalid packet sync or a wrong packet header checksum will for example show up as invalid in the [Packet View](#) or if they will be completely suppressed. If they show up, they will not contain valid data. Even the timestamp will not be correct but will be calculated from the highest RTC value processed so far. Only their position within the CH10 source will be correct.

The CH10 data type filter for **Unknown packets** will control if packets that have a data type not defined in IRIG 106 chapter 10 will be processed anyway if they are valid in other respects. If they are displayed, they will only show their channel specific data word but not contain any packet entries.

4.7.12 UDP Properties

This wizard page configures a UDP network connection.

You must always configure a UDP **Port** number.

You can define an **IP Address** (IP4 only). This may also be a broadcast or multicast address. If you don't enter anything here it is treated as the address for the computer you are running FLIDAS on ("localhost"). The IP 127.0.0.1 is a synonym for "localhost" and will appear in some situations even if you entered nothing.

If you are configuring a data transmission, the IP address is mandatory and identifies the target computer or network. If you are configuring a data reception the IP address is ignored

If you use IP multicast the IP address is mandatory for both transmission and reception.

If you configured a multicast IP address, you can select a network interface. You can also leave this to default to let your operating system pick it automatically.

4.7.13 CH10 Data Stream Format

The CH10 Data Stream Format Wizard Page appears when you want to transmit or receive a data stream. It controls which format defined in IRIG 106 chapter 10 is used for data streaming. The options are different for transmission and reception.

Transmission:

- **Format:** The format version number of the format from IRIG 106 chapter 10. Formats 1 and 3 are supported since format 2 is proprietary (ask for support if you need it). *Format 3 has been introduced in 2017 and may not yet be supported by your devices.*
- **Source ID length** (only on format 3): Controls how many bits are used to encode a source ID for the transmission. As in the standard, 4 times as many bits are used:

Source ID length	Bits used	possible sources
0	0	1
1	4	16
2	8	256
3	12	4096
4	16	65536

- **Source ID** (only on format 3): Controls which source ID is encoded in the transmission.

Reception:

You can only define the **Source ID** here. If you do, only a format 3 will be accepted and the data stream is filtered for this source ID. If you do not provide a source ID FLIDAS will lock to the first source ID it finds on the network if it runs a format 3 and otherwise to format 1.

4.7.14 MATLAB File Properties

This Wizard Page configures some options about MATLAB files.

You can choose between version 4 or 5 (compressed) **MAT file formats**. For MAT 5 files you can also choose to **Enforce a MAT5 default comment in the description section** of the file. Otherwise it will contain a description of the export conditions instead. The default comment seems to be required by some post processing software but is not required by the MAT file format specification.

The user can choose the **Measurement format** so the actual measurements can be stored as 32- or 64-bit floating-point numbers (IEEE 754 formats). This does not apply for the timestamps. These are subject to other configurations in the wizard.

If the measurements are created using parameters, then the name of the row vectors will be derived from the parameter name and the user may additionally choose to **Include the mapping ID in the parameter name**.

4.7.15 Pseudo Random Bit Stream Type Selection

This wizard page allows a pseudo random bit stream type selection.

The selected algorithm will generate a reproducible bit stream that always repeats after a specified number of bits ($2^n - 1$). Also each group of the defined bit size "n" will represent a different combination of bits and all combinations are taken exactly one time without the combination with only 0 bits.

These algorithms are used in bit error testing where a transmitter sends this cyclic bit stream and a receiver verifies if it receives this bit stream by generating the same bit stream for comparison.

The available algorithms are:

Algorithm	Bit size "n"	Number of different bit combinations	Specification
PN11	11	2047	ITU recommendation O.150
PN15	15	32767	ITU recommendation O.150

4.7.16 IENA Properties

This wizard page allows configuring some filters on where to find IENA packets in the Ethernet data (if your IENA data is recorded on the IENA layer in message data packets only the "Message reordering time frame" can be set here). Only the data passing the filter will be processed as IENA packets. The Airbus IENA specification requires specific values for the filters but real-world applications seem to deviate from it so you can configure the filters:

- **Destination MAC address:** The specification requires the destination MAC addresses to start with 01:00:5E:01:01. The last octet is missing as it will be different depending on where the data of the multicast frames should be transferred to

(diffusion group). You can leave this out to have no filtering, or define the first five octets in hexadecimal.

- **Source IP address:** The specification requires the source IP addresses to start with 172.28. The last two octets are missing as the 3rd will identify the aircraft where the data is coming from and the 4th the equipment. You can leave this out to have no filtering, or define the first two octets in decimal.
- **Destination IP address:** The specification requires the destination IP addresses to start with 235.1.1 (conforming to the multicast MAC address). The last octet is missing and normally corresponds to the destination MAC address last octet (the diffusion group). If both differ, the displayed diffusion group will be derived from the destination IP address. You can leave this out to have no filtering, or define the first three octets in decimal.
- **Source port:** The specification requires the source UDP port to be 50000 + the Airbus system reference. You can add a comma separated list of port numbers here but the filtered result will not distinguish between the ports. Simply all listed ports pass the filter. You can leave this out to have no filtering, or define the ports in decimal.
- **Destination port:** The specification requires the destination UDP port to be 51000. You can add a comma separated list of port numbers here but the filtered result will not distinguish between the ports. Simply all listed ports pass the filter. You can leave this out to have no filtering, or define the ports in decimal.
- **Message reordering time frame:** This is a special FLIDAS setting. The IENA packets contain own time stamping but FLIDAS processes the data ordered by the time stamps from the Ethernet frames or message data packet entries with their CH10 time stamping of the recorder. You can define a time frame here within which FLIDAS will reorder the data by its IENA time stamps. A setting of 100 ms means that FLIDAS will buffer and sort 100 ms of IENA time stamped IENA packets before it starts processing the oldest one. Bigger values may degrade performance. A value of 0 means no reordering

The button **Set defaults** sets all values to the IENA specification values and the message reordering time frame to a reasonable value (100 ms).

Note: If you do not know the IENA filter settings for your Ethernet data, try to use the Ethernet Communication Summary to get details about your Ethernet data.

Note: If your CH10 recorder is only recording the payload of the MAC frames and not the full MAC frames, then FLIDAS assumes that the destination MAC address passes the filter and that all MAC frames contain IP V4 data.

4.7.17 iNET-X Properties

This wizard page allows configuring some filters on where to find iNET-X packets in the Ethernet data. Only the data passing the filter will be processed as iNET-X packets. You can configure the filters:

- **Source IP address:** You can enter 0 - 4 octets of the source IP address for filtering. For example, "192.168" would only process iNET-X packets with a source IP starting with 192.168. Leave empty to process everything.

- **Destination IP address:** You can enter 0 - 4 octets of the destination IP address for filtering. For example, "192.168" would only process iNET-X packets with a destination IP starting with 192.168. Leave empty to process everything.
- **Source port:** You can decide which source UDP ports should be processed. You can use a comma separated list of multiple ports. You can leave this out to have no filtering, or define the ports in decimal.
- **Destination port:** You can decide which destination UDP ports should be processed. You can use a comma separated list of multiple ports. You can leave this out to have no filtering, or define the ports in decimal.
- **Message reordering time frame:** This is a special FLIDAS setting. The iNET-X packets contain own time stamping but FLIDAS processes the data ordered by the time stamps from the Ethernet frames with their CH10 time stamping of the recorder. You can define a time frame here within which FLIDAS will reorder the data by its iNET-X time stamps. A setting of 100 ms means that FLIDAS will buffer and sort 100 ms of iNET-X time stamped iNET-X packets before it starts processing the oldest one. Bigger values may degrade performance. A value of 0 means no reordering

The setting **Display stream ID in hex** controls whether the FLIDAS features will display the stream ID as hexadecimal or decimal number.

Note: iNET-X packets which exceed the maximum allowed size of 1472 bytes on iNET-X level are also filtered out. Also, FLIDAS will apply some more filtering based on some fixed fields in the iNET-X protocol to filter out more non-iNET-X data. So, even without any user defined filtering you will probably see only your iNET-X data if there is not too much other traffic.

Note: If you do not know the iNET-X filter settings for your Ethernet data, try to use the Ethernet Communication Summary to get details about your Ethernet data.

Note: If your CH10 recorder is only recording the payload of the MAC frames and not the full MAC frames, then FLIDAS assumes that all MAC frames contain IP V4 data.

4.7.18 IRIG 106 Chapter 7 Properties

This wizard page configures properties for IRIG 106 chapter 7 data within a PCM channel.

Chapter 7 data comes with a **Stream ID** to distinguish different streams of data that may be transported on the same PCM stream. The stream ID is between 0 and 15 and may be corrupted due to bit errors. You can configure the following options about how FLIDAS filters the data it should process:

- **Ignore** means FLIDAS takes any data independent of stream ID and processes it. This is helpful if there is only one stream. If different stream IDs appear this will be due to bit errors and it is still safe to process everything.
- **Lock to first** means FLIDAS will process the first data of a valid looking transport packet it receives. Further data will only be processed if it has the same stream ID. If there are several streams present this makes sure that you don't mix them. However, if there is a bit error in the stream ID data from the selected stream might appear to belong to another one and be filtered out.
- **Lock to** lets you enter a stream ID between 0 and 15. Only data with this stream ID will be processed. If there are several streams present this makes sure that you don't

mix them. However, if there is a bit error in the stream ID data from the selected stream might appear to belong to another one and be filtered out.

The **Ignore** option will be safest if you only have one stream.

4.7.19 Data Representation

This wizard page configures how raw data will be displayed. There are two options:

- **ASCII:** Each byte of raw data will be converted to a 7-Bit ASCII character. Bytes with the MSB set will be undefined.
- **Hex:** Each byte of raw data will be converted to two hex digits and each group will be separated by a space.

4.7.20 Time Channel Offsets

When you open a CH10 file or create a live data stream you can manipulate its time stamps by applying an offset. The use case is for distributed CH10 multiplexer sources where the different sources may use different time bases like UTC, GPS, TAI or even local time. The CH10 or TMATS data does not contain enough information to know about this. By applying a time offset you can make sure that all time stamps refer to the same time system. You can also use this if you just have a single source and want to offset the time (for example to another time zone).

You can choose a predefined setup in the drop-down box. The predefinitions are done in the predefined time channel offset preferences but the format is described here. Choose (**manual definition**) from the drop-down box to define the setup manually here instead of using a predefined setup. You can also choose a predefined setup to show the definition in the following two lines and then switch to manual definition to modify it based on the original definition.

You have two lines to manually configure time offsets. The first line configures a time offset in seconds (without limit), the second one in hours and minutes with a granularity of 15 minutes and limited to +/- 24 hours. Although you could do the time offsetting just with seconds, the hour and minute line offers a convenient, less error-prone method for applying time zone offsets.

Both lines have the offsets for different domains separated by a space. The same domain can appear in both lines. The possible domains are:

- **mx:** (e.g., m2) MSID x. x is a decimal number. All time channels belonging to this MSID will add the given offset to their time. If you have a single source CH10 file or live data stream, m0 will cover all time channels.
- **sptp:** "Secondary header PTP". This time will be added to all secondary header time stamps in PTP (IEEE 1588) format. It also applies to intra packet time stamps (packet entry time stamps) in this format. Note that PTP time stamps use TAI as time reference.
- **sch4:** "Secondary header chapter 4". This time will be added to all secondary header time stamps in IRIG 106 chapter 4 time format. It also applies to intra packet time stamps (packet entry time stamps) in this format. Note that Chapter 4 time is

insufficiently specified and FLIDAS treats it as time since new year instead of disregarding all chapter 4 time stamps.

The domain is then followed by a colon and the time offset:

- The time offset for the first line is just the number of seconds as an integral number. You can use negative numbers.
- The time offset for the second line is an hour and minute value separated by a colon. Both values must be integral and only the hours may be negative. The value is limited to +/- 24 hours. You can leave away the minute part if you want but need to skip the colon then as well.

Example: A CH10 file merges CH10 data from an airborne system with data from a ground system. The airborne system is on MSID 1 and uses UTC time. The ground recording (MSID 0) is using GPS+5:30 time (India). A specific sensor in the aircraft provides CH10 data as well with secondary header PTP time. The displayed time should be UTC. The time offsets would be written in these two lines:

Time channel offset (seconds): m0:-18 sptp:-37

Time channel offset (hours): m0:-5:30

The first line corrects the GPS based time from the ground recording down by 18 seconds leading to a UTC based time (still with India time zone offset). It also corrects the secondary header PTP time (TAI based) down by 37 seconds leading to UTC time (at least in April 2021).

The second line removes the 5 hour 30 minutes time offset from the ground recording (from India time zone).

4.7.21 WAV File Properties

This wizard page configures properties of a WAV (audio) file.

You can select if the audio data should use a sample size of 8 or 16 bits.

You can control how gaps in the source audio data is exported. If for a period of time there is no data, you can either fill it with silence or just skip that time period resulting in a shorter file without added silence.

4.7.22 Safran CH10 File Segmentation

If you are loading a CH10 file from a Safran Data Systems recorder (formerly Zodiac Data Systems, formerly Heim), it might be configured to split a CH10 recording into multiple files that have a sequential numbering. FLIDAS allows loading this as if it was one file. To do so there is this additional configuration page for the loading that only appears if you are loading a CH10 file with a specific naming (file name ends with "_rxxxxf000.*" where x is a single digit) and there is another file with the same name except it ends with "f001.*". This means if you are loading the first segment of a file that looks like a Safran Data Systems CH10 file and there is at least a following segment.

The configuration page then allows enabling the support to **Load multiple segments**. You can also enter a maximum count of segments you want to load. If you leave the field empty it will load all segments (with unbroken sequence).

Remember you can also cancel the loading at any time and use the data you have been loading up to then.

The FLIDAS code tries to hide the fact that it is a segmented file from almost all non-low-level code. This means for example that file positions reported for a CH10 packet will relate to the position in the whole sequence of files and not tell you in which file the packet is. Also, if there is a reading error inside the CH10 file sequence, the error message will tell you the name of the first file of the sequence, even if the error was in a following file. Also, if you are creating XML from a CH10 file or filter the content of the CH10 file to a new one, the result will always be a single file without segmentation.

4.8 Exporters

An Exporter exports data from CH10 files to various file formats.


Playing audio or video is also considered as some kind of export (it exports to the speakers / screen instead of a file).

To start an export right-click the CH10 file in the Repository View and select **Export > (type of exporter)**. Another way is to select and right-click some channels or subchannels in the Channel View to see a list of exporters appropriate to the selection. If your subchannel is not declared within TMATS (or declared with a wrong number) you may still be able to start the exporter from the communication summary of the CH10 data type. The communication summary is also the place to start an enclosed protocol exporter.

Exporters relying on parameters are started from the Parameter Plug-in Control View or the Parameter Set View.

After starting a wizard can be used to configure the options. For most exporters this includes:

- selecting one or more channels or subchannels. If you have already preselected the exact number of needed channels or subchannels in the Channel View this option will be suppressed. If there is only one possible channel or subchannel it will be automatically preselected.
- selecting a time slice
- selecting a target file or a target directory if it creates separate export files automatically
- probably more options explained at the specific exporter

After finishing the wizard, the **Cancel** button can be used to stop the export. You can also send the export to the background once it is running. You will be informed when it finishes but can continue working until then. If you want to bring up a running export job to cancel it or see the status, click at the symbol  to the lower right of the window.

Note that any kind of exporting to network drives is significantly slower compared to local drives. Consider exporting to a local drive and copying the result to the network drive later.

4.8.1 TMATS to Text File

This free exporter exports the TMATS record inside the CH10 file to a simple text file. This works for ASCII and XML TMATS records.

4.8.2 Filtered CH10

This free exporter exports a CH10 file to a new CH10 file and allows filtering channels and time slices.

The exporter can be used to reduce the size of files and to filter out classified data when sending in CH10 files for problem reporting. Note that TMATS will not be filtered or modified, so classified information which may be included inside will still be available.

TMATS records will always be exported. Recording indices, recording events and completely broken packets will always be suppressed.

Time channels on all used MSIDs (see [distributed CH10 multiplexer sources](#)) are automatically included. Any time sliced export will make sure the last time packet of the time channels before the beginning of the time slice will be exported with the data.

Packets up to 100 ms before the start time will also be exported because these packets may contain packet entries that reach into the time slice.

Note: The exported file will formally not conform to the IRIG 106 CH10 specification. However, this is true for many "CH10" files and the exported files can still be used with FLIDAS and probably most other CH10 software. Full compliance with CH10 would require the following adaptations which the exporter doesn't apply:

- *Recording indices and recording events must be adapted. If you don't use any of them there is no problem. Otherwise you will lose the features that depend on them.*
- *TMATS must be adapted to indicate that the file has been changed and in which way. Also, suppressed channels need to be disabled in TMATS.*

4.8.3 MSID Channel Splitter

This free exporter splits a CH10 file in multiple files, separated by its MSIDs.

The purpose is to help with files which are formally no valid CH10 files because they mix data from different CH10 sources in one file without adapting them properly. See the concepts section to [learn more about this issue](#).

Each created file will just contain channels from one MSID. The idea is that all channels from the same MSID will use a common RTC (real time counter) and therefore absolute time calculation will be possible like in any normal CH10 file. In addition it is expected that the TMATS packets on channel 0 will contain information on channels of all MSIDs. Each created file will also contain the initial TMATS packets from channel 0 (with RTC adapted to be 1 second before first packet for all MSID > 0). The order of the packets will be the same as in the original file and corrupted packets will be suppressed. Recording indices will also be

suppressed in the exported files. Since recording events formally require recording indices, they are suppressed as well.

You can select the directory to export into and the exporter will create files using the CH10 file name and extends it with "_MSIDxxx.ch10", where "xxx" is the MSID.

By default all MSIDs in the file are selected. There is a wizard page where you can change this selection. The available MSIDs are displayed on the left side and you can select them and move them to the right side with the right arrow button. All MSIDs on the right side will be part of the export. You can remove MSIDs from the export by selecting them and clicking the left arrow button. MSIDs can also be moved between the sides with a double-click. Multi selection of MSIDs to move them with the arrow buttons is possible.

Note: The exported files will formally not conform to the IRIG 106 CH10 specification (but also the input file did not conform). However, this is true for many "CH10" files and the exported files can still be used with FLIDAS and probably most other CH10 software. Full compliance with CH10 would require the following adaptations which the exporter doesn't apply:

- *Recording indices and recording events must be adapted. If you don't use any of them there is no problem. Otherwise you will lose the features that depend on them.*
- *TMATS must be adapted to indicate that the file has been changed and in which way. Also, suppressed channels need to be disabled in TMATS.*

Note: Other particularities to expect:

- *FLIDAS does not change the order of packets. If for any MSID there is no time packet before the dynamic data the result will be a formally invalid file. However, FLIDAS can process files also if the time packet is not the first dynamic packet.*
- *Recording indices and events are suppressed. If these are on channel 0 and there is other computer-generated data remaining then this will have sequence counter errors. However, these sequence counter errors do not prevent you from processing your data channels correctly in FLIDAS. Note that having recording indices or events on channel 0 is invalid since IRIG 106-2013.*

4.8.4 Replace TMATS in CH10

This exporter exports a CH10 file to a new CH10 file but replaces the TMATS data inside with an external TMATS file. This works for ASCII and XML TMATS records. The exporter is available in all commercial versions.

This can be useful in the same situations as TMATS overriding but saves you from remembering to select the TMATS override file and you can also use the CH10 file with other CH10 tools without TMATS overriding.

The exporter can also inject a TMATS record into the beginning of a CH10 file that had no TMATS record before (in this case it would have been an invalid CH10 file before).

You should not export to the original file but need to select a new target file (i.e., overwriting is not possible).

Note: You can also use TMATS defaults in some situations instead.

4.8.5 MPEG Video to MPEG File

This free exporter exports a single MPEG video channel to an MPEG transport stream (*.ts file) or MPEG program stream (*.mpg). Both file types can be played with simple players like Windows Media Player if the proper video codecs are installed.

The exporting of **program streams** is not tested (due to lack of sample files). We are quite confident that it works so we included it. If you require this feature, try the free version before you buy a license.

If a time sliced export will be done, the video packet entries (MPEG packets) outside of the time frame will be suppressed. There will be no further effort to recode the MPEG stream.

Note that exporting of video channels runs a lot faster if the recorder doesn't use intra packet timestamps.

4.8.6 Replay Audio from Analog

This free exporter plays a single analog subchannel on the local audio hardware.

The playing relies on the availability of local audio hardware that supports playing with the given parameters.

The playing also relies on several other attributes:

- Sample rate: This is taken from TMATS. If undeclared it will be estimated by the number of samples in the first packet and the time difference between first and second packet.
- Bit length of samples: This is taken from TMATS. If undeclared it will be taken from the first packet. Since most audio hardware supports only 8 or 16 bits, the samples are converted to these two options which might lose precision.
- Signed / unsigned: If TMATS declares the analog format to be "unsigned" it will use unsigned. All other cases will expect signed samples

Decoding of analog data is heavily depending on TMATS attributes. If some are missing, you can probably provide them with TMATS overriding or TMATS defaults.

4.8.7 Replay Audio from CVSD

This parameter exporter replays a single CVSD audio parameter contained in a PCM stream on the local audio hardware. The exporter is only available in the Parameter or Parameter Streaming Version.

The playing relies on the availability of local audio hardware that supports playing with the given parameters.

4.8.8 MPEG TS Replay (UDP Streaming)

This exporter plays an MPEG video (MPEG transport stream) using an external video player. It is available in all commercial versions.

The exporter will stream the video as a UDP stream to a video player running on the local computer to a configurable UDP port (1234 by default). You can use any player that is able to play MPEG transport streams transmitted on raw UDP.

An example of a free tool that can do this is VLC. In VLC open a network video stream from the address `udp://@` (if you configured a UDP port address, use `udp://@:PORT` and replace **PORT** by your UDP port). Another free tool is mpv. A simple command line would be `mpv udp://127.0.0.1:1234`. Replace **1234** by your selected UDP port if you configured one.

You can configure a command line for starting up the external video player. The command line should include the options to start it up in reception mode for the UDP stream. If you use `$PORT` in the command line, it will be replaced by the UDP port address you have set up. An additional delay can be configured between the command to start up the video player and the actual sending of UDP packets since FLIDAS has no way of knowing when the video player is ready. You can optionally select to close the video player on completion of the replay. If you want to start up your video player manually you should not provide a command line and have it ready when you click **Finish**.

Note: If you use VLC video player the command line will look like this if you use the default UDP port (adapt the installation path): `C:\Program Files (x86)\VideoLAN\VLC\vlc.exe udp://@`

If you configure a different UDP port (works with default port as well) it will be:
`C:\Program Files (x86)\VideoLAN\VLC\vlc.exe udp://@:$PORT`

Note: If you need to execute a batch (.bat) file on Windows you cannot start the batch file itself you must start the command shell. To execute a batch file test.bat in c:\temp you need to use this command line:* `cmd /C c:\temp\test.bat`

This feature must be understood as a video preview feature. UDP streams itself are not protected against information loss so videos replay may be corrupted depending on your local computer workload and overall performance.

4.8.9 Analog to WAV

This exporter exports one or more analog subchannels of the same channel to a wave file. The exporter does not care if the data represents audio, so you can also convert other things for post processing in WAV. The exporter is available in all commercial versions.

You can select the directory to export into and the exporter will create files using the subchannel names.

The exporting also relies on several other attributes:

- **Sample rate:** This is taken from TMATS. If undeclared it will be estimated by the number of samples in the first packet and the time difference between first and second packet.

- Bit length of samples: This is taken from TMATS. If undeclared it will be taken from the first packet. Since it is not clear if regular WAV software supports sample sizes > 32 bit, the samples are converted to 32 bits if they have more (by shifting LSBs out of the sample) which might lose precision.
- Signed / unsigned: If TMATS declares the analog format to be "unsigned" it will use unsigned. All other cases will expect signed samples

Decoding of analog data is heavily depending on TMATS attributes. If some are missing, you can probably provide them with [TMATS overriding](#) or [TMATS defaults](#).

WAV files only know signed numbers. Therefore, unsigned samples will be stored as signed values with one bit more to cover the full range. For example, 16 bit unsigned values will be stored as 17 bit signed and therefore occupy 3 bytes per sample.

If any sample with more than 32 bits needs to be stored, only the upper 32 bits will be stored. This will lose precision and you need to shift the stored bits to the left by the number of lost bits to get the magnitude.

4.8.10 Analog to MAT

This exporter exports a single analog subchannel to a MAT file. The file format is defined for MATLAB but is also common to other scientific tools like the free [GNU Octave](#). The exporter is available in all commercial versions.

The exporter uses the [MATLAB File Properties Wizard Page](#).

The measurements will be exported to a row vector named **y**.

The timestamps will be exported as 64-bit floating-point values to a row vector named **t**. The timestamps are optional and further configured in the [Time Representation Wizard Page](#).

FLIDAS creates temporary files in the same directory as the export file during the export so the space needed will be temporarily up to double as required for the data.

Due to restrictions of the MAT file format, no more than $2^{31} - 1$ (2147483647) samples can be exported.

Decoding of analog data is heavily depending on TMATS attributes. If some are missing, you can probably provide them with [TMATS overriding](#) or [TMATS defaults](#).

If the sample rate is not provided in TMATS it will be estimated by the number of samples in the first packet and the time difference between first and second packet.

4.8.11 Ethernet to PCAP

This exporter exports an Ethernet subchannel (one physical network inside an Ethernet channel) into a PCAP file. This file type is supported by many Ethernet analyzers, for example the free software [Wireshark](#). The exporter is available in all commercial versions.

Due to limitations of the PCAP format, timestamps will be reduced to microsecond's precision and frames that indicated an error in the packet entry will not explicitly indicate this in the PCAP file. If the raw data stored in the CH10 file for frames with errors indicates an error itself, other applications might detect this themselves.

If the CH10 recorder only recorded the Ethernet frame's payload, the PCAP file will assume that the payload is IPv4 or IPv6.

Note: The timestamps of the Ethernet frames are not corrected according to the definition of the timestamp position indicated in the CH10 Ethernet packet definition. This would only be possible if the speed of the transmission is known which is not available most of the time.

4.8.12 PCM Throughput Payload

This exporter exports the payload of one PCM channel in throughput mode. It is available in all commercial versions.

The PCM bit stream is exported in transmission order and byte wise. Therefore, the first transmitted bit is stored as MSB of the first byte, the 9th transmitted bit is stored as MSB of the second byte. This means that FLIDAS has already done byte swapping from the CH10 storage structure.

Corrupted PCM packets will be skipped, i.e., there may be gaps in the bit stream.

Start and end time of the exported time range will be respected with a precision of the byte and not the bit level.

4.8.13 Payload to File

This flexible exporter exports the payload of various protocols to a file. The currently supported protocols are UART or message data subchannels or UDP packets (enclosed in IP4 packets in Ethernet format 0 packet entries, no fragmentation support). The exporter is available in all commercial versions.

The export is totally unaware of the type of the payload so you may use it for whatever type of data your system under test offers. Examples are:

- NMEA 0183 sentences on an UART subchannel or UDP packets can be stored in a text file
- Videos streamed inside UDP can be extracted to video files

The payload is simply saved in sequence to a file. No external timestamps or metadata will be stored, just the raw data of the protocol.

The exporter is a CH10 data type exporter as well as an encapsulated protocol exporter because it can work directly as on a UART subchannel or on UDP encapsulated in Ethernet subchannels.

The UART or message data export is started from the [Repository View](#), the [Channel View](#) or the [UART Communication Summary](#) or the [Message Data Communication Summary](#). The UDP export is started from the [Ethernet Communication Summary](#).

4.8.14 General CSV Exporting

There are several exporters that allow exporting one or more channels or subchannels (also from different channels) or parameter sets into a text file in the CSV format. This can be read from different tools like Microsoft Excel and the free [Open Office Calc](#) or the free [CSVed](#).

All these exporters use the [CSV Properties Wizard Page](#).

All CSV exports can configure none, one or more timestamp columns using the [Time Representation Wizard Page](#). The timestamps can be of numeric or string type.

4.8.14.1 ARINC 429 to CSV

This exporter exports one or more ARINC 429 subchannels. It is available in all commercial versions.

Besides the [optional time columns](#) the export has these columns:

Column title in CSV	Description
Channel#	The number of the CH10 channel this subchannel was part of
Subchannel#	The number of the subchannel within the channel (1-based)
Label [oct]	The octal label number of the ARINC 429 message
SDI	The decimal ARINC 429 source destination identifier
SSM	The decimal ARINC 429 sign status matrix
Parity	The ARINC 429 parity bit
Data bits [dec]	The decimal 19 bit data content of the ARINC 429 message
Format error	true when there was a format error, false otherwise
Parity error	true when there was a parity error, false otherwise

4.8.14.2 MIL-STD 1553 to CSV

This commercial exporter exports one or more MIL-STD 1553 channels. It is available in all commercial versions.

This exporter has an additional configuration page with the configurations shown in the following table. The table also shows the column titles that are used in the CSV file.

Configuration option	Description	Column title in CSV
Message data	Adds a column with the hexadecimal data words of the message	Data

Decode command words	Adds six additional columns with the interpretation of the receive and transmit command words. RT address, subaddress and word count will be decoded.	Rx RT / Rx SA / Rx WC / Tx RT / Tx SA / Tx WC
Decode status words	Adds 18 additional columns with the interpretation of the receive and transmit status words. The RT address and all flags will be decoded	<p>All columns will start with <i>Rx St:</i> for the receive status word and <i>Tx St:</i> for the transmit status word. This is followed by:</p> <ul style="list-style-type: none"> • RT • Msg Err • Instr • SRQ • Brdc Rx • Bsy • Subsys • Dyn BC acc • Term
Gap timing	Adds two additional columns with timing information about up to two gaps in the transmission (time between data/command word and 1st and 2nd status word). The time will be in 100 nanoseconds.	1st gap [100ns] / 2nd gap [100ns]
Protocol errors	Adds six columns to show the error indicators from the packet entry. Note that the IRIG 106 standard gives no exact definition about the meaning.	Format Error / Invalid Word / Message Error / Response Timeout / Sync Type Error / Word Count Error

Besides the optional time columns and optional columns from above the export has these columns:

Column title in CSV	Description
Channel#	The number of the CH10 channel this bus was recorded with
Bus	Bus A or B from the dual redundant bus system
Message type	<p>One of the following types of messages:</p> <ul style="list-style-type: none"> • BCRT: Bus controller to remote terminal • RTBC: Remote terminal to bus controller • RTRT: Remote terminal to remote terminal • MCNoData: Mode code without data word • MCTxData: Mode code with one data word transmitted by a remote terminal • MCRxData: Mode code with one data word received by a remote terminal

	<ul style="list-style-type: none"> • BCRTBroadcast: Bus controller to remote terminals broadcast • RTRTBroadcast: Remote terminal to remote terminals broadcast • MCBroadcastNoData: Mode code broadcast without data word • MCBroadcastRxData: Mode code with one data word received by several remote terminals
Rx Cmd	The hexadecimal receive command word or empty if there was no receive command word
Tx Cmd	The hexadecimal transmit command word or empty if there was no transmit command word
Rx Status	The hexadecimal receive status word or empty if there was no receive status word
Tx Status	The hexadecimal transmit status word or empty if there was no transmit status word
Mode Code	If the message was a mode command, this column contains the name of the mode code
# data words	The amount of MIL-STD 1553 data words that were stored in the packet entry. This does not necessarily match the word count from the command words.

4.8.14.3 PCM to CSV

This exporter exports one PCM channel to a CSV file. It is available in all commercial versions.

You can additionally configure:

- If minor frame losses should be exported too. These will create a line without actual data in the export but show the synchronization status and any defined timestamps of the loss. Note this is only detected for PCM throughput mode.
- If PCM words are exported too. If you are just interested in the minor frame timing and the synchronization status you can suppress the minor frame content.

Besides the optional time columns the export has these columns:

Column title in CSV	Description
Minor Frame Sync	The state of the minor frame synchronization
Major Frame Sync	The state of the major frame synchronization
FS (*) [hex]	The frame synchronization pattern in hexadecimal. The (*) shows the number of bits.
* (**) ... * (**)	Any number of defined subframe ID counters in decimal in sequence. The * is the counter name and the (**) is the number of bits.

W1 (*), W2 (*) ... Wn (*)	The decimal data words. The (*) shows the number of bits.
------------------------------	---

4.8.14.4 CAN Bus to CSV

This exporter exports one or more CAN bus subchannels. It is available in all commercial versions.

Besides the optional time columns the export has these columns:

Column title in CSV	Description
Channel#	The number of the CH10 channel this subchannel was part of
Subchannel#	The number of the subchannel within the channel (1-based)
Identifier [dec]	The decimal identifier
# data words	The number of data bytes contained as payload in the message
Data [hex]	The hexadecimal data bytes contained as payload in the message
RTR	true if the message was a remote transfer request, false otherwise
Data error	true if there was a data error, false otherwise
Format error	true when there was a format error, false otherwise

4.8.14.5 Analog to CSV

This exporter exports multiple analog subchannels of the same channel to a CSV file. It is available in all commercial versions.

The exporter allows overriding some TMATS configuration about the subchannel decoding. All options will be initialized from TMATS if available:

- Analog format (*R-x\AF-n-m*): You can choose from the analog formats defined in TMATS (like two's complement, offset binary ...) and there is an additional option for Sypris recorders (inverted offset binary)
- Mask (*R-x\AMSK-n-m*): A hexadecimal mask that can be used to mask out bits of the measurement (masks must be most significant bit first)
- Reverse bits (*R-x\AMTO-n-m*): By default FLIDAS assumes that the measurements use most significant bit first order. You can reverse this here.

The export configuration can also define if subchannels with lower sampling frequency repeat old samples at the sampling times of the higher frequent subchannels or if they leave the measurements empty. Assume subchannel 1 samples with 100 Hz and subchannel 2 with 50 Hz. You can choose between these two types of results:

Export only new samples enabled:

Time	Subchannel 1	Subchannel 2
0 ms	1	

10 ms	2	1
20 ms	3	
30 ms	4	2
40 ms	5	
50 ms	6	3

Export only new samples disabled:

Time	Subchannel 1	Subchannel 2
0 ms	1	
10 ms	2	1
20 ms	3	1
30 ms	4	2
40 ms	5	2
50 ms	6	3

Besides the optional time columns the export has one column for each active and selected subchannel that displays the subchannel name and the chosen decoding options.

4.8.14.6 Discrete to CSV

This commercial exporter exports a single discrete channel (containing up to 32 discretes). It is available in all commercial versions.

FLIDAS will always right align the discretes before doing further processing and masking. This means even if your recorder stores 8 discretes with MSB alignment in the 32-bit word, FLIDAS normalizes this to an LSB alignment.

This exporter has an additional configuration page with the following options:

- **Bit mask for discrete events:** This LSB aligned, hexadecimal bit mask can be used to suppress discretes from processing. If a bit is unset the corresponding discrete will not be exported and have no influence at all. If more bits to the left are set where no discrete events are present these are ignored. To apply no masking use FFFFFFFF which is set by default.
- **Separate columns for each discrete:** If this option is enabled, the exporter will create a separate column for each discrete bit selected for the export. The columns will be labeled "Bit n" where n is the bit number of the discrete event where 0 is the rightmost. Without this option there is a single column labeled "Discrete event" containing the decimal state of all the selected discrete bits in a single value.
- **Export only changes:** Suppresses a line in the export if the discrete state of all selected bits hasn't changed.

Besides the optional time columns and columns described above the export has these columns:

Column title in CSV	Description
Channel#	The number of the CH10 channel that records the discrete events

4.8.14.7 UART to CSV

This exporter exports one or more UART subchannels to CSV. It is available in all commercial versions.

Besides the optional time columns the export has these columns:

Column title in CSV	Description
Channel#	The number of the CH10 channel this subchannel was part of
Subchannel#	The number of the subchannel within the channel (1-based)
Parity error	true when there was a parity error, false otherwise
Data length	The number of bytes of the UART data
Data [Hex] or Data [ASCII]	The data of the UART packet entry. The format can be controlled using the <u>data representation</u> configuration wizard page

Note: IRIG 106 chapter 10 does not require your recorder aligning UART packet entries with anything you might consider a message start or end. However, there might be additional guarantees of your recorder on top of IRIG 106 chapter 10.

4.8.14.8 NMEA 0183 to CSV

This exporter exports one UART subchannel, TSPI format 0 channel or UDP packets and interprets them according to the NMEA 0183 protocol. For UART and UDP you must make sure yourself that the source really conforms to this protocol since TMATS has no indication. TSPI will be filtered for NMEA 0183 or NMEA 0183 HS. The exporter is available in all commercial versions.

Note: The NMEA 0183 protocol contains navigation data amongst others and is supported by many GNSS like GPS, Galileo, GLONASS and Beidou.

The timestamp of each NMEA 0183 sentence is taken from the containing packet entry. For UART a packet entry may contain several NMEA 0183 sentence fragments and may contain transmission pauses, so the timestamp is not accurate. However, the sentence has been transmitted between this timestamp and the timestamp of the next packet entry. Note that timestamps on UART and TSPI packet entries are optional. If the hardware does not record this, all packet entries of one packet share the same timestamp with the packet.

Besides the optional time columns the export has these columns:

Column title in CSV	Description
---------------------	-------------

Error	An indication of two error situations: A wrong checksum and a wrong prefix (sentences are expected to begin with "\$" or "!").
Checksum [hex]	The checksum included in the sentence in hex
Data type	The data type, i.e., the type of the sentence
(Without title)	There will be one untitled column for each field of the sentence

Note: Checksums are optional on NMEA 0183 but if a sentence of a specific type with a checksum is seen on the NMEA 0183 source, all following sentences of that type are supposed to have one or are considered to have a checksum error. This makes sure incomplete sentences are invalid.

The UART or TSPI export is started from the [Repository View](#) or the [Channel View](#). For UART you can also start it from the [UART Communication Summary](#). The UDP export is started from the [Ethernet Communication Summary](#).

4.8.14.9 Message Data to CSV

This exporter exports one Message Data subchannel to CSV. It is available in all commercial versions.

Segmented messages inside the selected time slice will be joined to a single line and indicate if segments are missing.

Besides the [optional time columns](#) the export has these columns:

Column title in CSV	Description
Channel#	The number of the CH10 channel this subchannel was part of
Subchannel#	The number of the subchannel within the channel (1-based)
Data error	true when there was a data error in any of the segments, false otherwise
Format error	true when there was a format error in any of the segments, false otherwise
Segments missing	true when not all segments of the message were recorded, false otherwise
Data length	The number of bytes of all joined segments
Data [Hex] or Data [ASCII]	The data of all joined message segments. The format can be controlled using the data representation configuration wizard page

4.8.14.10 Fibre Channel to CSV

This exporter exports one or more fibre channel (FS or PH layer) channels to CSV. It is available in all commercial versions.

All exported channels must be of the same type (FS or PH layer).

Besides the [optional time columns](#) the export has these columns:

Column title in CSV	Description
Channel#	The number of the CH10 channel
Mode (only PH layer)	The recording mode used on the recorder, either stripped or non-stripped
Content (only PH layer)	The type of content either complete data frame or complete link control frame
Topology (only PH layer)	The topology where passive is the only defined meaning
Framing error	true when there was a framing error, false otherwise
CRC error	true when there was a CRC error, false otherwise
Overrun error	true when there was an overrun error, false otherwise
SOF delimiter	The start of frame delimiter like SOFc1 or SOFi3
EOF delimiter	The end of frame delimiter like EOFn or EOFt
Frame length	The number of bytes of the fibre channel frame
Frame header and data [Hex] or Frame header and data [ASCII]	The frame header and data of the fibre channel frame. The format can be controlled using the data representation configuration wizard page

4.8.14.11 Sampled parameters to CSV

This parameter exporter will take samples of a parameter set over time and stores the values with a timestamp in one or multiple CSV files. It is only available in the [Parameter or Parameter Streaming Version](#).

You can [configure the sampling mode](#) to use. If you export to a single file, you can configure the sampling period in milliseconds or sample asynchronously as both described in [parameter sampling](#). If you export to multiple files, you can use the separate timestamp sampling modes. In this case you need to start the **Sampled Parameters to Multiple CSV Exporter**.

Column headers will always contain the parameter name and you can optionally add mapping ID, path, engineering unit and comment. You can also choose to distribute these attributes over several lines.

4.8.14.12 ARINC 429 Message-Based Parameters to CSV

This exporter exports ARINC 429 protocol information of one or more subchannels to a CSV file in the same way as the [ARINC 429 to CSV Exporter](#). It extends its columns with one parameter decoded from each message. It is only available in the [Parameter or Parameter Streaming Version](#).

The exporter doesn't use a parameter set but instead uses the ICDs mapped to the exported subchannels with the active ICD mapping. For each message it will decode one parameter that applies to this message (label and SDI match) and has the longest bit length. The

exported protocol data will be extended by three columns for the parameter name, engineering unit, value and parameter path.

4.8.14.13 IENA to CSV

This commercial exporter exports IENA protocol information on an Ethernet or message data subchannel to a CSV file. It is available in all commercial versions.

For message data subchannels the assumption is that the recorded data is the IENA layer (i.e., UDP payload).

The IENA properties configuration wizard page allows you defining where to find the IENA packets in your data.

This exporter has an additional configuration page **Configure IENA export content** with the configurations shown in the following table. The table also shows the column titles that are used in the CSV file.

Configuration option	Description	Column title in CSV
Include Airbus specific fields	Some of the IENA protocol information seems to be used only by the original Airbus users but no other users of the IENA protocol. You can choose here to ignore this information. If you include it additional columns will show up in the export.	<ul style="list-style-type: none"> • Aircraft • Equipment • Diffusion group • N4 transmission prohibited • Never been synchronized • No external synchronization • ETR flag
Include IENA raw time	Will add a column with the raw value of the IENA time stamp in microseconds.	IENA raw time
Include IENA decoded time	Will add a column with the IENA time stamp decoded to day of year down to microseconds	IENA decoded time
Include payload	Will add a single column with the space separated hexadecimal bytes of the IENA payload (after sequence number, before end field). For Ethernet based IENA recordings with IP fragmentation only the first fragment's payload is included. IP fragmentation is not recommended for IENA anyway as it increases latency due to reassembly.	Payload

Besides the optional time columns and optional columns from above the export has these columns:

Column title in CSV	Description
Key [hex]	The hexadecimal IENA key
Parameter type	Parameter type for this IENA key (combination of positional / message / delay field flags). Possible options are P, D, N, M, Q.
Words per parameter	The number of 16-bit words that are used for each parameter in the IENA packet
IENA payload word length	Number of 16-bit words contained in the IENA payload for this IENA packet
Sequence number	The sequence number of this IENA packet
Discarded by CUB	True or false depending if this IENA flag is set in the key status

The normal time stamps exported to the CSV file are related to the reception time of the first Ethernet frame / message data packet entry related to the IENA packet. However, the IENA packets are reordered according to the IENA time stamps within the limits of the **Message reordering time frame** given on the IENA properties wizard page.

4.8.14.14 iNET-X to CSV

This commercial exporter exports iNET-X protocol information on an Ethernet subchannel to a CSV file. It is available in all commercial versions.

The iNET-X properties configuration wizard page allows defining where to find the iNET-X packets in your data and how stream IDs are displayed.

This exporter has an additional configuration page **Configure iNET-X export content** with the configurations shown in the following table. The table also shows the column titles that are used in the CSV file.

Configuration option	Description	Column title in CSV
Include source / destination	Controls if IP and UDP port addresses will be included in the export. Usually the stream ID is enough. If you include this information additional columns will show up in the export.	<ul style="list-style-type: none"> • Source IP • Destination IP • Source Port • Destination Port

Besides the optional time columns and optional columns from above the export has these columns:

Column title in CSV	Description
PTP time	The PTP time stored in the iNET-X packet header

Stream ID [hex] or Stream ID [dec]	The hexadecimal or decimal stream ID depending on your setting in the iNET-X properties wizard page
Sequence number	The sequence number of this iNET-X packet
Packet length	The packet length of the iNET-X packet, i.e., excluding MAC, IP and UDP header
Error bit	TRUE or FALSE whether the iNET-X error bit was set
Lost FIFOs	The lost count value of the iNET-X packet header
Time out	TRUE or FALSE whether the iNET-X time out bit was set
Payload	All iNET-X payload bytes in a single column in hexadecimal with a space as separator between each byte

The normal time stamps exported to the CSV file are related to the reception time of the Ethernet frame containing the iNET-X packet. However, the iNET-X packets are reordered according to the iNET-X time stamps within the limits of the **Message reordering time frame** given on the iNET-X properties wizard page.

4.8.14.15 TSPI / CTS to CSV

This exporter exports a TSPI / CTS channel to CSV. It is available in all commercial versions.

Besides the optional time columns the export has these columns:

Column title in CSV	Description
Channel#	The number of the CH10 channel this subchannel was part of
Data length	The number of bytes of the TSPI / CTS data
Data [Hex] or Data [ASCII]	The data of the TSPI / CTS packet entry. The format can be controlled using the <u>data representation</u> configuration wizard page

Note that there should be an individual TSPI format 1 exporter to a data transfer file as soon as the specification problems with format 1 are fixed. If you have sample data, you can help improving support for format 1 in FLIDAS.

4.8.15 Sampled Parameters to MAT

This parameter exporter takes samples of a parameter set over time and stores the values with a timestamp in one or multiple MAT files (ready to be used in e.g., MATLAB). Only numeric or enumeration parameters are supported. The exporter is only available in the Parameter or Parameter Streaming Version.

Two variants of this exporter exist, one that exports all parameters with a single timestamp vector into one MAT file while the other creates separate MAT files for each parameter with

separate timestamp vectors. The differences for the export variant with several export files are described at the end of this section.

The exporter uses the [MATLAB File Properties Wizard Page](#).

The [sampling mode can be configured](#) to a sampling period in milliseconds or to sample asynchronously as both described in [parameter sampling](#).

The timestamps will be exported as 64-bit floating-point values to a row vector named **t**. The timestamps are optional and further configured in the [Time Representation Wizard Page](#).

There will be one value of all parameters for each timestamp so if you are exporting parameters with different frequency of occurrence, the infrequent one will have as many values as the frequent one.

Each measurement will be stored in an own row vector with one value per timestamp. Invalid values will be exported coded as "Not a number" (NaN).

The measurement's variable name will be derived from the parameter name. FLIDAS will replace unsupported characters for variable names or replace the whole name by "Var" extended with a sequential number if it cannot be turned into a valid variable name.

Enumeration parameters will be exported as numeric using their raw value.

FLIDAS creates temporary files in the same directory as the export file during the export so the space needed will be temporarily up to double as required for the data.

4.8.15.1.1 Exporting to Multiple MAT Files

If you use the exporter variant which exports to separate MAT files, each with one parameter and separate timestamps the following differences apply:

- You need to start it as a separate exporter titled **Sampled Parameters to Multiple MAT**
- Only the separate timestamp [sampling modes](#) are available.
- The timestamp vector names will be "t_" followed by the parameter vector name. So, you can load several files without conflicting names.
- Of course, infrequent parameters have less timestamps than frequent ones. This is the purpose of having separate timestamp vectors.
- You need to select an export directory rather than a file. The file names will be chosen based on the parameter names

4.8.16 CVSD to WAV

This parameter exporter exports a single CVSD audio parameter contained in a PCM stream to a wave file. The exporter is only available in the [Parameter or Parameter Streaming Version](#).

You can configure the created WAV file in the [WAV File Properties Wizard Page](#).

4.8.17 PCM Numeric Parameter to WAV

This parameter exporter exports a single numeric parameter contained in a PCM stream to a wave (audio) file. The exporter is only available in the [Parameter or Parameter Streaming Version](#).

You can configure the created WAV file in the [WAV File Properties Wizard Page](#).

The PCM parameter you are using must satisfy some conditions:

- It must be numeric
- Minimum and Maximum value for the parameter must be available. This can be due to configuration in the ICD or because it could be calculated from the setup automatically. It will only be an issue in special situations.

The export will then scale the parameter's values in a way that the full value range of the parameter is mapped into the full value range of the WAV file's samples.

4.8.18 Byte Array Parameter to Binary File

This parameter exporter will export the raw data decoded from a Byte Array parameter (currently only [PCM Simple CSV ICD](#)) to a binary file. It is only available in the [Parameter or Parameter Streaming Version](#).

The use case for this is for example if you have blocks of an MPEG transport stream video in a PCM frame and want to extract it to a binary file. This exporter just handles binary data and does not know about MPEG. However, usually video players can replay the binary file (if you name it *.ts) and are able to find the proper synchronization in the data that is not synchronized to a specific starting position of an MPEG packet by the exporter. Probably there are many other use cases where you can use this to extract specific types of raw data.

4.8.19 IRIG 106 Chapter 7 Exporting

There are several exporters that allow exporting [IRIG 106 chapter 7](#) data from a PCM channel.

All these exporters use the [IRIG 106 Chapter 7 Properties Wizard Page](#).

4.8.19.1 IRIG 106 Chapter 7 to CH10

This commercial exporter exports IRIG 106 chapter 10 data contained inside an IRIG 106 Chapter 7 PCM stream to a CH10 file.

The exported data is not necessarily a valid CH10 file, just a chapter 10 packet dump. FLIDAS can still load these files. However, most likely you will not find TMATS data in the beginning of the file. You can use the [Replace TMATS in CH10](#) exporter to inject an external TMATS file into the chapter 10 file.

The exported data may include chapter 10 packets with checksum errors. FLIDAS will detect these when loading the file and handle the data accordingly. But FLIDAS will not suppress them from being exported from the chapter 7 data.

If you define a time slice for the export this will relate to the time stamps of the chapter 7 PCM data, not the encapsulated chapter 10 data.

4.8.19.2 IRIG 106 Chapter 7 to Ethernet PCAP

This commercial exporter exports Ethernet MAC frames contained inside an IRIG 106 Chapter 7 PCM stream to a PCAP file.

The time stamps of the MAC frames relate to the PCM minor frame they started in.

4.8.19.3 IRIG 106 Chapter 7 to IP PCAP

This commercial exporter exports Internet Protocol (IP) packets contained inside an IRIG 106 Chapter 7 PCM stream to a PCAP file.

The time stamps of the IP packets relate to the PCM minor frame they started in.

4.8.20 Further Exporters

Further exporters are already available under additional licenses or can be ordered as dedicated development.

4.9 Protocol Viewers

See the [protocol viewers concept](#) for a general explanation of protocol viewers and how to start them. They are [configured in a similar way like exporters](#) and allow selecting a [time slice](#) and probably more options.

The available protocol viewers are shown in the subsections of this chapter.

4.9.1 Communication Summaries

Communication summaries are [protocol viewers](#) that analyze a whole channel or subchannel for the message types occurring on protocol level and accumulate statistics on each message type and problems on the communication media. They are also starting point for other exporters and protocol viewers.


Communication summaries are part of all commercial FLIDAS versions and are available for a variety of CH10 data types. They are described in the following subchapters. The basic operation and content are common for them.

Communication summaries are opened from the [Channel View](#) when you right-click a channel or subchannel, select **Protocol Viewers** and the according entry from the sub menu. CH10 data types that have a subchannel level can have a communication summary for the subchannel or the channel, where the channel summary simply provides the information for all subchannels with a single processing step.

The communication summary can be limited to a time slice.

Communication summaries often collect detailed information on special events of the protocol and not only accumulated data. If there are a lot of special events, the amount of collected data may slow down the display of the events on the special events page. You can therefore configure to limit the number of special events for an individual group of special events using the wizard page **Configure special event reference properties**. You can also provide no limit.

Since the collection of the data looks through the whole CH10 file, the action can take some time and therefore shows a progress bar and can be cancelled. The summary will then just show the information collected until the canceling.

You can also send the data processing to the background once it is running. The summary will open when it finishes and you can continue working until then. If you want to bring up a running data processing job to cancel it or see the status, click at the symbol  to the lower right of the Window.

The result will have several pages that you can select with the tabs on top.

4.9.1.1 Messages Page

The messages page will show a table with one line for each detected message type. The line will show at least these columns:

- The attributes that identify the message type
- Count: The number of messages
- The number of all error conditions of the packet entry explicitly defined in IRIG 106 chapter 10 if they relate to the specific message type. Messages that deal more with problems outside of the message like a corruption of the transmission medium are displayed on a separate page.
- Avg period: The average time between two of these messages in milliseconds. This is calculated by dividing the count by the processing duration (see processing info page). It therefore takes into account times without transmission of this specific message in the beginning and end and can therefore be inconsistent with the following columns for min/max period.
- Min / max period: The minimum and maximum time between two of these messages in milliseconds.

Lines of message types with any error condition are marked in red.

Some communication summaries show a hierarchy of data. They have the subchannel level on top and one or more message level below. The higher level always shows the summary of all elements below.

If you select and right-click one or more entries in some of the communication summaries you will be able to start exporters or protocol viewers on this type of data. This is especially helpful if your TMATS setup has no (or wrong) definition of the subchannels. The communication summary will detect the available subchannels and allow you starting an export on them with a right-click. Note that you don't need to wait until the full

communication summary has been processed. Often it is enough to look into the file for a second and then cancel it to see the subchannels found in the first second.

A right-click on any subchannel element also allows seeing the subchannel TMATS properties by selecting **Properties**.

Some communication summaries detect enclosed protocols inside the regular chapter 10 data. These allow specific exporter or protocol viewer actions on these enclosed protocols.

To see how to export all the visible data of the table to a CSV text file read the section about [table exporting](#).

4.9.1.2 Special Events Page

The special events page provides details on some types of protocol errors and other significant events in the protocol. A table shows the special events grouped by subchannels, types and possibly other criteria. Groups are suppressed in general if no special events are contained. Each group shows these columns:

- **Count:** Shows the number of special events in this category. If you limited the maximum number of errors per category and the limit suppressed special events then the number will be followed by "(more)".
- **Description:** A general description of the group of special events

Each special event group can be expanded to show either subgroups or individual special events. The individual special events will show these columns:

- **Time:** The timestamp of the special event
- **Description:** More information on this special event if not already explained enough on the group above.

You can right-click each individual special event. The upcoming menu shows you more options for analysis depending on the type of event:

- **Show Packet Entry:** Shows the packet entry in the [Packet Entry View](#) of its containing packet.
- **Show Packet:** Shows the packet containing the special event in the [Packet View](#). Only the packets of the same CH10 channel will show so you can inspect in this context. This option will only be available when the analyzed communication protocol is not split into packet entries in IRIG 106 chapter 10 (time packets are an example).
- **Show Data:** This option is only available if the special event does not refer to a packet entry or packet but something further decoded. An example is PCM where it would refer to a minor frame which is not necessarily a packet entry (in PCM throughput). In this example it will show the [Packet Entry View](#) of some packets and highlight the minor frame in the minor frame area of the view.

A double-click on the event takes you to the details of the packet entry / packet / data as well.

As on most tables in FLIDAS you can export this table to a CSV file to get some kind of report.

4.9.1.3 Processing Info Page

The processing info page shows some of the circumstances of the processing:

- Time slice start / end: The user selected time slice for the summary
- Processing start / end: Timestamp of first and last processed packet entry on the channel or subchannel. This can differ from time slice start/end for example if the channel contains pauses or if the recording on this channel was not started at the same time as the recorder.
- Processing duration: Time difference between first and last processed packet entry. Therefore, does not include the duration of the last packet entry itself
- If the processing was canceled by the user or processed to the end
- If the processing was limited to a single subchannel it will be indicated
- Count of packets on the channel that have been excluded from the summary or which have been used but are unreliable because of:
 - Lost packets: Data not used. The count shows the minimum number of lost packets. Due to the sequence counter mechanism in each packet, the number can be higher by integral multiples of 256, e.g., 5 lost packets or 261 lost packets cannot be distinguished.
 - Data checksum error: Data not used. Requires that the user has chosen to check data checksums when processing the CH10 source.
 - RTC sync error: Data is used but time related information (like periods) is unreliable, if no absolute secondary header timing is used
 - Secondary header checksum error: Data is used but the packet time stamps that play a role in time slicing are unreliable. If packets pass the time filter the packet entry timing is not affected though for most data types where the packet entries carry own time stamps.
 - Data overflow error: Data is used but may not be complete

These error counts will relate to the whole CH10 file even when you are time slicing (unless you cancelled - then only up to the processing point).

4.9.1.4 Other Pages

Some communication summaries have additional pages like "Bus Overview" (summary of errors, bus load etc. on bus level) or a transmitter overview when the transmission medium has different transmitters.

4.9.1.5 ARINC 429 Communication Summary

The ARINC 429 Communication Summary is a communication summary to see a summary of ARINC 429 message counts, message periods and errors occurred within one channel or subchannel.

You can open it in the Channel View when you right-click an ARINC 429 channel or subchannel and select **Protocol Viewer > View ARINC 429 Communication Summary**.

The screenshot shows a window titled "ARINC 429 (2048) summary:TTC-ARINC-429_110_194626". It contains a table with columns: Messages, Bus Overview, Special Events, and Processing Info. The table lists messages with subchannel, label, SDI, count, and periods. A context menu is open over the table, showing options like "Export", "Protocol Viewers", "Export Table Content to CSV", "Copy Table Cell to Clipboard", "Use Monospace Font", "Expand All", "Collapse All", and "Properties".

Subchannel	Label	SDI	Count	Avg period	Min period	Max period
	266	11	91	249.844	249.994	250.016
	273	11	91	249.844	249.994	250.016
	304	10	91	249.844	249.994	250.016
	326	1	91	249.844	249.994	250.016
	372	0	91	249.844	249.994	250.016
	374	1	91	249.844	249.994	250.016
2			759	29.955	2.879	905.280

All messages on the same subchannel with the same label and SDI are considered to be of the same message type.

Despite the regular columns this data shows up:

- Subchannel: The ARINC 429 subchannel number (1 ... 256)
- Label: The octal label number
- SDI: The binary source destination identifier

The data has a fixed ordering with ascending order on these columns:

1. Subchannel
2. Label
3. SDI

The entries for the subchannels allow you starting exporters or protocol viewers or see the properties of that subchannel with a right-click.

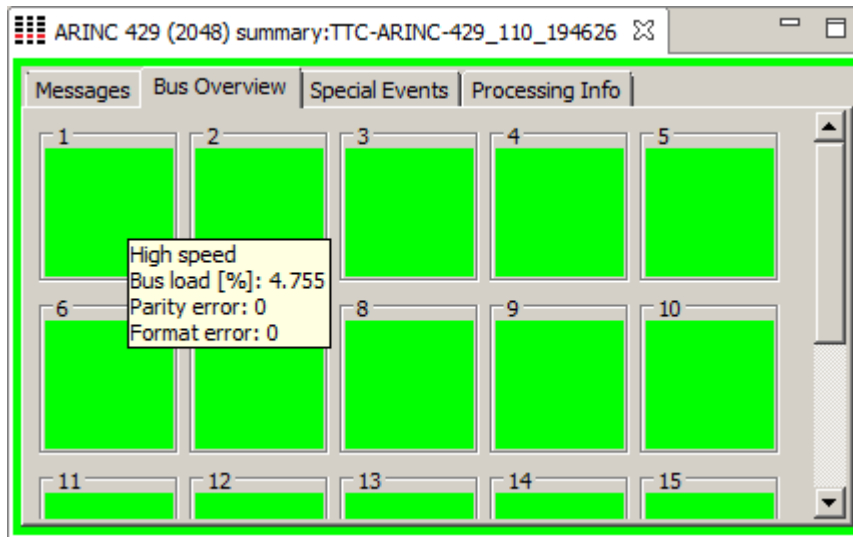
The additional **Bus Overview** page shows a diagram with one block for each subchannel with some additional information:

- Subchannel number and name from TMATS
- The block will have red color if any error occurred, green otherwise

Tool tips with further information pop up if you hover with the mouse over one subchannel block:

- High or low speed

- Bus load in percent (for low speed two values for 12.5 or 14.5 kBaud buses)
- Number of detected parity and format errors



The **Special Events** page provides information on each special event in two groups for each subchannel: Parity and format errors.

4.9.1.6 Ethernet Communication Summary

The Ethernet Communication Summary is a communication summary to see a summary of Ethernet frame counts, frame periods and errors occurred within one channel or subchannel. There is further decoding for IP4 and UDP packets.

You can open it in the Channel View when you right-click an Ethernet channel or subchannel and select **Protocol Viewer > View Ethernet Communication Summary**.

Subchannel / type	Source	Destination	Payload type	Count	Data length error	Data CRC error	Frame error	Frame CRC error	Avg period	Min period	Max period	Bit rate
IP4	10.14.1.7	237.31.0.32	UDP	118685	0	0	0	0	3.759	0.135	11.088	3123859
UDP	0	10001		118685	0	0	0	0	3.759	0.135	11.088	3123859
MAC	00:0D:A8:15:6F:09	01:00:5E:00:00:01	IP4	4								
IP4	10.30.5.1	224.0.0.1	IGMP	4								
MAC	00:0D:A8:15:6F:0A	01:00:5E:00:01:81	IP4	1115								
IP4	10.30.5.2	224.0.1.129	UDP	1115								
UDP	319	319	PTP (Event)	446								
UDP	320	320	PTP (General)	669								
MAC	00:0D:A8:B5:6F:0A	01:80:C2:00:00:0E	LLDP	14								
MAC	00:14:50:00:1B:68	01:00:5E:1F:00:20	IP4	892								
IP4	10.78.11.3	236.31.0.32	UDP	892								
UDP	5006	41089		892	0	0	0	0	500.195	178.059	825.582	10840

All frames on the same subchannel with same source and destination MAC address and ether type field are considered to be of the same message type. For this purpose all ether type fields below 0x600 (i.e., ether type not used) are considered to be of the same type.

If IP version 4 packets are detected as payload, these are considered as own message types below the Ethernet frame identified by source and destination IP and the enclosed protocol. *Note that the statistics are for the IP4 packets and not the datagrams. In case of fragmentation this means the statistics will be calculated from the fragments.*

If UDP packets are detected as payload of an IP4 packet these are considered as own message types below the IP4 packet identified by source and destination port. *Note that FLIDAS will associate data from a fragmented IP4 datagram correctly to the corresponding UDP message type. The first fragment (fragment offset = 0) will then be used for the count, ports and periods. Every fragment and duplicate are taken into account for the bit rate and error counts. Statistics for fragments where no first fragment is found is omitted.*

Despite the regular columns this data shows up:

- Subchannel / type: For subchannels the Ethernet network number (1 ... 256). Otherwise the type of data displayed, either **MAC** (frame), **IP4** (packet) or **UDP** (packet)
- Source / destination: The hexadecimal source and destination MAC addresses, decimal IP address or decimal UDP ports (for Ethernet, IP4 and UDP level)
- Payload type:
 - For the Ethernet frame level, the ether type field:
 - **(IEEE802.3 length)** to indicate ether type is not used and the field contains length information
 - **IP4** to indicate IP version 4 packets
 - **IP6** to indicate IP version 6 packets
 - **ARP** to indicate Address Resolution Protocol packets
 - **LLDP** to indicate Link Layer Discovery Protocol packets
 - Otherwise the hexadecimal ether type
 - For the IP4 packet level, the enclosed protocol:
 - **TCP** (Transmission Control Protocol)
 - **UDP** (User Datagram Protocol)
 - **IGMP** (Internet Group Management Protocol)
 - **ICMP** (Internet Control Message Protocol)
 - Otherwise the hexadecimal protocol
 - For the UDP packet level, the enclosed protocol:
 - **PTP (General)** (IEEE 1588 Precision Time Protocol (General Messages))
 - **PTP (Event)** (IEEE 1588 Precision Time Protocol (Event Messages))
 - **NTP** (Network Time Protocol)
- Data length error, data CRC error, frame error, frame CRC error: The number of errors detected by the recorder
- Bit rate: Number of bits for this message type (or all messages on subchannel level) per second. The number of bits includes all bit times from preamble to inter frame gap.

The data has a fixed ordering with ascending order on these columns:

1. Subchannel
2. Source
3. Destination
4. Payload type

The entries for the subchannels allow you starting exporters or protocol viewers or see the properties of that subchannel with a right-click.

The entries on the UDP level allow you starting exporters or protocol viewers on the UDP payload. This allows for example to export a video streamed on UDP to an MPEG file or interpret UDP payload as NMEA 0183 protocol.

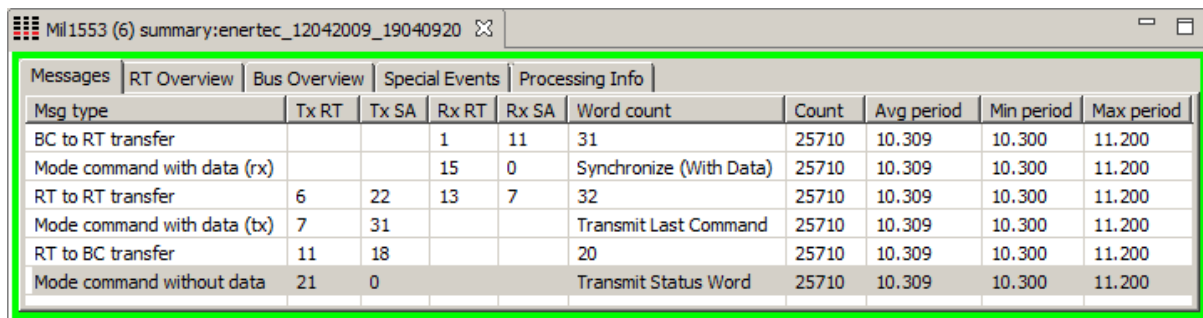
Note: If the CH10 recorder only recorded the Ethernet frame's payload a further classification is not possible due to missing information. They will show "unknown" for source, destination and payload type and will be grouped to one message type per subchannel. For further information you must configure your CH10 recorder to record full MAC frames.

The **Special Events** page provides information on each special event in four groups for each subchannel: Data length, data CRC, frame and frame CRC errors.

4.9.1.7 MIL-STD 1553 Communication Summary

The MIL-STD 1553 Communication Summary is a communication summary to see a summary of MIL-STD 1553 message counts, message periods and errors occurred within one channel.

You can open it in the Channel View when you right-click a MIL-STD 1553 channel (regular MIL-STD 1553 only, i.e., format 1) and select **Protocol Viewer > View MIL-STD 1553 Communication Summary**.



The screenshot shows a window titled "Mil1553 (6) summary:enertec_12042009_19040920". It contains a table with the following data:

Msg type	Tx RT	Tx SA	Rx RT	Rx SA	Word count	Count	Avg period	Min period	Max period
BC to RT transfer			1	11	31	25710	10.309	10.300	11.200
Mode command with data (rx)			15	0	Synchronize (With Data)	25710	10.309	10.300	11.200
RT to RT transfer	6	22	13	7	32	25710	10.309	10.300	11.200
Mode command with data (tx)	7	31			Transmit Last Command	25710	10.309	10.300	11.200
RT to BC transfer	11	18			20	25710	10.309	10.300	11.200
Mode command without data	21	0			Transmit Status Word	25710	10.309	10.300	11.200

All messages with both command words equal are considered to be of the same message type. Messages where the command words couldn't be decoded have one additional message type.

Despite the regular columns this data shows up:

- Message type: A classification of the message
- Tx RT: The transmit RT address
- Tx SA: The transmit RT subaddress
- Rx RT: The receive RT address
- Rx SA: The receive RT subaddress
- Word count: The word count taken from receive command if available or from the transmit command otherwise. In case of mode codes this shows the mode code type

The data has a fixed ordering with ascending order on these columns:

1. Tx RT (empty first)

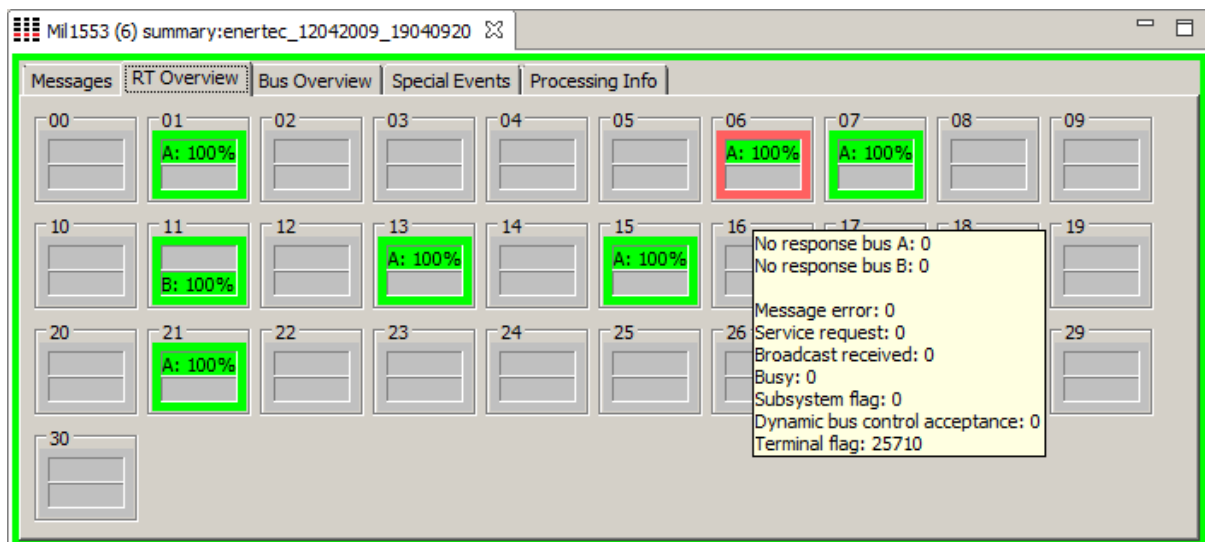
2. Rx RT (empty first)
3. Tx SA
4. Rx SA

The additional **RT Overview** page shows a diagram with one block for each RT with some additional information:

- RT number
- Percent of successful responses of the RT on bus A and B. I.e., how often responded the RT when it was required to respond. This indication has an additional color code:
 - Gray for no activity commanded on the bus
 - Red for at least one missing response on the bus
 - Green if there was activity on this bus and the RT always responded
- The RT block will have a color code:
 - Gray if there was no commanded activity on this RT
 - Red if any error was detected (no response, message error, busy, subsystem / terminal flag)
 - Green if there was activity on this RT but no error

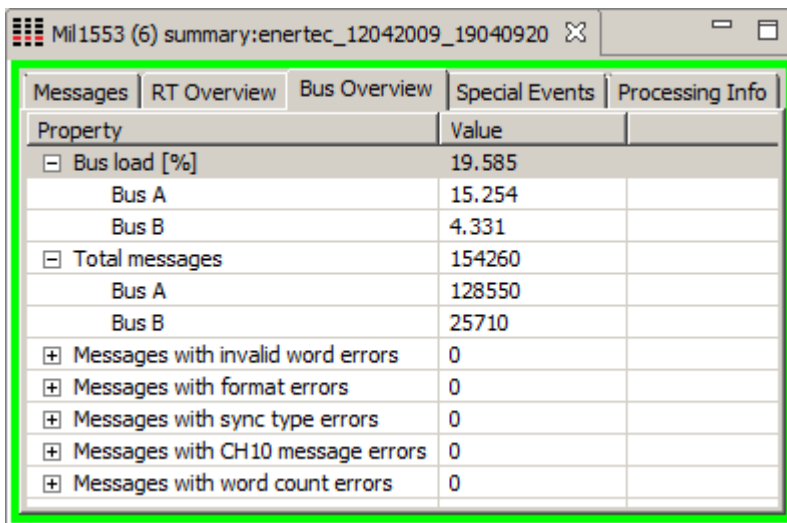
Tool tips with further information pop up if you hover with the mouse over the RT block:

- Number of no response errors separate for bus A and B
- Number of occurrences of status word flags:
 - Message error
 - Service request
 - Broadcast received
 - Busy
 - Subsystem flag
 - Dynamic bus controller acceptance
 - Terminal flag



The additional **Bus Overview** page shows a table with some statistics on the whole bus system. All statistics are available for bus A and B separate and combined:

- Bus load in %: The percentage of the available time covered with transmissions on the bus. The time includes required gaps in the message format and the minimum inter message gap.
- Total messages
- The count of messages with errors indicated explicitly in the CH10 file format for packet entries:
 - Invalid word
 - Format
 - Sync type
 - Word count error
 - Message error (*Note: The use of this error flag is especially unclear in IRIG 106 CH10. It seems to be used by at least one vendor to indicate that there was any error, e.g., a sync type error is additionally indicating a message error. It is not expected that this error should be related to the message error flag in the status word as this would relate to the previous message.*)
 - *Note: The response time out error is not listed here as this error is allocated to the individual RT in the RT Overview page. The errors above are not allocated to RTs as these errors could make a decoding of the message and therefore decoding of receive and transmit RT unsafe. They could also indicate errors on the transmission medium or the bus controller side so the RT wouldn't be causing the problem.*



Property	Value
[-] Bus load [%]	19.585
Bus A	15.254
Bus B	4.331
[-] Total messages	154260
Bus A	128550
Bus B	25710
[+] Messages with invalid word errors	0
[+] Messages with format errors	0
[+] Messages with sync type errors	0
[+] Messages with CH10 message errors	0
[+] Messages with word count errors	0

The **Special Events** page provides information on each special event. The following groups exists:

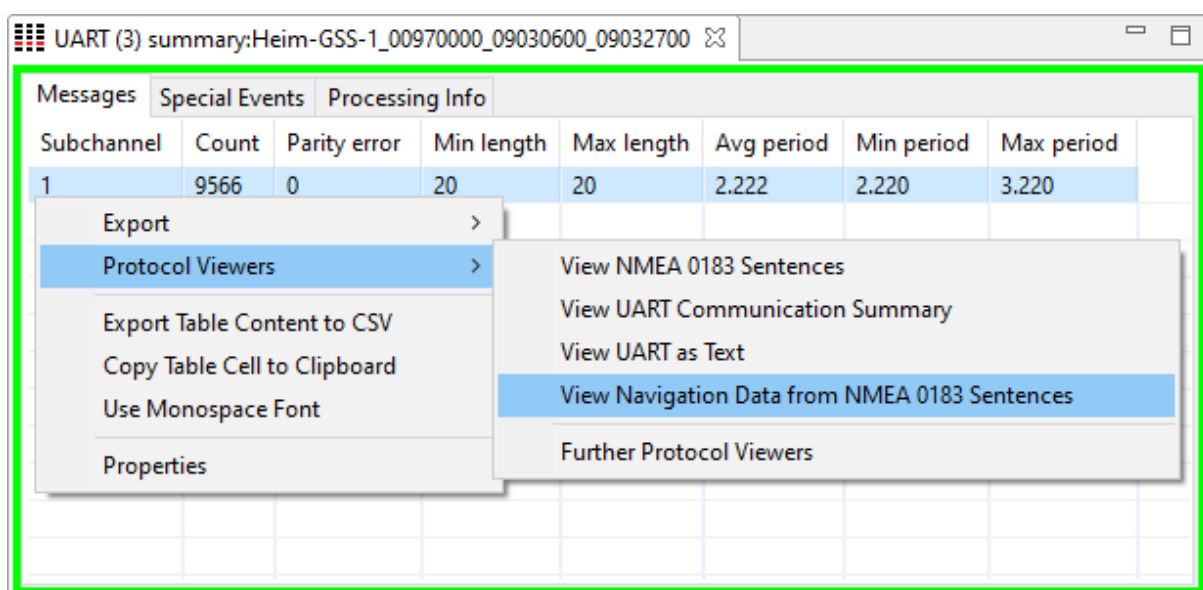
- **Invalid messages:** This group contains subgroups for each error condition directly captured in the IRIG 106 CH10 format: Invalid word, format, sync type, word count, message error (see also the note for the bus overview page).
- One group for each RT and the broadcast RT if any special events exist. Different special events have been allocated to the same subgroup when they are likely to be analyzed together in chronological order. There are the following subgroups:
 - **Response timeout:** All response timeouts categorized as transmit or receive timeout
 - **Status word: Message error:** All messages with a status word with a message error bit set

- **Status word: Busy:** All messages with a status word with a busy bit set
- **Status word: Subsystem flag:** All messages with a status word with a subsystem flag bit set
- **Status word: Terminal flag:** All messages with a status word with a terminal flag bit set
- **Mode code: Transmit last command:** All mode code messages of type transmit last command
- **Built in tests:** All mode code messages of type initiate self-test or transmit built in test word
- **Transmitter shutdown handling:** All mode code messages of type transmitter shutdown, override transmitter shutdown, selected transmitter shutdown or override selected transmitter shutdown
- **Mode code: Reset remote terminal:** All mode code messages of type reset remote terminal
- **Terminal flag manipulation:** All mode code messages of type inhibit terminal flag bit or override inhibit terminal flag bit
- **Dynamic bus control handling:** All messages with a status word with a dynamic bus control acceptance bit set and all mode code messages of type dynamic bus control
- **Service request / Transmit vector word:** All messages with a status word with a service request bit set and all mode code messages of type transmit vector word

4.9.1.8 UART Communication Summary

The UART Communication Summary is a communication summary to see a summary of UART message counts, message periods and errors occurred within one channel or subchannel.

You can open it in the Channel View when you right-click an UART channel or subchannel and select **Protocol Viewer > View UART Communication Summary**.



All messages on the same subchannel are considered to be of the same message type. Further differentiation would be up to the specific protocol running on the UART subchannel (can be ordered as customer extension).

Despite the regular columns this data shows up:

- Subchannel: The UART subchannel (1 ... 16348)
- Parity error: The number of parity errors detected by the recorder
- Min / max length: The minimum and maximum number of data bytes of all messages on the subchannel

Note that the UART packet entries not necessarily have a timestamp encoded. If not, they share the timestamp with the enclosing packet and therefore the calculated periods may be inaccurate.

The data is ordered by the subchannel number.

The entries for the subchannels allow you starting exporters or protocol viewers or see the properties of that subchannel with a right-click.

The **Special Events** page provides information on each parity error on each subchannel.

4.9.1.9 MPEG Transport Stream Summary

The MPEG Transport Stream Summary is a communication summary to see a summary of the MPEG video stream transport packet counts, packet periods and errors occurred within one channel. The summary is only available for MPEG transport streams and not program streams.

You can open it in the Channel View when you right-click a video channel (for format 2 not if program stream included) and select **Protocol Viewer > View MPEG Transport Stream Summary**.

Video (43) summary: Heim-GSS-1_00970000_09030600_09032700

PIDs		Special Events	Processing Info									
PID	Type	Count	Stream type	Has PCR	Program number	Priority flag	Transport error	Continuity counter error	Sync error	Payload unit starts	Avg period	Bit rate
0000	Program Association Table	226		No		0	0	0	0	226	93.791	0.016
0010	Program Map Table	226		No	1	0	0	0	0	226	93.791	0.016
0021	Video	62405	ITU-T Rec. H.262 ISO/IEC...	Yes	1	0	0	0	0	638	0.340	4.428
0031	Audio	2848	ISO/IEC 11172 Audio	No	1	0	0	0	0	178	7.443	0.202
1FFF	Null Packets	5079		No		0	0	0	0	0	4.173	0.360

Each PID (packet identifier) is considered a separate message type.

Despite the regular columns this data shows up:

- PID: The packet identifier in hexadecimal
- Type: A classification of the PID in:
 - Program association table
 - Program map table
 - Null packets
 - Audio - only if this PID is mapped inside of a PMT and is recognized as audio
 - Video - only if this PID is mapped inside of a PMT and is recognized as video

- Stream type: The stream type as listed in a PMT for the PID
- Has PCR: An indication if the PID is listed as program clock reference for a program
- Program number: The number of the associated program for all PIDs listed in a PMT or the PMT itself
- Priority flag: The number of times the priority flag was set inside the transport packets of this PID
- Transport error, continuity counter error, sync error: The number of errors detected in the transport stream. Sync error refers to the transport packet sync byte. Continuity counter errors mean that the continuity counter changed in an undefined way without a discontinuity flag on the same transport packet.
- Payload unit starts: The number of times a TS packet with a payload unit start indicator was found. This often matches the number of video frames on video PIDs but may be something different depending on the recorder.
- Bit rate [MBit/s]: The bit rate of transport stream packets of this PID in Megabits per second.

The minimum and maximum period columns normally present in communication summaries are suppressed here.

The data is ordered with ascending PIDs.

The **Special Events** page provides information on each special event in three groups for each PID: Transport, continuity counter and sync errors.

4.9.1.10 CAN Bus Communication Summary

The CAN Bus Communication Summary is a communication summary to see a summary of CAN bus message counts, message periods and errors occurred within one channel or subchannel.

You can open it in the Channel View when you right-click a CAN bus channel or subchannel and select **Protocol Viewer > View CAN Bus Communication Summary**.

The screenshot shows a window titled "CAN (8,1) summary:20110424153818_r0001f000". It contains a table with columns: Subchannel, Identifier, RTR, Count, Data error, Format error, Avg period, Min period, Max period, Bit rate, and Bus load. The first row (Subchannel 1) has values: 17896, 0, 0, 1.000, 0.089, 6.128, 79008, 7.901. Two rows with Identifier 000 and 7FF are shown below. A context menu is open over the table, listing options like "Export", "Protocol Viewers", "Export Table Content to CSV", "Copy Table Cell to Clipboard", "Use Monospace Font", "Expand All", "Collapse All", and "Properties". A sub-menu for "Export" is also visible, showing "CAN to CSV" and "Further Exporters".

Subchannel	Identifier	RTR	Count	Data error	Format error	Avg period	Min period	Max period	Bit rate	Bus load
1			17896	0	0	1.000	0.089	6.128	79008	7.901
	000							13	43893	4.389
	7FF							94	35115	3.512

All messages on the same subchannel with the same identifier and remote transfer request state are considered to be of the same message type.

Despite the regular columns this data shows up:

- Subchannel: The CAN bus subchannel number (1 ... 256)
- Identifier: The hexadecimal identifier
- RTR: **X** if the message type is a remote transfer request
- Data error, format error: The number of errors detected by the recorder
- Bit rate: Number of bits for this message type (or all messages on subchannel level) per second. The number of bits include all bit times from start of frame to the inter frame space.
- Bus load: Percent of the time the bus was occupied with the message type (or all messages on subchannel level). This is only available if the TMATS attributes indicate the bit rate of the bus.

The data has a fixed ordering with ascending order on these columns:

1. Subchannel
2. Identifier
3. RTR

The entries for the subchannels allow you starting exporters or protocol viewers or see the properties of that subchannel with a right-click.

The **Special Events** page provides information on each special event in two groups for each subchannel: Data and format errors.

4.9.1.11 PCM Communication Summary

The PCM Communication Summary is a communication summary to see a summary of PCM minor frame counts, minor frame durations and errors occurred within one channel.

You can open it in the Channel View when you right-click a PCM channel and select **Protocol Viewer > View PCM Communication Summary**.

All PCM minor frames are considered to be of the same message type. Therefore, the summary displays at most one line.

Despite the regular columns this data shows up:

- Expected duration: The expected duration of a single minor frame in milliseconds
- Time discontinuity: Number of times the time between two subsequent locked minor frames was not within 90 and 110 % of the expected duration. Normally this means that the frame synchronization was lost.
- Subframe ID discontinuity: Shows the number of times that the first subframe ID within the minor frame (if defined) had a gap in the sequence. This is only counted in a sequence of locked minor frames i.e., if there was no time discontinuity.
- Frame lock loss: The number of times the PCM frame synchronization was lost (see general note below).
- Avg / min / max duration: Shows the average, minimum and maximum duration of a minor frame. This is only calculated for locked minor frames that are directly followed by another locked minor frame so the duration is known. If the minor frame

is followed by a time discontinuity the duration is unknown and not taken into account for these values. If you want this taken into account look at the regular columns for avg / min / max period.

- Time w/o minor frame lock: Gives the percentage of the processing duration not covered by locked minor frames.

The **Special Events** page provides information on each special event in three groups: Time discontinuity, subframe ID discontinuity, frame lock losses (see note below). The events are counted under the same conditions as on the summary.

Note that the time range taken for the analysis is limited to the period from the time stamp of the first to the last locked or "checking" minor frame or frame lock loss (for PCM throughput mode). This especially means that longer periods without frame lock in the beginning or the end of the selected time range are not analyzed. You can see the actual processing duration on the processing info page of the view.

Note: Frame lock losses are not directly reported in the CH10 data. There are two cases:

- For PCM throughput mode the FLIDAS decoding provides a frame lock loss according to the frame sync criteria in the TMATS P-Group.
- For PCM packed or unpacked mode the CH10 hardware itself does the frame synchronization. It will provide minor frames in "check" and "lock" state. FLIDAS will not count the "check" state as minor frame in the statistics above. However, it is also not counted as sync loss. Still it will show up as time discontinuity as the sequence of valid (i.e., locked) minor frames has a gap.

4.9.1.12 Message Data Communication Summary

The Message Data Communication Summary is a communication summary to see a summary of message data message counts, message periods and errors occurred within one channel or subchannel.

You can open it in the Channel View when you right-click a message data channel or subchannel and select **Protocol Viewer > View Message Data Communication Summary**.

All messages on the same subchannel are considered to be of the same message type. Further differentiation would be up to the specific protocol running on the message data subchannel (can be ordered as customer extension).

Despite the regular columns this data shows up:

- Subchannel: The message data subchannel (1 ... 16348)
- Format error: The number of format errors detected by the recorder
- Data error: The number of data errors detected by the recorder
- Incomplete message error: The number of messages where not all segments were recorded
- Min / max length: The minimum and maximum number of data bytes of any message (including incomplete) on the subchannel
- Bit rate: The bits per second that were recorded on this subchannel

The data is ordered by the subchannel number.

The entries for the subchannels allow you starting exporters or protocol viewers or see the properties of that subchannel with a right-click.

The **Special Events** page provides information on each data, format or incomplete message error on each subchannel.

4.9.1.13 Time Communication Summary

The Time Communication Summary is a communication summary to see a summary of time packet counts, periods, RTC statistics and errors occurred within one channel.

You can open it in the Channel View when you right-click a time channel and select **Protocol Viewer > View Time Communication Summary**.

All time packets are considered to be of the same message type. Therefore, the summary displays at most one line.

Despite the regular columns this data shows up:

- Period failed: Shows the number of times that the time between two time packets was greater than one second. If this occurs at all the line is also displayed in red to indicate the error situation.
- Avg / min / max RTC increase: Shows the average, minimum and maximum increase of the 10 MHz real time counter (RTC) between two time packets. A perfectly stable RTC would show three times the same value.

The regular column **Avg period** is suppressed here to avoid misinterpretations.

The **Special Events** page provides information on the time packets where the period failed (i.e., was greater than one second).

4.9.1.14 Recording Event Communication Summary

The Recording Event Communication Summary is a communication summary to see a summary of recording event counts and periods occurred within the recording events channel. It also shows the complete sequence of recording events.

You can open it in the Channel View when you right-click a recording event channel and select **Protocol Viewer > View Recording Event Communication Summary**.

Despite the regular columns this data shows up:

- Event name: The event ID from TMATS
- Event description: The event description from TMATS

The data has a fixed ordering on the event number column.

As you are providing a time slice for the summary in the beginning or select the whole recording by default, recording events occurred before the recording start are excluded from the summary, even if they are stored in the data.

The additional **Recording Event Sequence** page shows the complete recording events in the selected time slice in sequence. There is a drop-down selection box where you can select to filter for a specific event number. The table on the bottom of the page shows this data:

- The timestamp
- The event number
- The event name (ID) from TMATS
- The event count
- The additional timestamp if it was defined in the packet
- The event description from TMATS

4.9.1.15 IENA Communication Summary

The IENA Communication Summary is a communication summary to see a summary of IENA packet counts, types, periods and errors occurred within one subchannel. IENA is originally an Airbus format but is now adopted also in other projects, for example in connection with Curtis-Wright / Acra FTI.

You can open it in the Channel View when you right-click an Ethernet or Message Data subchannel and select **Protocol Viewer > View IENA Communication Summary**. For Subchannels that are not properly TMATS declared you can start it the same way after discovering the subchannels in an Ethernet or Message Data communication summary.

For message data subchannels the assumption is that the recorded data is the IENA layer (i.e., UDP payload).

The IENA properties configuration wizard page allows you defining where to find the IENA packets in your Ethernet data.

All IENA packets with the same IENA key are considered to be of the same message type.

Despite the regular columns this data shows up:

- Aircraft: The 3rd octet of the source IP address in decimal. According to the Airbus specification this is identifying the aircraft type and number. In other applications it might be meaningless. For message data subchannels it is not recorded.
- Equipment: The 4th octet of the source IP address in decimal. According to the Airbus specification this is identifying the equipment type and location. In other applications it might be meaningless. For message data subchannels it is not recorded.
- Diffusion group: The 4th octet of the destination IP address in decimal. According to the Airbus specification this is the diffusion group identifier. In other applications it might be meaningless. For message data subchannels it is not recorded.
- Key: The hexadecimal IENA key
- Parameter type: Parameter type for this IENA key (combination of positional / message / delay field flags). Possible options are P, D, N, M, Q.

- Positional, Message key structure, Uses delay field: These three columns contain an X if the corresponding key status flags are set. In combination they make up the previous Parameter type column.
- Words per parameter: The number of 16-bit words that are used for each parameter in the IENA packet
- Discarded by CUB: This column contains an X if the corresponding key status flag is set.
- N4 transmission prohibited: This column contains an X if the corresponding key status flag is set. This is probably only used in Airbus applications.
- Minimum / Maximum IENA payload word length: Displays how many 16-bit words are contained in the IENA payload for this IENA key minimum and maximum. You can see here if all IENA packets of this key have a constant length.
- Lost IENA packets: The number of IENA packets of this key that have been lost based on the sequence counter.
- # Never been synchronized: The number of IENA packet of this key that have the N2 status flag set that indicates that the system has never been synchronized since the last reset. This is probably only relevant in Airbus applications.
- # Local synchronization loss: The number of IENA packet of this key that have the N2 status flag set that indicates that the system is not synchronized to an external source. This is probably only relevant in Airbus applications.
- # ETR flags: The number of IENA packet of this key that have the N2 status flag set that indicates a bad latency. This is probably only relevant in Airbus applications.
- Bit rate: Number of bits for this message type per second on IENA level (i.e., without Ethernet / IP / UDP).

Furthermore the columns for Min/Max period have been split to four columns to distinguish between the min/max periods based on the embedded IENA time stamps and the time stamps where CH10 data was recorded (first or only Ethernet or message data fragment). You can get an idea here how well the Ethernet network performs.

The data has a fixed ordering with ascending order on these IENA keys.

The **Processing Info** page adds all the configuration of the IENA properties configuration wizard page and the total bit rate in bits per second on IENA level for all summarized IENA packets.

The **Special Events** page provides information on each special event in three groups for each IENA key: IENA packets with ETR flags, lost messages (i.e., inconsistent sequence counter) and synchronization events. The synchronization events provide information on both N2 status synchronization flags in a way that not each flag is reported, but only the changes in the flags. If there is a synchronization error with the first IENA packet of the key it will be indicated as well.

4.9.1.16 iNET-X Communication Summary

The iNET-X Communication Summary is a communication summary to see a summary of iNET-X packet counts, types, periods and errors occurred within one Ethernet subchannel.

You can open it in the Channel View when you right-click an Ethernet subchannel and select **Protocol Viewer > View iNET-X Communication Summary**. For Subchannels that are not

properly TMATS declared you can start it the same way after discovering the subchannels in an Ethernet communication summary.

The iNET-X properties configuration wizard page allows defining where to find the iNET-X packets in your Ethernet data.

All iNET-X packets with the same stream ID, source / destination IP and destination port are considered to be of the same message type. Source ports will not be used to distinguish and will not be visible.

Despite the regular columns this data shows up:

- Source IP: The source IP of the iNET-X packet.
- Destination IP: The destination IP of the iNET-X packet.
- Destination UDP port: The destination UDP port of the iNET-X packet.
- Stream ID: Hexadecimal or decimal stream ID of the iNET-X packet depending on your configuration on the iNET-X properties wizard page.
- Average / Minimum / Maximum packet length: The average / minimum / maximum length of all iNET-X packets (whole iNET-X layer).
- Lost iNET-X packets: The number of iNET-X packets of this stream ID that have been lost based on the sequence counter.
- Lost FIFOs: The sum of the LostCount fields in the iNET-X protocol (number of FIFOs dropped without being sent due to too low transmission rate)
- # Error bits: The number of iNET-X packets that had the error bit set.
- # Timeout bits: The number of iNET-X packets that had the timeout bit set (No error. Transmission of packet was due to a timeout rather than a filled FIFO).
- Bit rate: Number of bits for this message type per second on iNET-X level (i.e., without Ethernet / IP / UDP).
- Average / Min / Max CH10-PTP time difference:
 These indicators should enable you to get a feeling on the network delays. The idea is that the iNET-X timestamp (PTP) is from the oldest data in the iNET-X packet while the CH10 timestamp is the timestamp of the recording of the Ethernet packet. The difference is the time span of the data in the iNET-X packet and the network delay. You can usually see the time span of the data if you look at the periods of the iNET-X data. A 400ms period packet could contain a time span of 400 ms of data (take care. If you have the data of a single MIL-1553 message transmitted every 400ms the period will be 400ms but the time span only the transmission time of the message). Another source of differences are leap seconds. If your CH10 time source is UTC based there will be an offset of 37 leap seconds (by 2022) between CH10 and iNET-X time. FLIDAS will assume that the time synchronization between CH10 and iNET-X will be pretty precise and only be the source or leap seconds. Therefore, FLIDAS will remove any time difference in the area of seconds from the indicator and make sure that the time difference is between -500 and +500 ms. FLIDAS then displays the average, minimum and maximum value of all these differences. If you subtract the duration of the packet data from this value you should have a network delay indicator.
Note that Ethernet frames may also be time stamped on the last bit. If you look at your CH10 Ethernet packet properties FLIDAS will tell you where the time stamp is.
Note that you could use this information also to define a value for the message reordering time frame in the iNET-X properties configuration wizard page

Furthermore the columns for Min/Max period have been split to four columns to distinguish between the min/max periods based on the embedded iNET-X time stamps (PTP) and the time stamps where CH10 data was recorded (Ethernet frame).

The data is initially ordered with ascending stream IDs. By clicking at the headers of Source / Destination IP, Destination UDP port and Stream ID you can select another ascending order column.

The **Processing Info** page adds all the configuration of the iNET-X properties configuration wizard page and the following information:

- Total bit rate in bits per second on iNET-X level for all summarized iNET-X packets
- Highest / lowest iNET-X timestamp, to see the time range in iNET-X PTP time
- The number of UDP packets that have been filtered out besides the IP / port filtering. This happens if FLIDAS tries to detect if a UDP packet on the network is really iNET-X and rejects it. This can happen for UDP packets that exceed the maximum iNET-X packet length, if the iNET-X control field does not indicate version 1 and option count 1 or if the reserved bytes of the payload information field are not 0.

The additional **Sources** page shows a table with some additional information for each source (summarized by source IP address):

- Source IP
- The count of iNET-X packets of this source
- Lost iNET-X packets: The number of iNET-X packets that have been lost based on the sequence counter.
- Lost FIFOs: The sum of the LostCount fields in the iNET-X protocol (number of FIFOs dropped without being sent due to too low transmission rate)
- # Error bits: The number of iNET-X packets that had the error bit set.
- # Timeout bits: The number of iNET-X packets that had the timeout bit set (No error. Transmission of packet was due to a timeout rather than a filled FIFO).
- Bit rate: Number of bits transmitted by the source per second on iNET-X level (i.e., without Ethernet / IP / UDP).
- First iNET-X/CH10 time: The timestamp of the first processed iNET-X packet of this source in iNET-X PTP time and CH10 time.
- First timestamps delta: The difference between iNET-X and CH10 time (iNET-X minus CH10). This can be used to check if the source had a proper time initialization. Note that iNET-X time (PTP based on TAI time) and CH10 time may differ in leap seconds, for example 37 seconds in 2022 if CH10 time is UTC. Also, CH10 time is related to the recorded Ethernet packet while iNET-X time is related to the first data in the packet.

The **Special Events** page provides information on each special event in three groups for each iNET-X stream ID: iNET-X packets with error bits, lost iNET-X packets (i.e., inconsistent sequence counter) and lost FIFOs.

4.9.1.17 CH7 Communication Summary

The CH7 Communication Summary is a communication summary to see statistics of an IRIG 106 chapter 7 PCM stream. It summarizes content types, contained CH10 channels, errors, bit rates etc. It works on the source packet level of CH7.

You can open it in the Channel View when you right-click a PCM channel and select **Protocol Viewer > View CH7 Communication Summary**. You have to know yourself if this PCM channel contains CH7 data.

This communication summary breaks down different types of source packets to three hierarchical levels:

1. by the stream ID of the CH7 transport packets
2. then by the content type
3. then for Chapter 11 (formerly chapter 10) content type by the channel ID

In the sense of the communication summary description the "messages page" will be called **CH7 Streams** and the lowest hierarchy level of the above break down will be a "message type". Since the CH7 Streams page is quite different from the regular messages page it will be described here separately.

For all timing calculations source packets will have the time stamp of the PCM minor frame they are started in.

The CH7 Streams page has the following columns but the content might be different depending on the hierarchy level:

Column title	Hierarchy level		
	Stream	Content type	CH10 channel
Stream ID / content type	stream ID taken from the transport packet	content type	channel ID and data type format. The data type format is taken from the first valid source packet of the channel and is considered the same within the channel
Version	CH7 protocol version taken from the first transport packet of this stream. Is considered the same within the stream.	-	-
# TP	number of transport packets (one for each frame locked PCM minor frame belonging to the stream ID if it did not look invalid)	-	-
# total SP	total number of valid source packets		
# LL SP	number of valid source packets build from low-latency encapsulation packets		
# standard SP	number of valid source packets build from non low-latency encapsulation packets		

Bit rate	bit rate of valid source packet's payload in MBit/s		
Avg SP payload size	average size of valid source packet's payload in bytes		
# CH7 out of sync	number of all CH7 protocol related errors that caused the processing to stop and restart at a safe position. Usually caused by bit errors or PCM minor frame lock loss. Transport packets that look invalid increment this number on every stream ID that was detected up to the time of encountering the bad transport packet.	-	-
# Dropped EPs	number of encapsulation packets started but not ending up in any valid source packet. Excludes up to one incomplete source packet in front or at end of decoding (non low-latency). EPs are usually dropped with aborted source packets (see below)	-	-
# Aborted SPs	number of source packets started but not ending up in a valid one (e.g., mixed content types or wrong fragment order). Excludes up to one incomplete source packet in front or at end of decoding (non low-latency)	-	-
Avg period	average period of valid source packets in milliseconds		
Min / max periods	-	-	Minimum and maximum time between two valid source packets in milliseconds.

The CH7 stream line is colored in red in case of detected errors on that stream.

The **Special Events** page provides information on two types of special event:

- PCM minor frame lock loss
- When a source packet is aborted (see column description on CH7 Streams page). The events are organized by stream ID.

The events are counted under the same conditions as on the summary.

4.9.2 NMEA 0183 Protocol Viewer

The NMEA 0183 Protocol Viewer is a Protocol Viewer to see the NMEA 0183 sentences transmitted on an UART subchannel, TSPI format 0 channel or on UDP packets.

You can open it in the Channel View when you right-click a UART subchannel (not channel) or TSPI format 0 channel and select **Protocol Viewer > View NMEA 0183 Sentences**. In a similar way you can open it from the UART Communication Summary. For UDP packets it is opened from the Ethernet Communication Summary. For UART and UDP there is no indication in the CH10 file if it really contains NMEA 0183 sentences. The NMEA 0183 Protocol Viewer will attempt the interpretation but you should make sure that the data really has this format. For TSPI format 0 it will be filtered for NMEA 0183 and NMEA 0183 HS data.

Time	Errors	Checksum	#Parameters	Data type	Param 1	Param 2	Param 3	Param 4	Param 5	Param 6
5/3/13 11:04:23 AM 577:400:000		6	17	GPGLA	M	3	26	09	08	05
5/3/13 11:04:23 AM 577:400:000		50	14	GPGLA	110423.00	5057.4893178	N	00711.3172866	E	1
5/3/13 11:04:24 AM 047:596:095		74	12	GNRMC	110424.00	A	5057.4892746	N	00711.3173277	E
5/3/13 11:04:24 AM 047:596:095		7A	19	GPGLV	3	1	10	26	66	297
5/3/13 11:04:24 AM 314:271:668		77	19	GPGLV	3	2	10	28	47	134
5/3/13 11:04:24 AM 314:271:668		7C	11	GPGLV	3	3	10	21	10	323
5/3/13 11:04:24 AM 580:947:341		6	17	GPGLA	M	3	26	09	08	05
5/3/13 11:04:24 AM 580:947:341		59	14	GPGLA	110424.00	5057.4892746	N	00711.3173277	E	1
5/3/13 11:04:25 AM 045:536:909		70	12	GNRMC	110425.00	A	5057.4892439	N	00711.3173663	E
5/3/13 11:04:25 AM 045:536:909		78	19	GPGLV	3	1	10	26	67	297

Opening might take some time, since the data from the whole CH10 file is loaded. A progress indication will show up and you can cancel the loading to see just the data that has been loaded so far.


The data that shows up is:

- A timestamp taken from the packet entry that contained the sentence. Since a packet entry may contain several NMEA 0183 sentence fragments and may contain transmission pauses (UART only), the timestamp is not accurate. However, the sentence was transmitted between this timestamp and the timestamp of the next packet entry. Note that timestamps on UART and TSPI packet entries are optional. If the hardware does not record this, all packet entries of one packet share the same timestamp.
- An indication of two error situations: A wrong checksum and a wrong prefix (sentences are expected to begin with "\$" or "!"). In both cases the line will also have a red background.
- The checksum included in the sentence in hex
- The number of parameters (excluding data type and checksum)
- The data type, i.e., the type of the sentence
- One column for each parameter

Note: Checksums are optional on NMEA 0183 but if a sentence of a specific type with a checksum is seen on the NMEA 0183 source, all following sentences of that type are supposed to have one or are considered to have a checksum error. This makes sure incomplete sentences are invalid.

To see how to export all the visible data of the table to a CSV text file read the section about table exporting.

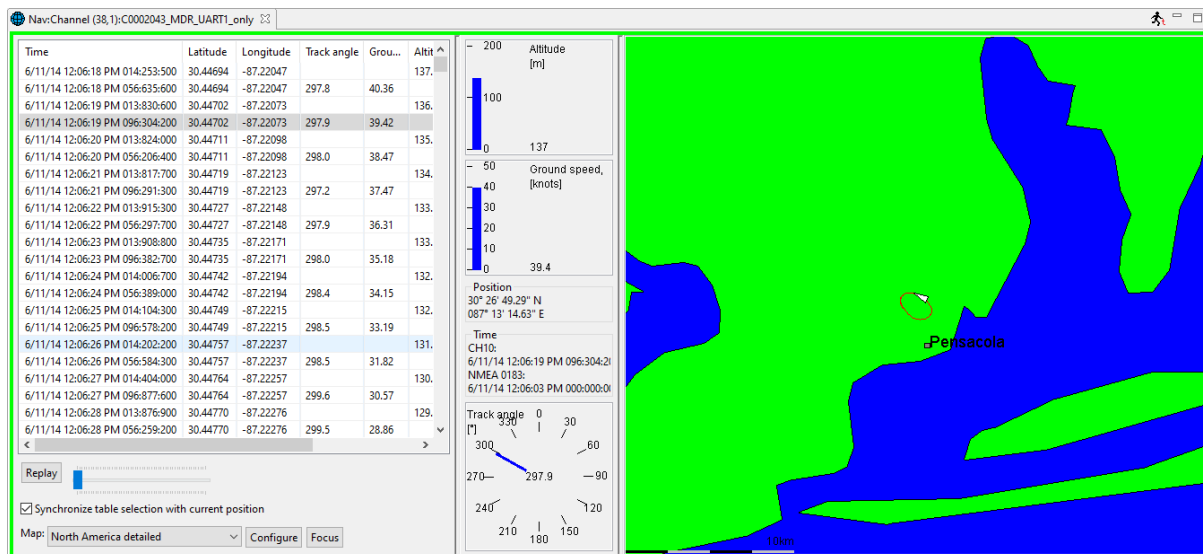
To create a time marker for a NMEA 0183 sentence, right-click it and select **Create Time Marker**.

You can jump to the NMEA 0183 sentence at a time marker or recording event. To do so use the  icon on the right top of the view. The table will locate the NMEA 0183 sentence right before or at the time (or after if the time is before first NMEA 0183 sentence).

4.9.3 Navigation Data View

The Navigation Data View is a protocol viewer to analyze and replay navigation data from NMEA 0183 sentences from a UART subchannel, TSPI format 0 channel or UDP packets. It is only available in the Parameter or Parameter Streaming Version

You can open it in the Channel View when you right-click a UART subchannel (not channel) or TSPI format 0 channel and select **Protocol Viewer > View Navigation Data from NMEA 0183 Sentences**. Similar options are found in the UART Communication Summary or for UDP packets in the Ethernet Communication Summary. For UART and UDP there is no indication in the CH10 file if it really contains NMEA 0183 sentences. The Navigation Data View will attempt the interpretation but you should make sure that the data really has this format. For TSPI format 0 it will be filtered for NMEA 0183 and NMEA 0183 HS data.



Opening might take some time, since the data from the whole CH10 file is loaded. A progress indication will show up and you can cancel the loading to see just the data that has been loaded so far.

The screen will consist of four areas:

- A map to the right
- An instrument panel in the middle
- A sequence of navigation data to the upper left
- A control panel to the lower left

You can resize the three columns by dragging the vertical separator line between them.

The Navigation Data View allows several things:

- The engineering values extracted from the NMEA 0183 sentences can be inspected over the time sequence
- The flight path defined by the navigation data can be seen on the map
- Specific points in time can be selected to visualize the navigation data at this point in time on the map and in the instrument panel
- The time sequence of navigation data can be replayed, i.e., you can follow the aircraft on the map and in the instrument panels

4.9.3.1 The Sequence of Navigation Data

This area contains a table with one line for each interpreted NMEA 0183 sentence. The following sentences are currently supported and used to extract the listed data:

- RMC
 - Timestamp
 - Latitude / Longitude
 - Speed over ground
 - Track over ground
- GGA
 - Latitude / Longitude
 - Altitude above mean sea level

Some sentences are not included:

- Sentences with checksum error. Checksums are optional on NMEA 0183 but if a sentence of a specific type with a checksum is seen on the NMEA 0183 source, all following sentences of that type are supposed to have one or are considered to have a checksum error. This makes sure incomplete sentences are invalid.
- Sentences with invalid data (RMC without active state or GGA with 0 satellite fix)
- Sentences with serious parsing errors (like required columns are not present)


If only single parameters cannot be parsed (like a latitude in wrong format) the sentence is included but marked red and parameters show up indicating "NaN" ("not a number" - computer coding for invalid).

The line contains one column for each extracted parameter and a first column with the timestamp of the packet entry that contained this NMEA 0183 sentence. Since a packet entry may contain several NMEA 0183 sentence fragments and may contain transmission pauses (UART only), the timestamp is not accurate. However, the sentence was transmitted between this timestamp and the timestamp of the next packet entry. Note that timestamps on UART or TSPI packet entries are optional. If the hardware does not record this, all packet entries of one packet share the same timestamp.

You can select the current position by double-clicking a line or right-clicking a line and selecting **Navigate to Selection**. This will update the map and instrument panel to reflect this position. Parameters that are not contained in the selected sentence are taken from the previous sentences. However, contained invalid values will be used to indicate this state.

To see how to export all the visible data of the table to a CSV text file read the section about [table exporting](#).

To create a time marker for a NMEA 0183 sentence, right-click it and select **Create Time Marker**.

You can jump to the NMEA 0183 sentence at a time marker or recording event. To do so use the  icon on the right top of the view. The table will locate the NMEA 0183 sentence right before or at the time (or after if the time is before first NMEA 0183 sentence). This will not set the location on the map. To do so you need to navigate to the selection as described above.

4.9.3.2 The Map

The map shows the flight path described by the navigation data with a red line.

The map shows the current selected position as a white symbol with black outline. The center of the symbol is the current selected position. If the navigation data contains a valid track angle, the symbol will be a triangle pointing in that direction. Otherwise it is a square. If no current position is selected the symbol disappears.

The map can be shifted by dragging it with the mouse. You can zoom in and out with the mouse wheel or the + and - key on the keyboard. Note that the map must have the focus for this, i.e., you need to click inside the map before.

To center on the current selected position you can use the **Focus** button on the control panel.

The default map covers the whole world in a coarse scale. You can switch to different maps and configure their content (see the control panel description). You can also add own maps.

The lower part of the map contains a scale. The scale can be toggled in the map configuration.

4.9.3.3 Instrument Panel

The instrument panel visualizes the current selected position as automatically set by the replay or explicitly set by a user within the sequence of navigation data.

4.9.3.4 Control Panel

The control panel offers configuration for the map and replay control.

The **Replay** button opens a replay dialog. You can select the speed of the replay (1 = real time, bigger means faster, lower slower). You can also configure to start replay from the beginning or the currently selected position.

The replay will then update the current position (and therefore map and instrument panel) using the time information from the packet entries that contained the NMEA 0183 sentences. The system will try to make at least one update of the timestamp to indicate progress every second (real time). It will also update once for each NMEA 0183 sentence. Note that this is not a hard real-time system and without a real-time operating system no timing guarantees can be given. However, we found no performance issues here.

You can stop the replay by clicking at **Stop**.

A slider next to the replay button visualizes the selected position. You can move it to another position as a different means to select a current position.

The option **Synchronize table selection with current position** controls whether a change in the current position (by replay or slider) will also highlight and show the line in the table.

A drop-down list named **Map** allows choosing a different map set. See the section below for details about the map.

The **Configure** button changes some display options of the map. Three options allow disabling rivers, cities or provinces on the map. This can be used to tweak the performance. A fourth option allows changing the scale on the lower part of the map to use kilometers, nautical miles or miles or disable it completely.

The **Focus** button moves the current selected position to the center of the map.

4.9.4 PCM BERT Viewer

The PCM BERT Viewer is a protocol viewer to verify a pseudo random bit stream on a PCM throughput channel for a bit error rate test (BERT). This is the type of test data often created by bit error test equipment. It is only available in the commercial versions.

You can open it in the Channel View when you right-click a PCM throughput channel and select **Protocol Viewer > View PCM BERT**.

In the wizard you can select the **Pseudo random bit stream type**.

The result includes this information:

- **Processed duration:** The period of time that has been processed. This includes the initial bits when trying to acquire a pattern synchronization.
- **Bits processed:** The number of bits during the processed duration
- **Bit rate:** The average bit rate in the processed duration in bits per second
- **Total bit errors:** The number of bit errors according to the pattern synchronization described below.
- **Bit error probability:** The probability for a single bit in the processed duration to be in error. This is calculated by the number of bit errors divided by the number of bits that have been checked
- **Bit errors per second:** The number of bit errors that have been found in average in a second. Note that bit errors have not been checked in the whole duration of the process due to the bits required to acquire a pattern synchronization.
- **Errored seconds:** The number of one second periods from the beginning that had at least one bit error. The last one second period might just be a fraction of a second.
- **Errored second ratio:** The ratio of errored seconds to the processed seconds (limited to 1 max)
- **Sync losses:** How often has the pattern synchronization been lost during the process
- **Bits unchecked:** How many bits have not been checked during the processed duration due to the synchronization process

- **Unchecked bits ratio:** Ratio of the bits unchecked to the bits processed. If this is high, all the values are not meaningful. The value has a red background if this is higher than 0.01

The checking process will always process full 16-bit words. Bit errors are only counted when the process is synchronized to the pattern. The bits processed while not synchronized are counted as "bits unchecked". A synchronization is found if a 16-bit word following another 16-bit word conforms to the expected pattern compared to the first word (i.e., there are no bit errors). A pattern synchronization is lost if at least four 16-bit words in a row had at least 3 bit errors each. All the bit errors until this loss of synchronization are counted. An immediate loss of pattern synchronization with an immediate new synchronization is declared if the current word has bit errors according to the current pattern synchronization but conforms with the pattern derived from the last word (that had bit errors with the current pattern sync). In this case the new pattern synchronization is picked up and no bit errors are counted for the current word (but for the last word so a bit error is guaranteed).

If the processing encounters any invalid PCM packets (e.g., lost CH10 packets) in the time slice it will stop processing, display a message and the results up to this time.

4.9.5 UART Text Protocol Viewer

The UART Text Protocol Viewer is a protocol viewer to interpret the data of a single UART subchannel as text.

You can open it in the Channel View when you right-click a UART subchannel (not Channel) and select **Protocol Viewer > View UART as Text**. In a similar way you can open it from the UART Communication Summary. There is no indication if the data is really text. The UART Text Protocol Viewer will attempt the interpretation but you should make sure that the data really is text.

You can configure the **Character encoding** that should be used to decode the data. The available character encodings depend on your local computer but US-ASCII (7 bit) will always be available and interpret 8-bit wise (i.e., if the MSB is set it will show an unknown character symbol).

FLIDAS will convert any kind of recognized line break to one that will be visible in the display (this especially means that Windows (CR LF), Unix (LF) or Macintosh (CR) style line breaks should be visible on all systems).

The displayed text will also include text taken from packet entries with parity errors. When interpreting this data as text it is usually still human readable if there are single parity errors.

Opening might take some time, since the data from the whole CH10 file is loaded. A progress indication will show up and you can cancel the loading to see just the data that has been loaded so far.

The displayed text allows searching inside by entering your keyword in the **Search term** field on top and clicking **Next** or **Previous** for forward or reverse search from the cursor position. The search will look for the exact, case sensitive search term entered.

The keyboard shortcut **Ctrl+F** will position the cursor in the search term field and hitting return there will automatically search for the next match.

Note that IRIG 106 chapter 10 only defines a storage format for 8-bit UART. All other (if stored) is undefined.

4.9.6 Recording Gap Viewer

The Recording Gap Viewer is a protocol viewer to detect and visualize recording gaps on one or many CH10 channels. (Formally it is not a protocol viewer as it looks into CH10 rather than any recorded protocol. But it uses the same technical infrastructure.) It can also export the recording gaps to a CSV file. It is only available in the commercial versions.

You can open it in the Channel View when you right-click one or many CH10 channels and select **Protocol Viewer > View Recording Gaps**. Disabled, TMATS and recording index channels will be excluded if you select them anyway.

In the wizard you can select the number of seconds (at least one) you consider a recording gap. Note that CH10 does not forbid time ranges without data on channels, except for time channels where packets must exist at least once a second (unless the first time packet indicates the absence of a time source).

The **Recording Gaps** page shows a table of all channels over time indicating the detected gaps in red and activity on the channel in green.

You can show the Packet View of a channel around the gap time if you right-click the table cell and select **Show in Packet View**. This also works for the time ranges with activity. If you select **Show in Packet View** on the first column with the time stamps, it shows you the Packet View with packets of all channels. Remember that this will then show the order of packets in the CH10 file which is not strictly ordered between channels (one channel may be up to a second further in time than another channel).

You can create a Time Marker for a table cell if you right-click it and select **Create Time Marker**.

The **Processing Info** page has the same content as for Communication Summaries and also shows your configuration for the minimum gap length.

The exact rules of how gaps are determined and what the time stamps mean are given in the end. Normally you don't need to fully understand them if you just want to find a section of at least one second of a channel without data. For general understanding here is an explanation of the below Recording Gap Viewer with a configured maximum gap time of 2 seconds.

Time	Mil1553Channel1 (32)	Mil1553Channel2 (33)
9/7/23, 1:00:00 AM	1:00:00 AM 000:000:100	
9/7/23, 1:00:02 AM		1:00:02 AM 000:000:000 (3)
9/7/23, 1:00:03 AM		1:00:03 AM 000:000:000
9/7/23, 1:00:05 AM	1:00:05 AM 000:000:000 (13)	1:00:05 AM 000:000:000 (13)
9/7/23, 1:00:16 AM	1:00:16 AM 000:000:000	1:00:16 AM 000:000:000
9/7/23, 1:00:18 AM	1:00:18 AM 000:000:000 (2.1)	

- There are two Mil 1553 channels displayed in the second and third column. The column headers show their name and channel ID.
- The first column shows full second time stamps of changes. Note that it does not show every second but only the times where something happened.
- 9/7/23, 1:00:00 AM: Mil1553Channel1 first started producing data. Around the same time there was no information on Mil1553Channel2. This means there was no activity yet but it did not time out.
- 9/7/23, 1:00:02 AM: While Mil1553Channel1 is still active, Mil1553Channel2 timed out after the configured 2 seconds. It indicates the exact time with the 2 seconds after the recording started. It reports the gap lasted 3 seconds. These are the two seconds already expired and the following second until activity starts on this channel. The time column is yellow as now some channels are on, some are off.
- 9/7/23, 1:00:03 AM: While Mil1553Channel1 is still active, Mil1553Channel2 sees first data after the gap of 3 seconds. Maybe this channel just started recording for the first time. All channels are on, so the time column is green.
- 9/7/23, 1:00:05 AM: Two seconds later both channels time out. The last data on each channel was 2 seconds before the time in each channel's cell. The total gap on both channels lasts 13 seconds and probably the whole recording was paused. The time column in red indicates that all channels are off.
- 9/7/23, 1:00:16 AM: Eleven seconds later both channels become active again at the same time
- 9/7/23, 1:00:18 AM: Mil1553Channel1 times out after two seconds of inactivity. The inactivity lasts for 2.1 seconds. This is the time where the CH10 file ends. Mil1553Channel2 keeps active until the very end. In fact its last recorded packet is the last packet in the CH10 file and defines the recording end time.

On the **Export** page you can export the collected data to a CSV file. This follows the same rules as a normal [CSV Exporter](#) except that the data to export is already collected and configured by the Recording Gap Viewer. See here for the [selection of the target file](#).

Besides the [optional time columns](#) the export has these columns:

Column title in CSV	Description
Channel#	The number of the CH10 channel
Channel name	The name of the CH10 channel
State	Either On or Off
Duration [ns]	The duration of the gap in nanoseconds in case the state was On

4.9.6.1 Gap Detection Rules

Here now the more detailed description of the gap detection for the once interested.

Recording gaps will be detected by the time difference between two CH10 packets on that channel. The content of the packets will not be decoded. So if there is a time difference of 2 seconds between two packets on the same channel and you configure 2 seconds as something you consider a gap it will count as gap even though the first packet may have content of up to 100 ms (so there may have been only 1.9 seconds without data). So do not expect the results to be absolutely precise. It can guide you to find times where you disabled recording on one or many channels, disconnected a cable etc.

Packets with header checksum error or wrong packet sync are considered not existing for the gap detection (i.e. no activity).

The first column shows the time for which the gap has been reported leaving away the sub-second precision.

A recording gap is reported for the time when the configured maximum gap length has expired from the previous packet's time stamp on that channel. It is also reported when this time expired since recording start without any activity on that channel. The cell in the table will then contain the exact time, while the time column in front contains the rounded down time (unless you just look at a time slice where the time stamps are adapted to be in range if necessary).

A cell indicating a gap will also give you the length of the gap in seconds in parentheses after the time.

Whenever a channel continues operation, the exact time is noted in the cell of the table and the rounded down time is reported in the first column.

The time column will be green if all channels are online, red if all channels are offline and yellow if some are offline and some are online. It can also have no color if some channels do not have a state yet (in beginning).

Time stamps may appear twice in the time column. This happens when a channel goes to the offline state and becomes online again within the same second.

Note that the Recording Gap Viewer does not look for gaps on subchannels, so only a complete gap on all subchannels of a channel is detected as a gap.

4.10 Parameter Viewers

See the [parameter viewers concept](#) for a general explanation of parameter viewers and how to start them. They are [configured in a similar way like Exporters](#) and allow [selecting a CH10 file](#), [a time slice](#) and probably more options.

The available parameter viewers are shown in the subsections of this chapter.

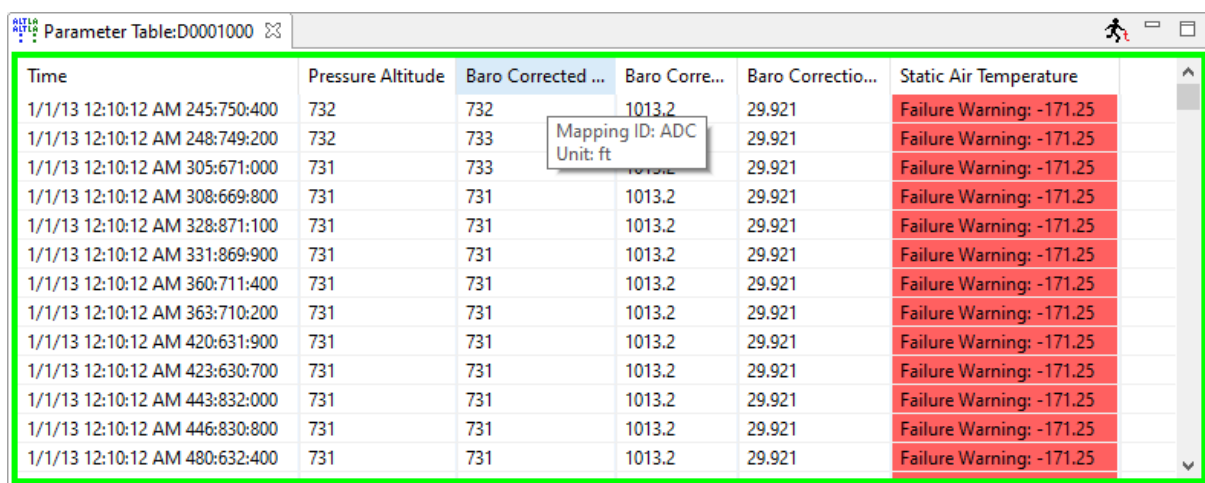
4.10.1.1 Performance

Displaying many measurements at once can degrade performance and exhaust memory. This can be caused by displaying many parameters at once or parameters over a long time or with high sampling frequency. If you encounter this problem, you can take less parameters at a time or use synchronous sampling with lower sampling rate instead of asynchronous. However, on regular PCs several million measurements should still lead to acceptable performance. In general, the [FAQ](#) offers a solution for memory problems.

You can [configure warning limits](#) to warn you before you potentially run in a problem.

4.10.2 Parameter Table Viewer

This parameter viewer will take samples of a parameter set over a period of time and display the values with the timestamp in a table.



Time	Pressure Altitude	Baro Corrected ...	Baro Corre...	Baro Correctio...	Static Air Temperature
1/1/13 12:10:12 AM 245:750:400	732	732	1013.2	29.921	Failure Warning: -171.25
1/1/13 12:10:12 AM 248:749:200	732	733	1013.2	29.921	Failure Warning: -171.25
1/1/13 12:10:12 AM 305:671:000	731	733	1013.2	29.921	Failure Warning: -171.25
1/1/13 12:10:12 AM 308:669:800	731	731	1013.2	29.921	Failure Warning: -171.25
1/1/13 12:10:12 AM 328:871:100	731	731	1013.2	29.921	Failure Warning: -171.25
1/1/13 12:10:12 AM 331:869:900	731	731	1013.2	29.921	Failure Warning: -171.25
1/1/13 12:10:12 AM 360:711:400	731	731	1013.2	29.921	Failure Warning: -171.25
1/1/13 12:10:12 AM 363:710:200	731	731	1013.2	29.921	Failure Warning: -171.25
1/1/13 12:10:12 AM 420:631:900	731	731	1013.2	29.921	Failure Warning: -171.25
1/1/13 12:10:12 AM 423:630:700	731	731	1013.2	29.921	Failure Warning: -171.25
1/1/13 12:10:12 AM 443:832:000	731	731	1013.2	29.921	Failure Warning: -171.25
1/1/13 12:10:12 AM 446:830:800	731	731	1013.2	29.921	Failure Warning: -171.25
1/1/13 12:10:12 AM 480:632:400	731	731	1013.2	29.921	Failure Warning: -171.25


The sampling period can be configured in milliseconds or you can sample asynchronously as both described in [parameter sampling](#).

All invalid or not measured values will be indicated additionally with a red background color.

The column headers will contain the parameter name. If you hover with the mouse over it to show the tool tip you will see the mapping ID, parameter path, engineering unit and comment.

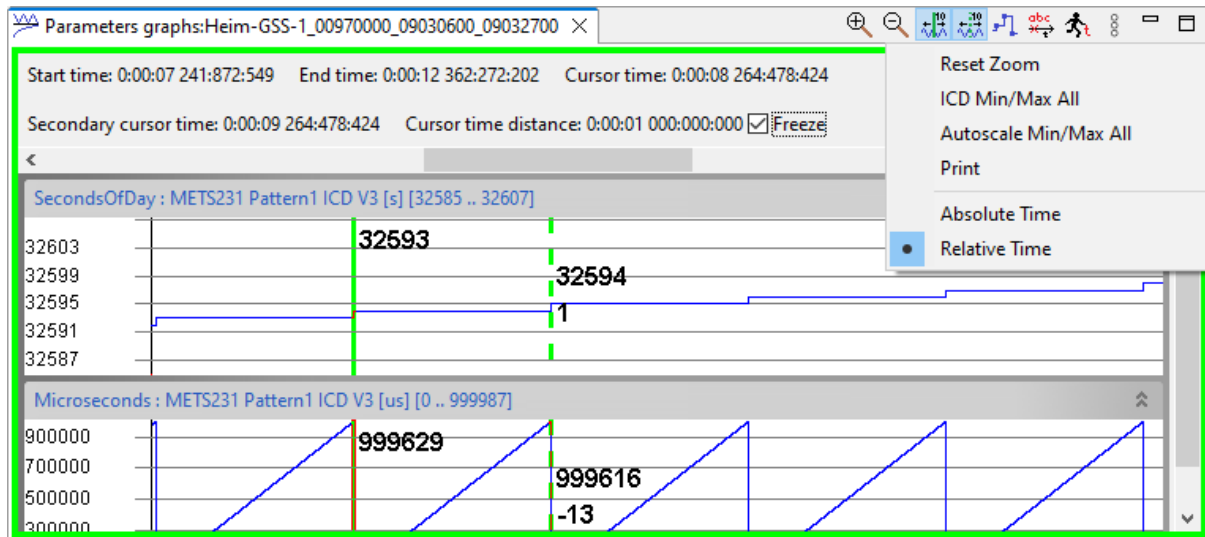
To see how to export all the visible data of the table to a CSV text file read the section about [table exporting](#).

To create a [time marker](#) for a sampling point, right-click the line and select **Create Time Marker**.

You can jump to the line of samples at a [time marker or recording event](#). To do so use the  icon on the right top of the view. The table will locate the line of samples right before or at the time (or after if the time is before first line of samples).

4.10.3 Parameter Graph Viewer

This parameter viewer will take samples of a parameter set with numerical or enumeration parameters over a period of time and display the values as separate graphs.



The sampling period can be configured in milliseconds or you can sample asynchronously as both described in [parameter sampling](#). The "separate timestamps full" mode is only useful if you want to step through every sample with the cursor as described below.

Enumeration parameters will only display their raw value in the graph, but the cursors will allow inspecting the enumeration value. Non-numerical and non-enumeration parameters will not be displayed at all.

You can use the **page up** and **page down** keys on the keyboard to go up and down through the graphs.

4.10.3.1 Stripes

Each graph will be displayed in its own stripe.

Values between two samples will not be interpolated. The graph will stay at the same level between two samples and then change instantly to the new value.



Invalid values will be indicated as red line on the bottom of each stripe.

The title bars above each stripe shows the parameter's


- Name
- Mapping ID
- Path
- Comment
- Engineering unit
- Minimum and maximum value within the selected time slice
- Configured minimum and maximum value for the value scale if different from the minimum / maximum above

Each single stripe can be collapsed or expanded with a left-click on its title bar. A right-click on any title bar opens a context menu to expand / collapse all stripes at once.

There are several ways to configure the minimum and maximum value on the value scale (i.e., zoom on the value axis):

- You can manually configure the minimum and maximum for each graph when you right-click the graph and select **User Min/Max**. A dialog pops up that shows the minimum and maximum value in the time range of the graph and the user configured limits. You can change them here. If you leave minimum or maximum empty it will use the minimum or maximum value in the time range of the graph.
- You can configure the minimum and maximum for each graph to use the minimum and maximum value within the selected time slice when you right-click the graph and select **Autoscale Min/Max**
- You can configure the minimum and maximum for each graph to use the minimum and maximum value defined in the ICD for the parameter when you right-click the graph and select **ICD Min/Max**. This might not be available for some parameters.
- You can configure all graphs to use autoscale min / max when you open the view's pull-down menu by clicking at the three dots icon  at the top right of the view and select **Autoscale Min/Max All**.
- You can configure all graphs to use ICD min / max when you open the view's pull-down menu by clicking at the three dots icon  at the top right of the view and select **ICD Min/Max All**.
- You can set the mode that is used when opening a new Parameter Graph View in the Parameter Plug-in Preferences.

4.10.3.2 Curve Marks


You can enable curve marks if you click at the  icon on the top right of the view. Curve marks will be displayed at the position of the actual samples of each curve. Click it again to disable it.

If you have a lot of samples (multiple million), curve marks will take a lot of performance. Therefore, they only show up if there are less than a specified limit of samples in the currently visible area of a stripe (i.e. considering zoom). So, if they do not show up, you can zoom in until you see them. Since your screen will rarely be wider than a few thousand pixels, displaying more than a thousand curve marks doesn't give a recognizable result anyway. The maximum number of curve marks per stripe to display can be configured in the Parameter Plug-in Preferences.

Curve marks will be displayed for the samples. Remember that first the sampling mode and then the Parameter Graph Viewer itself will remove repeated equal samples unless you use the sampling mode "Separate Time Stamps Full". This sampling mode will therefore also use the most memory and performance. Try to avoid it on large data sets unless you need to see the actual sample positions.

4.10.3.3 Time Information


The area on top shows the start and end time of the visible time slice. It also shows the time at the primary and secondary cursor position and the time between both.


To toggle between absolute and relative time open the view's pull-down menu by clicking at the three dots icon  at the top right of the view. A menu will appear where you can select absolute or relative time. Relative time will be related to the first measurement.

4.10.3.4 The Cursors

You can use two cursors to navigate within the stripes to get more information about a specific point in time. The upper number on each cursor will show the value at this time while the time will be displayed in the area above all stripes.

There are two cursors, a primary with a full line and a secondary with a dashed line. The secondary additionally shows the difference in value between both cursors' points in time with the lower number on the cursor.

To enable/disable the primary cursor use the  icon on the right top of the view.

Only if the primary cursor is enabled you can enable/disable the secondary cursor using the  icon on the right top of the view.

You can drag the cursors to move them to a new point in time. The cursors will show up at the same position within all stripes. The position will not lock to a timestamp of a measurement but can be freely moved through time.



To center the primary cursor disable it and enable it again. If you have lost the secondary cursor, you can also bring it back into view if you disable/enable it.


You can freeze the secondary cursor to a fixed distance to the primary cursor. To do so, check the check box labeled **freeze** behind the **Cursor time distance** on top. A dialog will pop up where you can enter the time distance you want to keep. It is initialized to the current distance. The format that you need to enter the time distance is the same as for the duration on the [Time Slice Selection Wizard Page](#). If you want to change the distance later, you must disable and re-enable the freeze check box. When you move either primary or secondary cursor in frozen mode, the other cursor will automatically adjust to keep the distance.

You can also move the selected primary cursor with the cursor keys on your keyboard. The selected one is the one of the graphs you last clicked at and shown in red. If you move it, it will go to the next or previous time stamped sample of its stripe. Of course, the cursor moves in the other stripes as well but the selected primary cursor is used to determine the position of the next sample. If you have enabled curve marks you will step from mark to mark but also if you don't display them the same rules for the samples apply (i.e. repeated samples are removed unless you use "Separate Time Stamps Full" sampling mode).

4.10.3.5 Zoom

You can zoom into and out of the data along the time axis using the mouse wheel when the mouse pointer is on one of the graphs. The zoom will be centered on the point where the mouse is pointing.


You can zoom into and out of the data along the time axis using the  and  icons on the right top of the view. This will zoom to the middle of the current time range.


To completely zoom out open the View's pull-down menu by clicking at the three dots icon  at the top right of the view and select **Reset Zoom**.

The zoom will always be to the same time range on all graphs.


In zoom mode you can move in time using the scroll bar below the time information.

4.10.3.6 Time Markers

To create a time marker for the time of the primary cursor use the  icon on the right top of the view.

To jump with the primary cursor to a time marker or recording event use the  icon on the right top of the view.

4.10.3.7 Printing

To print the currently expanded stripes open the view's pull-down menu by clicking at the three dots icon  at the top right of the view and select **Print**. You may print up to 4 stripes at a time.

Printing will use the current time slice, the selected relative / absolute time mode and the user configured minimum / maximum value for the value scales. In absolute time mode the date from the first printed measurement will be printed left of the time axis while the scale below the time axis only contains the time of the day without date. In relative time mode the date and time of the beginning of your parameter processing will be displayed left of the time axis. This is not necessarily the first printed measurement if you used zoom. The timestamps below the time axis display the time relative to the timestamp left of the time axis.

Printing will always use a single page and print a black and white curve. Every single measurement within the zoomed area will be send to the printer. This may create a large workload for the printer if your time slice contains high frequency samples which results in slow printing. Also, when FLIDAS reports that printing finished, the print job might still be processed by your operating system print service. Use operating system tools to see the status of the processing then.

A line of text above each curve will show the same information as the title bar above each curve on the screen. Two lines of text on the top of the page will show the CH10 source name and the printed time slice.

FLIDAS will normally use the smallest possible line for the curve to display even very small changes. However, if you are printing with a very high resolution you might need to increase the line width to still see it without a magnifier. There is a printing line width setting in the General Preferences. This will only apply to the curves and not to the axes.

4.10.3.8 Performance

An addition to the general warning on performance with parameter viewers: If you face performance problems but successfully displayed the graph, you can zoom in so less samples are displayed at a time.

4.11 ICD Formats

The subsections of this chapter define the formats used for ICDs. For most of them it is the user's responsibility to define them externally according to this format specification. You can also order specific support for your company internal existing formats.

All ICD support relies at least on the Parameter or Parameter Streaming Version.

4.11.1 CSV ICD Formats

Several ICD formats are defined by a CSV text file that can be edited with Excel for example. The common topics between these are described here.

Column delimiters and decimal separator (Locale) must conform to the configuration defined in the General Preferences.

The first line contains the version number of the format. It starts with V and is directly followed (no whitespace) by the version number of the file format as described in the individual ICD format chapter.

The second line is ignored and is supposed to contain the column headers from the column description table in the individual ICD format chapter.

Some column definitions are common between CSV ICD formats, so they are described here:

Column header	Description	Empty allowed?
Name	The name of the parameter. The combination of parameter path (see section below) and name must be unique within this ICD	No
Unit	The engineering unit of the parameter	Yes
Comment	Any comment you want to store for this parameter. The comments can for example be exported with the export files. A use for a comment could be to store only a short name as parameter name and store a more descriptive name as comment.	Yes
MSB pos (0 based)	The position of the most significant bit of the parameter. The counting starts with 0 for the rightmost and least significant bit within the data.	No
Length	The number of bits occupied by this parameter including a possible sign bit. The range of data will therefore be [MSB pos ... (MSB pos - length + 1)].	No
Encoding (BNR / UBNR / BCD / Float / Enum / ASCII / MIL1750AFloat / DECDFloat)	Defines how the data is encoded into the raw bits: <ul style="list-style-type: none"> BNR: Two's complement encoding. Note that the length column must include the sign bit. UBNR: Unsigned binary. There will be no sign bit 	No

	<ul style="list-style-type: none"> • BCD: Binary coded decimal • Float: Encoded as IEEE 754 single or double precision floating-point value. Must have a length of 32 or 64 bits. • MIL1750AFloat: Floating-point value encoded as MIL-STD 1750A floating-point number. Must have a length of 32 or 48 bits. The sign bit will be mapped to the MSB of the bit position you define and there will be no implicit byte swapping. • DECDFloat: Floating-point value encoded as DEC / VAX "D" floating-point number. Must have a length of 64 bits. The sign bit will be mapped to the MSB of the bit position you define and there will be no implicit byte swapping. • Enum: The raw value will use a mapping to enumeration values defined in the enumdef column. Some features of FLIDAS will use the raw number instead (for example, when displaying the values in a graph). <i>Note that the raw values are internally handled as double or single precision floating-point numbers in some places. This means that enumerations of more than 23 or 52 raw bits will not work properly on places requiring the raw number.</i> • ASCII: The 7 bits will be interpreted as a single ASCII character. 	
Min	Defines the minimum and maximum value for a parameter with numeric encoding (for enums it defines the min/max for the raw value).	
Max	<p>The minimum and maximum can be used when FLIDAS displays a graph of the parameter values over time for example to automatically define the displayed value range.</p> <p>If minimum, maximum or both remain undefined FLIDAS will try to calculate it. This works exactly for polynomials up to the 3rd degree. For higher polynomials approximation with the Laguerre method is calculated which might not result in the optimal result in rare cases. Manually provide the values then if you need them.</p>	Yes. For encodings without support for min / max no effect. For others min / max is calculated. You can also leave just one empty to use automatic calculation for this one.

	<p>The manual range can also be bigger or smaller than the mathematical values if needed.</p> <p>There will be no calculated minimum and maximum for floating point encodings if they are left empty. These are usually quite unlimited in the raw value range which usually leads to quite unlimited value ranges (or at least value ranges not suitable for covering in a graph).</p> <p>For enumerations the minimum and maximum depend on the number of bits, not on the used enumeration values. So, when there is no user defined min / max it will cover the whole possible raw value range to also display invalid enumeration values (i.e., a 3 bit enumeration with values of 1 = red and 3 = green will have a range of 0 ... 7 rather than 1 ... 3).</p> <p>If no min / max is present for a numeric value, features like a graph will provide other options like automatic scaling (min / max from observed values) or user definition during graph display.</p>	
Enumdef	<p>Defines a mapping from raw values to enumeration values. The definition consists of pairs of raw values (decimal or hexadecimal) and the enumeration value separated by "=". Hexadecimal raw values must be prefixed with "0x". The pairs themselves are separated by "#". The last pair must not be followed by "#". You don't need to fully specify the mapping. Other values will be considered invalid. Example: <i>0=Red#1=Amber#2=Green#0xFF=Yellow</i> defines the mapping:</p> <ul style="list-style-type: none"> • 0: Red • 1: Amber • 2: Green • 3, 4, ..., 254: Invalid • 255: Yellow • 256, 257, ...: Invalid 	Yes (if no Enum encoding)
Coefficient 0, Coefficient 1, ..., Coefficient n	<p>These are separate columns that define how to calculate the engineering value from the raw value. Each column defines a coefficient in an n-th degree polynomial curve to be applied to the raw value. So, for a conversion of $a + b*x + c*x^2$ you would set coefficient 0 to a, coefficient 1 to b and coefficient 2 to c. In the simplest form where you just want to scale the value you would set coefficient 0 to 0 (or leave it empty) and coefficient 1 to the scaling factor.</p>	Yes (defaults to 0). If no column contains any value it defaults to an unscaled raw value (Coefficient 1 = 1, rest 0)

CSV ICDs can define a hierarchy of parameters as described in the [ICD concept](#). Every line within a CSV ICD file that has only 3 columns will not be treated as a parameter definition but will open or close hierarchical groups of parameters.

Let's look at an example first. The following table displays the first three columns of a CSV ICD and a comment.

Hierarchy group name	Action	Target hierarchy level	Comment (this column does not appear in CSV!)
			Parameter definitions here will be directly below the ICD
Engine	>	1	Opens a new hierarchical group called "Engine" directly below the ICD
			Parameter definitions here will be inside "Engine" and have a path of "Engine".
Performance	>	2	Opens a new hierarchical group called "Performance" directly below the "Engine" group
			Parameter definitions here will be inside "Performance" and have a path of "Engine#Performance".
Limit exceedance	>	2	Opens a new hierarchical group called "Limit exceedance" directly below the "Engine" group. Note that the previous group "Performance" has not been formally closed but the target hierarchy level of 2 made it clear that the "Performance" group should be closed.
			Parameter definitions here will be inside "Limit exceedance" and have a path of "Engine#Limit exceedance".
	<	0	Formally closes all open groups until the target hierarchy level of 0 is reached. Since 0 is the ICD level, the groups "Limit exceedance" and "Engine" are closed.
			Parameter definitions here will be directly below the ICD

The above definition with some parameters could look like below. The hierarchy groups are displayed in **bold** and parameters in regular with their path in *italics*:

- Avionics bus MIL-STD 1553 ICD
 - NavigationParameter1 (*empty path*)
 - **Engine**
 - EngineParameter1 (*Engine*)
 - **Performance**
 - EnginePerformanceParameter1 (*Engine#Performance*)
 - **Limit Exceedance**
 - EngineLimitExceedanceParameter1 (*Engine#Limit Exceedance*)
 - AutoPilotParameter2 (*empty path*)

- ILSPParameter3 (*empty path*)

The columns in the CSV file for hierarchy group definition have the following meanings:

- The first column defines the name of the group. This is ignored and can be empty when closing groups.
- The second column can contain either the character ">" to open a new group or "<" to close a group.
- The third column defines the level within the existing hierarchy where 0 refers to the ICD itself while 1 is the first hierarchy level.

When opening a group you may only define a target hierarchy level up to one above your current level. The new group will be inserted at the specified position in your current path. If this means reducing the current path, then the existing groups are automatically closed.

When closing a group you may only define a target hierarchy level between 0 and one below your current level.

4.11.1.1 ARINC 429 Simple CSV ICD Format

This ICD format is the basic format for ARINC 429. The general comments on [CSV ICD formats](#) apply here too.

The version number for the file format described here is 4. Therefore, the first line must read "V4".

The following table describes the meaning of each column:

Column header	Description	Empty allowed?
Name	See CSV ICD formats	
Unit	See CSV ICD formats	
Comment	See CSV ICD formats	
Label [oct]	The octal label number of the ARINC 429 message	No
SDI	The decimal source destination identifier of the ARINC 429 message. If you leave this empty the parameter will contain the values of all possible SDIs. You can specify a specific decimal SDI to make the parameter limited to this SDI. You can also use a comma separated list of decimal SDIs to use the same parameter definition but create separate parameters for each of these SDIs. If the column contains any SDI definition, the parameter name that will be used will automatically be	Yes

	extended by a suffix indicating the binary SDI (like "parameter.01").	
Ignore SSM	<p>If you put anything in this column (like an "x") the decoding of the parameter will ignore the meaning of the SSM (sign-status matrix). This means invalid values will not be indicated as such and negative BCD values will be shown with their positive absolute value.</p> <p>If you leave the column empty, FLIDAS will automatically decode and apply the SSM according to the ARINC 429 specification. BNR and UBNR encodings will use the BNR SSM definition, BCD encodings the BCD SSM, enumeration and discrete encodings the discrete SSM and ASCII encodings the ISO #5 data SSM. As far as possible the displayed and exported values will convey invalid values together with the reason why they are invalid.</p>	Yes (will respect SSM)
MSB pos (31...0)	<p>See CSV ICD formats</p> <p>The position is defined within the 32-bit ARINC 429 word. The regular positions are:</p> <ul style="list-style-type: none"> • Label: 7 ... 0 • SDI: 9 ... 8 • Data: 28 ... 10 • SSM: 30 ... 29 • Parity: 31 <p>You can still define parameters outside of the data range. For BNR encoding the MSB will be the sign bit.</p>	
Length	See CSV ICD formats	
Encoding (BNR/UBNR/BCD/Enum/Disc/ASCII)	<p>See CSV ICD formats</p> <p>BCD encodings will use the sign defined in the SSM (sign-status-matrix) as defined in the ARINC 429 specification.</p> <p>The Disc encoding uses the same SSM as Enum but represents the raw numeric value. This can be used if you are exporting to numeric-only export files.</p>	
Min	See CSV ICD formats	
Max	Minimum and maximum can be defined for the Disc encoding as well.	

Enumdef	See CSV ICD formats	
Coefficient 0, Coefficient 1, ..., Coefficient n	See CSV ICD formats	Also ignored for Disc encoding

4.11.1.2 MIL-STD 1553 Simple CSV ICD Format

This ICD format is the basic format for MIL-STD 1553. The general comments on [CSV ICD formats](#) apply here too.

The ICD format allows decoding parameters from either MIL-STD 1553 RT subaddresses, associated data words from mode commands or RT status words.

The version number for the file format described here is 4. Therefore, the first line must read "V4".

The following table describes the meaning of each column:

Column header	Description	Empty allowed?
Name	See CSV ICD formats	
Unit	See CSV ICD formats	
Comment	See CSV ICD formats	
Type (Tx/Rx/SW/Mode)	Defines the place where to locate the data words: <ul style="list-style-type: none"> • Tx: The transmit subaddress of an RT • Rx: The receive subaddress of an RT • Mode: The data word associated with a mode command • SW: The status word of an RT 	No
RT	The decimal RT address	No
SA/Mode code	The decimal RT subaddress for type "Tx" or "Rx". The decimal mode code number for type "Mode".	Yes (for type "SW")
Error accepting	You can put an "x" here to also decode from faulty messages. In result messages that are only faulty because of a response timeout or a word count error (or a CH10 message error which has undefined meaning and is not the same as a MIL 1553 message error) will be used for decoding parameters. In these cases the message is incomplete but FLIDAS could distinguish between command, data and status words.	Yes (false by default)

	<p>An example is a BC - RT message where the RT does not confirm with a receive status word. This likely means the RT is offline. However, it could be interesting do look for the data that the BC sent to the RT. So, if you accept the no response error you can decode the data.</p> <p>If you decode a parameter from data words that are not present (e.g., because of a word count error) you will get an invalid value indicating a "message short" error.</p> <p>If you decode parameters from a status word from a message where the status word is required but missing (e.g., due to a no response error) you will get an invalid value indicating a "no status word" error.</p> <p>Note that you can define the same parameter twice if you want, one time with and one time without error accepting.</p>																
Endianness B/L	<p>The word order used for parameters that span over a data word boundary. B means big-endian, L little-endian.</p> <p>The following example shows a 16 bit signal with MSB pos 3 in word 1. The content is displayed as hex digits.</p> <table><tr><th>Word#</th><th>Bits 15 ... 12</th><th>Bits 11 ... 8</th><th>Bits 7 ... 4</th><th>Bits 3 ... 0</th></tr><tr><td>1</td><td></td><td></td><td></td><td>A</td></tr><tr><td>2</td><td>B</td><td>C</td><td>D</td><td></td></tr></table> <p>The interpretation would be: Big-endian: ABCD Little-endian: BCDA Big-endian concatenates the content from the table from top to bottom and left to right. Little-endian concatenates the content from the table from bottom to top and left to right.</p>	Word#	Bits 15 ... 12	Bits 11 ... 8	Bits 7 ... 4	Bits 3 ... 0	1				A	2	B	C	D		Yes (uses big-endian by default). Not used at all for type "SW" or "Mode".
Word#	Bits 15 ... 12	Bits 11 ... 8	Bits 7 ... 4	Bits 3 ... 0													
1				A													
2	B	C	D														
Word (1...32)	<p>The first position of a data word that contains a part of the parameter (for big-endian this is the word containing the most significant bit of the parameter for little-endian it is the word containing the least significant bit of the parameter). The first data word has the position 1.</p>	Yes (if type "SW" or "Mode")															

MSB pos (15...0)	See CSV ICD formats . The position of the most significant bit within the word defined by the word column. Must be between 15 and 0.
Length	See CSV ICD formats
Encoding (BNR / UBNR / BCD / Float / Enum / ASCII / MIL1750AFloat / DECDFloat)	See CSV ICD formats
Min	See CSV ICD formats
Max	See CSV ICD formats
Enumdef	See CSV ICD formats
Coefficient 0, Coefficient 1, ..., Coefficient n	See CSV ICD formats

If a MIL-STD 1553 message is correctly received by the CH10 recorder but the parameter reaches outside the valid data word range, the parameter will have an invalid value indicating a "message short" error.

4.11.1.3 PCM Simple CSV ICD Format

This ICD format is the basic format for PCM. The general comments on [CSV ICD formats](#) apply here too. If you need other features consider using the [PCM TMATS ICD](#).

The version number for the file format described here is 3. Therefore, the first line must read "V3".

The following table describes the meaning of each column:

Column header	Description	Empty allowed?
Name	See CSV ICD formats	
Unit	See CSV ICD formats	
Comment	See CSV ICD formats	
Subframe ID counter name	The name of the subframe ID counter that the Frame position and Frame interval refers to. The subframe ID counter must be defined in TMATS. Only required if any of these two attributes are used.	Yes (will use first subframe ID counter)
Frame position	The word position inside the subframe of the associated subframe ID counter for the first occurrence of this parameter. The first word position inside the subframe is 1.	Yes (will default to 1)

Frame interval	The interval in word positions inside the subframe of the associated subframe ID counter between consecutive occurrences of this parameter. 1 means the parameter exists in every position in the subframe (i.e., is not subcommutated). 0 means it is only defined in the single word position specified by Frame position .	Yes (will default to 1 i.e., no subcommutation. This differs from the default in TMATS where a default was introduced after this ICD has been established.)
Word (1...n)	The number of the word containing the parameter. The first word of the minor frame (i.e., first after frame sync) is numbered 1.	No
Word interval	<p>For supercommutated parameters this defines the interval in word positions between two consecutive measurements inside the same minor frame. This means the measurements must be evenly spaced and will continue to the end of the minor frame. A value of 0 (or empty) means there is no supercommutation of this parameter.</p> <p>The timestamps of the supercommutated measurements will also be evenly spaced, starting with the timestamp of the minor frame sync. This means several supercommutated parameters with the same number of measurements per minor frame will always be recorded at the same time, for example the start and middle of the minor frame when there are two measurements.</p>	Yes (will default to 0 i.e., no supercommutation)
MSB pos (1...n)	The number of the most significant bit within the word for the parameter. 1 means the leftmost (front) bit. This deviates from other ICD formats to conform with IRIG 106 chapter 4.	No
Length	<p>The number of bits occupied by this parameter. The range of data will therefore be [MSB pos ... (MSB pos + length - 1)]. For BNR encoding this includes the sign bit.</p> <p>You can define lengths bigger than the number of bits in the defined PCM</p>	No

	word. In this case the surplus bits will be taken from the MSB side of the following PCM word(s).	
Encoding (BNR / UBNR / BCD / Float / Enum / CVSD / ASCII / ByteArray / MIL1750AFloat / DECDFloat)	<p>See CSV ICD formats</p> <p>CVSD can be used to decode IRIG 106 chapter 5 digital audio. It cannot be used with regular display features but only with specialized CVSD functionality (like Replay audio from CVSD, CVSD to WAV Exporter and CVSD live replay). The software decoding does not reach the criteria defined in IRIG 106 appendix F especially when it comes to the attenuation of the low pass filter. However, it produces a well recognizable audio signal.</p> <p>ByteArray defines some sequential PCM words to contain a sequence of raw data in in full byte sizes. This can then be used in specialized features for example in some cases of PCM Video. Byte Arrays must always have a length that is a multiple of 8 bits. They must start at MSB position 1. The decoding will then take 8-bit blocks from the MSB to LSB and continue in the next PCM word. All PCM words that contain bytes of this parameter must have a length that is a multiple of 8. It is possible to use supercommutation to have several Byte Array blocks per minor frame but the more blocks you have the more performance overhead you create for processing. A standard feature for Byte Arrays is to extract this data to a binary file. Customer specific plug-ins can be things like specific video replay.</p>	
Min	See CSV ICD formats	
Max	See CSV ICD formats	
Enumdef	See CSV ICD formats	
Coefficient 0, Coefficient 1, ..., Coefficient n	See CSV ICD formats	

If a parameter reaches outside the valid data word range, the parameter will have an invalid value indicating a "frame short" error.

This PCM ICD file format does not yet support fragmented measurements in non-adjacent PCM words.

4.11.1.4 Analog Simple CSV ICD Format

This ICD format is the basic format for analog channels. The general comments on [CSV ICD formats](#) apply here too.

The version number for the file format described here is 2. Therefore, the first line must read "V2".

Even though analog channels have subchannels this ICD must be mapped to a channel.

This ICD allows overriding some of the TMATS settings for analog subchannels that are missing very often. It also offers a solution for the IRIG 106 specification problem of missing TMATS properties for disabled analog subchannels.

This ICD expects that data is stored MSB first (TMATS: *R-x\AMTO-n-m:M* or *R-x\AMTO-n-m:D*) without required bit masking (TMATS: *R-x\AMSK-n-m:FW*). If you have other requirements, contact us.

You **must** define one parameter for each enabled subchannel. You can also define parameters for disabled subchannels.

The following table describes the meaning of each column:

Column header	Description	Empty allowed?
Name	See <u>CSV ICD formats</u>	
Unit	See <u>CSV ICD formats</u>	
Comment	See <u>CSV ICD formats</u>	
Disabled (X)	If you decide to define parameters for disabled subchannels you must indicate this with an "X" in this column	Yes (if subchannel enabled)
Subchannel	The number of the subchannel that contains the parameter. The number must be 1 ... 256.	No
Sample factor	Same definition as the TMATS attribute <i>R-x\ASF-n-m</i> . It is used to calculate the sample rate of the subchannel. FLIDAS uses the sample rate of the channel defined by the TMATS attribute <i>R-x\ASR-n</i> and multiplies this by $1/(2^n)$ where "n" is the value in this column. This means a value of 0 indicates the same sample rate as the channel, 1 indicates half, 2 a quarter.	No
Length	The number of bits occupied by this parameter including a possible sign bit.	No
Encoding (BNR / UBNR / Float / OffsBNR / SignMagPlus0 / SignMagPlus1 / OnesCompl)	See <u>CSV ICD formats</u> for definition of BNR, UBNR and Float. There is support for other encodings because TMATS (<i>R-x\AF-n-m</i>) allows defining these for analog subchannels too: <ul style="list-style-type: none"> • OffsBNR: Offset binary (implemented with an offset of $2^{\text{length} - 1}$) • SignMagPlus0: Sign and magnitude binary where a sign bit of 0 indicates positive values. The sign bit is assumed at the msb side of the raw value. • SignMagPlus1: Sign and magnitude binary where a sign bit of 1 indicates positive values. 	No

	<p>The sign bit is assumed at the msb side of the raw value.</p> <ul style="list-style-type: none"> OnesCompl: One's complement 	
Min	See CSV ICD formats	
Max	See CSV ICD formats	
Coefficient 0, Coefficient 1, ..., Coefficient n	See CSV ICD formats	

4.11.1.4.1 Full Live Decoding Parameters

When loading the ICD you will get two parameters for each parameter definition in the CSV file. A normal parameter and one with the same name suffixed by **.full** which stands for full live decoding. You will need the full live decoding parameter only in the context where you process parameters from a [live data stream](#) or if you replay data into an [instrument panel](#) (and therefore create "live" updates of data). The reason is that analog data is often sampled in high frequency. This creates a lot of data and comes with a serious performance impact that you may want to avoid when working with data that updates live on the screen. The screen does not update fast enough to display the changes anyway. In these live contexts (live data stream or instrument panel file replay) you have two options:

1. Use the regular parameter which gives you just the first sample out of a packet for each subchannel. Packets are created at least every 100 ms so it gives you at least 10 samples per second. Most of the time your hardware will create packets more often than every 100 ms so you get more samples. If you are just displaying a current value on the screen, you should use this option.
2. Use the full live decoding parameter. This will decode all samples of the packet (all together at the time of the packet). This group of samples can be used whenever you want to display a history of values (like a curve of the last 2 seconds). The parameters will not be accepted in other situations. Take care about performance in these situations. Decoding takes performance every time the data comes in. Displaying takes performance every time the display is updated and increases with the displayed time span. Remember you can also use the regular parameter in these situations if you can accept the fewer samples.

In non-live contexts (like [parameter export](#) or [parameter viewers](#)) the regular parameter also decodes all data and the full live decoding parameter will be ignored. The full live decoding parameter is therefore only needed in live contexts and only if the display shows a history of data and you can't cope with the regular parameter with fewer samples per second.

4.11.1.5 Discrete Simple CSV ICD Format

This ICD format is the basic format for discretely. The general comments on [CSV ICD formats](#) apply here too.

The version number for the file format described here is 1. Therefore, the first line must read "V1".

The following table describes the meaning of each column:

Column header	Description	Empty allowed?
Name	See CSV ICD formats	
Comment	See CSV ICD formats	
Position (31...0)	The position of the discrete within the 32-bit word stored in the CH10 packet entry. The counting starts with 0 for the rightmost bit. Note that the data is automatically LSB aligned by FLIDAS. I.e., if your recorder records only 8 bits in MSB alignment then FLIDAS will shift these 24 bits to the right so the actual bits are in positions 7 ... 0.	No
Encoding (Disc/Enum)	See CSV ICD formats Disc will represent the discrete as a numeric value (0 or 1). This encoding can then be used if you need numbers for exporting for example. Note that you can define the same bit twice as Enum and Disc if you need.	
Enumdef	See CSV ICD formats	

4.11.1.6 CAN Bus Simple CSV ICD Format

This ICD format is the basic format for CAN bus. The general comments on [CSV ICD formats](#) apply here too.

The version number for the file format described here is 3. Therefore, the first line must read "V3".

The following table describes the meaning of each column:

Column header	Description	Empty allowed?						
Name	See CSV ICD formats							
Unit	See CSV ICD formats							
Comment	See CSV ICD formats							
Identifier	The decimal identifier of the message	No						
Endianness B/L	<p>The byte order used for parameters that span over a data byte boundary. B means big-endian, L little-endian.</p> <p>The following example shows an 8 bit parameter with MSB pos 3 in byte 0. The content is displayed as hex digits.</p> <table border="1"> <tr> <th>Byte#</th><th>Bits 7 ... 4</th><th>Bits 3 ... 0</th></tr> <tr> <td>0</td><td></td><td>A</td></tr> </table>	Byte#	Bits 7 ... 4	Bits 3 ... 0	0		A	Yes (uses big-endian by default)
Byte#	Bits 7 ... 4	Bits 3 ... 0						
0		A						

	1 B	
	The interpretation would be: Big-endian: AB Little-endian: BA Big-endian concatenates the content from the table from top to bottom and left to right. Little-endian concatenates the content from the table from bottom to top and left to right.	
Byte (0...7)	The first position of a data byte that contains a part of the parameter (for big-endian this is the byte containing the most significant bit of the parameter for little-endian it is the byte containing the least significant bit of the parameter). The first data byte has the position 0.	
MSB pos (7...0)	See CSV ICD formats . The position of the most significant bit within the byte defined by the byte column. Must be between 7 and 0.	
Length	See CSV ICD formats	
Encoding (BNR / UBNR / BCD / Float / Enum / ASCII / MIL1750AFloat / DECDFloat)	See CSV ICD formats	
Min	See CSV ICD formats	
Max	See CSV ICD formats	
Enumdef	See CSV ICD formats	
Coefficient 0, Coefficient 1, ..., Coefficient n	See CSV ICD formats	

If a CAN bus message is correctly received by the CH10 recorder but the parameter reaches outside the valid data byte range, the parameter will have an invalid value indicating a "message short" error.

4.11.1.7 Ethernet UDP Simple CSV ICD Format

This ICD format is the basic format for UDP on Ethernet. The general comments on [CSV ICD formats](#) apply here too.

The ICD format only supports UDP on IPv4. Also, no VLAN frame tagging (IEEE 802.1Q) or IP fragmentation is supported. If your recording hardware uses the payload only recording mode for Ethernet, FLIDAS simply assumes that the data is UDP on IPv4. It cannot verify this definitely.

FLIDAS will not check the UDP checksum to improve performance. The Ethernet CRC has been checked by the recorder before. Since bit corruption will most likely only occur on the transmission media the UDP consistency is already protected by the Ethernet frame CRC.

FLIDAS will provide the whole UDP payload for decoding and start byte counting with 0 for the first byte.

Since UDP doesn't define anything about how application data is transferred the ICD might not satisfy all your needs. In this case you could order a more specific ICD as customization.

The version number for the file format described here is 2. Therefore, the first line must start with "V2".

To decode the data properly one usually needs a message ID to distinguish different messages. You can define the byte position of your message ID and the length in full bytes (1-4). The IDs are supposed to be big-endian and you can handle different endianness by defining the filter for the message ID on each parameter in reversed bytes. To define the position of your message IDs within the UDP packet you add two columns behind the version number. The first indicates the 0-based byte position, the second the byte length. Example:

```
V2;0;2
```

This means it is a file format version 2 with a message ID position of 0 (first byte) and a 2 byte (16 bit) ID. The semicolon depends on your CSV delimiter settings.

In case you don't use a message ID, you can leave out the last two columns.

The following table describes the meaning of each column for the separate parameter definitions following the version line. Remember that the line with the column headers follows the version line before the parameters:

Column header	Description	Empty allowed?
Name	See CSV ICD formats	
Unit	See CSV ICD formats	
Comment	See CSV ICD formats	
Source IP	The value of the source IP from the IPv4 header. Must be in the form 'a.b.c.d' where a, b, c and d are decimal numbers from 0 to 255.	Yes, if you need no filtering
Destination IP	The value of the destination IP from the IPv4 header. See also source IP.	
Source Port	The value of the source port from the UDP header. Must be decimal from 0 to 65535.	Yes, if you need no filtering
Destination Port	The value of the destination port from the UDP header. See also source port.	
Message ID	The value of the bits from the message ID that identify this type of message. It can either be a non-negative decimal or hexadecimal where hex	Yes, if you need no

	values are prepended by '0x'. For example, '0xff' means 255 decimal.	message ID filtering									
Endianness B/L	<p>The byte order used for parameters that span over a data byte boundary. B means big-endian, L little-endian.</p> <p>The following example shows an 8 bit parameter with MSB pos 3 in byte 0. The content is displayed as hex digits.</p> <table border="1"> <thead> <tr> <th>Byte#</th><th>Bits 7 ...4</th><th>Bits 3 ... 0</th></tr> </thead> <tbody> <tr> <td>0</td><td></td><td>A</td></tr> <tr> <td>1</td><td>B</td><td></td></tr> </tbody> </table> <p>The interpretation would be: Big-endian: AB Little-endian: BA Big-endian concatenates the content from the table from top to bottom and left to right. Little-endian concatenates the content from the table from bottom to top and left to right.</p>	Byte#	Bits 7 ...4	Bits 3 ... 0	0		A	1	B		Yes (uses big-endian by default)
Byte#	Bits 7 ...4	Bits 3 ... 0									
0		A									
1	B										
Byte (0 based)	The first position of a data byte that contains a part of the parameter (for big-endian this is the byte containing the most significant bit of the parameter for little-endian it is the byte containing the least significant bit of the parameter). The first byte has the position 0.	No									
MSB pos (7...0)	See CSV ICD formats . The position of the most significant bit within the byte defined by the byte column. Must be between 7 and 0.										
Length	See CSV ICD formats										
Encoding (BNR / UBNR / BCD / Float / Enum / ASCII / MIL1750AFloat / DECDFloat)	See CSV ICD formats										
Min	See CSV ICD formats										
Max	See CSV ICD formats										
Enumdef	See CSV ICD formats										
Coefficient 0, Coefficient 1, ..., Coefficient n	See CSV ICD formats										

If a UDP packet is correctly received by the CH10 recorder but the parameter reaches outside the valid data byte range, the parameter will have an invalid value indicating a "message short" error.

Note: In general, FLIDAS does not support IP fragmentation for this ICD. However, FLIDAS will process first fragments and only ignore later fragments. This means you could safely access parameters that are in the first fragment. Parameters in further fragments will result in invalid values indicating a "Message short" error.

4.11.1.8 Recording Events Simple CSV ICD Format

This ICD format is the basic format for recording events. The general comments on [CSV ICD formats](#) apply here too.

The decoded values will be the event counts from the recording event packet entries.

The version number for the file format described here is 1. Therefore, the first line must read "V1".

The following table describes the meaning of each column:

Column header	Description	Empty allowed?
Name	See CSV ICD formats	
Unit	See CSV ICD formats	
Comment	See CSV ICD formats	
Event number (0...4095)	The event number as it appears in the recording event packet entries. This means 0 is the first.	No
Offset	A value that will be added to the event count from the recording event packet entries. Since CH10 starts counting the first event with 0 you can put a 1 here if you want to start counting with 1.	Yes (uses 0)
Min	See CSV ICD formats	
Max	See CSV ICD formats	

4.11.1.9 UART ASCII Simple CSV ICD Format

This ICD format is the basic format for text-based UART communication. The general comments on [CSV ICD formats](#) apply here too.

UART communication in general is only defined on a very low level and applications are free to be creative with the protocol they put on top. Therefore, an ICD format cannot support all possible protocols. This ICD format supports specific types of UART communication with the following limitations that hopefully apply to a wider range of protocols:

- The communication is text-based. Therefore, parameters will not be found at fixed positions.
- The parameters will be separated by a single character as delimiter. The section between two field delimiters is called a "field".
- Messages are separated by specific characters (may be control characters like carriage return or line feed).

- Different types of messages with a different set of parameters can be distinguished by a fixed value in one field with a fixed field index.
- The timestamps are taken from the packet entry that contained the start of the message. Since a packet entry may contain several message fragments and may contain transmission pauses, the timestamp is not accurate. However, the sentence was transmitted between this timestamp and the timestamp of the next packet entry. Note that timestamps on UART packet entries are optional. If the hardware does not record this, all packet entries of one packet share the same timestamp. Therefore, timestamps may be up to 100 ms off.
- Fractional values - if they need to be decoded as numbers instead of text - must use a "." as decimal separator and can use scientific notation (like 3.4E-2).
- UART data will be 8 data bits per word. This limitation is from IRIG 106 chapter 10. Storage of other data is simply undefined.
- The data will be encoded as 7-bit ASCII characters. This means the 8th bit on top is invalid.
- FLIDAS does not support multiple streams of differently formatted data on a single subchannel.

If you have other requirements, contact us. It might be possible to weaken some limitations or create a new ICD format.

The version number for the file format described here is 1. Therefore, the first line must start with "V1".

The first line contains more data in columns described here:

Column number	Description
1 (version number)	Must be V1
2 (message prefix)	Each message can be prefixed with some fixed characters that will be removed before processing the message. The prefix will be used to check if the message is valid. The value must be given in hexadecimal and each two digits represent an ASCII code. There may be several or none. There should be no whitespaces. For NMEA 0183 data (e.g., GPS) for example the prefix character would be a "\$" and it would be represented by the hex value "24".
3 (message ID field number)	The field that contains the message ID. The message ID is the fixed value used to identify the message type. The counting starts with 0 on the first column. You can leave it empty if you don't need a message ID. Then all messages will be decoded the same way.
4 (message delimiter)	IRIG 106 Chapter 10 does not define where one message ends and the next starts. You must define one or several characters as delimiter. Each character would be represented by two hex digits without any whitespaces. For NMEA 0183 data (e.g., GPS) for example the delimiter would be carriage return followed by line feed which would be "0D0A" in hexadecimal.
5 (field delimiter)	The character that separates two fields of data in the message. It must be a single character which is represented by its hexadecimal ASCII code in two

	digits. For NMEA 0183 data (e.g., GPS) it would be the "," character represented by "2C" in hexadecimal.
--	--

A sample line for NMEA 0183 (e.g., GPS) using ";" as CSV delimiter could look like this:

```
V1;24;0;0D0A;2C
```

This means it is a file format version 1 with a message prefix of "\$", a message ID in the field 0 (first one). Messages are separated by carriage return followed by line feed and the fields are separated by a ";". The semicolon depends on your CSV delimiter settings.

The following table describes the meaning of each column following the first line:

Column header	Description	Empty allowed?
Name	See CSV ICD formats	
Unit	See CSV ICD formats	
Comment	See CSV ICD formats	
Message ID	The exact value that must be in the message ID field to decode the parameter. Note that if you have defined a message prefix and if the message ID is in the first field, the prefix will be removed before the comparison.	Yes (decodes from all messages)
Field number	The number of the field that contains the parameter value. The first field is field 0	No
Encoding (Integer/Float/Text)	This defines how the text in the field should be interpreted: <ul style="list-style-type: none"> • Integer: Interprets the text as integral number • Float: Interprets the text as fractional number. The decimal separator must be a "." and scientific notation like "3.4E-3" may be used. • Text: Just takes the text as it is. It will not be interpreted in any way and can especially not be used in drawing time-value graphs. 	No
Min	See CSV ICD formats	
Max	Defining minimum and maximum is possible for integer and float encoding. There will be no calculation of minimum and maximum if it is not defined here.	

If a message is correctly received by the CH10 recorder but the parameter reaches outside the valid field range, the parameter will have an invalid value indicating a "message short" error. If a numeric parameter value cannot be interpreted as a number the parameter will have an invalid value reporting the text of the field with an error message.

Since IRIG 106 Chapter 10 does not know how to separate UART messages, recordings may start in the middle of a message. This means that especially the first message may be corrupted and decoded wrong. If you have defined a message prefix, this can be detected and the decoding will be suppressed.

4.11.1.10 UART Message-Aligned Binary Simple CSV ICD Format

This ICD format supports decoding of binary data in CH10 UART data type. The general comments on CSV ICD formats apply here too.

There are some important assumptions that are not guaranteed by CH10. If they do not hold in your situation, you should not use it. But the requests for this type of ICD were so common that it was released anyway to let you decide yourself if you can use it. Probably it is OK in many situations.

For this ICD FLIDAS assumes that a UART packet entry contains the data of a full UART message, i.e. the first byte is the start of the message and the last is the end. Also it assumes that data words are 8 bit long. For more details see the limitations in the end of this chapter.

If we assume that a packet entry matches a message, we still have no definition of the protocol and don't know how application data is transferred. So this ICD might not satisfy all your needs. In this case you could order a more specific ICD as customization. Also there are customer specific plug-ins that join multiple packet entries to what makes up a message in their environment.

The version number for the file format described here is 1. Therefore, the first line must start with "V1".

To decode the data properly one usually needs a message ID to distinguish different messages. You can define the byte position of your message ID and the length in full bytes (1-4). The IDs are supposed to be big-endian and you can handle different endianness by defining the filter for the message ID on each parameter in reversed bytes. To define the position of your message IDs within the message data you add two columns behind the version number. The first indicates the 0-based byte position, the second the byte length. Example:

```
V1;0;2
```

This means it is a file format version 1 with a message ID position of 0 (first byte) and a 2 byte (16 bit) ID. The semicolon depends on your CSV delimiter settings.

In case you don't use a message ID, you can leave out the last two columns.

The following table describes the meaning of each column for the separate parameter definitions following the version line. Remember that the line with the column headers follows the version line before the parameters:

Column header	Description	Empty allowed?									
Name	See CSV ICD formats										
Unit	See CSV ICD formats										
Comment	See CSV ICD formats										
Message ID	The value of the bits from the message ID that identify this type of message. It can either be a non-negative decimal or hexadecimal where hex values are prepended by '0x'. For example, '0xff' means 255 decimal.	Yes, if you need no message ID filtering									
Endianness B/L	<p>The byte order used for parameters that span over a data byte boundary. B means big-endian, L little-endian.</p> <p>The following example shows an 8 bit parameter with MSB pos 3 in byte 0. The content is displayed as hex digits.</p> <table border="1"> <thead> <tr> <th>Byte#</th><th>Bits 7 ...4</th><th>Bits 3 ... 0</th></tr> </thead> <tbody> <tr> <td>0</td><td></td><td>A</td></tr> <tr> <td>1</td><td>B</td><td></td></tr> </tbody> </table> <p>The interpretation would be: Big-endian: AB Little-endian: BA Big-endian concatenates the content from the table from top to bottom and left to right. Little-endian concatenates the content from the table from bottom to top and left to right.</p>	Byte#	Bits 7 ...4	Bits 3 ... 0	0		A	1	B		Yes (uses big-endian by default)
Byte#	Bits 7 ...4	Bits 3 ... 0									
0		A									
1	B										
Byte (0 based)	The first position of a data byte that contains a part of the parameter (for big-endian this is the byte containing the most significant bit of the parameter for little-endian it is the byte containing the least significant bit of the parameter). The first byte has the position 0.	No									
MSB pos (7...0)	See CSV ICD formats . The position of the most significant bit within the byte defined by the byte column. Must be between 7 and 0.										
Length	See CSV ICD formats										
Encoding (BNR / UBNR / BCD / Float / Enum / ASCII / MIL1750AFloat / DECDFloat)	See CSV ICD formats										
Min	See CSV ICD formats										
Max	See CSV ICD formats										

Enumdef	See CSV ICD formats
Coefficient 0, Coefficient 1, ..., Coefficient n	See CSV ICD formats

If the parameter reaches outside the valid data of the packet entry, the parameter will have an invalid value indicating a "message short" error.

4.11.1.10.1 Limitations

The problem is that UART does not define what a message is. It transmits single data words. CH10 records groups of these data words and stores them in a UART packet entry. How it groups them is up to the recorder manufacturer. Typically you would expect that you have something like a message which would start after a gap in transmission and end on the next gap. CH10 does not say anything about this. So if you look at a CH10 UART packet entry you will not know if the first byte is the start of a message, the middle or the end. You will not know if the recorded block of data has gaps itself in the middle.

However, at least there are CH10 recorders that give additional guarantees on this. If you know your recorder you can check if the assumptions are OK for you. For example on some Safran Data Systems Recorders (DATaRec4 series) you can define a message gap (minimum number or bit times without transmission that start the creation of a new packet entry (vendor specific TMATS: V-x\ZDS\R\UMG-n)). Possibly other recorders can configure this as well or at least have some guarantees you can rely on.

You need to be in control of when a new packet entry is started:

- Required: After a gap that is big enough to indicate a message start in your protocol.
- Forbidden: On gaps that are small enough that they may happen inside your message: You need to be able to prevent this
- Forbidden: After 65535 bytes as a packet entry cannot contain more. This is not likely to happen as on typical baud rates you cannot transmit this much data within the 100 ms packet duration limit. You would need something around 7 MBaud.
- Forbidden: After 100 ms after the first bit in the packet (not packet entry) expired. The new packet entry will then start in a new packet. This means you need to make sure a single message does not span more than 100 ms and the recorder should move the complete message to a new packet if the current packet cannot contain it anymore rather than storing the first segment and putting the second segment in the next packet.

This means especially if your messages exceed 100 ms you cannot get it to work. 100 ms on an 9.6 KBaud connection is 80-96 bytes message length, on 115.2 KBaud it is already 960 - 1152 bytes. In practice messages often seem to take more time than the calculated values.

A note on UART word sizes: TMATS allows 7-9 bit UART word sizes. CH10 does not support storing 9 bit word sizes. CH10 also does not define how 7 bits should be stored (1 bit padding or packed with filler bits in the end). Therefore, FLIDAS only supports the 8 bit per word option.

4.11.1.11 Message Data Binary Simple CSV ICD Format

This ICD format is the basic format for the CH10 message data type with binary data. The general comments on [CSV ICD formats](#) apply here too.

Since message data could be any type of protocol and doesn't define anything about how application data is transferred the ICD might not satisfy all your needs. In this case you could order a more specific ICD as customization.

The version number for the file format described here is 2. Therefore, the first line must start with "V2".

To decode the data properly one usually needs a message ID to distinguish different messages. You can define the byte position of your message ID and the length in full bytes (1-4). The IDs are supposed to be big-endian and you can handle different endianness by defining the filter for the message ID on each parameter in reversed bytes. To define the position of your message IDs within the message data you add two columns behind the version number. The first indicates the 0-based byte position, the second the byte length. Example:

```
V2;0;2
```

This means it is a file format version 2 with a message ID position of 0 (first byte) and a 2 byte (16 bit) ID. The semicolon depends on your CSV delimiter settings.

In case you don't use a message ID, you can leave out the last two columns.

The following table describes the meaning of each column for the separate parameter definitions following the version line. Remember that the line with the column headers follows the version line before the parameters:

Column header	Description	Empty allowed?
Name	See CSV ICD formats	
Unit	See CSV ICD formats	
Comment	See CSV ICD formats	
Message ID	The value of the bits from the message ID that identify this type of message. It can either be a non-negative decimal or hexadecimal where hex values are prepended by '0x'. For example, '0xff' means 255 decimal.	Yes, if you need no message ID filtering
Endianness B/L	The byte order used for parameters that span over a data byte boundary. B means big-endian, L little-endian.	Yes (uses big-endian by default)

	<p>The following example shows an 8 bit parameter with MSB pos 3 in byte 0. The content is displayed as hex digits.</p> <table border="1"> <thead> <tr> <th>Byte#</th><th>Bits 7 ...4</th><th>Bits 3 ... 0</th></tr> </thead> <tbody> <tr> <td>0</td><td></td><td>A</td></tr> <tr> <td>1</td><td>B</td><td></td></tr> </tbody> </table> <p>The interpretation would be: Big-endian: AB Little-endian: BA Big-endian concatenates the content from the table from top to bottom and left to right. Little-endian concatenates the content from the table from bottom to top and left to right.</p>	Byte#	Bits 7 ...4	Bits 3 ... 0	0		A	1	B		
Byte#	Bits 7 ...4	Bits 3 ... 0									
0		A									
1	B										
Byte (0 based)	The first position of a data byte that contains a part of the parameter (for big-endian this is the byte containing the most significant bit of the parameter for little-endian it is the byte containing the least significant bit of the parameter). The first byte has the position 0.	No									
MSB pos (7...0)	See CSV ICD formats . The position of the most significant bit within the byte defined by the byte column. Must be between 7 and 0.										
Length	See CSV ICD formats										
Encoding (BNR / UBNR / BCD / Float / Enum / ASCII / MIL1750AFloat / DECDFloat)	See CSV ICD formats										
Min	See CSV ICD formats										
Max	See CSV ICD formats										
Enumdef	See CSV ICD formats										
Coefficient 0, Coefficient 1, ..., Coefficient n	See CSV ICD formats										

If the parameter reaches outside the valid data of the message data block, the parameter will have an invalid value indicating a "message short" error.

For segmented message data packet entries these are joined to complete messages first. If any segment is missing or any error is indicated in the packet entries, the message is not used for decoding.

4.11.1.12 User Defined Computer-Generated Data Binary Simple CSV ICD Format

This ICD format is the basic format for the CH10 user defined computer-generated data type with binary data. The general comments on [CSV ICD formats](#) apply here too.

Since user defined computer-generated data is not further defined it could use any kind of storage and the ICD might not satisfy all your needs. In this case you could order a more specific ICD as customization.

The version number for the file format described here is 1. Therefore, the first line must start with "V1".

To decode the data properly one usually needs a message ID to distinguish different messages. You can define the byte position of your message ID and the length in full bytes (1-4). The IDs are supposed to be big-endian and you can handle different endianness by defining the filter for the message ID on each parameter in reversed bytes. To define the position of your message IDs within the message data you add two columns behind the version number. The first indicates the 0-based byte position, the second the byte length. Example:

```
V1;0;2
```

This means it is a file format version 1 with a message ID position of 0 (first byte) and a 2 byte (16 bit) ID. The semicolon depends on your CSV delimiter settings.

In case you don't use a message ID, you can leave out the last two columns.

The following table describes the meaning of each column for the separate parameter definitions following the version line. Remember that the line with the column headers follows the version line before the parameters:

Column header	Description	Empty allowed?
Name	See CSV ICD formats	
Unit	See CSV ICD formats	
Comment	See CSV ICD formats	
Message ID	The value of the bits from the message ID that identify this type of message. It can either be a non-negative decimal or hexadecimal where hex values are prepended by '0x'. For example, '0xff' means 255 decimal.	Yes, if you need no message ID filtering
Endianness B/L	The byte order used for parameters that span over a data byte boundary. B means big-endian, L little-endian.	Yes (uses big-endian by default)

	<p>The following example shows an 8 bit parameter with MSB pos 3 in byte 0. The content is displayed as hex digits.</p> <table border="1"> <thead> <tr> <th>Byte#</th><th>Bits 7 ...4</th><th>Bits 3 ... 0</th></tr> </thead> <tbody> <tr> <td>0</td><td></td><td>A</td></tr> <tr> <td>1</td><td>B</td><td></td></tr> </tbody> </table> <p>The interpretation would be: Big-endian: AB Little-endian: BA Big-endian concatenates the content from the table from top to bottom and left to right. Little-endian concatenates the content from the table from bottom to top and left to right.</p>	Byte#	Bits 7 ...4	Bits 3 ... 0	0		A	1	B		
Byte#	Bits 7 ...4	Bits 3 ... 0									
0		A									
1	B										
Byte (0 based)	The first position of a data byte that contains a part of the parameter (for big-endian this is the byte containing the most significant bit of the parameter for little-endian it is the byte containing the least significant bit of the parameter). The first byte has the position 0.	No									
MSB pos (7...0)	See CSV ICD formats . The position of the most significant bit within the byte defined by the byte column. Must be between 7 and 0.										
Length	See CSV ICD formats										
Encoding (BNR / UBNR / BCD / Float / Enum / ASCII / MIL1750AFloat / DECDFloat)	See CSV ICD formats										
Min	See CSV ICD formats										
Max	See CSV ICD formats										
Enumdef	See CSV ICD formats										
Coefficient 0, Coefficient 1, ..., Coefficient n	See CSV ICD formats										

If the parameter reaches outside the valid data of the user defined computer-generated data block, the parameter will have an invalid value indicating a "message short" error.

The data bytes will be everything that follows the channel specific data word of the CH10 user defined computer-generated data packet.

4.11.2 NMEA 0183 on UART or TSPI Hardcoded ICD Format

This ICD format is a predefined ICD for GNSS (e.g., GPS, Galileo, GLONASS and Beidou) information in the NMEA 0183 protocol. It supports data on a UART subchannel or on a

TSPI format 0 channel. There is no need for you to enter the ICD, instead it automatically decodes the essential parameters given below. If you need other parameters, ask for extending it.

The parameters have hardcoded minimum and maximum values that can be used e.g., in graph displays. If you require a broader range for your application, you can write an email and ask for change. Note that you can also use auto scaling in graph displays (minimum and maximum of observed values) or configure the minimum and maximum for each graph.

The decoded parameters are:

Parameter	Description	Decoded from which sentence?	Minimum	Maximum
Latitude	The latitude in degrees (positive is north)	RMC, GGA, GLL	-90	90
Longitude	The longitude in degrees (positive is east)	RMC, GGA, GLL	-180	180
Track True	Track angle in degrees true	RMC, VTG	0	360
Track Magnetic	Track angle in degrees magnetic	VTG	0	360
Ground Speed kts	Speed over ground in knots	RMC, VTG	0	2500
Ground Speed km/h	Speed over ground in km/h	VTG	0	4500
Altitude	Altitude in meters above mean sea level	GGA	-500	25000
Magnetic Variation	Magnetic variation in degrees	RMC	-180	180
UTC Date	UTC date in the format ddmmyy	RMC		
UTC Time	UTC time in the format hhmmss.ss	RMC, GGA, GLL		
Fix Quality	Quality of GPS fix (GPS Fix, DGPS Fix ...)	GGA	0	8
Satellites Used	Number of satellites used for position fix	GGA	0	40
HDOP	Horizontal dilution of precision	GGA, GSA	0	100
VDOP	Vertical dilution of precision	GSA	0	100
PDOP	Position (3D) dilution of precision	GSA	0	100

NMEA 0183 sentences with invalid checksum will not be evaluated at all for parameter decoding. In some cases where the checksum is valid but the payload itself indicates invalid data (e.g., RMC without active state) the parameter's value will indicate invalid data.

Note: Checksums are optional on NMEA 0183 but if a sentence of a specific type with a checksum is seen on the NMEA 0183 source, all following sentences of that type are supposed to have one or are considered to have a checksum error. This makes sure incomplete sentences are invalid.

The parameter's timestamps will be synchronized to the time of the UART or TSPI packet entry that contained them rather than the time contained in some of the NMEA 0183 sentences. Note that timestamps on UART and TSPI packet entries are optional. If the hardware does not record this, all packet entries of one packet share the same timestamp (i.e., may be off up to 100 ms from the reception time). On UART a single packet entry may contain several NMEA 0183 sentence fragments and may contain transmission pauses, which also contributes to the inaccuracy.

Note that TSPI format 0 can contain other data than NMEA 0183 or NMEA 0183 HS, which will be ignored.

4.11.3 PCM TMATS ICD Format

This ICD format is an alternative to the PCM Simple CSV ICD Format. It is based on the ASCII TMATS format described in IRIG 106 chapter 9 but does not provide all the TMATS features. Typically this means you can still use existing ICDs but may lose parameters with unsupported features.

You need to link an external TMATS file in the ICD mapping. It will not be taken from a CH10 file. The reasoning behind this is that an ICD mapping can be active independently to any opened CH10 files and may apply its parameters to several files with the same parameter setup. If you have the information in a CH10 file you can of course extract the TMATS information to an external file and use it inside an ICD mapping.

The TMATS file must contain information from the P, D and C groups and they must be properly linked. Some P-group information will be used to create the parameters. However, when you apply the parameters to a CH10 file the P-group settings of this CH10 file will be active for general decommutation. This means both should not contradict each other!

When you map the TMATS file in the ICD mapping you need to provide two more options:

- Data link name: The name of the data link that links P and D group by the P-d\DLN and D-x\DLN value.
- Measurement list name: The name of the measurement list to load as parameters. This is the value from D-x\MLN-y. If you leave this empty the first measurement list will be used.

If you need any unsupported option contact Data Bus Tools. Often the implementation is easily possible and hasn't been done because it seemed a rare option. In other cases IRIG 106 chapter 9 may be too imprecise to implement it. In this case it could be implemented if you can provide samples (and the implementation then potentially only matches your sample's interpretation and might conflict with others). The implementation will most likely be within regular versions i.e., will need no funding from your side.

Note that for this ICD you cannot manually define the minimum and maximum values of parameters like in the PCM Simple CSV ICD Format. This means they will be calculated under the same conditions as with the CSV ICD.

Note that the size of the TMATS file influences the loading time of the ICD. If you encounter problems try to remove unused attribute groups and P-/D-/C-groups that refer to other data links.

4.11.3.1 Additional Notes on Features

4.11.3.1.1 D-Group Features

4.11.3.1.1.1 Bit Masks

FLIDAS uses msb position and length (like in the PCM Simple CSV ICD Format) to describe what is taken from D-x\WFM-y-n-m-e "bit mask". If you use a "full word" bit mask the P-Group must define the attributes to calculate the specific word size. Remember that msb position 1 is the msb of a PCM word (complying with IRIG 106 chapter 4).

4.11.3.1.1.2 Measurement Locations

D-x\LT-y-n "measurement location type" only supports the "word and frame" option.

Only single measurement locations are supported. Note that a single location can specify supercommutation through the word interval.

If a parameter reaches outside the valid data word range during decommutation, the parameter will have an invalid value indicating a "frame short" error.

4.11.3.1.1.3 Word and Frame Intervals

Word and frame intervals always run to the end and the options D-x\EWP-y-n-m-e "end word position" and D-x\EFP-y-n-m-e "end frame position" may not be used (they have also been removed from the standard in 2017). *If they are used anyway FLIDAS will check if this is redundant information (as the values in fact indicate the real end of the minor / major frame) and accept it in this case. Other values will be rejected.*

The default value for an undefined frame interval (D-x\FI-y-n-m-e) is 0 like specified in IRIG 106 Chapter 9. This differs from the default value of the PCM Simple CSV ICD Format that was specified before TMATS introduced a default.

4.11.3.1.1.4 Fragmentation and Transfer Orders

Fragmented measurements are supported with some limitations on rare cases or places where the TMATS specification is really unclear to avoid invalid implementations.

Limitations:

- Fragments of a measurement must be in the same minor frame (i.e., frame position and frame interval must be the same for all fragments of a measurement)
- The fragment transfer order must be the same for all fragments of a measurement.

Transfer order is defined on three levels: The PCM word level, the measurement level, the fragment level. Each can define bit order as msb or lsb first. All combinations are valid and can also be different for different fragments of the same measurement. But this is not likely to happen (if a sensor produces measurements with lsb first, then probably the fragments will not be msb first). It does not seem 100% sure how these combinations must be interpreted. The usual case (all msb first) is clear.

When looking into all combinations it seems that the order of processing may be different but the outcome is the same for some of them. It seems that the measurement transfer order does not have an influence on decoding at all except for the optional parity bit location.

The processing for msb first transfer ordering on all levels is tested well. For all other cases, take care and check the results or review the processing described here.

FLIDAS processes the fragmented measurements like this:

1. The PCM words are already in msb first order (FLIDAS always corrects the bit order to msb first when reading the minor frame raw data).
2. If the declared PCM word and fragment transfer order matches:
 1. FLIDAS fills the measurement from left to right with the bits of the fragments with increasing fragment positions.

otherwise:

2. FLIDAS fills the measurement from right to left with the bits of the fragments with increasing fragment positions.
 3. The measurement bits are reversed
3. Measurement level parity is checked if defined and the parity bit is removed then:
 - If measurement transfer order is msb first and the parity is leading the measurement or if measurement transfer order is lsb first and the parity is trailing the measurement the parity bit will be located on the msb side.
 - otherwise the parity bit will be located on the lsb side
4. Conversions are applied.

4.11.3.1.1.5 Parity

Parity is supported on the measurement level. If a parameter has an invalid parity, it will have an invalid value indicating a "parity error".

4.11.3.1.1.6 Simultaneous Sampling

The simultaneous sampling settings will not be used.

4.11.3.1.2 C-Group Features

4.11.3.1.2.1 Binary and Floating Point Formats

The following binary formats (C-d\BFM) are supported and "unsigned binary" is used if undefined:

- UNS: Unsigned binary

- TWO: Two's complement
- ONE: One's complement
- BCD: Binary coded decimal
- OFF: Offset binary (implemented with an offset of $2^{\text{length} - 1}$)
- SIG: Sign magnitude binary (plus is 0) (the sign bit is assumed at the msb side of the raw value)
- SIM: Sign magnitude binary (plus is 1) (the sign bit is assumed at the msb side of the raw value)
- FPT: Floating point
- BWT: Bit weight

The following floating point formats (C-d\FPF) are supported and:

- IEEE_32
- IEEE_64
- 1750A_32
- 1750A_48
- DEC_64

4.11.3.1.2.2 Data Conversion Types

The following data conversion types (C-d\DCT) are supported and "none" is used if undefined:

- None
- Coefficients
- Discrete
- PCM time
- Pair set
- Digital voice

4.11.3.1.2.3 PCM Time

PCM chapter 4 time comes in two types, BCD and binary. TMATS does not explicitly have an option to define the type. The assumption is that a binary format C-d\BFM of "BCD" will select the BCD type and all other settings lead to the binary type. However, TMATS tools seem to exist that define PCM BCD time using a binary format of unsigned, giving no chance to detect it for FLIDAS.

FLIDAS will provide PCM time as a seconds count. Since microseconds, low and high order word are three different parameters, you have three numbers that adding up result in the seconds time. So, you can use this for external processing and sum it up or in live display to see if the values keep incrementing.

4.11.3.1.2.4 Pair Sets

The polynomial curve fit option (C-d\PS1:Y) is not supported. For the linear interpolated type the interpolation is done between the pairs. Before the first pair and after the last pair, the values stay at the engineering unit value of the first/last pair. This is unspecified in chapter 9 but is what a user recommended.

4.11.3.1.2.5 Discretes (Enumerations)

The binary format must be an integral one. This excludes floating point values but also bit weights as these could have fractional values.

4.11.3.1.2.6 Digital Voice

The binary format C-d\BFM will be ignored here. The encoding method (C-d\VOI\E) must be "CVSD".

4.11.3.1.2.7 Comments

The comments for the parameters are taken from C-d\MN1 "description".

4.12 CH10 File Loading

Before you can work with a CH10 file you need to load the file by selecting **File > Open CH10 File** or using the keyboard shortcut **Alt+O**.

After you have selected the CH10 file a wizard pops up for the further configuration. The configuration controls which data is loaded from the CH10 file. It can be used to increase the performance of loading and overall handling of the file. It will also help you focus on CH10 data types important for your current task.

The settings belong to several groups located on different pages:

- An optional page only appearing in special situations setting up Safran CH10 file segmentation
- TMATS override settings
- CH10 file loading properties consisting of:
 - **Check for data checksum errors** controls if packet's data will be checked for checksum errors while processing the CH10 source (the packet header and secondary header will always be checked). Invalid packets will be shown red in the Packet View. If you skip this check, the processing of CH10 sources runs faster but checksums are ignored. You can use this in an environment where errors are unlikely to occur.
 - **Try finding next packet on wrong packet sync** controls the behavior when a packet with a wrong header checksum or packet sync has been found in a CH10 file. You can either skip the further loading of the file or let FLIDAS try to find the next packet. When many corrupt packets are in the file, this option will decrease loading performance significantly and finding the packets is based on a heuristic approach (the next place with a valid sync, header, secondary header and data checksum will be considered the next packet).
 - **Ignore MSIDs**: Normally FLIDAS will support MSIDs (see distributed CH10 multiplexer sources). This means that it will expect a separate time channel for each MSID for RTC timing. If you choose to ignore MSIDs, FLIDAS will treat the file as if all channels belong to the same MSID 0 and therefore like a regular CH10 file. If your files do not use MSIDs at all, both modes should be the same. But if you use MSIDs and there is only one time channel and the RTC is synchronized between sources, this option should be used.

-
- **Ignore secondary header time:** FLIDAS will decode CH10 packet time from the secondary header if available. Otherwise it will use regular RTC timing. Using this option you can use the RTC timing instead of secondary header time. Note that a packet with RTC sync error will neither decode RTC time, nor secondary header ERTC time but use a fallback time. If your CH10 sources do not use secondary headers (which is usually the case) this option makes no difference.
 - **Partial reading** allows reading just a fragment of a file. For example if you have a file of many GB size and you are just interested in something occurring close to the end of the file. You can select a byte position where to start and/or end reading. Anything in that range will be read. Just make an approximation of the position. You can prepend "M" to the number to indicate 2^{20} bytes or "G" for 2^{30} bytes. The first TMATS record in the file will be read before reading from the **Start position**. The first packet after or at the start position will be the first non-TMATS packet. Any packet that started before the **End position** will be read completely.
 - **User property file:** Enables the use of a property file stored for the CH10 file. The file is used to persist time markers you created for the next time you use the CH10 file. If you enable it, you need to provide the name of a file. You can click on **Default** to use the default file name which would be the CH10 file name extended by ".flidas" stored in the same directory. It is recommended to use this file name. If you use another one and save the wizard page setup, you need to enter the name manually every time. The **Browse** button allows selecting a property file. If you forget enabling the property file here, you can later enable the property file for the current session by right-clicking the CH10 file in the Repository and selecting "Save Property File". This only works if you changed the time markers in any way.
 - **Start year of recording** to define the year that the recording was started in. This is used when the timestamps inside the CH10 file do not contain year and month information. The value that is found here when the dialog opens is the minimum of the current year and the CH10 file's file modification date (in your file system not in the recorder media). It is further corrected by the TMATS "original recording date" when processing timestamps if you check the option **Try to get start year from TMATS**.
 - **Maximum number of unexpected processing errors before termination** controls how many unexpected errors are allowed to occur during the processing of a CH10 source. If this threshold is met, the processing will terminate. You can still use the data processed so far. Only errors not explicitly handled by FLIDAS that occur during CH10 file loading or initial CH10 packet processing on live data streams count against this threshold.
 - **Data Source ID:** Sometimes the TMATS data in the CH10 file might contain several R-x sections describing the recording hardware setup. In this case you can enter the data source ID (R-x\ID, G\DSI-n) to select which one FLIDAS shall use. The value is case insensitive. If you enter nothing, FLIDAS will take the R-x group with the lowest "x" (which is the only one most of the time anyway).
- CH10 data type filters
 - Time channel offsets

You can later see which configuration the CH10 file has been loaded with if you right-click the file in the Repository View, select **Properties** and go to the **CH10 Data Type Filters** and the **General File Information** page.

4.13 Selecting Time Markers or Recording Events

Some places in FLIDAS allow selecting a time. Often you can select time markers or recording events. There is a common dialog to choose them.

The dialog has two pages (or just one if only time markers are allowed). You can simply choose a time marker or a recording event on the separate pages and finish the dialog with OK. You can also double-click the line in the table instead. In case of recording events there is an option below the table to choose if you want to use the regular time of the event or the additional time stamp that is optionally stored with the recording event. Both are visible in the table.

4.14 Default Table Actions

FLIDAS offers some generic features for many tables on the GUI.

4.14.1 Saving Individual Column Setups

Most of the tables allow reordering the columns when you drag the column header and place it between other column headers. You can also resize the columns when you drag the area between two column headers.

Most of the tables also have a context menu when you right-click their header. Here you can enable/disable each column. You can also save the setup you created. You can also reset the column order or widths to its default.

If your columns are too small in general (e.g., on Linux), you can apply a scaling factor to it which is found in the General Preferences. This will only apply to not individually configured columns.

If you want to remove all individual configured column widths (not order) you can use the button on the GUI Scaling (troubleshooting) Preferences

4.14.2 Table Cell Copy

This feature allows copying table cells to the clipboard to reuse the text in other applications.

The feature is accessible by right-clicking into a table and selecting **Copy Table Cell to Clipboard**. A cross-hair cursor will appear and you must click into the cell you want to copy. The cross-hair cursor will disappear without copying if you hit any key on the keyboard or switch to another GUI element.

4.14.3 Table Exporting

This feature allows exporting tables to a text file in the CSV format. It is only available in the commercial FLIDAS versions.

The exporting is accessible by right-clicking into a table and selecting **Export Table Content to CSV**. The following dialogs allow entering a file name and a CSV column delimiter that is used to separate two columns in the text file from each other. The default column delimiter can be configured in the General Preferences.

When the exported table contains a hierarchy (i.e., a tree structure with parent-children relationships), the CSV file will contain a leading column indicating the hierarchy level starting with 0 for the root level.

While exporting the process can be cancelled. The data exported so far will be available.

4.14.4 Table Font Switching

This feature allows switching the font in a table between a default and a monospaced font. This may be useful if you have a table with raw data inside one column and want the characters inside the column aligned below each other.

You can toggle the fonts by right-clicking into a table and selecting **Use Monospace Font**.

4.14.5 Expanding / Collapsing a Tree

If the table has a tree like structure (has parent lines with children lines below), you can expand the tree structure down to the lowest level with **Expand All** or collapse it to only show the top level with **Collapse All**.

4.15 CH10 Packet Body Exporting

You can export the packet body of a CH10 packet to a file. The function is available in all commercial versions.

You start the export when you right-click a packet in the Packet View or Packet Entry View and select **Export Binary Packet Body**. You can then select an export file name and then a byte offset in the packet body of the CH10 packet to start the export. A value of 0 exports the whole body. A value of 4 skips the channel specific data word for most data types for example.

4.16 TMATS Overriding

TMATS overriding can be used to suppress the TMATS setup of a CH10 source and use an external TMATS file instead.

This can be useful for example if the TMATS setup is wrong or if some attributes are missing. FLIDAS doesn't care if the attributes are newer than the IRIG 106 revision used for the original TMATS setup. An example would be providing missing PCM or analog TMATS attributes that are needed to properly use some FLIDAS features. If you don't have a matching TMATS file the process would be to export the real TMATS setup from a CH10 file, correct it and reopen the CH10 source with the corrected external TMATS file. For live data streams you need to create a short recording to have a CH10 file to export the TMATS from.

For live data streams TMATS overriding has the additional benefit that the live data stream is instantly created even if the data stream is not yet received. In conclusion you can prepare the [live processors](#) you are going to use while you are still offline and be ready for processing when the data stream starts.

TMATS overriding can be done in the [CH10 file loading wizard](#) or when [creating the live data stream](#). You can use the file extension filter options in the TMATS file selection dialog to control which types of TMATS files you will see. ASCII and XML files are supported and the type is auto detected based on the content rather than the file extension.

If TMATS overriding has taken place, it will be shown in the [General CH10 Source Information Properties](#).

Only the configuration is replaced. The TMATS packets shown for the CH10 source for example in the [Packet View](#) will still be the real ones and if you export the TMATS setup from a CH10 file you will export the real setup.

You can permanently override the TMATS setup of a CH10 file with the [Replace TMATS in CH10 Exporter](#).

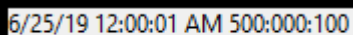
[TMATS Default Preferences](#) may also be an alternative to TMATS overriding.

4.17 Gauges

Gauges are used to visualize current parameter values on an [instrument panel](#) or control the replay of parameters. Their general operation is described in the [Instrument Panel View](#).

The following subchapters describe the different gauges available.

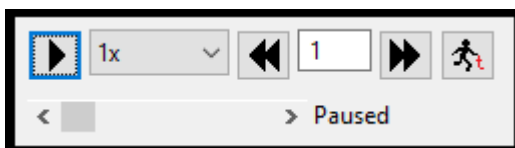
4.17.1 Clock



This gauge does not use a parameter. Instead it takes the highest processed CH10 packet time (or packet entry if decoded) of the active live data stream and displays it. For CH10 files the time will progress (if replay is running) even if there is a gap without data. For live data streams the time will stall in this case until new data comes in.






You can select a [time representation](#) from the edit mode context menu to change the time display format.

4.17.2 Replay Control




This gauge is needed if you want to use instrument panels with a CH10 file rather than a live data stream. It allows controlling the replay of the file. The current position of the replay will be the same across all instrument panels of that CH10 file. You can have a replay control gauge on an instrument panel even with a live data stream but the buttons will be disabled. This way you can share the same instrument panel for different use cases.

The control elements have these functions:

-  starts the replay if paused. The button turns into a pause button afterwards.
-  pauses the replay if running. The button turns into a play button afterwards.
- The drop-down box allows selecting a speed up or slow down of the replay. If you replay fast, you might overload the system, so take care. Overloading will be indicated like described for the active data stream tasks in the [Repository View](#)
-  moves the replay position forwards. The field before it can be edited to set up the number of seconds to move (fractional number allowed). Keeps it running or paused as before.
-  moves the replay position backwards. The field after it can be edited to set up the number of seconds to move (fractional number allowed). Keeps it running or paused as before.
-  Opens a dialog to choose a new replay position from [time markers or recording events](#) which include start and end of file. Keeps it running or paused as before.
- The scrollbar can be used to freely move the replay position in the file. The timestamp will be shown. Keeps it running or paused as before. The scrollbar can only be used to set the position not to see the position. I.e., the scrollbar will not update its position while the replay runs.
- A text to the bottom right will show the current replay status like Running, Paused, Searching and At End.

Whenever you move the replay position, the current values will reset to invalid (not measured) and then refresh from the next valid value in the replay. In pause mode the reset will be postponed until you run the replay again. So, FLIDAS will not search for the last valid values before the new position.

4.17.3 Simple Value



This gauge accepts any parameter and will just display the value as text.

Invalid values will be red.

The parameters name and unit will be displayed before the value.

4.17.4 Table

Name	Value	Unit
Pitch Angle	-4.998779	deg
Roll Angle	19.995117	deg
WindSpeed	141	knots

This gauge accepts any type and number of parameters and will display the values as text in a table.

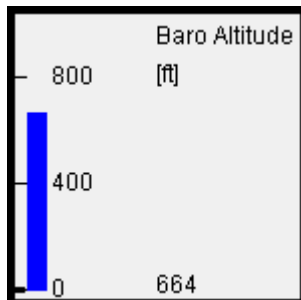
There are columns for parameter name, value, unit, mapping ID, path and comment. You can resize the columns and hide columns by reducing the size to 0. You need to drag the space between the column headers to resize. You can show or hide the column headers in the edit mode context menu when you select **Show Table Header**.

When you drag parameters on the table all parameters will be added to the table. To remove parameters select them and either use the keyboard shortcut **Del** or select **Delete Selection** from the edit mode context menu.

Invalid values will be red.

The parameter name column will have a red background if the parameter has not been found (like an "unlinked parameter" in a parameter set).

4.17.5 Bar Graph

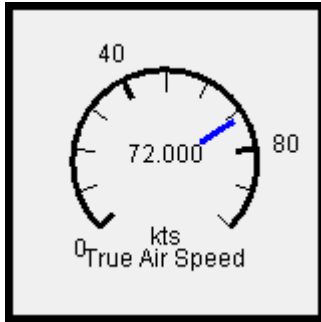


This gauge accepts one numeric parameter. It will visualize the value in a vertical bar with a scale in the configured range. The value will also be displayed on the bottom.

If the value is outside the configured range, a small red bar will only be displayed on top (exceedance) or bottom (undercut). For invalid values, there will be no bar at all and the value display on the bottom will be red as well.

The parameters name and unit will be displayed on the top.

4.17.6 Dial



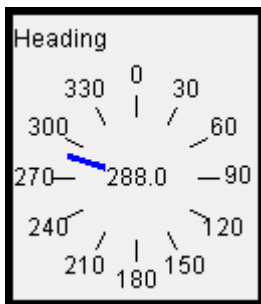
This gauge accepts one numeric parameter. It will visualize the value in a dial with a scale in the configured range. The value will also be displayed in the middle.

If the value is outside the configured range, the needle will be red and displayed on the left (undercut) or right (exceedance) end of the scale. For invalid values, there will be no needle at all and the value display in the middle will be red as well.

The parameters name and unit will be displayed on the bottom.

The scale as well as the displayed value will be divided by factors of 1000 (or multiplied by factors of 1000) to result in better readable values. The factor will be displayed with the unit on the bottom.

4.17.7 Compass Rose

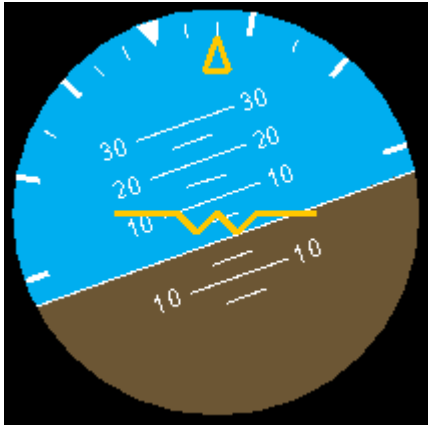


This gauge accepts one numeric parameter. It will visualize a value on a compass rose with a fixed range of 0 to 360 (degree). The value will also be displayed in the middle.

Values outside the range of 0 to 360 will be handled accordingly, i.e., 720 will result in the same needle position than 360 and -90 will have the same position than 270. For invalid values, there will be no needle at all and the value display in the middle will be red as well.

The parameters name will be displayed on the top. The unit will not be displayed and will be treated as degree.

4.17.8 Attitude Indicator



The attitude indicator or artificial horizon displays pitch and roll values as it would be displayed in an aircraft.

Both parameters must be numeric and in the range of +/- 180 degree. If any parameter value is invalid, it will be displayed in the 0 position but the scale will not be displayed.

4.17.9 Altimeter

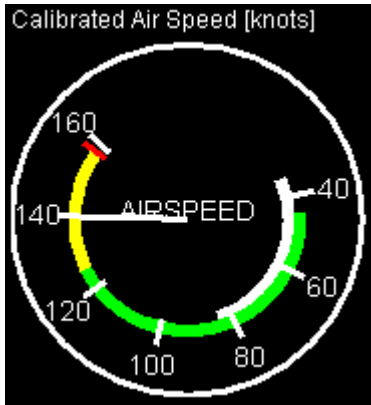


The altimeter displays altitude with three pointers as it would be displayed in an aircraft. It can optionally also display the barometric correction applied in the altimeter.

The altitude parameter must be numeric. Its value divided by 100, 1000 and 10000 will be applied to the appropriate pointers. So, there is no limitation on the range from technical side. Although the instrument normally uses feet you can assign an altitude in meters as well as the instrument just shows the raw number without any unit. If the altitude parameter value is invalid, the pointers will not be shown.

The baro correction parameter must be numeric. Its numeric value will be shown in a box to the right of the instrument. It is just a display and no calculation is applied. So it is irrelevant if you assign a millibar value or inches of HG. Values above 100 will be shown without fractional digits like 1013 (assuming millibar) and values below 100 with two fractional digits like 29.92 (assuming inches of HG). If the parameter value is invalid, the numeric value will not be shown.

4.17.10 Airspeed Indicator



The airspeed indicator displays airspeed as it would be displayed in an aircraft.

The airspeed parameter must be numeric. Although the instrument normally uses knots you can assign an airspeed in meters per second, miles per hour or kilometers per hour as well as the instrument just shows the raw number without any unit. If the airspeed parameter value is invalid, the pointer will not be shown.

You can configure some things in the edit mode context menu if you select **Configure Speeds**. You can configure the range and spacing of the numbers displayed on the scale as well as the scale ticks and the V-speeds that are needed to define the flap operating range, normal operating range, caution range, never exceed speed and the two speeds for multi-engine aircraft. The V-speeds and scale tick configuration are optional and lead to the presence or absence of the colored markings or ticks.

The pointer will linearly progress over the defined range (from the top clockwise until 30 degrees before the top). It will keep linearly progressing afterwards for 20 degrees above the defined range and then block in a 350 degree position for higher speeds.

4.17.11Map



A map gauge shows the position of an object based on its latitude and longitude. It can optionally also show the direction of the object.

See also the general description of maps in FLIDAS.

All three parameters must be numeric. Latitude must be in the range +/- 90 degree and longitude +/- 180 degree. Direction must be in the range of 0 ... 360 degrees if used.

The map shows the current latitude/longitude as a white symbol with black outline. The center of the symbol is the position. If there is a valid track angle, the symbol will be a triangle pointing in that direction. Otherwise it is a square. If no valid position is available the symbol disappears.

The map can be zoomed in and out with the mouse wheel if the mouse pointer is over the map or with the + and - key if you click in the map with the mouse before. You must be in in edit mode to zoom. The zoom level will be saved with the instrument panel.

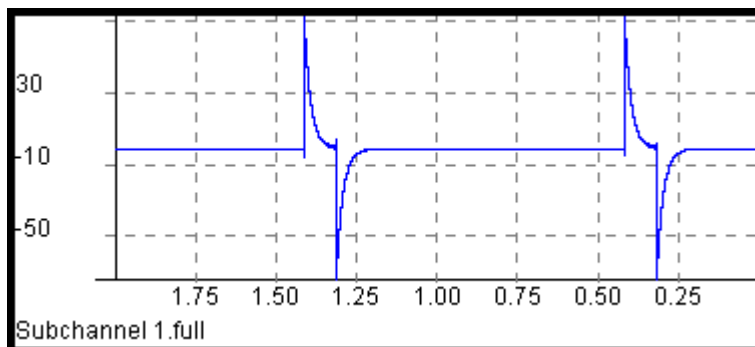
The lower part of the map contains a scale.

The edit mode context menu contains two additional options:

- **Select Map** allows selecting the displayed map. The option will be saved with the instrument panel.
- **Configure Map** allows changing additional things like the type of scale shown on the map and additional options depending on the map
- **Zoom In/Zoom Out** zooms into or out of the map like described above for the mouse wheel and keyboard.

Note that if you select a user defined map the display may show a white canvas only until the current position is set since the current position may not be in the user defined map.

4.17.12Strip Chart



This gauge shows a graph of a single numeric or enumeration parameter over time.

Enumeration parameters will only display their raw value in the graph.

A line on the bottom shows the parameter name. The graph itself shows samples of the parameter over time. *Note that if you use analog parameters this is a place where you can use full live decoding parameters.*

A time axis in seconds runs from the bottom right (time of last sample) to the left (i.e., the past). A value axis on the left shows the maximum value on top and the minimum on the bottom. Maximum and minimum can be set on the edit mode context menu.

Invalid values will appear in red on the bottom axis.

The edit mode context menu has some additional options:

- **Time span** sets the time span in milliseconds to display in the graph. Using a too long time span for a high frequency measurement will massively degrade performance. Several ten-thousands of measurements should not cause a problem in normal systems. A too short time is not useful since the screen plot will only be updated in intervals so you would only see some segments of the values.
- **Auto Scale** can be enabled or disabled to influence the scale on the value axis. If enabled the axis ranges between minimum and maximum value received since selecting the parameter for the display. If disabled it will use the configured minimum and maximum.

4.18 Maps

Several features of FLIDAS use maps and the default maps are described here. You can also [add own maps](#).

The map projection for the default maps is an equidistant azimuthal projection. This means that angles and distances from the center of the map are displayed correctly. Areas to the edge of the screen appear larger than in the middle. These effects matter only significantly if you show a large area. North will always be to the top.

The maps currently available are public domain and taken from the [Natural Earth Project](#). The maps serve the purpose of giving you a reference when analyzing your flight data. Data Bus Tools does not necessarily share the same political view as the map data authors about the location of borders.

The coarse maps are based on the 50m scale available on the Natural Earth Project. These show a good performance. Maps with more details tend to become slower when shifting or re-scaling the map (only then) and require larger downloads. We offer additional maps for those who require them. These can be individually built (also using a bitmap-based map instead of vector maps).

There is one prepared set of maps that you can order as a free option (please order directly with the base product). It is based on the 10m scale maps from the Natural Earth Project. Besides the better scale it also adds the borders for provinces in most countries. Since this world map can become a performance issue the package also contains a separation of the data into continents (simply suppressing the rest). The mechanism used for this is not completely reliable (due to overseas provinces) but the whole world map is always present as a fallback.

Some maps allow configuring what they show (like if rivers, cities or province borders are displayed) and this can impact the performance. Also, you can configure which scale they show (based on km, nm or mi).

4.19 Data Streaming

4.19.1 Create a Live Data Stream

You can create a live data stream when you click **File > Create Live Data Stream** or use the keyboard shortcut **Alt+L**. The live data stream is part of the free FLIDAS version but not all features for working with live data streams are.

The data stream must use the UDP transfer method as defined in IRIG 106 chapter 10. It must also include a time channel.

You must configure the UDP connection on the UDP Properties Wizard Page.

You must configure the UDP data streaming format on the CH10 Data Stream Format Wizard Page.

You can either select an external TMATS file (so called TMATS overriding) to define the data in the data stream or take the TMATS from the data stream. In the latter case all incoming data will be ignored until FLIDAS finds a TMATS packet on the data stream. If you use TMATS overriding you have the benefit that you can prepare all the live processors in advance before any data is received.

The configuration **Synchronize channels** chooses if you want to synchronize the timing of the different CH10 channels. Different CH10 channels may have a different delay between the occurrence of the recorded data and the availability of the data on the network. This is allowed to be up to one second and can be different on each channel. To compensate this FLIDAS can buffer and delay all incoming data for one additional second and process them in time order. This way two events that occurred at the same time on different channels will be visible almost at the same time. If this is not important for you, you can leave this out and save system resources and achieve a lower delay. For example, if you are just interested in the data of one CH10 channel or you just want to verify proper operation of separate FTI equipment you might not need the synchronization. Without synchronization the delay is dominated by the recorder delay which may be up to a second. With synchronization FLIDAS adds another second leading to about two seconds. Of course, there are other delays to add like network and untimely processing in this non real-time environment which are normally negligible compared to the other delays.

Another group of settings is like those when loading a CH10 file:

- **Check for data checksum errors:** If this is deactivated, then packets with data checksum errors will be treated like valid packets. Omitting the checksum check improves processing speed.
- **Ignore MSIDs:** Normally FLIDAS will support MSIDs (see distributed CH10 multiplexer sources). This means that it will expect a separate time channel for each MSID for RTC timing. If you choose to ignore MSIDs, FLIDAS will treat the live data stream as if all channels belong to the same MSID 0. If your files do not use MSIDs at all, both modes should be the same. But if you use MSIDs and there is only one time channel and the RTC is synchronized between sources, this option should be used.
- **Ignore secondary header time:** FLIDAS will decode CH10 packet time from the secondary header if available. Otherwise it will use regular RTC timing. Using this option you can use the RTC timing instead of secondary header time. Note that a packet with RTC sync error will neither decode RTC time, nor secondary header

ERTC time but use a fallback time. If your CH10 sources do not use secondary headers (which is usually the case) this option makes no difference.

- **Start year of recording (4 digits):** Defines the year that the data stream belongs to (in case this is an older recording being replayed). This is used when the timestamps inside the live data stream does not contain year and month information. The default value is the current year. It is further corrected by the TMATS "original recording date" when processing timestamps if you check the option **Try to get start year from TMATS**.
- **Maximum number of unexpected errors before termination:** Controls how many unexpected errors are allowed to occur during the reception of the live data stream. If this threshold is met, the reception will terminate. Only errors not explicitly handled by FLIDAS count against this threshold.
- **Data Source ID:** Sometimes the TMATS data in the CH10 file might contain several R-x sections describing the recording hardware setup. In this case you can enter the data source ID (R-x\ID, G\DSI-n) to select which one FLIDAS shall use. The value is case insensitive. If you enter nothing, FLIDAS will take the R-x group with the lowest "x" (which is the only one most of the time anyway).

You can use CH10 data type filters to suppress some data from being processed by the live data stream.

You can use time channel offsets to configure timing offsets if you receive data from distributed CH10 multiplexer sources.

FLIDAS will skip all incoming data until TMATS has been initialized (with TMATS override or from data stream) and a first valid time packet has been received.

Once the live data stream is initialized you can see it in the Repository View where you can invoke some actions on it.

Once a live data stream has tuned into an incoming data stream you cannot reuse it for another data stream. This means if you stop your device providing the data stream and start it over you must also create a new live data stream.

The live data stream is also visible as data stream task in the Repository View. You can stop it there to abort the background processing. However, the live data stream will still be visible in the Repository View and all connected views will still exist. There is just no new data coming in anymore. You can use this if you still want to see the data collected in the live processors. The normal way would be to close the live data stream in the Repository View which also closes the data stream task and the associated views.

The Data Streaming Preferences offer more configuration options for the live data stream.

4.19.2 Record a Data Stream

You can record a data stream when you click **File > Record Data Stream** or use the keyboard shortcut **Alt+R**. The recording is part of the free FLIDAS version.

The data stream must use the UDP transfer method as defined in IRIG 106 chapter 10.

You must configure the UDP connection on the [UDP Properties Wizard Page](#).

You must configure the UDP data streaming format on the [CH10 Data Stream Format Wizard Page](#).

Another configuration is whether you want to record everything or want to synchronize to TMATS and time. Since a valid CH10 file must start with a TMATS packet followed by a time packet this mode will skip all incoming data until TMATS and time has been received and saved. Only then other packets will be recorded.

Once the recording is running you can see a [data stream task](#) in the [Repository View](#). To finish the recording you must go there and stop it.

The [Data Streaming Preferences](#) offer more configuration options for the recorded data stream.

Note: The recording will not modify the incoming TMATS in any way to create a valid file (like updating the recording date). Also, TMATS included in streaming configuration packets will stay as is and not be converted to setup record packets. FLIDAS can read these files properly but they are not formally valid.

4.19.3 Publish a Data Stream

You can [publish a CH10 file as data stream](#) if you right-click it in the [Repository View](#) and select **Publish Data Stream**. The publishing is part of the free FLIDAS version.

The data stream will use the UDP transfer method as defined in IRIG 106 chapter 10.

You must configure the UDP connection on the [UDP Properties Wizard Page](#).

You must configure the UDP data streaming format on the [CH10 Data Stream Format Wizard Page](#).

You can [select which CH10 channels to include in the publishing](#). By default this is everything except computer-generated data channels.

You can select to **Always send TMATS** to make sure the TMATS record is sent in the beginning no matter if channel 0 is selected or not. In this case the first group of TMATS packets are included from channel 0. Other packets from channel 0 are sent in the normal time order if the channel is selected. **Always send time** does the same for a single time packet of all time channels of the selected channels MSIDs (see [distributed CH10 multiplexer sources](#)).

You can select to **Loop** the transmission. This will continuously repeat the same sequence of CH10 packets after each transmission. There will be a gap after each transmission to make the start of retransmission detectable. The duration of the length can be configured but will not be respected precisely. Note that the repetition of the data will cause the time to jump back what FLIDAS considers invalid. Don't loop if you want to receive the live data stream with FLIDAS.

You can select a time slice for the publishing as well. Note that even if you include channel 0 for publishing, the first group of TMATS packets might not be in the time range and therefore not be transmitted. If you need them use the "always send TMATS" option.

Once the publishing is running you can see a data stream task in the Repository View. This task will end when the publishing of the file has finished. You can also stop the publishing in the Repository View before.

The Data Streaming Preferences offer more configuration options for the published data stream.

Note: The publishing will reorder the CH10 packets from the CH10 file by time and try to achieve a real-time replay as close as possible. As an exception initial TMATS packet(s) are not transmitted at their original timestamp but directly before the first dynamic packet.

Note: This functionality was developed to allow testing FLIDAS data stream reception. It was not tested with other data stream reception soft- or hardware although at least one customer uses it successfully.

Note: To test the data streaming functionality you can publish a data stream on the same computer that you receive it. You must configure the local host IP address 127.0.0.1. If you are checking the performance, remember that the performance is affected also by the publishing which is unnecessary if you receive a real data stream. Since publishing is part of the free version you can of course also use a second computer with FLIDAS for publishing.

4.20 Analyzing Data Stream Problems

Receiving and processing a CH10 UDP data stream is an error-prone task because it is influenced by many factors compared to simply opening a CH10 file. This section shall give you an overview of what is happening and at which places you can look to solve problems. The section will not explain everything in detail but rather expect you are already familiar to some level with FLIDAS.

4.20.1 Data Flow

1. The CH10 data will be transmitted by your CH10 hardware. It must be properly set up to transmit the data to a network.
2. The data is flowing on your network, possibly being send through network switches and routers to the computer running FLIDAS.
3. In FLIDAS the data is received by a component internally called "data stream receiver". It will receive UDP packets, combine and reorder them to blocks of data representing CH10 packets.
4. The data stream receiver then hands over the collected data blocks to either:
 - a. A data stream recorder. This is the simplest case and the blocks are just saved to a file or (depending on the options) filtered out until a TMATS and initial time packet has been saved.
 - b. A CH10 live data stream. Depending on the options it will either directly process the incoming data as CH10 packets or will reorder and delay them to time sequence if the option is set. Also, packets may be dropped until TMATS was initialized and until a time source for the packet was found on the stream.

4.20.2 Analysis: Try with Minimum Setup

To analyze problems you should try with a minimal and simple setup and if this is running expand it step by step to your real setup locating the source of the problem.

The simplest setup would be:

- For the transmission skip the CH10 hardware. Use FLIDAS to publish a recorded CH10 file and publish to IP 127.0.0.1. This skips the CH10 hardware and the whole network problems as well. This mode is described in the data steaming tutorial. Note that if you are repeating tests, you will jump back in time every time you replay the file again. A data stream in FLIDAS does not support jumping back in time so you must close the existing receiving data stream and create a new one every time. If your old replay is still running you must also close this.
- For the reception use a data stream recorder and use the mode without synchronization to TMATS and time. This will just save the data of the reconstructed CH10 packets including all checksum errors etc.
- Use the UDP data stream format 1 (i.e., enter no source ID) for reception and transmission. This format is well established.
- Reset your Data Streaming Preferences to the default values (exporting your current setup before or taking notes of it)

Note that these features are also available in the free FLIDAS version. You could setup a separate free FLIDAS in a new directory that uses the default setup, avoiding any possible misconfiguration.

To extend the setup step by step you could:

- Move the FLIDAS publishing to another computer and use the network for transmission. Note that the free version of FLIDAS can publish CH10 data streams.
- Use your real CH10 hardware with a direct network connection (i.e., direct cabling) and an IP unicast (i.e., the IP of the target computer, no broadcast, no multicast. Depending on your network setup the unicast might not be possible if recorder and computer are configured to different subnets). Try to stay with a UDP data stream format 1
- Use your real CH10 hardware on the real network (i.e., possibly including switches / routers).
- Switch to IP multicast (only if needed in your final configuration, otherwise avoid multicast). Then also select the network interface instead of relying on a default.
- Use the FLIDAS recording mode that synchronizes to TMATS and time packets instead of the simple mode.
- Use a live data stream without synchronize channels option instead of the recording.
- Use a live data stream as before but turn on synchronize channels option.
- Change Data Streaming Preferences step by step to your desired settings
- If you need UDP data stream format 3:
 - Change to UDP data stream format 3 on transmit and receive if you want to use this format. Only transmit one source and leave the source empty on the FLIDAS receive side for auto selection
 - Configure FLIDAS reception to the specific source you want to receive
 - Enable transmission of several sources at the same time if required

4.20.3 Analysis: What to Check

General checks:

- Make sure you have the error log view open. Some potentially repetitive errors are just logged there and do not pop up as errors. Critical situations to look for are of course the time when you start the reception or transmission.
- Watch the "Status" column in the Repository View for any live data streams. Also, be aware of the details of each Data Stream Tasks that you can see if you right-click it and select **Status**
- Consider if your CH10 hardware or computer had any recent hard-, firm- or software changes that might be related.
- Look for detected overloading of data stream tasks.
- Look for general overloading of your computer for example using the Windows Task Manager.

Then you should verify if you receive anything in FLIDAS. To do so use the simple setup with the FLIDAS recording extending it when necessary to reproduce the problem. In the Repository View you can see the recording data stream task below the item **Other active data stream tasks**. Look for its status. If you see UDP packets are listed to come in then the network connection works in general. If not:

- Check the target IP of the published CH10 data stream. As said unicast is most reliable
- Check the target UDP port of the published CH10 data stream and the FLIDAS port configuration of the reception
- Check if all network elements are working correctly (cables attached, switches configured etc.). If you can, use a network monitoring software like Wireshark then monitor if the UDP traffic on the selected port arrives on the computer.
- Check if any firewalls in the network or on the local computer block the traffic to the computer or to the FLIDAS application.
- Make sure you use IPv4. IPv6 is currently not supported by FLIDAS.

If you receive UDP packets but no CH10 packets are created in the status of the FLIDAS recorder, then check if the UDP data streaming format is the format you expect and the source ID for a format 3 is correctly chosen.

If you receive UDP packets but you see a lot of UDP packets dropped in the status of the FLIDAS recorder, then either:

- You have configured FLIDAS to drop UDP packets not arriving in sequence. Unless you are transmitting from and to FLIDAS on the same computer or you have a very controlled network environment this is not a good idea. Disable this option in the Data Streaming Preferences.
- Your network is quite overloaded and not able to carry the data stream through the network. FLIDAS is missing UDP packets resulting in incomplete CH10 packets. Depending on the amount you might still be able to see data passing through. Try to reduce the CH10 load on the network, the load from other network use or the load on the computer that is not able to handle the incoming data.

If you can record CH10 data but cannot setup a live data stream then the data might be corrupted or essential data (TMATS, time) is missing. Take a recording without synchronization to TMATS and time and open the recorded file.

- Check if there are a lot of errors (Repository View > Show Invalid Packets). Checksum errors might indicate that there is corruption on the network or dropped UDP packets belonging to a part of the CH10 packet (see above answer for this problem).
- Check if it contains TMATS packets. Look for this in the Channel View on the TMATS channel. In a data stream recorded like this it might not be the first packet. If there is no TMATS at all you must either configure your hardware to transmit TMATS packets periodically (like every 30 seconds) or you must set up live data streams in FLIDAS with an external TMATS file.
- Check if it contains time packets. Look for this in the Channel View on the time channel. FLIDAS needs at least an initial time packet to set up a live data stream. This must also be valid (i.e., no checksum error, packet sync error, RTC sync error etc.). Also, check if the time packets indicate a reasonable time (e.g., not a time from 1970 or in the future). If several sources are involved, you need to understand that different channels might use different time channels. So, packets might even be dropped if a time channel is present but it is not the right one. See [distributed CH10 multiplexer sources](#) for details. You can use the channel view to look up which time channel is used for which channel (unless the channel uses secondary header timing). You can also see there if the time channels received packets. Note that the use of MSIDs and secondary headers could be suppressed when opening the recorded CH10 file or creating the live data stream.
- Make sure that you have not configured any CH10 Data Type Filters to suppress Packets on the Live Data Stream
- Try to make a recording with synchronization to TMATS and time. If you can, check in the recorded TMATS if the "original recording date" (R-x\RI4) is a current date.

If your reception is running well, you should now be able to see CH10 packets coming in on the [Channel Activity Live Processor](#) of a live data stream.

If you are still having problems, ask for support and try to provide information on how far you succeeded with this. Consider if you can provide data like the recorded CH10 data (like one second) and/or a Wireshark network recording (limited to the relevant port). In any case don't send confidential data.

4.21 Live Processors

See the [live processor concept](#) for a general explanation of live processors.

To start a live processor, right-click a channel or subchannel in the [Channel View](#) and select it from the **Live Processor** menu. The configuration is then [like the exporter configuration](#).

The available live processors are shown in the subsections of this chapter.

4.21.1 Channel Activity Live Processor


This free live processor displays statistics on selected channels of a live data stream.


You can start it for selected channels in the Channel View. You can also display the activity of all channels if you start it from the Repository View. Note that channels not detected until the start of the live processor will not be added later.

The view will contain a table with a line for each selected channel and a summary line on top separated by a black line. The table will display the following data limited to the time since the live processor was started:

- Number: The number of the channel
- Name: The name of the channel
- Type: The CH10 data type of the channel
- Valid packets: The number of valid packets received on that channel
- Data size: The sum of the bytes of the received packets on the channel. This is the size of the CH10 packets data section. It includes invalid packets but not lost ones.
- Data bit rate: The average bit rate in MBit/s calculated from the data size and the time span from the first to the last CH10 packet processed by the live processor so far
- Invalid packets: The number of invalid CH10 packets on the channel. This includes data checksum errors (if checked), secondary header checksum errors, RTC sync errors and data overflow errors
- Lost packets: The number of CH10 packets lost on the channel. (This is an estimation since 257 lost packets look identical to 1 lost packet etc.)
- Time: The last CH10 packet time received on that channel (for the summary: the highest time of all channels)

If there was any activity on the channel within a period of about two seconds, the line will be green. If there was an error (invalid or lost packet) on the channel in that time period, the line will be red instead.

There is a  pause button to pause the updating of the table. If paused the table will use an italics font.

You can create time markers by pushing the  button on the upper right. They will refer to the current time of the live data stream when you clicked at the icon or the time you paused if the pause mode is on.

4.21.2 MPEG TS Replay Live Processor (UDP Streaming)

This live processor plays an MPEG video (MPEG transport stream) from a live data stream using an external video player. It is only available in the Data or Parameter Streaming Version.

The live processor is like the MPEG TS Replay (UDP streaming) Exporter with the exception that you cannot configure a delay for the replay. Instead the MPEG transport stream will be streamed out as soon as available. Your external video player might introduce additional delays for example buffering a period of time before starting replay. For the VLC player you can configure this in the advanced settings for the input ("network caching").

To stop the video replay you must close the data stream task in the Repository View. It will also end if you close the live data stream itself. Just closing the video player does not stop the video streaming as FLIDAS has no information if your external video player has closed.

4.21.3 UART Replay Live Processor (UDP Streaming)

This live processor packages UART raw data from a live data stream into a UDP stream. It is only available in the [Data or Parameter Streaming Version](#). This is a simple solution to externally process live UART data without the need to directly implement the IRIG 106 CH10 UDP protocol.

You can configure a [target IPv4 address and UDP port](#).

Data will be transmitted as fast as possible when it is coming in. The complete UDP payload is the sequence of UART data words (with a 0 padding bit on the MSB for 7-bit UART data words). Data blocks with parity error will not be streamed. Note that CH10 recordings do not detect message borders, so the data block can start anywhere in a message and end anywhere in the same or following message(s). Also, FLIDAS will slice the data into UDP sized blocks if necessary.

To stop the replay you must close the data stream task in the [Repository View](#). It will also end if you close the live data stream itself.

4.21.4 PCM Throughput Replay Live Processor (UDP Streaming)

This live processor packages PCM throughput raw data from a live data stream into a UDP stream. It is only available in the [Data or Parameter Streaming Version](#). This is a simple solution to externally process live PCM data without the need to directly implement the IRIG 106 CH10 UDP protocol.

You can configure a [target IPv4 address and UDP port](#).

Data will be transmitted as fast as possible when it is coming in. The complete UDP payload is just the raw PCM bits.


To stop the replay you must close the data stream task in the [Repository View](#). It will also end if you close the live data stream itself.

4.21.5 PCM Streams Live Processor

This live processor shows the live status of one or more incoming PCM streams. It is only available in the [Data or Parameter Streaming Version](#).

The live processor will show this information about each selected PCM channel and continuously update it:


- Channel ID
- Channel name
- Minor frames: The number of properly decommutated minor frames. Note that this requires some decommutation settings in TMATS for PCM throughput mode.
- Lost packets: The number of lost CH10 packets from this channel.

You can create [time markers](#) by pushing the  button on the upper right. They will refer to the current time of the live data stream when you clicked at the icon.

4.21.6 PCM BERT Live Processor

This live processor verifies if an incoming PCM throughput live data stream conforms with a pseudo random bit stream for a bit error rate test (BERT). This is the type of test data often created by bit error test equipment. It is only available in the [Data Streaming or Parameter Streaming Version](#).

The general output is described with the [PCM BERT Viewer](#) but the live processor updates the current values while the live data stream comes in.

You can create [time markers](#) by pushing the  button on the upper right. They will refer to the current time of the live data stream when you clicked at the icon.

4.21.7 CH10 from CH7 Publishing

This live processor takes the IRIG 106 chapter 10 content from a PCM channel with [chapter 7](#) data in a live data stream and publishes it as IRIG 106 chapter 10 UDP stream. It is only available in the [Data or Parameter Streaming Version](#).

While you receive the live data stream with the CH7 data on one UDP port you can set up a second live data stream (which adds to the delay) that receives the CH10 UDP stream that is created by this functionality on another UDP port. Remember that you need TMATS information for this stream as well.

The live processor uses the [IRIG 106 Chapter 7 Properties Wizard Page](#).

For setting up the UDP publishing the same things like for normal [data stream publishing](#) apply except that the packets are not reordered but sent as soon as decoded from CH7. An active publishing can also only be ended and seen in the [Repository View](#).

4.21.8 TmNS from CH7 Transmission

This live processor takes the TmNS (IRIG 106 chapter 24) messages from a PCM channel with [chapter 7](#) data in a live data stream and transmits each message in a UDP packet. It is only available in the [Data or Parameter Streaming Version](#).

Each TmNS message goes into one UDP packet (which may be subject to IP fragmentation). TmNS messages which are too big will be suppressed (i.e., > 65507 bytes)

You can configure a [target IP address and UDP port](#).

The live processor uses the [IRIG 106 Chapter 7 Properties Wizard Page](#).

4.21.9 CH7 Live Processor

This live processor shows live statistics on an incoming [IRIG 106 chapter 7](#) PCM stream. It is only available in the [Data or Parameter Streaming Version](#).

The CH7 live processor sticks to the encapsulation packet level of CH7. It does not aggregate them to source packets or detect errors on higher levels. If you need this, record the stream and use the [CH7 Communication Summary](#) on the CH10 file for a deeper analysis.

A table will show one line for each stream ID. The information on each stream ID can be expanded to show a summary broken down for each content type. The columns of the table show the following:

- **Stream ID / content type:** The 4 bit stream ID for the stream ID lines or the content type (like CH10 packets, MAC frames) for the content type summary lines
- **Version:** The version of the stream (only on stream ID lines)
- **# TP:** The number of the transport packets (formerly packet telemetry data frames) (only on stream ID lines)
- **# LLEP:** The number of low-latency encapsulation packets (formerly low-latency packet telemetry data packets)
- **# Standard EP** The number of standard (i.e., non low-latency) encapsulation packets (formerly packet telemetry data packets)
- **Bit rate:** The bit rate of processed EP's payload in MBit/s. This misses EPs broken due to massive protocol errors.
- **Avg EP payload size:** The average size of EP payloads (low-latency and standard) in bytes
- **# CH7 out of sync:** The number of errors that have been detected when analyzing this stream ID (not for content lines). These are internal FLIDAS statistics but indicate that the decoding might not run well. This can especially be seen when bit errors occur on PCM reception and the decoding cannot safely detect the beginning and end of EPs anymore. Stream lines with this error will be shown in red.

A line above the table shows more that applies to all stream IDs:

- **Reset:** Push this button to reset the statistics and start with fresh data. For example, the throughput will be the average for the whole time since the last reset (or start)
- **Minor frame sync losses:** The number of times the frame sync has been lost (only displayed in PCM throughput mode). This will also cause decoding inaccuracies on the stream IDs.
- **CH7 segments byte length:** The total byte size of all CH7 segments within one minor frame

4.21.10 Analog Audio Live Replay

This live processor replays audio from a single analog subchannel to the local audio hardware.

The playing relies on the availability of local audio hardware that supports playing with the given configuration.

The replay has no visual representation and can only be stopped by stopping the associated data stream task in the [Repository View](#).

Often the local audio hardware is less precise than the analog recording and does not replay exactly at the sample rate it is commanded to. To keep the replay synchronous with the CH10


stream FLIDAS might need to drop or duplicate samples from time to time which may lead to small audible sound artifacts. The sound quality might be less than if you export the data from a recorded stream.

To keep the audio hardware buffers filled with audio for replay, FLIDAS creates an additional delay to the CH10 stream. Currently this is around 300 ms on top of the regular delays but may change without further notice.

4.21.11 Raw Data Live Processors

The commercial raw data live processors visualize raw data of a specific CH10 data type from a live data stream and continuously update it. They also allow plotting a graph of selected raw data over time. There are different raw data live processors but all have some things in common.

There is a  pause button to pause the updating. If paused tables will have another color.

You can create time markers by pushing the  button on the upper right. They will refer to the current time of the live data stream when you clicked at the icon or the time you paused if the pause mode is on.

The view consists of two areas and you can redistribute the space between them by dragging the area in between with the mouse.

4.21.11.1 The Raw Data Area

The upper area of the view somehow represents the raw data or protocol level of the specific CH10 data type. This is individual in each raw data live processor.

4.21.11.2 The Graph Area

The lower area shows a graphical plot of a single raw data word over time if you selected one. A time axis in seconds runs from the bottom right (time of last sample) to the left (i.e., the past). A value axis on the left shows the maximum value on top and the minimum on the bottom. Invalid values will appear in red on the bottom axis

On the right is a two page configuration area for the graph. The configuration will be displayed in green when it is active. As soon as you change something it will either turn red or yellow. Red means the value is invalid. Yellow means it is valid but has not been applied yet. If you click the **Apply** button you will also see error messages for any invalid field in red. The **Disable** button will disable the plotting at all. Each configuration page separately applies its settings.

A word filter page allows selecting the raw data word to filter. It is always specific to the type of raw data live processor.

The **Curve Setup** configuration page has these options:

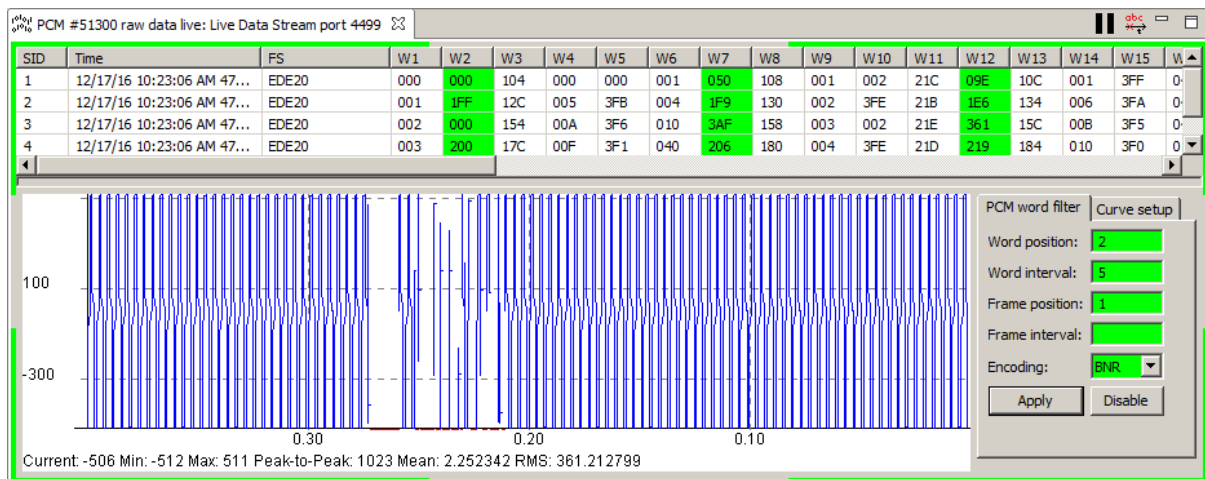
- Define the time span of the display in milliseconds. Using a too long time span for a high frequency measurement might have a huge performance impact. Several ten-

thousands of measurements should not cause a problem in normal systems. A too short time is not useful since the screen plot will only be updated in intervals so you would only see some segments of the values.

- Choose the range of numbers on the value axis (i.e., scale on the value axis):
 - Auto: Axis ranges between minimum and maximum value received since applying the current raw data word filter
 - Full range: Axis ranges between minimum and maximum possible value of the raw data word
 - User: Define the minimum and maximum by entering them into the min and max fields
- Enable statistics for the selected parameter for the measurements in the time span. The statistics will then show at the lower border of the plot:
 - Current value
 - Minimum
 - Maximum
 - Peak-to-peak value
 - Mean value
 - Root mean square (RMS) value
- You can choose how to label the time axis. Relative time will display the seconds that have passed since the sample while absolute time will print time stamps (without date) on the axis. The labels will include milliseconds when needed.

4.21.11.3 PCM Raw Data Live Processor

This raw data live processor shows raw data from an incoming PCM stream and continuously updates it. It displays a full major frame with its minor frames and allows a graphical plot of a selected PCM word. The live processor is only available in the Data or Parameter Streaming Version.



4.21.11.3.1 The Minor Frame Area

The upper area of the view shows a table with one row for each minor frame of the current major frame. If you have not defined a subframe ID counter in TMATS this area will just show the most recent minor frame. If you have several subframe IDs it will use the first one. The columns of the table show:

- The subframe ID if a subframe ID counter is defined
- The time of the start of the minor frame
- The hexadecimal frame sync pattern
- A column with the hexadecimal content of each word of the minor frame. *Note that you can limit this in case of performance problems using the [General Preferences](#).*

4.21.11.3.2 The PCM Word Filter Configuration

The **PCM Word Filter** configuration page allows selecting the PCM word to filter and two different options for the used encoding. See the [PCM Simple CSV ICD Format](#) definition for an explanation of the fields:

- Word position
- Word interval
- Frame position
- Frame interval
- Encoding (only BNR/UBNR)

If you activated a PCM word filter the according word positions will also be marked with a green background in the minor frame area.

4.21.11.4 MIL-STD 1553 Raw Data Live Processor

This [raw data live processor](#) shows raw data and protocol information from a MIL-STD 1553 channel and continuously updates it. It displays RT activity, subaddresses content, mode code information and allows a graphical plot of a selected word. The live processor is only available in the [Data or Parameter Streaming Version](#).

4.21.11.4.1 The MIL-STD 1553 Data Area

The upper area consists of three pages showing current information on the MIL-STD 1553 bus.

4.21.11.4.1.1 RT Overview

This page shows status information on 31 RTs and the bus as a whole. Each RT is represented by a box with the RT number on top. The complete bus has a separate box at the end. Each box may contain abbreviations for errors that have occurred within the most recent time (about two seconds) and that are allocated to the RT or bus. If any error is indicated the box is also in red color. Otherwise and if there has been any activity at all, the box is colored green. Activity here means that the RT reacted on a command with at least a status word. For the bus it means that there was activity on any RT.

The following abbreviations for errors on the RT are used:

- **R:** No response error: The RT was addressed with a command that required a reply but didn't reply.
- **T:** Terminal flag: The RT transmitted a status word with a terminal flag
- **Su:** Subsystem flag: The RT transmitted a status word with a subsystem flag
- **B:** Busy flag: The RT transmitted a status word with a busy flag

- **M:** Message error flag: The RT transmitted a status word with a message error flag

The error abbreviations for the complete bus include all the errors from the RTs above if they occurred on any RT. Some additional errors are there that cannot be allocated to an RT because they prevent a proper decoding of the message and identification of the involved RTs. All these errors are directly detected by the CH10 recorder and their description is found in IRIG 106 chapter 10. These errors are always displayed before the RT specific error indications. The additional abbreviations are:

- **I:** Invalid word error
- **F:** Format error
- **W:** Word count error
- **Sy:** Sync type error
- **C:** Message error (more precise "CH10 message error" i.e., the obscure error flag from IRIG 106 chapter 10 in contrast to the message error flag in the MIL-STD 1553 status word)

4.21.11.4.1.2 Subaddresses

This page shows a table with the content of the receive or transmit subaddresses of a single RT. The RT and the receive or transmit subaddresses can be selected from the drop-down list on top. The table will show a line for each subaddress and data from the last message to or from this subaddress. The data shown is:

- Subaddress number
- Timestamp of last message
- Other RT (i.e., the RT transmitting to or receiving from this RT. BC means bus controller)
- Period: The time span in milliseconds between the last and before last message on this subaddress
- W1 ... W32: The hexadecimal data words

4.21.11.4.1.3 Mode Codes

This page shows a table with information on mode codes. Each line will represent an RT that received a mode code. The columns indicate the hexadecimal data word and the time of the last mode code separately for the following three mode codes:

- Transmit vector word
- Synchronize with data
- Transmit BIT word

4.21.11.4.2 The MIL-STD 1553 Word Filter Configuration

The **MIL-STD 1553 Word Filter** configuration page allows selecting the MIL-STD 1553 word to filter and two different options for the used encoding. The fields are:

- Word position (1 ... 32)
- RT (0 ... 31)
- SA (1 ... 30)

- Rx/Tx/Mode code: Selects if you want to look into the receive or transmit subaddresses of the selected RT or into a mode code data word
- Mode code: If you have selected the mode code option you can select the mode code here. Only mode codes with data words are available.
- Encoding (only BNR/UBNR)

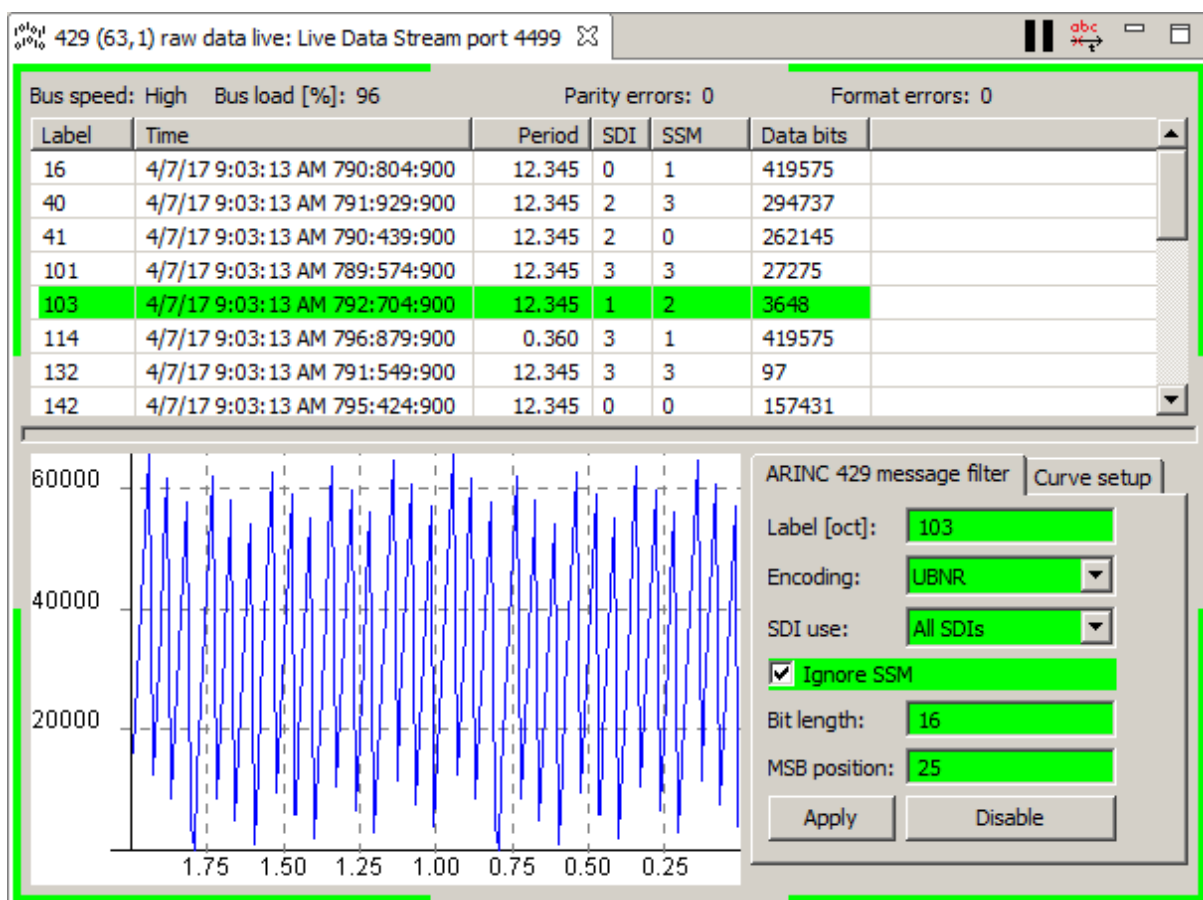
If you activated a MIL-STD 1553 word filter the according word positions will also be marked with a green background in the related subaddress or mode code page.

4.21.11.4.3 The Graph Area

In the graph the only invalid values (i.e., the ones displayed in red) that may occur are if the word position of the raw word filter reaches out of the range of the actual transmitted data words.

4.21.11.5 ARINC 429 Raw Data Live Processor

This raw data live processor shows raw data from an ARINC 429 subchannel and continuously updates it. It displays ARINC 429 messages, the bus load and error and timing statistics and allows a graphical plot of the data bits of a selected ARINC 429 message. The live processor is only available in the Data or Parameter Streaming Version.



4.21.11.5.1 The Message Area

The upper area of the view shows a table with one row for each ARINC 429 message type that has been seen. All messages with the same label (independent of SDI) are considered to be of the same message type. The rows are ordered ascending by the label. Messages with parity or format error are excluded as the label is not known for sure. The columns of the table show:

- Label: The octal label number
- Timestamp of last message of this type
- Period: The time span in milliseconds between the last and before last message of this type. If the last message of this type was more than 2 seconds ago or if there was no activity on the bus at all for about two seconds, the period shows "timeout".
- SDI: The decimal source destination identifier of the last message of this type
- SSM: The decimal sign status matrix of the last message of this type. In fact these are only the two most significant bits of the SSM so they will never include the sign bit of a BNR SSM.
- Data: The raw decimal 19 data bits of the last message of this type

Above the table there are some information on the whole bus:

- The bus speed (high or low)
- The bus load in %. For low-speed buses this gives a range calculated for 12.5 and 14.5 kBaud. In case of no processed data during about two seconds it will show a timeout with a red background.
- Two separate displays show the number of parity and format errors detected by the recording hardware. In addition the background will be red if there was an error processed in about the last two seconds.

4.21.11.5.2 The ARINC 429 Message Filter Configuration

The **ARINC 429 Message Filter** configuration page allows selecting the ARINC 429 message to filter for the graphical display and some processing options:

- The octal label number
- The encoding type:
 - BNR: Two's complement signed binary. Uses the BNR SSM. The sign bit (normally 3rd bit of SSM) is considered part of the data bits and must be included in the MSB position and bit length.
 - UBNR: Unsigned binary. Uses the BNR SSM.
 - BCD: Binary coded decimal. Uses the BCD SSM and picks up the sign from there. If your bit count is not a multiple of 4 the remaining bits will be used for the first digit.
 - Disc: Unsigned binary. Uses the discrete SSM.
- The SDI to filter for. You can either look at messages for the selected label with any SDI or limit to a specific SDI or a specific SDI with broadcast ("all-call"). You can also declare the SDI bits as additional data bits and therefore not filter for specific values.
- You can choose to ignore the SSM (for BNR encoding only the two most significant bits of the SSM). If you don't then the graph will consider data with the SSM encodings for "functional test", "failure warning" and "no computed data" as invalid.

- The bit length of the data to decode. This limits the number of bits considered to be part of the data. For BNR encodings the sign bit is considered part of the number of data bits. Normally you can use up to 19 data bits but when you declare the SDI as data bits in the filter you can use up to 21 bits.
- The MSB position of the data to decode, i.e., the leftmost data bit (including the sign for BNR).

If you activated an ARINC 429 message filter the message type will also be marked with a green background in the message area if this message type was already seen.

4.21.11.6 Analog Raw Data Live Processor

This raw data live processor shows raw data from several analog subchannels within the same channel and continuously updates it. It displays information on the last packet entry of the selected subchannels. For a single selected subchannel it can either plot the raw data or the results of a fast Fourier transform (FFT limited to magnitude) to visualize the frequency spectrum of the signal. The live processor is only available in the Data or Parameter Streaming Version.

4.21.11.6.1 The Packet Entry Area

The upper area of the view shows a table with one row for each selected subchannel. The columns of the table show:

- The number of the subchannel
- The name of the subchannel
- The time of the last processed packet entry
- The minimum and maximum values within the last packet entry
- The sample rate in samples per second. This is taken from TMATS if available or calculated from the real data on the first two packets otherwise.

4.21.11.6.2 The Analog Subchannel Filter Configuration

The **Analog Subchannel Filter** configuration page allows selecting the subchannel for the graph display. If it is active the subchannel will also be marked with a green background in the packet entry area.

4.21.11.6.3 The FFT Display

Instead of displaying the raw data samples you can also display the frequency spectrum of the incoming data. It will be calculated with a fast Fourier transform (FFT). The frequencies from 0 to half the sampling rate of the subchannel will be visible on the horizontal axis. There is no vertical axis but the vertical position of the plotted values is proportional to the amplitude of this frequency. If statistics are enabled, the highest frequency with its amplitude is displayed too. The display will continuously update as the regular sample display.

You can enable the FFT on the **Curve Setup** tab.

Notes:

- *The FFT will use the FFT algorithm on the most recent 1024 samples if that much data is available. Otherwise it will use 512, 256, 128 ... samples. If that much data is available depends also on the selected time span. However, the FFT is only calculated over the specified number of samples not the selected time span.*
- *FLIDAS is limited to 1024 samples here. Using significantly more would slow FLIDAS down without producing better results as several calculated frequencies would map to the same pixels on the screen.*
- *Keep in mind the FFT is calculated on the raw data not a parameter. If your signal requires a simple scaling to calculate the engineering unit the result will be the same as it doesn't change the frequency spectrum but only scales the amplitudes which show no scale on the vertical axis anyway. If your signal requires an offset too the displayed FFT will differ in the frequency 0 output which indicates the offset. If your signal requires a polynomial to convert to engineering units the FFT will not be useful for you.*

4.21.11.6.4 General Notes

Using high frequency analog channels has a high performance impact and can make FLIDAS unusable (note the possibility to automatically shut down overloaded Data Stream Tasks). There are two performance influencing factors here:

- *The work for generally decoding the data on the channel always applies if you use the channel here. Even if you only select one subchannel, all subchannels must be demultiplexed.*
- *The performance and memory impact for painting increases with the time span and sampling rate of the data that you configure for the curve.*

As a rule of thumb today's regular PCs should be able to handle a time span of a few seconds of a 100 kHz sampling easily.

To cope with higher sampling rates it is at least possible to reduce the display time span further than on other raw data live processors. However, the displayed time span might then be shorter than the display update rate. This means you will no longer see the full history of the data on your channel then but only snapshots inside. This could be used to detect general signal characteristics of high frequency measurements. However, still the whole data is being decoded and pressures performance even if only a small slice is being displayed.

4.21.11.7 Discrete Raw Data Live Processor

This raw data live processor shows raw data from a discrete channel and continuously updates it. It displays the current state of the discrete inputs and allows a graphical plot of a selected discrete. The live processor is only available in the Data or Parameter Streaming Version.

4.21.11.7.1 The Discrete Status Area

The upper area consists of a check box for every discrete with the bits labeled from up to 31 down to 0. If the discrete is currently set the check box will be checked.

4.21.11.7.2 The Discrete Filter Configuration

The **Discrete Filter** configuration page allows selecting the discrete bit to display in the graph area. The bits are numbered from up to 31 down to 0.

If you activated a discrete filter the according check box will be highlighted in green on the discrete status area.

4.21.11.7.3 The Curve Configuration

In the curve configuration you cannot scale the values as on other raw data live processors. The displayed value range is fixed between 0 and 1.

4.21.11.7.4 The Graph Area

The graph has no value scale. If the discrete is 1 (or on) the graph will be on top, otherwise on the bottom.

4.22 Parameter Live Processors

See the [parameter live processor concept](#) for a general explanation of parameter live processors.

Before being able to start a parameter live processor you must activate an [ICD mapping](#) in the [Parameter Plug-in Control View](#).

Parameter live processors may allow the removal or adding of further parameters while they are running. In general, the Parameter Set View or the [Parameter List View](#) can be used to drag and drop parameters.

The live parameter provider will process the CH10 packets in time order but then completely and not enforce a time order of packet entries from packets on different channels. This means you may see a parameter from time $x + 100$ milliseconds on one channel before you see a parameter of time x on another channel. A reordering of more than 100 ms is not possible as a packet may only contain up to 100 ms of data (and will most of the time contain much less). However, if the parameter live processor displays the time of the sample as text or on a time axis it will display the correct time. You will only see it later or earlier.

The available parameter live processors are shown in the subsections of this chapter.


4.22.1 Parameter Table Live Processor


This parameter live processor shows the current values of selected parameters in a table and periodically updates them.

The Parameter Table Live Processor will show this information about each parameter:

- Parameter mapping ID
- Parameter path
- Parameter name
- Current value

You can right-click one or more parameters and select **Remove** to remove them from the table. Alternatively, you can use the keyboard shortcut **Del**.


There is a  pause button to pause the updating of the table. If paused the table will have another color.


You can create time markers by pushing the  button on the upper right. They will refer to the current time of the live data stream when you clicked at the icon or the time you paused if the pause mode is on.

4.22.2 Parameter Graph Live Processor

This parameter live processor shows a graph of a single selected numeric or enumeration parameter over time. If opened on a parameter set it will only display the first Parameter.

You can drag & drop parameters to the parameter graph live processor to display another parameter.

There is a  pause button to freeze the current graph.

You can create time markers by pushing the  button on the upper right. They will refer to the current time of the live data stream when you clicked at the icon or the time you paused if the pause mode is on.

Enumeration parameters will only display their raw value in the graph.

The live processor has two pages, one with the configuration and one with the graph.

4.22.2.1 Graph Page

The tab on top of the graph page shows the parameter name. The graph itself shows samples of the parameter over time. *Note that if you use analog parameters this is a place where you can use full live decoding parameters.*

A time axis in seconds runs from the bottom right (time of last sample) to the left (i.e., the past). A value axis on the left shows the maximum value on top and the minimum on the bottom. Maximum and minimum can be selected on the Curve Setup Page.

Invalid values will appear in red on the bottom axis.

A line with statistics on the samples in the selected time span can be shown below the graph. It contains:

- Current value
- Minimum
- Maximum
- Peak-to-peak value
- Mean value
- Root mean square (RMS) value

4.22.2.2 Curve Setup Page

This area allows some configurations for the graph. The settings are in green if they are active, otherwise they have not yet been applied by using the **Apply** button. Settings in red are invalid, while yellow settings can be applied. If you have invalid settings error messages will appear as soon as you try to apply.

One configuration allows setting the time span in milliseconds to display in the graph. Using a too long time span for a high frequency measurement will massively degrade performance. Several ten-thousands of measurements should not cause a problem in normal systems. A too short time is not useful since the screen plot will only be updated in intervals so you would only see some segments of the values.

Another configuration allows choosing the range of numbers on the value axis (i.e., scale on the value axis):

- Auto: Axis ranges between minimum and maximum value received since selecting the parameter for the display
- ICD: Take the minimum and maximum from the values calculated or defined in the ICD. Might not be available for each parameter.
- User: Define the minimum and maximum by entering them into the min and max fields

The initial mode for the value axis scaling can be set in the [Parameter Plug-in Preferences](#).

You can optionally enable the statistics display.

You can choose how to label the time axis. Relative time will display the seconds that have passed since the sample while absolute time will print time stamps (without date) on the axis. The labels will include milliseconds when needed.

4.22.3 CVSD Live Replay

This parameter live processor replays audio from a single selected CVSD parameter to the local audio hardware.

The playing relies on the availability of local audio hardware that supports playing with the given configuration.

The replay has no visual representation and can only be stopped by stopping the associated data stream task in the [Repository View](#).

Often the local audio hardware is less precise than the PCM stream and does not replay exactly at the sample rate it is commanded to. To keep the replay synchronous with the CH10 stream FLIDAS needs to drop or duplicate samples from time to time which leads to small audible sound artifacts. The sound quality will be less than if you export the data from a recorded stream.

To keep the audio hardware buffers filled with audio for replay, FLIDAS creates an additional delay to the CH10 stream. Currently this is around 300 ms on top of the regular delays but may change without further notice.

4.23 XML CH10 Mapping

The IRIG 106 standards body the RCC (Range Commanders Council) requested proposals for a mapping between XML language and CH10 files with their change request CR031. Data Bus Tools made a proposal for a specification and provided an implementation integrated in FLIDAS. The specification is official part of the RCC's "IRIG 106 Chapter 10 Programmers Handbook" (RCC document 123-16) in Appendix P/Q since August 2016 (moved to appendix Q/R in RCC 123-20).

The specification can be downloaded from:

- The official RCC site: www.trmc.osd.mil/wiki/display/publicRCC/100s (public access to this server is changing from time to time due to US DoD policy although the standard is approved for public release with unlimited distribution).
- A document mirror on www.irig106.org/wiki/rcc_mirror.
- The Data Bus Tools website on www.databustools.de/IRIG106/XMLCH10mapping.html also contains an excerpt of the relevant part for convenience.

4.23.1 Intention of Specification

The intention of this specification is to have a way to define IRIG 106 chapter 10 files for testing purposes in a readable XML format. This would allow tools to convert between XML and CH10 back and forth. The focus of XML is readability and not memory efficiency or processing efficiency. Therefore, the area of application for this specification is the generation of smaller test files (although nothing technically prevents to have large files).

The conversion from XML to CH10 opens these opportunities for example:

- Set up of test data for the verification of CH10 analysis SW
- Preparation of exactly specified data sets for replay

The conversion from CH10 to XML opens these opportunities for example:

- CH10 files (for example, from recorders under development) could be converted to XML for closer inspection and to verify possible corrections (by converting the manually corrected XML file back to CH10).
- Specific error cases can be created for test cases by converting real files to XML, injecting the error and converting back for replay.

4.23.2 Official and Extended Implementations in FLIDAS

FLIDAS up to version 3.6 included an implementation for the XML CH10 Mapping specification version 0.3.1 (and is the only software to our knowledge to do this so far). It is available in the free FLIDAS version and still available on the website. There are finer

grained options to control the conversion between CH10 and XML in all commercial FLIDAS versions.

We enhance the specification independent of the current four year release cycles of the RCC's document so we might do changes to the format that are incompatible to the released format. There is no guarantee that these will be included in further official RCC documents. But since this technology is new there should be room to evolve when new use cases appear. If you actively use it for your testing and rely on it you should consider keeping downloads of a proper release of the free FLIDAS version for your XML format version. If you use a commercial version, you can still use this functionality with the free version. Please let us know new use cases and requirements for this functionality and we will check if it can be supported.

So far (FLIDAS Version 4.15.2, XML CH10 mapping specification version 0.3.7) all changes are extensions in a way that XML files following the RCC specification can still be processed. However, if you convert CH10 files to XML features of the FLIDAS extended format might appear in some cases.

4.23.3 Operation

Using some of the features of the free FLIDAS version you can immediately see the results of your conversions. Any commercial version of FLIDAS supports further features and especially the low-level inspection of CH10 data structures and is far more useful when checking the results of the XML conversions.

4.23.3.1 CH10 to XML Conversion

After loading a CH10 file use the export function, i.e., right-click the file and select **Export > CH10 to XML**. Note that XML files require more space than the CH10 files on your storage and large files might require some time for exporting. The XML CH10 mapping is not intended to work with gigabytes of data.

The commercial FLIDAS versions contain further options to control the conversion process that you can configure in the export wizard.

You can open the file in an XML or text editor or also with browsers like Internet Explorer or Firefox. Using a real XML editor and applying the XSD file with the XML schema definition provides further advantages like validation and guidance in modifying the data. The free Eclipse IDE for example includes XML editors that can be used like this. Our website includes setup instructions how a free Eclipse XML editor environment can be created. Go to www.databustools.de/IRIG106/XMLCH10mapping.html for the download of the instructions.

4.23.3.2 XML to CH10 Conversion

Go to the main menu and select **File > Convert XML to CH10**. Select the XML file and a CH10 file to create. You can choose to validate your XML file to learn about errors in the file (on the XML schema level - no CH10 specification errors!). This will take more processing time than the conversion itself so if you are sure your file is OK or if it has been generated with the CH10 to XML conversion, you can skip the validation.

You can export the internally used XSD file (the one used for validation) to your file system by clicking **File > Export Resources** and select the **XML CH10 Mapping XSD File**. You will need this to learn about the syntax of the XML files and to set up an external XML editor if you use one. It will be the same file as offered on the website in a version matching the FLIDAS version.

The commercial FLIDAS versions contain further options to control the conversion process that you can configure in the wizard.

4.23.4 Samples and Other Information

A source for further information is our website www.databustools.de/IRIG106/XMLCH10mapping.html. You will find sample files for download there, the XML schema definition (XSD file), some description about the specification and a guide how to set up a free environment where you can edit and validate XML definitions. These things are not part of the regular delivery since they are related to the specification that is now published by the RCC so that also competing companies can provide implementations.

4.23.5 Verifying Created CH10 Files

The XML CH10 mapping intentionally lets you create all kind of errors. You can use the commercial FLIDAS feature CH10 File Verification View to verify that the result conforms with IRIG 106 chapter 10 to some extent or deviates from it in the form you intended.

4.23.6 Notes on Implemented CH10 to XML Conversion

Although the specification only defines some data types in detail (the original RCC request was limited), the conversion will convert all packets. It will simply define the binary data in XML for the unsupported types. Packet headers and secondary headers are generated anyway.

The conversion will try to make use of auto-values if possible. For example, it will not define the checksums and filler data if they are correct. If they are wrong however it will generate explicit definitions:

- Checksums will then be defined relative to the expected checksum. I.e., if your checksum is 1 above the expected one it will define it as "+1". In contrast to an explicit checksum definition (like 0x3F21) this guarantees that if you now modify the data, it will still generate a wrong checksum. If you want to fix the checksum you can of course just remove the checksum attribute. Note that the idea of this specification is to create test data. Therefore, a wrong checksum is likely intentional and shall be kept wrong and not turn out correct by accident.
- Fillers will be explicitly defined if they are different from 0x00 and 0xFF or if they do not cause the packet length to be a multiple of 4 (32-bit alignment). The benefit is that you can modify the content of packets with correct fillers without the need to manually update the filler definition.

Converting CH10 to XML and back to CH10 doesn't always give the exact same binary file by default (but files with the identical interpretation). You can use some of the commercial options to create binary identical files. Differences created by default are results of:

- XML replaces all line breaks in TMATS by a single line feed, even if they included a carriage return. Also, ASCII 0x00 values must be removed. In conclusion, also the file size differs often.
- Filler definitions are not always explicitly defined if they were correct. Since fillers can be 0x00 or 0xFF but the conversion generates them automatically as 0x00 there can be a difference. Also, some recorders create more fillers than necessary due to their internal architecture. FLIDAS will reduce them to the minimum by default as they do not affect the interpretation of the file.
- Also invalid file formats might not create identical files. In some cases the file format is invalid but in an aspect that FLIDAS does not care about and therefore does not detect (FLIDAS is built to use the data not verify the files). In these cases undetected errors might not be represented in the XML file and therefore not converted back to CH10.

4.23.7 XML CH10 Mapping Commercial Additions

There are some additions to the XML to CH10 conversion in both directions that only exist in the commercial version.

4.23.7.1 CH10 to XML Conversion Commercial Additions

The XML specifies the CH10 file to create exactly. However, CH10 files can be exported to different XML files that all convert back to the same CH10. The commercial FLIDAS versions contain additional options to configure the XML that is being created from a CH10 file. The options influence how easy you can modify the created files or what types of modifications you can apply in case you create them for modification.

In the wizard for the CH10 to XML export you can configure these options:

- **Timestamp comments on each packet:** This prepends a comment with a timestamp to every CH10 packet definition in the XML code. This helps navigating the XML file.
- **Separate TMATS file(s):** Instead of putting the TMATS content into the XML an external TMATS file will be created that is included from the XML file. It will have the same name as the created XML file with an addition of the part number (there may be several TMATS packets) and with a "tma" or "xml" extension depending on the TMATS type. If you move one of the files to another directory you must move the other file too or update the path that refers to the TMATS file inside the XML file. The benefit of using an external TMATS file is that exactly this content is included in the created CH10 file. This includes precise line break definition and other special characters.
- **Exact packet fillers:** This option makes sure that the fillers in the packet trailer are exactly created when converting back to CH10. While the normal mode just makes sure that valid fillers are converted to valid (but possibly different) fillers while invalid fillers are recreated exactly, this mode also precisely recreates fillers that use 0xFF instead of 0x00 as bit pattern or that are longer than 3 bytes. In effect you lose

maintainability if you intend to modify the created XML code because you must then manually modify the fillers too instead of managing them automatically.

- **Channel relative packet RTCs:** This mode will create RTC definitions for each packet that are relative to the last packet on the same channel. This helps making the time difference easier visible and easier to modify.
- **Explicit packet entry timestamps:** Packet entries often have a timestamp. FLIDAS creates explicit XML definitions of packet entries only for the data types supported in the XML format. For these the timestamp would normally be created as a raw value for the 8 byte timestamp independent of its type. When activating this option an explicit timestamp type dependent definition would be created instead (like defining the RTC, hour, minute, second etc.). This mode is slightly slower and might create larger XML files but helps you reading and modifying the file easier. Never the less both modes are precise.
- **Relative packet entry timestamps for RTC / ERTC:** This option is only in effect when using "Explicit packet entry timestamps" (see option above). When activated it will use relative instead of absolute definitions for RTC and ERTC packet entry timestamps. This means they are defined as offset to the packet timestamp itself which makes it easier to read and modify in some situations.
- **Detailed export for:** This option controls if a more detailed XML format will be used for the selected data types data section of the packets. This often means that CSDW (channel specific data words) are explicitly decoded to its attributes and packet entries will be created in a structured way rather than raw data. By default this is enabled for all supported data types and active in the free version. You can turn it off for some data types to get smaller and faster to convert files if you are not interested in this data type.

Note: If you are converting from CH10 to XML and then back to CH10 you might not create binary identical files but only equivalent files. If you want identical files you need to activate these options:

- *Separate TMATS file(s)*
- *Exact packet fillers*

Depending on your CH10 recorder one or both options may be unnecessary.

Both modes have negative effects on maintainability in the cases where you want to apply changes to the XML. So if you need to verify FLIDAS conversion you can use the binary identity modes and later inspect the differences caused by deactivating these modes. The differences are limited to special characters within TMATS and the generation of useless filler data.

If your CH10 file has an invalid format it might still not create binary identical files.

4.23.7.2 XML to CH10 Conversion Commercial Additions

When you convert an XML file to CH10 you might want to keep the original XML file included in the CH10 file for reference. This can be done by enabling the option **Include XML as user defined computer-generated data** on the **XML to CH10 export options** page of the XML to CH10 export wizard. In result the export will add a computer-generated user-defined data packet directly after the first explicitly defined time packet. It will only be a

single packet, so XML files with more than 512 KBytes (-28 bytes) will not be included. To generate the packet you need to specify the channel ID of a computer-generated user-defined data channel on the wizard page. You can then later export the XML from the generated CH10 file using the CH10 packet body export function.

5 System Administration Topics

5.1 Portable Installation

Besides the regular installation it is even possible to install FLIDAS on a USB stick together with Java. This way you can have it with you, including your configuration and you are guaranteed to get the same Java environment on every computer without installing Java there.

To use this installation mode follow these steps:

1. Install FLIDAS on one computer regularly
2. Create a directory "jre" (lower case!) in the same directory where the FLIDAS executable is located (i.e., next to plugins directory)
3. Locate your local Java installation directory. You need to copy the files inside this directory into the new created "jre" directory. In result you should have a "bin" and "lib" directory directly below "jre".
4. If you start FLIDAS now, you will use the Java version from the JRE directory.
5. You can now copy the whole directory to an USB stick

FLIDAS has different version for each operating system. So the USB stick will be limited to one type of operating system (usually Windows).

Another aspect you should consider is that the Java version in the "jre" directory is no longer managed by your operating system. Therefore, if your administrator installs new releases of Java, you will still use the old one. You should therefore make sure this way of using FLIDAS is in conformance with your company's IT rules.

5.2 Backup

FLIDAS is easy to handle from a system administrators' point of view. It is fully contained in a directory and stores no data somewhere else (provided you do not write protect the configuration directory).

If you need to backup FLIDAS you can simply backup the full directory. If something fails, restore the old directory at the same or other location and go on.

If the backup takes up too much space, you can just backup this directory one time initially (after a first start up which does some configuration) and afterwards only backup the settings. These are contained in the installation directory below *configuration/.settings*

Note that users will also create their own data (like ICDs, exports, ICD mappings, parameter sets). These are contained wherever the user stores these and must be backed up separately. It is possible though to keep them in the installation directory.

5.3 Update

If you need to update FLIDAS, follow this process:

1. Install the new version as the initial installation but do not start it yet. If you want to keep the directory name the same, rename the old directory before installing the new.

2. Copy the folder *configuration/settings* from the old to the new installation directory

You can now normally start up. If you are happy with the new installation, you can delete the old one or keep it for a while for backup.

Note that if you store your self-defined data (like ICDs, ICD mappings etc.) inside the installation directory, you need to copy this too.

6 Security

FLIDAS will not try to transmit any data out of your company. The content of the CH10 sources will be loaded into memory for processing during that session. The configuration files will not include content information of the CH10 sources.

Data might be stored in other locations on your request and you are then responsible for the security of these new created files. This occurs in these situations:

- Explicitly requested data exports
- Explicitly created ICD mapping, parameter set or instrument panel files
- CH10 file properties can be stored if enabled explicitly. They will include time markers of CH10 files you loaded. By default they will be stored in the same directory like the CH10 file unless you configure something else.

In normal mode FLIDAS will only read from your CH10 files without modifying them. However, it is advised to create backups of your data. It is possible to activate a modification mode but this needs a specific command line option and cannot be started accidentally. See the [Hex Packet View](#) for more information.

The help system may open a network connection on some systems. This connection will always be to the same computer that FLIDAS is running on (the "local host"). This is because the help system uses traditional web browser technology. Some links of the manual may point to websites in the Internet. Blocking the help system's network connection will not limit the features of FLIDAS.

If you use [data streaming](#) to publish CH10 files or TmNS messages or use raw data streaming on the network FLIDAS may transmit data outside of your company. This is if you configure the destination address outside of your company or your company's network hardware chooses to route the data outside the local network.

7 FAQ

Can't start FLIDAS. I only see a window saying "Java was started but returned exit code=13": Your installed Java version does not match the system requirements. Follow the [installation chapter](#) for proper Java installation.

I want to check if my download went OK: On the download website there is a checksum and a file size given for each of the downloads. If you want to verify if the download was successful, you can compare both. If you have no tool to check the MD5 checksum, you can use Microsoft's free [File Checksum Integrity Verifier utility](#).

Out of memory: If you encounter memory problems with FLIDAS you should consider:

- Reduce the memory consumption with the [CH10 data type filters](#).
- Reduce the memory consumption of [parameter viewers](#) by reducing the [parameter set size](#).
- Close surplus CH10 sources in the [Repository View](#) and unnecessary windows
- By default Java allows applications only using a part of the available memory. Increase the memory available to Java by starting it with command line options. Since Java is started implicitly when you start the FLIDAS executable, you need to add the command line options (each as a separate line) to the end of the FLIDAS.ini file. An example for Oracle Java (and most current systems which are derived from it) is the option `-Xmx1024m`. The value 1024 is the maximum number of megabytes to use. You can also increase this further. You can even increase this over your physical memory limits and Java will use virtual memory (i.e. swap data in memory to the disk).

I am using a protocol viewer, exporter or live processor on a single subchannel and don't get the data I expect or no data at all: The TMATS section and the recorded data section of a CH10 source define numbers for each subchannel. While the recorded data section (CH10) had an explicit definition of the numbering the TMATS specification (CH9) failed to provide an explicit definition until IRIG 106-2015. Therefore, different manufacturers implemented different styles of numbering and sometimes even vary between recorder models and firmware versions. In effect FLIDAS needs help in mapping the properties defined for a subchannel in TMATS to the actual data for the subchannel. This help is provided using the [Subchannel Number Preferences](#). If you don't set this up properly and try to work with data from one subchannel it might access data from another (maybe not existing) subchannel.

When you create a [communication summary](#) for the channel you will be able to see which subchannels are available by deeper analysis of the data. You can also start an exporter or protocol viewer from there but if the action needs TMATS attributes it might use the attributes of a wrong subchannel if you have a wrong configuration.

Decoded analog data is not correct: To decode analog data FLIDAS needs to know how it is encoded (two's complement, BCD, offset binary ...). This is configured with the TMATS attribute `R-x\AF-n-m`. However, this is missing often, especially because the attribute was removed from the 2009 revision of TMATS by accident and reintroduced in 2011. Recorders using TMATS 2009 which strictly conform to the standard may not use the attribute.

Solutions:

- Upgrade to a newer TMATS revision. This is the best solution if you can technically implement it as it will solve many other issues too.
- Use TMATS defaults. This solves the problem without further actions if you only use recorders that use the same encoding.
- Use TMATS overriding or TMATS replacement. Both requires re-occurring actions

Data streaming with data loss: If you encounter data loss on any recording data streams or live data streams you may consider the following:

- Try to configure the Data Streaming Preferences. Consider setting them to default if you changed them. After the default values increasing the receive buffer size is a good idea.
- Check if FLIDAS detects an overload situation with the indicator described in the Repository View
- Does your computer have the performance to handle the incoming traffic (and does it have it consistently or are there background processes occupying the CPU for longer periods from time to time)?
- Do you have a firewall blocking network traffic?
- Is your network connection on the whole path between transmitter and receiver fast enough to cope with the traffic (types of network interfaces, cables, switches and routers, network congestion etc.)?
- In general, be aware that the data streaming uses UDP protocol which by specification is allowed to drop data all along its transmission path (routers, switches, network interfaces, operating system processing etc.) and neither the operating systems you use nor FLIDAS itself is real-time capable. So, ultimately you have to live with the risk of data loss. Never the less realistic system setups can run for hours without any data loss.

CH10 files in Windows do not open on a double-click anymore: When you have configured Windows to automatically open CH10 files with FLIDAS and it stopped working, then you have probably put FLIDAS to a different directory than on initial configuration. Some Windows versions then malfunction and do not allow selecting the new FLIDAS directory for opening. The following procedure helped on Windows 7 64-bit. Try it at your own risk:

1. Run the command **regedit** by typing it in the Windows start menu and hitting return
2. Navigate to **Computer\HKEY_CLASSES_ROOT\Applications\FLIDAS.exe**
3. Delete this key and everything below
4. Reconfigure the Windows CH10 file association with FLIDAS

My data export runs really slow

Most kinds of exporting to network drives are significantly slower compared to local drives. Consider exporting to a local drive and copying the result to the network drive later.

Graphics output is corrupted or small and I am using an operating system scaling other than 100%

When you set up windows to scale applications to other scales than 100% it will try to modify the graphics output. Although FLIDAS 4.10 should fix the problems it seems that

problems may depend on system settings like Windows version, display types, multiple display configurations, configured windows scaling and Java version. There is a [GUI Scaling \(troubleshooting\) Preferences](#) to help you dealing with the case that you still have problems.

Icons are missing in the chapter navigation of the external help browser

When you open the external help browser using **Help > Display Help**, you might find that the icons from the chapter navigation are replaced by alternative texts. Internally this uses the Internet Explorer installed on all Windows systems (like other Windows functionality seems to do as well) to display the locally installed help. For some reason the Internet Explorer does not display this well on some systems (either because SVG graphics are poorly supported or JavaScript is restricted). In these cases you can change the browser to start by default to Microsoft Edge. To do so:

- Open the FLIDAS.ini file inside the FLIDAS installation directory with a text editor.
- Add this line after the line that says "-vmargs"

```
-Dorg.eclipse.swt.browser.DefaultType=edge
```

The text must directly start in front of the line. The hyphen is part of the line!

8 Open IRIG 106 Issues

Some issues in the IRIG 106 specification are not described detailed enough to implement or leave room for different interpretations. FLIDAS will try to raise warnings if it encounters any of these situations. The following subchapters will describe some of the problems and how FLIDAS will implement it. All these issues have been reported to the RCC for improvement of the standard.

8.1 Storage of LSB First PCM Data Words in Unpacked or Packed PCM Mode

TMATS allows declaring PCM data words with LSB first bit order (TMATS: P-d\F2). This is reversed to the "normal" processing order in a computer so the words must be reversed for processing in software.

There would be two options to implements this:

1. The recorder reverses the PCM data word before storing it in the CH10 file
2. The recorder stores the unreversed PCM data word in the CH10 file. Processing software must reverse it.

FLIDAS supports option #2 and indicates a warning if it encounters the situation.

The problem does not affect throughput mode since the recorder doesn't know the positions of PCM data words inside the stream and can therefore not correct it.

Discussions with some recorder vendors showed that LSB first doesn't seem to be used by anyone. Some vendors implemented one of the options but haven't seen it being used so far.

8.2 Storage of PCM Data Words in Unpacked PCM Mode with Word Sizes Above 16 bits

CH10 doesn't specify how PCM data words are stored in unpacked mode when they have more than 16 bits.

FLIDAS implements a solution proposed by Zodiac Data Systems: Each word is padded with 0 on the left until it has a length that is a multiple of 16. Then as many full 16-bit fragments are stored as needed.

It seems that different implementations for this exist in the industry.

8.3 Disabled Analog Subchannels

The following problem has been solved with IRIG 106-15 but still applies to CH10 sources that do not use TMATS according to IRIG 106-15.

To decode analog subchannels it is required to know if subchannels are disabled. CH10 refers to TMATS for this definition but the TMATS specification (CH9) does not provide the ability to store this information.

FLIDAS will decode analog data as if all subchannels are enabled. Some recorder vendors even ceased to use analog subchannels and use channels now instead because of this problem.

If you are working with parameters the Analog Simple CSV ICD Format allows defining disabled subchannels and therefore proper decoding (only on the parameter level!)

You can also use TMATS overriding to include the new IRIG106-15 attributes (or you may even be able to include them directly in the recorder setup).

8.4 Chapter 4 time as secondary header time or intra packet time stamps

Chapter 11 defines that chapter 4 binary time can be used in secondary packet headers (for packet time) or for intra packet time stamps (for packet entry time). However, chapter 4 just defines this time as a counter of microseconds. Since no reference is given like this is the microseconds after new year, after Christmas, microseconds of the day, microseconds since start of recording etc. this time is basically unusable. Instead of just ignoring the time, FLIDAS will interpret it as microseconds since new year, the only interpretation seen so far in a real file.

8.5 TSPI / CTS format 1 is ambiguous

At least until the 2022 standard, TSPI / CTS format 1 data type packets storage is ambiguous. The tables shown and the text disagrees if the intra packet data header is 16 or 32 bit. Also possibly the optional static data / pod ID is supposed to be included in the intra packet data word or separate. Also, the definition of a "message" is unclear (message means static and/or dynamic lines? Is carriage return included?). There are multiple other issues as well. If anyone can provide information or even sample data, then at least support for an existing interpretation could be implemented.

9 Feedback

Don't hesitate to ask for new features or suggest improvements. Write an email to info@databustools.de or go to www.databustools.de for further contact information.

The CH10 file format has many options and many of them rely on interpretation. Different recorder manufacturers have chosen different interpretations. If your files cause any trouble reading, we would be happy to improve FLIDAS to cover your specific problem. The same applies to the support of other CH10 data types. Often the implementation is missing due to the lack of sample data so if you have any non-classified files and want support for these CH10 data types, provide them to get a free upgrade when your contribution leads to support of this CH10 data type.

As said before you may also request versions for other operating systems and we would be happy to hear about your experiences with it.

10 What's New

10.1 FLIDAS 4.15.2

This version improves CH7 decoding stability for corrupted streams and offers usability improvements.

New features in the parameter plug-in:

- Parameter graph viewer:
 - mark the positions of samples
 - Indicate currently selected stripe with red cursor
 - Scrolling with page up/down key synchronized to scroll by stripes.
- Parameter filtering in parameter set/list view:
 - optional regular expression filtering
 - optional case sensitive filtering

Basic commercial plug-in:

- CH7 decoding stability improved for corrupted PCM streams
- Display more information in the general file information properties if Safran Data Systems segmented files are loaded (filenames, size and time range for each segment). This can be used to find out which part of the file is located in which segment.

General new features:

- The default table actions (export to CSV, copy cell, toggle font, expand/collapse) have been added to many more tables

There are some more smaller bug fixes and usability improvements.

10.2 FLIDAS 4.15.1

This version supports a new, separately available plug-in for a specific Ethernet parameter decoding. It also has a lot of usability improvements.

New NDO plug-in:

- This plug-in decodes parameters from UDP packets that contain fixed sized data blocks (network data objects = NDO) starting at a fixed position that need to be decoded as separate messages.
- This is a separate plug-in as it will not be applicable to many projects.
- If you think you can use it or think it can be generalized to your use case, get in contact

New features in the parameter plug-in:

- More details on parameter and ICD configuration are available on the UI
- A bunch of improvements on the filtering on parameters in Parameter Set/List View:
 - filter for comments

- filter in hierarchy elements
 - filtered vs. complete selection
 - save filter setup
 - Apply filter when dragging tree segments
 - ...
- Drag & drop of parameters from the selected parameters area in the parameter set view
- A bug was fixed that only decoded the first occurrence of a PCM supercommutated parameter within a minor frame (only in PCM non-throughput mode in live data streaming)

General new features:

- When you need to select files or streams in a dialog you see its assigned color and compatibility with the active ICD mapping and possible problems.
- Other smaller improvements and bug fixes

10.3 FLIDAS 4.15

This version adds support for binary UART and user defined computer-generated data parameter decoding and introduces a recording gap view.

New features in the parameter plug-in:

- New ICDs for:
 - user defined computer-generated data
 - binary UART (this has been requested often but it comes with some assumptions that your recorder and system must satisfy)
- The parameter list and parameter set view can be filtered for parameter names
- The table gauge on instrument panels can show the comment field of the parameter
- Enumeration definitions for parameters can define the numeric value in hexadecimal
- You can use a "save as" for ICD Mapping, Parameter Set and Instrument Panels
- Performance warnings when working with large parameter sets or sample amounts in Parameter Graph or Parameter Table View can be configured
- A bug for parameter decoding on multiple subchannels of the same channel has been fixed. The bug could happen when:
 - You process files (streams were not affected) and
 - you use message data or UART data type and
 - Connected ICDs to multiple subchannels of the same channel. Many recorders seem to avoid this in the first place as subchanneling comes with some trouble. They often use a one subchannel per channel approach on CH10 data types that require subchannels.

In this case the decoded data of one subchannel was correct but the others not. If you were affected, it is unlikely that you would not have seen it.

New features in the basic commercial plug-in:

- A new Recording Gap View allows checking a selection of channels for gaps where no data has been recorded. The results are displayed in a colored table to show you

where the channels had a common or single longer pause in the data and see their points in time.

- There is a packet entry viewer for user defined computer-generated data
- A new CH10 verification check can check if CH10 packets use data type versions that are valid for their data type.

New features in the free version:

- User defined computer-generated data packets are now split into "packet entries" so the binary data can be seen on the GUI

There are some more smaller bug fixes and improvements on robustness and usability.

10.4 FLIDAS 4.14

This version upgrades to a new version of the user interface framework used for FLIDAS. There are also some other improvements.

FLIDAS upgraded to the latest version of the Eclipse RCP framework. As a result from this the required system configuration is now Windows 10 or later and Java 17 or later. **Java 32-bit is not supported anymore**, you will need a 64-bit Java installation. Note that even if you have a 64-bit Windows system you might still run Java 32-bit. Details about Java installation can be found in the installation chapter 2.1. It is usually an easy installation of free software you can do yourself without an administrator.

You may still be able to run FLIDAS with older Windows and Java (but 64-bit). A quick test worked out on Windows 7 and Java 11. However this is unofficial and without liability. Still you can report problems.

New features in the free version:

- Time markers can be persisted between sessions working on the same CH10 file. An additional properties file will be stored when you enable it during loading.
- The time markers can now be seen when you right-click the file and go to Properties > Time Markers. You can also edit them there and im-/export them to share with other applications.
- You can also create time markers for UDP live data streams with the intent to export them after your testing session.
- You can read just a part of a CH10 file. For example if you have a very big file and are just interested in a part in the middle you can configure approximate start and end positions in the file and FLIDAS will read the TMATS and then just the section you selected.
- FLIDAS will start with maximized window. This solves some strange behavior on some multiple screen configurations.

New features in the parameter plug-in:

- The Parameter Graph View now supports zooming with the mouse wheel to the mouse pointer.

There are some more smaller bug fixes and improvements on robustness and usability.

10.5 FLIDAS 4.13

This version primarily improves support for TMATS PCM ICDs.

New features in the parameter plug-in:

- Improvements on PCM TMATS ICD:
 - Support for fragmented measurements.
 - Parity supported on the parameter level.
 - Support for new binary formats:
 - Bit weights
 - One's complement
 - Offset binary
 - Sign magnitude binary in both options
 - Support for pair set data conversion type using the linear interpolation option
 - PCM time parameters
 - Enumerations can now use any integral binary format, not only unsigned binary
- The detailed properties displayed for each Parameter look a little bit different in the new version. This is because a lot of changes took place under cover for the PCM TMATS ICD improvements. These are things that are likely to lead to improvements in other ICDs as well in the future.

10.6 FLIDAS 4.12.1

This version is primarily a quick bug fix for the support for Safran Data Systems recorder CH10 file segmentation. Users that don't use these files don't need to update.

New features in the free version:

- There was a problem reading segmented Safran Data Systems CH10 files when one of the segments started at or after a 2 GByte offset. This is fixed.
- The reading of segmented Safran Data Systems CH10 files now supports also non default filenames as long as they end with "_rxxxxfxxx.*" (x is a single digit).

10.7 FLIDAS 4.12

This version primarily adds support for Safran Data Systems recorder CH10 file segmentation and audio from numeric PCM parameters.

New features in the free version:

- CH10 recordings from Safran Data Systems often segment the single recording into a sequence of files. FLIDAS can now load them as if they were one file. The files must match the default naming scheme.
- All functionalities that receive or transmit UDP data allow selecting the network interface card to use in the computer in case a multicast IP address is used.

New features in the parameter plug-in:

- Numeric PCM parameters can be exported to a WAV file. This is intended for audio exports but could be used for other things as well. The full numeric range of the parameter will be mapped to the full range of the WAV file samples.
- The CVSD to WAV export can be configured to skip gaps in the PCM reception or fill gaps with silence.

There are other smaller changes and corrections.

10.8 FLIDAS 4.11

This version primarily adds support for the latest IRIG106-22 standard and iNET-X.

New features in the free version:

- Adaptations to the IRIG 106-22 release. Some changes include:
 - CH10 verification checks for changes to test conformity of updated recorders
 - Decoding of Analog I/Q recordings (format 3). The analog packets of this format can be used where the regular packets can already be used. I.e., you can export and do parameter decoding for example. However, format 3 comes with high sampling rates, so you might run into performance issues if you try to process longer time frames)
 - The deprecated video formats (1/3) will be reported as invalid in 2022 if you verify a CH10 file. However, they are still supported with the previous FLIDAS features.
 - The XML CH10 mapping has been adapted to include the new release flags in the packet header and TMATS CSDW.
- The configuration of tables in FLIDAS was adapted. For all tables that can be configured, there is a context menu now that you can reach with a right-click on the table header. You can manage the visible columns, widths and order a lot easier than before. The save symbols previously used to save the column setup have moved to this menu as well.
- A bug in the XML to CH10 conversion of structurally defined Ethernet messages created wrong fillers and led to unreadable data in some cases. It has been fixed.

New features in the basic commercial plug-in:

- iNET-X data recorded as CH10 Ethernet can be used to:
 - export the protocol data to a CSV file
 - create an iNET-X communication summary listing message periods by stream ID, summarizing message counts and errors etc. As usual you can directly jump to the detected problems.

There are other smaller changes and corrections.

10.9 FLIDAS 4.10

This version primarily adds support for user interface scaling and basic support for TSPI format 1.

New features in the free version:

- Windows allows scaling the user interface (Windows start menu > Settings > System > Display > Scale & Layout). There were several problems how FLIDAS reacted to this external scaling that should be fixed now. If you find remaining problems, report them. Note that you should restart FLIDAS after changing the scaling. There is also a preference page for handling possible problems. You can also reset individual configuration of table column widths there.
- TSPI format 1 packets will be decoded into separate packet entries. This is only basic decoding as the specification in IRIG 106 still needs improvement.

New features in the basic commercial plug-in:

- TSPI format 1 packet entries can be displayed in a structured way.
- TSPI format 1 data can be exported to CSV.
- The XML CH10 mapping can include the XML source code into a generated CH10 file as computer-generated user defined data.
- The binary content of a selected packet can be exported to a file. This can be used in different situations but also to extract the XML source code mentioned in the previous item.

There are other smaller changes and corrections.

10.10 FLIDAS 4.9

This version primarily adds minimum and maximum definition to parameters.

New features in the parameter plug-in:

- Most ICDs now allow defining a minimum and maximum value for numeric or enumeration parameters. If these remain undefined, the values are calculated to the maximum range. For this purpose new columns have been added to the CSV ICDs, mostly creating a new file format version. Migration is easy by adding the two columns. Old file format versions stay supported but do not allow definition of minimum and maximum but still provide the calculated minimum and maximum. Affected ICDs are:
 - MIL 1553
 - ARINC 429
 - CAN bus
 - Analog
 - UDP
 - Message data
 - NMEA 0183 hardcoded (has fixed minimum and maximum)
 - PCM (CSV version only - TMATS version only has calculated minimum / maximum)
 - Recording events (unchanged version number)
 - UART ASCII (unchanged version number; no calculation of minimum and maximum if omitted)
- The parameter graph view can now scale the values to the minimum and maximum defined in the ICD.

- The following gauges can pick up the minimum and maximum defined in the ICD:
 - Strip chart
 - Bar graph
 - Dial
- There is a preference whether to prefer automatic scaling of values (from observed minimum and maximum) over ICD scaling (minimum and maximum defined in ICD).

New features in the parameter streaming plug-in:

- The parameter graph live processor can now scale the values to the minimum and maximum defined in the ICD.

New features in the free version:

- CH10 publishing can loop after the end of the file is reached. This creates a stream where the time jumps back in the end which is not supported by FLIDAS. However, a customer requested this for publishing to other receiving software.

There are other smaller changes and corrections.

If you have issues with licensing: The calculation of minimum and maximum values for parameters uses an external library for higher degree polynomials. So, the [Apache Commons Mathematics Library](#) has been added (in the commercial parameter or parameter streaming version). It uses an open-source license. The licensing information on FLIDAS has been updated.

10.11 FLIDAS 4.8

This version improves support for message data type and IENA.

New features in the basic commercial plug-in:

- IENA data in the message data type is supported in the following areas. The message data must directly contain the IENA packet without further wrapping.
 - IENA communication summary
 - IENA to CSV export
- The IENA to CSV export can now optionally export the payload bytes as well
- The IENA communication summary now displays the bit rate of each IENA key and in total
- The PCM communication summary now calculates the percentage of time that no frame lock existed

New features in the free version:

- The channel activity live processor now indicates recent activity or errors with color codes

New features in the parameter plug-in:

- There is a simple CSV ICD to decode message data type to parameters. A message ID is used to distinguish messages and parameters can be mapped to fixed positions.

There are other smaller changes and corrections.

10.12 FLIDAS 4.7

This version improves support for CH7 analysis and adds support for TSPI NMEA 0183.

New features in the basic commercial plug-in:

- The CH7 protocol can be summarized on the source packet level. You can see the streams, content types and even chapter 10 channels contained with their bit rate, protocol errors, low-latency status and more.
- TSPI / CTS format 0 / 2 data can be exported to CSV.
- The NMEA 0183 protocol viewer and NMEA 0183 to CSV exporter now support TSPI format 0 sources as well.
- The previous version improved the robustness of NMEA 0183 decoding by always requiring a checksum if a checksum has previously been detected on the input. This has been relaxed to require it only for the sentence types that had checksums before.
- The IRIG 106 Chapter 7 decoding was changed to detect more protocol errors and reject them earlier. In effect it is possible to decode more data on some streams but it also shows that some current hardware does not respect the specification (encapsulation packets of different source packets are mixing. The new CH7 summary will reveal this.).

New features in the free version:

- TSPI / CTS packet formats 0 and 2 are decoded to the packet entry level. Format 1 is not supported as the Chapter 11 definition is not done properly so it is undecodable without additional information. If you can provide sample data or can give information on it, please get in touch. The support can then easily be done.
- Time channel offsets can now be predefined in the preferences to easier reuse them when opening a CH10 file or creating a live data stream.
- The performance of exporting data to files on network drives has improved significantly for most exporters.

New features in the parameter plug-in:

- The Navigation Data View and NMEA 0183 hardcoded ICD now support TSPI format 0 (NMEA 0183) sources as well.
- Several improvements in NMEA 0183 hardcoded ICD:
 - HDOP, PDOP and VDOP parameters added
 - Latitude, Longitude and UTC time is now also decoded from GLL sentence
 - Some improvements on parameter validity depending on FAA mode etc.
 - The parameter "Satellites in View" has been renamed to "Satellites Used". If it is contained in your Parameter Sets or Instrument Panels you need to update it.
 - A bug was fixed that decoded latitude always with a positive sign, even for the southern hemisphere

New features in the commercial streaming plug-in:

- The CH7 Live processor had some changes:
 - Streams with detected errors show in red. (Note that the live processor processes data on the encapsulation packet level and not source packet level. The CH7 summary on a recorded file will detect more error types)
 - The throughput column has been renamed to bit rate and is measured in MBit/s

There are other smaller changes.

10.13 FLIDAS 4.6

This version improves support for distributed CH10 multiplexer sources

New features in the free version:

- FLIDAS now supports CH10 files and live data streams from sources with independent real time counter (RTC) timing. The source must be indicated in the upper bits of the channel ID (multiplexer source ID, MSID). For each MSID there must be a time channel. Live data streams will drop packets from each MSID until a valid time packet for the MSID was received. The TMATS configuration must still be done in a single TMATS with a single R-group (either from CH10 file / stream or as external TMATS file). When opening a file or creating a live data stream you can choose to ignore the separate MSID handling which essentially brings you back to the way FLIDAS worked before.
- FLIDAS now uses secondary header times instead of RTC time if secondary headers are present. For channels using secondary header timing, no other RTC based time channel is required. Secondary header checksum checking is now automatically enabled. You can disable secondary header time use when opening a CH10 file or creating a live data stream. This brings you back to the old FLIDAS behavior.
- You can configure time offsets between time channels. This allows fixing time offsets between various time channels and secondary header time sources especially leap seconds offsets but also local time can be adjusted. Although this feature is targeted at multiple time source scenarios (i.e., distributed systems) you can also offset the time in a single time source scenario.
- The filtered CH10 exporter now includes all the time channels in the MSIDs of the selected channels.
- Some rare issues with live data stream timing are fixed
- RTC sync errors in the packet header now prevent decoding of RTC timing. Instead a fallback time will be used.

New features in the commercial streaming plug-in:

- You can replay audio from analog data types in a live data stream to your speakers.

New features in the basic commercial plug-in:

- The robustness of NMEA 0183 decoding has improved. If on any NMEA 0183 source a checksum is detected (by protocol it is optional) it will be required for the rest of the data or the sentence will be considered to have a wrong checksum.

There are other smaller changes and bug fixes.

10.14 FLIDAS 4.5

This version delivers first features to support CH10 files from independent CH10 multiplexer sources.

New features in the free version:

- CH10 files that merge data from independent CH10 multiplexer sources and use the multiplexer source IDs (MSIDs) to distinguish sources can be split into separate files that can then be properly processed by FLIDAS and possibly other software. The merged files are often no valid CH10 files as they have independent relative time counters (RTCs) and time decoding is not (yet) possible with FLIDAS. After the file is split the processing of time is possible. Files of this kind are appearing more, for example when merging ground recording and airborne CH10 data acquisition transmitted to the ground recorder as CH7 PCM. Within the IRIG 106 standard groups there are discussions (and Data Bus Tools is involved) to give recommendations on this. It is intended to deliver more support for this in FLIDAS.
- The "number of source bits" used to decode the multiplexer source IDs (MSIDs) are now shown in the General TMATS Properties.
- Many internal changes in FLIDAS reduce the dependency on the real time counter (RTC) in order to increase support for independent CH10 multiplexer sources in one file or stream. This will also help with a future "Version 2 Packet header" that is in discussion within the IRIG 106 standard group and that might drop the use for RTC completely for these packets in favor of an absolute time.
- CH10 files can use IRIG 106 Chapter 4 time stamps in secondary headers or packet entries. These are not properly defined in the standard and therefore cannot be safely used as time reference. In the past users could choose to interpret the time stamps as time since new year (choosing an option during file loading or live data stream set up). This option does no longer exist and the time will always be interpreted this way. This is the only implementation seen so far and there is no safe meaning for the time stamp anyway, not even the way of interpreting it as time without date like done before by default.
- Jump to a recording event in the Packet View of a single channel

New features in the parameter plug-in:

- Allow parameter decoding from incomplete MIL 1553 messages. Messages with a missing status word or incomplete word count often still come with decodable data or status words. You can now indicate in the ICD that you want to decode data also from faulty messages. This especially enables you to decode data from a transmitter when the receiver is offline and does not confirm reception.
- Bugfix for MIL 1553 status word parameter decoding. In the past, status words from RT addresses > 15 were not accessible.
- Jump to a recording event in the Replay Control Gauge and Parameter Graph View

- Jump to a time marker or recording event in the Navigation Data View and Parameter Table Viewer
- PCM parameters are now supported up to 64 bit

New features in the basic commercial plug-in:

- The Recording Event Packet Entry Viewer now also displays the name and description of events.
- The MPEG Transport Stream Summary no longer displays the minimum and maximum period between two transport stream packets of the same PID (as the timing in CH10 is not as expected here).
- The MPEG Transport Stream Summary displays the bit rate per PID.
- Jump to a time marker or recording event in the NMEA 0183 Protocol Viewer.
- The Hex Packet View does more coloring of sections for some packet types.

There are other smaller changes and bug fixes.

10.15 FLIDAS 4.4

This version improves usability for instrument panels and parameter graph viewer.

New features in the parameter plug-in:

- You can speed up or slow down the replay of data in instrument panels
- You can create time markers or jump to time markers during the replay of data to instrument panels
- In the parameter graph viewer you can step through the data sample by sample using the cursor keys. To support this a new sampling mode is available where every sample is included in the graph and not only the changed samples.

There are other smaller changes and bug fixes.

10.16 FLIDAS 4.3

This version mainly improves instrument panels and offers a new type of external live processing support.

New features in the parameter plug-in:

- There are some new gauges for instrument panels:
 - Table gauge
 - Airspeed indicator
 - Altimeter

New features in the data streaming plug-in:

- A customer wanted to process live PCM data outside of FLIDAS with an own application. He did not want to implement the access to the IRIG 106 CH10 UDP data stream. He asked for an easy solution to stream out data from FLIDAS. FLIDAS can now stream out raw data as a UDP stream. The payload of the UDP stream is just the

raw data without any additional protocol. This way own applications can just connect to the UDP stream and easily process the data. The feature is available for:

- PCM throughput channels
- UART subchannels

FLIDAS relies on customer requests for new features. So, if you are in need for some features that look generic enough for other user as well, suggest them. If you are a customer within the upgrade program you will also get your required feature for free if it is selected for implementation within your upgrade period. For non-generic feature requests you can of course also order customization plug-ins.

There are other smaller changes and bug fixes.

10.17 FLIDAS 4.2

This version mainly improves instrument panels.

New features in the parameter plug-in:

- A strip chart gauge has been added for instrument panels.
- Analog full live decoding parameters: In the past analog data was not fully decoded to parameters when processing live data (i.e., instrument panels or parameter live processors - raw data processing and non-live processing was always complete). The reason was that this potentially high sample rate data can cause performance problems for decoding of data and continuous display updates although the observer on the screen does not see the rapid changes anyway on current value displays. Only one sample was decoded from each CH10 packet (i.e., at least one every 100 ms). FLIDAS now adds the option to use full decoding in some places where it makes sense. For each analog parameter you will now see a second parameter with the extension ".full" which will do the full decoding. This makes sense only for displays that show a history of values over time (i.e., a curve). Still high sample rates and/or long time spans of display can cause performance issues. Use of these new parameters is optional and you can still use the limited decoding as well. The new strip chart gauge can use these parameters.

New features in the parameter streaming plug-in:

- The parameter graph live processor can now use analog full live decoding parameters (see above).

Free version:

- When publishing a CH10 file you can now publish a subset of channels or a time slice.
- The XML CH10 mapping (converting CH10 files to readable XML and back for manipulation, creation of artificial files etc.) has some new elements:
 - Ethernet packets now have a structured XML representation wrapping around the MAC frame.
 - PCM data now has a structured XML representation wrapping around throughput data or packed/unpacked minor frames.

- NMEA0183 messages can now be explicitly defined and use an auto generation for the checksum, carriage return and line feed. This is useful when manually defining or manipulating GPS data in UART messages.

New features in the basic commercial plug-in:

- When converting CH10 files to XML using the CH10 XML mapping, users can now suppress the generation of detailed XML for data types they choose. This leads to shorter, faster to convert files. The data will still be defined but as simple raw data.

There were several bug fixes, especially in the area of analog data decoding. Most of them were in really unlikely data formats that have never been seen in real files (sample bit sizes > 32 bit or unpacked LSB padded mode). Also, parameter decoding (not raw data processing) of 32 bit analog parameters was wrong when the MSB was set. It is unclear if there are any 32 bit analog CH10 devices on the market.

There is another bug fix about the treatment of PCM throughput parameters during CH10 file replay to instrument panels when changing the replay position.

There are other smaller changes and bug fixes.

10.18 FLIDAS 4.1

This version extends instrument panels to the use with CH10 files (not only live data streams) and adds new gauges.

New features in the parameter plug-in:

- Instrument panels can now also be used with CH10 files.
- An attitude indicator (artificial horizon) and a map gauge has been added for instrument panels.
- The parameter tutorial in the manual now teaches the use of instrument panels with CH10 files.

New features in the parameter streaming plug-in:

- The parameter graph live processor can now also display absolute timestamps instead of relative ones.

New features in the data streaming plug-in:

- All raw data live processors can now also display absolute timestamps on the curve instead of relative ones.

New features in the basic commercial plug-in:

- A bug was fixed in the NMEA 0183 sentence processing. When only the last two bytes of a sentence were recorded, the processing terminated before.

There are other smaller changes.

10.19 FLIDAS 4.0

This version adds user defined instrument panels with gauges to display live parameters.

New features in the parameter streaming plug-in:

- Users can create instrument panels and add gauges to it to display live parameter values. The instrument panels are saved and can be associated with an ICD mapping and reused on different files. Gauges available so far are:
 - clock to display the live data stream time
 - bar graph
 - dial
 - simple value display
 - compass rose

Note that you can replay CH10 files locally and receive them as live data stream to use instrument panels. Possibly future FLIDAS versions will work directly with instrument panels on CH10 files.

- In case you miss the save buttons in the ICD mapping view and the parameter set view: They moved to the save symbol in the upper right of the views.

New features in the parameter plug-in:

- Asynchronous parameter sampling produced several lines of results for the same time stamp if parameter existed at this time on different channels. Now these are joined to one line of result.

There are other smaller changes.

10.20 FLIDAS 3.18

This version mainly improves IRIG 106 chapter 7 support.

New features in the commercial data streaming plug-in:

- CH10 data from a live IRIG 106 chapter 7 PCM stream can be published as CH10 UDP stream on a network.
- TmNS messages (Telemetry Network Standard, IRIG 106 Chapter 24) from a live IRIG 106 chapter 7 PCM stream can be transmitted as UDP packets on a network.
- You can inspect a live IRIG 106 chapter 7 PCM stream and show information from it.

New features in the basic commercial plug-in:

- The hex packet view has been improved to show the detailed section of each position within a CH10 packet (like "these bytes store the RTC") as well as the selected offset within the section and packet.

Free version:

- The Active Data Stream Tasks View has been removed. Instead the Data Stream Tasks are organized below its source in the FLIDAS Repository.
- For specific uses of incoming CH10 UDP data streams suffering from power losses there were improvements. When a device continues transmission after a power loss it will transmit with inconsistent sequence numbers. FLIDAS now accepts this after a configurable gap so for example a recording can keep on recording the situation around the power loss. Before FLIDAS rejected the inconsistent sequence numbers. But after a power loss also the RTC values of the CH10 packets will be inconsistent. FLIDAS will try to detect this and report it although it can do nothing to correct it. A new verification check will also try to detect this situation.

In general, *Live Displays* have been renamed to *Live Processors* as more and more features do background processing rather than displaying something.

There are other smaller changes.

10.21 FLIDAS 3.17

This version mainly updates to the recently released 2019 version of IRIG 106.

Free version:

- MIL 1553 16PP194 packets will be decoded to packet entries.
- The new IRIG106-19 PCM DQE packets have been added to be decoded on the packet level.
- The XML CH10 mapping has been updated to support the IRIG106-19 revision flags
- The XML CH10 mapping does no longer allow to validate XML files against an external XSD file (the conversion was based on the internal XSD file anyway).

New features in the basic commercial plug-in:

- There is a CH10 format verification check for the PCM DQE packet introduced in IRIG106-19.

New features in the commercial parameter plug-in:

- The tutorial for parameter decoding has been updated in the manual. It is now based on PCM decoding and comes with sample CH10 and ICD files.

There are other smaller changes.

10.22 FLIDAS 3.16

This version mainly improves the quick look capabilities in live streaming.

New features in the commercial data streaming plug-in:

- Live display of discrete raw data including a graph of a single discrete.
- The video live streaming can be configured with a UDP port address, so you can see multiple videos at the same time in an external application.

- The raw data live displays for PCM, MIL-STD 1553, ARINC 429 and Analog can now apply a user defined scaling.

New features in the basic commercial plug-in:

- The video replay from file can be configured with a UDP port address to stream to an external application.
- Some PCM throughput recordings seem to stop the recording if no stable data is received (instead of recording more or less random data like other recorders). IRIG 106 does not define how this should be. FLIDAS will now try to detect gaps in the PCM stream, go out of frame sync and sync in again. In result the PCM bit rate must now be indicated in TMATS.

New features in the commercial parameter streaming plug-in:

- The parameter graph live displays can now apply a user defined scaling.

New features in the commercial parameter plug-in:

- Workaround for a bug caused from some OpenJDK Java versions: Printouts of the parameter graph in landscape orientation had an improperly positioned text output.

Free version:

- The Packet entry view can now be filtered to a specific subchannel.
- When opening a CH10 file or creating a CH10 live stream you can explicitly select which data source of the TMATS section you want to use in case there are more. If you configure nothing FLIDAS will use the first available data source.
- There are some more consistency checks on TMATS. The channel view will display the channel related problems as before but now distinguish between problems that might restrict FLIDAS features (displayed in red) and inconsistencies that will most likely not cause any problem for FLIDAS (displayed in yellow).
- Fixed leap year problem for recorders recording the year wrongly as 2000 in some cases.

FLIDAS now gives more details on [third-party licenses](#) that apply to the underlying Eclipse framework.

There are other smaller changes and bug fixes.

10.23 FLIDAS 3.15

This version mainly adapts FLIDAS to newer Java versions.

10.23.1Java topics

Summary:

- FLDIAS now runs on Java 9 and later as well

- The commercial FLIDAS license had to be changed in the area of support for old Java versions
- Java 8 as well as 32-bit Java support may be discontinued after September 2019

In more details:

In the not widely adopted Java 9 some things changed that made some older application not run out of the box as it was always experienced before. This was an attempt to straighten some things up in Java. In order to support the newer Java version, FLIDAS needed to migrate to a newer version of the underlying Eclipse Platform. It was migrated to the newest version that still officially supports 32-bit Java versions.

Another Java issue is that the Java update policy of Oracle changed. They have now completely open-sourced Java and it is easy to get releases from different distributors based on the same source code. The previous commercial FLIDAS license had a clause that newer FLIDAS releases may terminate support for Java versions that were beyond the Oracle's "end of public updates" date. This date has come for Java 8 now (but FLIDAS keeps supporting it for now). This date has also passed for Java 9 and 10. A definition based on "end of public updates" does not make sense anymore in the new Oracle Java update policy as this would mean it runs out for each version half a year after release. **The commercial FLIDAS license has changed now.** The support for any Java version may be discontinued one year after a new Oracle LTS (long term support) version is released. This should be every three years and will give everyone one year to migrate to the latest LTS version. And of course you are then still allowed to use older FLIDAS versions with the older Java versions (see license section in manual).

Although the support guarantees are based on the Oracle dates, you are not limited on Oracle Java. Oracle will require a commercial Java license for commercial use or the Oracle JDK for the future. You can get free versions based on the same source code for example even from Oracle from another website than before: jdk.java.net. An example for other builds of the same source code is adoptopenjdk.net. Both builds come without installer. You can even integrate them in the FLIDAS directory without installation if you follow the manual chapter 5.1 for portable installation. For this it does not matter if you download a "JDK" or "JRE".

Oracle is still contributing development even to the open source Java and there are other big companies like IBM supporting it. There is no worry about obsolescence.

Since Java 9 only 64-bit Java versions are distributed by Oracle. This means that by the new FLIDAS license the support for the last 32-bit version of Java 8 could be terminated one year after the JDK 11 LTS (long term support) version was released. **This means 32-bit FLIDAS support is not guaranteed beyond September 2019.** This does not mean it will be immediately dropped but there is no guarantee anymore.

To clarify this: This FLIDAS release just allows using Java 9 or later. You can still stay on Java 8 if you prefer.

10.23.2 Other changes

New features in the basic commercial plug-in:

- Users can configure time stamp output, especially for CSV exports.
- Communication summaries for message data channels to show cumulated errors, message periods, lengths etc.
- Payload exporting for message data subchannels.

Parameter plug-in:

- Fixed a bug where separate time stamp asynchronous sampling was not possible anymore for some types of data (used in multi-file CSV or MAT export).

Free version:

- Preferences can be ex- and imported.

There are other smaller changes and bug fixes.

10.24 FLIDAS 3.14

This version mainly improves raw data export and supports more data types.

New features in the free version:

- The packet entries of message data packets will be decoded.
- Fibre channel (PH layer, format 0) packet entries will be decoded (FS layer, format 1 was supported before already).

Basic commercial plug-in:

- Export UART packet entries with their properties to a CSV file.
- Export data from a message data channel to a CSV file. Segmented messages will be joined.
- Export fibre channel packet entries (FS or PH level) with their properties to a CSV file.
- The Ethernet to PCAP export has been extended to export also when only the Ethernet frame's payload was recorded. In this case the PCAP file assumes there is an IPv4 or IPv6 content.

There is one smaller bug fix.

10.25 FLIDAS 3.13

This version mainly improves support for embedded protocols (IENA + Chapter 7) and CH10 verification.

New features in the basic commercial plug-in:

- CH10 verification:
 - New data type specific checks for almost all data types (only for the channel specific data words for now)
 - Existing checks have been improved

-
- More general checks
 - IENA packets can be exported to a CSV file (no payload, just protocol information for analysis).
 - A new group of exporters extracts data from a PCM channel with IRIG 106 chapter 7 data. The support is up to the 2017 standard but without Reed-Solomon error correction (ask if you need this).
 - Export of CH10 data to a CH10 file
 - Export of Ethernet frames to a PCAP file
 - Export of IP packets to a PCAP file
 - The TMATS replace reporter now supports injecting a TMATS record in a file without a TMATS record.

Free version:

- The XML CH10 Mapping now allows the includeFile tag everywhere, where raw data could be used before. Before it was only possible on TMATS packets. Now it can be used for example to load image files into image data packets or even to merge whole CH10 files into other ones. XML files following the previous standard will still be converted.

There are some other smaller bug fixes and improvements.

10.26 FLIDAS 3.12

This version mainly improves protocol error analysis and localization and adds some IENA support.

New features in the basic commercial plug-in:

- Most communication summaries have been extended to collect information of protocol errors and special events. These are displayed with their time stamps and more details on a new page of the communication summary. It is possible to navigate from there into the actual CH10 packets / packet entries to see all the details and see the event in its temporal context. This feature has been added for these communication summaries:
 - MIL-STD 1553
 - ARINC 429
 - CAN bus
 - UART
 - Ethernet
 - PCM
 - Time
 - MPEG transport stream
 - IENA
- There is a new IENA Communication Summary to summarize IENA messages and errors. IENA is an Airbus Ethernet protocol but also used by Curtiss-Wright / Acra FTL. Take care to configure the MAC / IP / Port filters correctly for your data if you do not follow the Airbus specification.
- The PCM communication summary now also summarizes frame sync losses in PCM throughput mode.

Parameter plug-in:

- The support for some different floating-point encodings has been added: MIL-STD-1750A 32- and 48-bit floats and VAX / DEC "D" 64-bit floats are now supported for PCM, MIL-STD-1553, UDP and CAN bus ICD types.

Free version:

- You can configure the numeric output for some parts of FLIDAS in the General Preferences. This will provide scientific notation where very long numbers have been used before. You can manually switch it off if you require a compatible output to old FLIDAS versions.

There are some other smaller bug fixes and improvements.

10.27 FLIDAS 3.11

This version mainly adds an ICD format to decode text-based UART communication to engineering values.

New features in the parameter plug-in:

- Text-based UART communication can be decoded to parameters. Since there is no high-level specification how UART should be decoded the ICD makes some basic assumptions that hopefully cover a wider range of applications. If it does not fit to your UART protocol, feel free to write an email so the ICD can be further generalized.
- Most ICDs now support character parameters using 7-bit ASCII. The ICDs that implement this are:
 - MIL-STD 1553 Simple CSV
 - CAN Bus Simple CSV
 - Ethernet UDP Simple CSV
 - PCM Simple CSV
 - ARINC 429 Simple CSV
- In the Parameter Graph View you can manually configure the minimum / maximum value to be displayed for each curve.
- You can define a sequence of bytes within a PCM minor frame as a "Byte Array" with raw data in the PCM Simple CSV ICD. This gives access to the raw data for more specialized features. An already prototyped feature based on this is a PCM video live replay for a specific camera. Since this is a proprietary standard it will not be part of the standard FLDIAS. However, the Byte Array is a standard feature and may be used in other situations as well.
- The new Byte Array parameters from the PCM ICD can be exported to a binary file. FLDIAS does not care what type of data it is. In the above example with the camera it leads to a simple export functionality for the video that can be replayed with standard video players.

Basic commercial plug-in:

- When UART is displayed as text in the UART Text Protocol Viewer or UART Packet Entry Viewer, all line breaks will be displayed as line breaks, even if they are from other platforms (i.e., Linux / Mac style line breaks will also be displayed on Windows). Existing UART protocols seem to use all kind of possibilities to terminate a line.

Free version:

- Secondary header binary chapter 4 time stamps are treated differently now. They are not fully specified in IRIG 106, as regards to what the time stamp relates to. In the past FLIDAS just decoded the time and not the date part and created a time stamp related to the date of the rest of the recording. Now there is a GUI option when loading CH10 files or creating live data streams to assume that the time stamp contains the microseconds since new year and therefore also the day of year. For some recorders this might be the fact and some recordings contain chapter 4 time stamps significantly different from the rest of the file.
- The XML CH10 Mapping XSD file has been updated. For the above mentioned binary chapter 4 time stamps it is now possible to define a day component of the time stamp that is larger than 255. This is another deviation from the officially published XSD file in the RCC documentation. But still XML files conforming to the RCC documentation can be converted to CH10 with FLIDAS. FLIDAS will just also convert properly in more cases and offers language extensions.

There are some other smaller bug fixes and improvements.

10.28 FLIDAS 3.10

This version mainly adds CH10 file format verification and improves file debugging support.

New features in the basic commercial plug-in:

- The format of CH10 files can be verified (same functionality as CH10 validators). A lot of general checks have been implemented and some checks focusing on the new features of IRIG 106-15 and IRIG 106-17 to support verification of recent recorders. The difference to existing validators is that FLIDAS already has a lot of debugging features (like inspecting packet content and structured hex packet view) and now directly links from the GUI list with the detected errors to all these features for easy debugging. Existing customers had troubles with recorders before and had to spend a lot of time to investigate and prove these problems that can now be automatically detected. Of course, this is also useful to recorder developers themselves.
- The structured hex packet view has been improved in several ways:
 - It is now possible to open the view from a packet entry and highlight the data that belongs to it.
 - The packet trailer visually distinguishes filler and data checksum bytes
 - Recording indices and events now also show the separate packet entries
- There is a new packet entry viewer for fibre channel FS level (format 1).
- There is a summary for recording events.

Free version:

- Fibre channel FS level packets are now decoded to packet entries.
- Fixed potentially wrong display of time stamps in recording index root index.
- Fixed wrong display of a specific corrupted MIL-STD 1553 message type (RT-RT messages with no TX response displayed a fake RX status but were clearly indicated as invalid)
- Fixed wrong method of secondary header checksum calculation

There are some other smaller bug fixes and improvements.

10.29 FLIDAS 3.9

This version mainly improves support for Linux and adds features for UART decoding.

New features in the free version:

- FLIDAS has been adapted to the latest version of the underlying Eclipse RCP framework. This solved a lot of Linux GUI issues related to GTK3. Recently there has been a request of an existing customer for Linux use of FLIDAS.
- The CH10 UDP data stream format 3 (introduced in the IRIG106-2017 release) reception has been adapted to clarifying explanations of the author.

Basic commercial plug-in:

- When viewing UART packet entries the entries can be decoded to text for a quick look with different text encodings.
- The UART data of a complete subchannel can be shown in a separate text display with selectable text encodings.
- The Ethernet communication summary has been adapted to support IP4 fragmentation on the UDP packet analysis. Also, some new protocols are detected.

Parameter plug-in:

- The parameter graph view now shows the full selected time range instead of the time range from the first to the last sampling.

There are some other smaller bug fixes and GUI adaptations.

10.30 FLIDAS 3.8

This version offers support for PCM bit error rate tests and adds the missing IRIG 106-17 XML TMATS support.

New features in the free version:

- CH10 Files can be opened directly by double-clicking them in the Windows Explorer (see the installation section in the manual).
- The IRIG 106-17 XML TMATS specification has now been released by the RCC about 3 month later than the rest of the standard. FLIDAS now supports the changes.
- A bug has been fixed to allow to connect the same TMATS P section to multiple PCM Channels in the R section.

Basic commercial plug-in:

- A new PCM BERT Protocol Viewer checks a recorded PCM throughput Channel for conformance with a pseudo random bit stream from a bit error test. PN11 and PN15 are supported.

Data Streaming plug-in:

- A new PCM BERT Live Display checks an incoming PCM throughput Channel from a Live Data Stream for conformance with a pseudo random bit stream from a bit error test. PN11 and PN15 are supported.

Parameter plug-in:

- In some situations supercommutated PCM Parameters produced one value too much per minor frame. This value has been flagged as invalid before. Now the bug is fixed and the correct number of values will be created.
- Enumeration Parameters can now also be used with their raw value for display in the Parameter Graph View and for export in the different Parameter to MATLAB Exporters.

Parameter streaming plug-in:

- Enumeration Parameters can now also be used with their raw value for display in the Parameter Graph Live Display.

There are some other smaller bug fixes.

10.31 FLIDAS 3.7

This version adapts to the IRIG 106-17 release and offers some XML CH10 mapping extensions to allow hardware and software developers creating test data for IRIG 106-17.

Since the IRIG 106-17 specification has not been completely published (the XML TMATS is delayed), all features have only been implemented without the TMATS XML 2017 support. This will most likely be added with the next version.

New features in the free version:

- Support for network time packets (IRIG 106-17).
- Support for streaming configuration packets (IRIG 106-17).
- Support for fibre channel FS layer packets (IRIG 106-17). Only packet level decoding, no packet entries.
- Support for UDP Data Streaming format 3 (IRIG 106-17).
- Added IRIG Time Source property to normal time packets (IRIG 106-17).
- Correction for time packets: The standard requires that packets with a time format of "None" are invalid. The time stamp in these packets is now ignored.
- The maximum sizes of UDP packets in the UDP data streaming has been increased in IRIG 106-17. The FLIDAS preference for the default maximum reception size has been changed to work with new data stream publishers. However, this comes with a

performance overhead so you might want to reduce it if your publishers do not send large UDP packets. If they do and you have changed the default value in the past, then you might need to adapt your value manually to the new maximum now.

- Added features to XML CH10 Mapping. This means the implemented XML CH10 mapping now no longer conforms to the RCC published version but extends it. The RCC only updates this about every four years but we want to keep this new technology evolving. Since the existing RCC document is solely based on our specification we will offer the extensions to the RCC but there is no guarantee that they accept it. Files valid by the official specification remain valid under the extended one. We keep a free download of the FLIDAS version implementing the official release downloadable. The new features are:
 - Network time packets (IRIG 106-17)
 - added IRIG Time Source attribute to Time Packet definition (IRIG 106-17)
 - Added version markers for IRIG 106-17
 - Intra packet time stamp generation within raw data sections
 - Corrected use of ERTC definition. Now a relative definition of ERTC reaching backward in time is possible

Basic commercial plug-in:

- Support for alternate code complement PCM frame synchronization (IRIG 106-17).

10.32 FLIDAS 3.6

This version focuses on parameter decoding for new data types: CVSD, Ethernet UDP and discretes.

Especially the CVSD audio features could only be implemented because a customer provided sample data for it. Often the implementation of new features requires samples and will be implemented in the regular versions of FLIDAS without additional payment from customers. So, if there are generic feature requests that you can support with sample data, please provide it. The Ethernet UDP and discrete features were only implemented because of requests of a prospect customer also without payment.

The new features in more detail:

Parameter plug-in:

- The PCM CSV and TMATS ICDs have been extended to support CVSD audio parameters.
- CVSD audio can be replayed from CH10 files to the speakers.
- CVSD audio can be exported to wave files.
- There is a new CSV ICD format for discretes. In results this data type can be used with all parameter-based features.
- There is a new CSV ICD format for UDP on Ethernet. In results this data type can be used with all parameter-based features. Since UDP is not defining anything about the layout of avionics data please consult the manual of what can be achieved with it. It is intended to reach the same amount of flexibility as with TMATS definitions for Ethernet parameter decoding. Send feature requests if you need more.

Parameter streaming plug-in:

- CVSD audio can be replayed from a live data stream to the speakers.
- The PCM parameter decoding on Live Data Streams now supports supercommutation as well.

Basic commercial plug-in:

- Exporting of discrete raw data to CSV files.
- PCM to CSV raw data exporting can now include lines for the loss of minor frame synchronization (on PCM throughput mode)
- The PCM packet entry view now includes lines for the loss of minor frame synchronization (on PCM throughput mode)

Free version:

- A new manual section (in the Data Streaming reference) covers the analysis of data stream problems. Data streams strongly depend on preconditions external to FLIDAS so this gives guidance in tracking down the source of a problem if you cannot properly receive data

There is also one important bug-fix on the PCM decommutation in the packed mode. The last PCM word of each minor frame in packed mode has been decoded wrong if the bits of the minor frame were no multiple of the used alignment of the recorder (mostly 16-bit, sometimes 32-bit). In reality most people seem to use the throughput mode of PCM anyway.

10.33 FLIDAS 3.5

This version focuses on usability improvements and frequency spectrum analysis for analog live data. There are also new system requirements.

Free version:

- FLIDAS is now using the latest Eclipse framework under the surface. This was needed to catch up with bug-fixes provided there. This comes with a change in the system requirements as we announced last September. FLIDAS now requires **Java 8**. Keep in mind that Oracle stopped security updates for Java 7 two years ago. Also, the Eclipse framework is no longer tested on a **Windows XP / Vista** configuration so this is no longer officially supported by FLIDAS (Microsoft stopped security patches for Windows XP three years ago and for Vista lately). Although Eclipse is no longer tested under this configuration it will highly likely still run on it. Also, Java 8 may run on Windows XP but receives no support as Oracle says. If you run FLIDAS on this configuration you do it on your own risk. You may keep and use older version of FLIDAS as long as you comply with the license terms (concurrent use restriction).
- As a result of the new Eclipse framework the help system is back to normal available in a separate window so you can easier keep it open and navigate while working. In addition most dialogs are able to display context help inside the dialog so the help is not covered by the dialog.
- The workflow on exporting, protocol viewing etc. has become easier. You can now multi-select channels and subchannels before opening the wizards and the further

configuration will be skipped if it is not possible to include further channels / subchannels.

Basic commercial plug-in:

- The commercial FLIDAS license has changed (see manual for details). It now defines when Java / Windows configurations become unsupported for upgrades and grants customers the right to keep on using old versions on their old system configuration.

Data Streaming plug-in:

- Analog Raw Data Live Display: You can now analyze the frequency spectrum of incoming analog live data using an FFT.

There are also several bug-fixes and robustness improvements against invalid files. Some of the bug-fixes are:

- Use of CH10 Data Type filters prevented CH10 file loading.
- Lost CH10 analog packets could lead Analog Raw Data Live Display to stop working.
- Live Data Streams that haven't used a TMATS override file could display false overload indications and display no data in Raw Data Live Displays. This happened only under further rare conditions.

There has been a clarification on the exported time stamps created using the Time Representation Wizard Page. If these indicate nanoseconds since something then this will always be based on an 86400 seconds day and not be modified by leap seconds. If the date represented by this uses leap seconds or not depends on your time source.

10.34 FLIDAS 3.4

This version focuses on new Live Display capabilities and commercial features for the XML CH10 Mapping. It also improves the performance.

Free version:

- The performance of the initial file loading has improved in many situations. The performance increase depends on the individual properties of the CH10 files and the used computer environment. The focus was to improve performance of CH10 files that consist of a lot of small CH10 packets and during testing the loading times dropped here to 40%. Also, the loading time of CH10 files located on network drives dropped to 15% for small packet files and 70% for big packet files.
- A small but useful feature allows copying a single table cell of almost all tables to the clipboard so you can easily transfer it to other applications.
- Added a question mark symbol to most dialogs to get context help. This shows the same help page that you would have found with the F1 key before.

Basic commercial plug-in:

- Exporting CH10 files to XML (using the CH10 XML Mapping) has now additional options to influence the generated XML code. The options now also allow for a

binary identical conversion cycle CH10 > XML > CH10 even in the special cases where before only equivalent files have been created. Also, some options generate code that is easier modifiable if you want to manually modify the XML to create modified CH10 files.

Data Streaming plug-in:

- Analog Raw Data Live Display: This Live Display shows a graphical plot of a single Analog Subchannel from an incoming live stream.

There are further performance improvements in the area of creating MATLAB (MAT4) files and in analog sample handling.

10.35 FLIDAS 3.3

This version focuses on new Live Display capabilities and introduces TMATS ICD definitions.

Parameter plug-in:

- PCM ICDs can now also be defined in ASCII TMATS as an alternative to a CSV file. The TMATS must be mapped from a separate file. This is a first cut on TMATS ICDs. It does not support all the features as sometimes the TMATS specification is not clear enough for an implementation or the use of an option seems to be quite small. This ICD should be a starting point growing with user requests. Later perhaps other data types than PCM will also be supported by TMATS ICDs.
- The detailed Parameter definitions can be inspected in the Parameter List View and the Parameter Set View.
- Bug fix: Since last version the rarely used ARINC 429 Discrete data type was not working. It has been fixed.

Data Streaming plug-in:

- ARINC 429 Raw Data Live Display: This Live Display shows information on incoming ARINC 429 messages on a single Subchannel. A single raw value can be selected for a graphical plot.

Free version:

- Bug fix: If a PCM subframe ID counter uses LSB first ordering and the PCM stream does as well the processing has been wrong. To our knowledge no hardware implements LSB transfer order so it shouldn't be an issue.

There are some more smaller bug fixes and improvements.

10.36 FLIDAS 3.2

This version focuses on new Live Display capabilities and an official XML CH10 Mapping.

Parameter Streaming plug-in:

- Live display of single numeric Parameters in a graph.

Data Streaming plug-in:

- MIL-STD 1553 Raw Data Live Display: This Live Display shows an RT status overview, subaddress data and mode code information from a MIL-STD 1553B bus. A single raw value can be selected for a graphical plot.
- The PCM Raw Data Live Display has been improved with a value and time axis and an auto scaling mode optional to the full range of the value axis.

Free version:

- The XML CH10 Mapping proposed by Data Bus Tools in reply to a request from the IRIG 106 specification committee has now been officially accepted and published by the Range Commanders Council. The specification can be found in the RCC document 123-16 "IRIG-106 Chapter 10 Programmers Handbook" appendix P/Q.

The specification can be downloaded from:

- The official RCC site (public access to this server is changing from time to time due to US DoD policy although the standard is approved for public release with unlimited distribution).
- A document mirror on www.irig106.org
- The Data Bus Tools website also contains an excerpt of the relevant part for convenience.

Since this is published by the RCC now the FLIDAS functionality is no prototype feature anymore but part of the regular version.

Accidentally the RCC has published a newer version of the XML CH10 mapping than agreed so FLIDAS catches up with this to support the XML Version 0.3.1 which adds automatic parity generation for ARINC 429. To reuse your existing XML you must rename the *Data* attribute within *ARINC429Message* tags to *Raw32*. If you want auto parity generation you can switch to the *Raw31* attribute in combination with the *GenerateWrongParity* attribute.

- Some people have recorders that generate time stamps without years and have no proper year information even within the CH10 TMATS. For them there is a new option when opening a CH10 File or Live Data Stream to use the start year provided on the GUI even if there is year information in TMATS.
- CH10 Data Type Filters have been added to suppress the video formats 3 and 4 and fibre channel and also user defined data packets. If you upgrade from an existing version these packets are automatically filtered out. You have to enable this type of data and save the setting once within the dialog to open CH10 Files and once in the dialog to open Live Data Streams when you need it.
- Some dialogs ("Wizards") have been harmonized and have separate sections in the manual now. Especially the CH10 File opening now also uses these Wizards and looks similar to opening a Live Data Stream. This also means that some settings for opening CH10 Files that have been in the General Preferences have moved to these Wizards now and must be saved there as default configuration. You must one time

manually configure these defaults separately for CH10 File loading and Live Data Stream creation. The things to configure are on the CH10 data type filter pages and on the File loading / Live Data Stream Properties pages.

10.37 FLIDAS 3.1

This version focuses on Parameter decoding and PCM raw data capabilities on Live Data Streams.

Data Streaming plug-in:

- PCM Raw Data Live Display: This Live Display shows raw data from an incoming PCM stream and continuously updates it. It displays a full major frame with its minor frames and allows a graphical plot of a selected PCM word.

Parameter plug-in:

- Parameter decoding for Recording Events: You can define a simple ICD to use Recording Events together with other decoded Parameters. For example, you can correlate the Recording Events with other Parameters in a graphical plot.
- Parameter List View: There is a new View showing all available Parameters in a hierarchical tree. It is used for drag and drop of Parameters for Parameter Live Displays.

Parameter Streaming plug-in:

- This is a new Plug-in. It contains the Parameter related parts for Data Streaming. It will be part of the delivery to customers with the Data Streaming and Parameter Plug-ins.
- Parameter Table Live Display: Table display of Parameters coming from a Data Stream with continuous updating. Parameters can be from any CH10 Data Type that is already supported by FLIDAS for Parameter decoding.

10.38 FLIDAS 3.0

This version mainly creates the base for a new line of features: Processing live data from UDP Data Streams. Most of the first features are for free.

Free version:

- Recoding data from CH10 UDP Data Streams
- Publishing CH10 Files as CH10 UDP Data Streams. This feature is mainly for testing FLIDAS itself but can also be used to publish data to other software.
- Live Data Streams: This adds an incoming CH10 UDP Data Stream to the FLIDAS Repository similar to a CH10 File. The features available for processing the live data are called Live Displays and can be used concurrently.
- Live Display of statistics of incoming Data Streams per Channel
- The TMATS View will now show the content of the TMATS override file if one has been used.

- The existing help system had problems with the latest Java release. The online help changed to use only the help system formerly used for the context sensitive help (i.e., not in an external window).

Data Streaming plug-in (this will be a new commercial plug-in but is right now provided to existing customers for free as it has only a few features):

- Videos from an incoming CH10 UDP Data Stream can be streamed to an external Video player for live replay.
- Display PCM statistics from incoming CH10 UDP Data Streams to verify PCM operation.

Parameter plug-in:

- The line width for printing Parameter graphs can now be configured.

To get familiar with the new Data Streaming go through the new Data Streaming tutorial in the manual.

A general note on UDP Data Streaming: The UDP protocol is not guaranteed to operate without data loss and gives no guarantees on delays on Ethernet. These are results of the UDP and Ethernet specification. The operating systems used for FLIDAS are not real-time capable. This makes guarantees on the delay between occurrence of data and the display impossible. All of these restrictions apply to other CH10 software using UDP Data Streaming and Windows / Linux OS as well and are results of IRIG106 and your system environment. If your testing cannot sustain potential data loss and unknown delays it is up to your judgement to use it or not. The free version can already be used to test data losses and performance in your environment and you can ask for evaluation versions of the commercial features too. However, usually regular PCs can handle hours of reception without data loss and achieve a delay of one or two seconds depending on your choices on the synchronization mode. Both delays are mainly due to the requirements in the IRIG 106 CH10 Standard and FLIDAS usually adds only some milliseconds to the delay. FLIDAS will make use of multi-core CPUs and the use is recommended for Data Streaming.

11 Legal

11.1 Free License

This license applies to the free download of the tool from the website www.databustools.de. It does not apply to the version that contains the commercial features. See [the next chapter](#) for the commercial license.

This software is provided free of charge. You receive the non-exclusive right to produce copies of the software (this implies also executing it on your computer) but may not sell it.

The place of jurisdiction shall be Munich in Germany. German law shall apply to this license agreement.

11.1.1 Third-Party Software

FLIDAS contains third-party components that are free of charge. They are subject to different licenses. If you read this in the online help, you will find them further below, but they will not be included in the PDF manual. If you read this in the PDF manual you will find the third-party legal information on the Data Bus Tools website at https://www.databustools.de/IRIG106/licenses_4_27/third_party_licenses.html

11.2 FLIDAS End User License Agreement (EULA)

This EULA is a legal contract between a business customer ("you") and Data Bus Tools GmbH ("DBT"). The contract is not targeted at private customers.

This contract is about the use of the FLIDAS software (Flight Data Studio). This contract relates to the FLIDAS manual where the features and limitations are described.

The license granted below is given under the condition that the customer agrees to all of the terms of this contract and pays its license fee.

11.2.1 License Grants

The customer receives the permanent, non-exclusive right to:

- simultaneously execute N instances of FLIDAS (where N is the number of procured licenses)
- Keep as many copies of FLIDAS on a storage medium as required for the operation

11.2.2 Warranty

The customer's right to cancel the contract and claims for damages are limited to cases that are not insignificant.

The customer is responsible to keep backups of the original IRIG 106 Chapter 10 files.

11.2.3 System Environment for Upgrades

Some customer's order upgrades for FLIDAS. The upgrades might require another system environment than the original version and the customer agrees to accept it under these conditions:

- The support for Windows operating systems may be discontinued if Microsoft's official "End of extended support" date is reached.
- The support for Java versions may be discontinued one year after a new oracle LTS (long term support) version is released. For this definition it doesn't matter which distribution of Java the customer uses.

The new system environment only applies to new delivered FLIDAS versions and not old ones. Customers may keep and use copies of their latest supported FLIDAS releases for their system configuration as long as they comply with the license grants established above (simultaneous execution restriction).

11.2.4 Applicable Law and Place of Jurisdiction

This contract shall be governed by German law. The place of jurisdiction shall be Munich in Germany.

11.2.5 Changes to Contract

All changes to the regulations in this contract require written form.

11.2.6 Third-Party Software

FLIDAS contains third-party components that are free of charge. They are subject to different licenses. If you read this in the online help, you will find them further below, but they will not be included in the PDF manual. If you read this in the PDF manual you will find the third-party legal information on the Data Bus Tools website at https://www.databustools.de/IRIG106/licenses_4_27/third_party_licenses.html