

`mogasens_csv`

Generated by Doxygen 1.8.17

1 Namespace Index	1
1.1 Namespace List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Namespace Documentation	11
5.1 cl Namespace Reference	11
5.1.1 Typedef Documentation	12
5.1.1.1 column_type	12
5.1.1.2 Expected	13
5.1.2 Enumeration Type Documentation	13
5.1.2.1 Channel	13
5.1.2.2 Column	13
5.1.2.3 CsvFileKind	14
5.1.2.4 Sensor	14
5.1.3 Function Documentation	14
5.1.3.1 CL_SPECIALIZE_COL_TRAITS() [1/11]	14
5.1.3.2 CL_SPECIALIZE_COL_TRAITS() [2/11]	14
5.1.3.3 CL_SPECIALIZE_COL_TRAITS() [3/11]	15
5.1.3.4 CL_SPECIALIZE_COL_TRAITS() [4/11]	15
5.1.3.5 CL_SPECIALIZE_COL_TRAITS() [5/11]	15
5.1.3.6 CL_SPECIALIZE_COL_TRAITS() [6/11]	15
5.1.3.7 CL_SPECIALIZE_COL_TRAITS() [7/11]	15
5.1.3.8 CL_SPECIALIZE_COL_TRAITS() [8/11]	15
5.1.3.9 CL_SPECIALIZE_COL_TRAITS() [9/11]	16
5.1.3.10 CL_SPECIALIZE_COL_TRAITS() [10/11]	16
5.1.3.11 CL_SPECIALIZE_COL_TRAITS() [11/11]	16
5.1.3.12 dataSetAccessor()	16
5.1.3.13 dos2unix()	17
5.1.3.14 isAccelerometer()	17
5.1.3.15 isGyroscope()	17
5.1.3.16 operator<<() [1/4]	18
5.1.3.17 operator<<() [2/4]	18
5.1.3.18 operator<<() [3/4]	19
5.1.3.19 operator<<() [4/4]	19
5.1.3.20 readCsvFile()	20
5.1.3.21 s2n()	20

5.1.3.22 threshold()	20
5.1.3.23 to_string()	21
5.1.3.24 useUnbufferedIo()	21
5.1.4 Variable Documentation	22
5.1.4.1 accelerometerThreshold	22
5.1.4.2 channelCount	22
5.1.4.3 channels	22
5.1.4.4 column_index	23
5.1.4.5 data_set_accessor_v	23
5.1.4.6 gyroscopeThreshold	23
5.1.4.7 sensors	23
5.2 cl::fs Namespace Reference	23
5.2.1 Enumeration Type Documentation	24
5.2.1.1 DirectoryListingOption	24
5.2.2 Function Documentation	24
5.2.2.1 directoryListing()	24
5.2.2.2 formatError()	25
5.2.2.3 operator<()	25
5.2.2.4 operator<<()	26
5.2.2.5 operator==()	26
5.2.2.6 utf16ToUtf8()	26
5.2.2.7 utf8ToUtf16()	27
5.3 cs Namespace Reference	27
5.3.1 Enumeration Type Documentation	28
5.3.1.1 FilterKind	28
5.3.1.2 SegmentationKind	28
5.3.2 Function Documentation	29
5.3.2.1 CS_SPECIALIZE_DATA_SET_INFO() [1/20]	29
5.3.2.2 CS_SPECIALIZE_DATA_SET_INFO() [2/20]	29
5.3.2.3 CS_SPECIALIZE_DATA_SET_INFO() [3/20]	29
5.3.2.4 CS_SPECIALIZE_DATA_SET_INFO() [4/20]	29
5.3.2.5 CS_SPECIALIZE_DATA_SET_INFO() [5/20]	30
5.3.2.6 CS_SPECIALIZE_DATA_SET_INFO() [6/20]	30
5.3.2.7 CS_SPECIALIZE_DATA_SET_INFO() [7/20]	30
5.3.2.8 CS_SPECIALIZE_DATA_SET_INFO() [8/20]	30
5.3.2.9 CS_SPECIALIZE_DATA_SET_INFO() [9/20]	30
5.3.2.10 CS_SPECIALIZE_DATA_SET_INFO() [10/20]	30
5.3.2.11 CS_SPECIALIZE_DATA_SET_INFO() [11/20]	31
5.3.2.12 CS_SPECIALIZE_DATA_SET_INFO() [12/20]	31
5.3.2.13 CS_SPECIALIZE_DATA_SET_INFO() [13/20]	31
5.3.2.14 CS_SPECIALIZE_DATA_SET_INFO() [14/20]	31
5.3.2.15 CS_SPECIALIZE_DATA_SET_INFO() [15/20]	31

5.3.2.16 CS_SPECIALIZE_DATA_SET_INFO() [16/20]	31
5.3.2.17 CS_SPECIALIZE_DATA_SET_INFO() [17/20]	32
5.3.2.18 CS_SPECIALIZE_DATA_SET_INFO() [18/20]	32
5.3.2.19 CS_SPECIALIZE_DATA_SET_INFO() [19/20]	32
5.3.2.20 CS_SPECIALIZE_DATA_SET_INFO() [20/20]	32
5.3.2.21 logFiles()	32
5.3.2.22 operator"!="()	33
5.3.2.23 operator<<() [1/3]	33
5.3.2.24 operator<<() [2/3]	34
5.3.2.25 operator<<() [3/3]	34
5.3.2.26 operator==()	34
5.3.2.27 PL_DEFINE_EXCEPTION_TYPE()	34
5.3.2.28 repetitionCount()	34
5.3.3 Variable Documentation	35
5.3.3.1 logPath	35
5.3.3.2 oldLogPath	35
5.4 ctg Namespace Reference	35
5.4.1 Function Documentation	35
5.4.1.1 aboveThreshold()	36
5.4.1.2 averageComparisonValueCalculator()	36
5.4.1.3 halfMaximumComparisonValueCalculator()	37
5.4.1.4 isRelevant()	38
5.4.1.5 percentageOf()	39
5.4.1.6 runAboveThreshold()	40
5.5 fmc Namespace Reference	40
5.5.1 Function Documentation	41
5.5.1.1 adjustHardwareTimestamp()	41
5.5.1.2 convertToUnixLineEndings()	41
5.5.1.3 createBackupFile()	42
5.5.1.4 deleteNonBoschSensors()	43
5.5.1.5 deleteOutOfBoundsValues()	43
5.5.1.6 removeZerosFromField()	43
5.5.1.7 restoreFromBackup()	44
5.5.1.8 writeFile()	44
6 Class Documentation	47
6.1 cl::col_traits< Col > Struct Template Reference	47
6.1.1 Detailed Description	47
6.2 cs::CsvLineBuilder Class Reference	47
6.2.1 Detailed Description	48
6.2.2 Member Typedef Documentation	48
6.2.2.1 this_type	48

6.2.3 Constructor & Destructor Documentation	48
6.2.3.1 CsvLineBuilder()	48
6.2.4 Member Function Documentation	48
6.2.4.1 build()	48
6.2.4.2 dataSet()	49
6.2.4.3 deleteLowVariance()	50
6.2.4.4 deleteTooClose()	50
6.2.4.5 filter()	51
6.2.4.6 isOld()	52
6.2.4.7 kind()	52
6.2.4.8 pushUps()	53
6.2.4.9 segmentationPoints()	54
6.2.4.10 sensor()	54
6.2.4.11 skipWindow()	55
6.2.4.12 windowSize()	56
6.3 cl::data_set_accessor< Chan > Struct Template Reference	56
6.3.1 Detailed Description	56
6.4 cs::data_set_info< Tag > Struct Template Reference	57
6.4.1 Detailed Description	57
6.5 cl::DataPoint Class Reference	57
6.5.1 Detailed Description	57
6.5.2 Constructor & Destructor Documentation	57
6.5.2.1 DataPoint()	58
6.5.3 Member Function Documentation	58
6.5.3.1 channel()	58
6.5.3.2 fileName()	59
6.5.3.3 sensor()	59
6.5.3.4 time()	60
6.5.3.5 value()	60
6.5.4 Friends And Related Function Documentation	60
6.5.4.1 operator<<	61
6.6 cl::DataSet Class Reference	61
6.6.1 Detailed Description	62
6.6.2 Member Typedef Documentation	62
6.6.2.1 ChannelAccessor	62
6.6.2.2 size_type	62
6.6.3 Member Function Documentation	62
6.6.3.1 accelerometerAverage()	62
6.6.3.2 accelerometerMaximum()	63
6.6.3.3 accelerometerX()	63
6.6.3.4 accelerometerY()	64
6.6.3.5 accelerometerZ()	64

6.6.3.6 create()	65
6.6.3.7 extractId()	66
6.6.3.8 fileName()	66
6.6.3.9 gyroscopeAverage()	67
6.6.3.10 gyroscopeMaximum()	68
6.6.3.11 gyroscopeX()	68
6.6.3.12 gyroscopeY()	69
6.6.3.13 gyroscopeZ()	69
6.6.3.14 hardwareTimestamp()	70
6.6.3.15 rowCount()	70
6.6.3.16 time()	70
6.6.3.17 trigger()	71
6.7 cl::Error Class Reference	71
6.7.1 Detailed Description	72
6.7.2 Member Enumeration Documentation	72
6.7.2.1 Kind	72
6.7.3 Constructor & Destructor Documentation	72
6.7.3.1 Error()	72
6.7.4 Member Function Documentation	73
6.7.4.1 file()	73
6.7.4.2 function()	73
6.7.4.3 kind()	74
6.7.4.4 line()	74
6.7.4.5 message()	74
6.7.4.6 raise()	75
6.7.4.7 to_string()	75
6.7.5 Friends And Related Function Documentation	75
6.7.5.1 operator<<	75
6.8 cl::Exception Class Reference	75
6.8.1 Detailed Description	76
6.8.2 Member Typedef Documentation	76
6.8.2.1 base_type	76
6.8.3 Constructor & Destructor Documentation	77
6.8.3.1 Exception() [1/2]	77
6.8.3.2 Exception() [2/2]	77
6.8.4 Member Function Documentation	77
6.8.4.1 file()	77
6.8.4.2 function()	77
6.8.4.3 line()	78
6.9 cl::fs::File Class Reference	78
6.9.1 Detailed Description	78
6.9.2 Constructor & Destructor Documentation	78

6.9.2.1 File()	78
6.9.3 Member Function Documentation	79
6.9.3.1 copyTo()	79
6.9.3.2 create()	80
6.9.3.3 exists()	81
6.9.3.4 moveTo()	81
6.9.3.5 path()	82
6.9.3.6 remove()	83
6.9.3.7 size()	83
6.10 cl::fs::FileStream Class Reference	84
6.10.1 Detailed Description	85
6.10.2 Member Typedef Documentation	85
6.10.2.1 this_type	85
6.10.3 Member Enumeration Documentation	85
6.10.3.1 OpenMode	85
6.10.4 Constructor & Destructor Documentation	85
6.10.4.1 FileStream()	85
6.10.4.2 ~FileStream()	86
6.10.5 Member Function Documentation	86
6.10.5.1 create()	86
6.10.5.2 operator=()	87
6.10.5.3 PL_NONCOPYABLE()	87
6.10.5.4 readAll()	87
6.10.5.5 write()	87
6.11 cs::LogInfo Class Reference	87
6.11.1 Detailed Description	88
6.11.2 Constructor & Destructor Documentation	88
6.11.2.1 LogInfo()	88
6.11.3 Member Function Documentation	88
6.11.3.1 create()	88
6.11.3.2 deleteLowVariance()	89
6.11.3.3 deleteTooClose()	89
6.11.3.4 filterKind()	89
6.11.3.5 isInitialized()	89
6.11.3.6 logFilePath()	89
6.11.3.7 segmentationKind()	90
6.11.3.8 sensor()	90
6.11.3.9 skipWindow()	90
6.11.3.10 windowSize()	90
6.11.4 Friends And Related Function Documentation	90
6.11.4.1 operator"!="	90
6.11.4.2 operator<<	91

6.11.4.3 operator==	91
6.11.5 Member Data Documentation	91
6.11.5.1 invalidSensor	91
6.12 cs::LogLine Class Reference	91
6.12.1 Detailed Description	92
6.12.2 Member Function Documentation	92
6.12.2.1 fileName()	92
6.12.2.2 filePath()	92
6.12.2.3 parse()	93
6.12.2.4 segmentationPointCount()	93
6.12.2.5 sensor()	93
6.12.3 Member Data Documentation	93
6.12.3.1 invalidSensor	93
6.13 cl::fs::Path Class Reference	94
6.13.1 Detailed Description	94
6.13.2 Constructor & Destructor Documentation	94
6.13.2.1 Path()	94
6.13.3 Member Function Documentation	94
6.13.3.1 exists()	95
6.13.3.2 isDirectory()	95
6.13.3.3 isFile()	96
6.13.3.4 str()	96
6.13.4 Friends And Related Function Documentation	97
6.13.4.1 operator<	97
6.13.4.2 operator<<	97
6.13.4.3 operator==	97
7 File Documentation	99
7.1 compare_segmentation/CMakeLists.txt File Reference	99
7.1.1 Function Documentation	99
7.1.1.1 set()	99
7.2 compare_segmentation/test/CMakeLists.txt File Reference	99
7.2.1 Function Documentation	99
7.2.1.1 include()	100
7.3 counting/CMakeLists.txt File Reference	100
7.3.1 Function Documentation	100
7.3.1.1 set()	100
7.4 counting/test/CMakeLists.txt File Reference	100
7.4.1 Function Documentation	100
7.4.1.1 include()	100
7.5 csv_lib/CMakeLists.txt File Reference	101
7.5.1 Function Documentation	101

7.5.1.1 set()	101
7.6 csv_lib/test/CMakeLists.txt File Reference	101
7.6.1 Function Documentation	101
7.6.1.1 include()	101
7.7 fix_csv/CMakeLists.txt File Reference	102
7.7.1 Function Documentation	102
7.7.1.1 set()	102
7.8 fix_csv/test/CMakeLists.txt File Reference	102
7.8.1 Function Documentation	102
7.8.1.1 include()	102
7.9 compare_segmentation/include/csv_line.hpp File Reference	103
7.10 compare_segmentation/include/data_set_info.hpp File Reference	103
7.10.1 Macro Definition Documentation	105
7.10.1.1 CS_SPECIALIZE_DATA_SET_INFO	105
7.11 compare_segmentation/include/filter_kind.hpp File Reference	105
7.12 compare_segmentation/include/log_files.hpp File Reference	106
7.13 compare_segmentation/include/log_info.hpp File Reference	107
7.14 compare_segmentation/include/log_line.hpp File Reference	108
7.15 compare_segmentation/include/paths.hpp File Reference	109
7.16 compare_segmentation/include/segmentation_kind.hpp File Reference	110
7.17 compare_segmentation/src/csv_line.cpp File Reference	111
7.18 compare_segmentation/src/data_set_info.cpp File Reference	111
7.19 compare_segmentation/src/filter_kind.cpp File Reference	112
7.20 compare_segmentation/src/log_files.cpp File Reference	112
7.21 compare_segmentation/src/log_info.cpp File Reference	113
7.22 compare_segmentation/src/log_line.cpp File Reference	114
7.23 compare_segmentation/src/main.cpp File Reference	114
7.23.1 Function Documentation	115
7.23.1.1 main()	115
7.24 compare_segmentation/test/main.cpp File Reference	116
7.24.1 Function Documentation	117
7.24.1.1 main()	117
7.25 counting/src/main.cpp File Reference	117
7.25.1 Function Documentation	118
7.25.1.1 main()	118
7.26 counting/test/main.cpp File Reference	119
7.26.1 Function Documentation	120
7.26.1.1 main()	120
7.27 csv_lib/test/main.cpp File Reference	120
7.27.1 Function Documentation	121
7.27.1.1 main()	121
7.28 fix_csv/src/main.cpp File Reference	121

7.28.1 Function Documentation	122
7.28.1.1 main()	122
7.29 fix_csv/test/main.cpp File Reference	122
7.29.1 Function Documentation	123
7.29.1.1 main()	123
7.30 compare_segmentation/src/segmentation_kind.cpp File Reference	123
7.31 compare_segmentation/test/csv_line_test.cpp File Reference	124
7.31.1 Function Documentation	124
7.31.1.1 TEST()	124
7.32 compare_segmentation/test/data_set_info_test.cpp File Reference	124
7.32.1 Function Documentation	125
7.32.1.1 TEST()	125
7.33 compare_segmentation/test/log_files_test.cpp File Reference	126
7.33.1 Function Documentation	126
7.33.1.1 TEST() [1/3]	126
7.33.1.2 TEST() [2/3]	127
7.33.1.3 TEST() [3/3]	127
7.34 compare_segmentation/test/log_info_test.cpp File Reference	127
7.34.1 Function Documentation	128
7.34.1.1 TEST() [1/19]	128
7.34.1.2 TEST() [2/19]	129
7.34.1.3 TEST() [3/19]	129
7.34.1.4 TEST() [4/19]	130
7.34.1.5 TEST() [5/19]	130
7.34.1.6 TEST() [6/19]	131
7.34.1.7 TEST() [7/19]	131
7.34.1.8 TEST() [8/19]	132
7.34.1.9 TEST() [9/19]	132
7.34.1.10 TEST() [10/19]	133
7.34.1.11 TEST() [11/19]	133
7.34.1.12 TEST() [12/19]	134
7.34.1.13 TEST() [13/19]	134
7.34.1.14 TEST() [14/19]	135
7.34.1.15 TEST() [15/19]	135
7.34.1.16 TEST() [16/19]	136
7.34.1.17 TEST() [17/19]	136
7.34.1.18 TEST() [18/19]	137
7.34.1.19 TEST() [19/19]	137
7.35 compare_segmentation/test/log_line_test.cpp File Reference	137
7.35.1 Function Documentation	138
7.35.1.1 TEST() [1/4]	138
7.35.1.2 TEST() [2/4]	138

7.35.1.3 TEST() [3/4]	139
7.35.1.4 TEST() [4/4]	139
7.36 counting/include/above_threshold.hpp File Reference	139
7.37 counting/include/average_comparison_value_calculator.hpp File Reference	140
7.38 counting/include/half_maximum_comparison_value_calculator.hpp File Reference	141
7.39 counting/include/is_relevant.hpp File Reference	142
7.40 counting/include/percentage_of.hpp File Reference	143
7.41 counting/include/run_above_threshold.hpp File Reference	144
7.42 counting/src/above_threshold.cpp File Reference	145
7.42.1 Macro Definition Documentation	146
7.42.1.1 CL_CHANNEL_X	146
7.42.2 Variable Documentation	146
7.42.2.1 channel	147
7.42.2.2 channelAccessor	147
7.43 counting/src/average_comparison_value_calculator.cpp File Reference	147
7.44 counting/src/half_maximum_comparison_value_calculator.cpp File Reference	148
7.45 counting/src/run_above_threshold.cpp File Reference	148
7.46 counting/test/above_threshold_test.cpp File Reference	149
7.46.1 Macro Definition Documentation	149
7.46.1.1 EXPECT_LONG_DOUBLE_EQ	150
7.46.2 Function Documentation	150
7.46.2.1 TEST()	150
7.47 counting/test/percentage_of_test.cpp File Reference	151
7.47.1 Macro Definition Documentation	151
7.47.1.1 EXPECT_LONG_DOUBLE_EQ	151
7.47.2 Function Documentation	151
7.47.2.1 TEST()	152
7.48 csv_lib/include/cl/channel.hpp File Reference	152
7.48.1 Macro Definition Documentation	153
7.48.1.1 CL_CHANNEL	154
7.48.1.2 CL_CHANNEL_X [1/4]	154
7.48.1.3 CL_CHANNEL_X [2/4]	154
7.48.1.4 CL_CHANNEL_X [3/4]	154
7.48.1.5 CL_CHANNEL_X [4/4]	155
7.49 csv_lib/include/cl/column.hpp File Reference	155
7.49.1 Macro Definition Documentation	156
7.49.1.1 CL_SPECIALIZE_COL_TRAITS	157
7.50 csv_lib/include/cl/data_point.hpp File Reference	157
7.51 csv_lib/include/cl/data_set.hpp File Reference	158
7.52 csv_lib/include/cl/dos2unix.hpp File Reference	159
7.53 csv_lib/include/cl/error.hpp File Reference	160
7.53.1 Macro Definition Documentation	160

7.53.1.1 CL_ERROR_KIND	161
7.53.1.2 CL_ERROR_KIND_X	161
7.53.1.3 CL_UNEXPECTED	161
7.54 csv_lib/include/cl/exception.hpp File Reference	161
7.55 csv_lib/include/cl/fs/directory_listing.hpp File Reference	162
7.56 csv_lib/include/cl/fs/file.hpp File Reference	163
7.57 csv_lib/include/cl/fs/file_stream.hpp File Reference	164
7.58 csv_lib/include/cl/fs/path.hpp File Reference	165
7.59 csv_lib/include/cl/fs/separator.hpp File Reference	166
7.59.1 Macro Definition Documentation	166
7.59.1.1 CL_FS_SEPARATOR	166
7.60 csv_lib/include/cl/fs/windows.hpp File Reference	167
7.61 csv_lib/include/cl/read_csv_file.hpp File Reference	168
7.62 csv_lib/include/cl/s2n.hpp File Reference	169
7.63 csv_lib/include/cl/sensor.hpp File Reference	169
7.63.1 Macro Definition Documentation	171
7.63.1.1 CL_SENSOR	171
7.63.1.2 CL_SENSOR_X [1/2]	171
7.63.1.3 CL_SENSOR_X [2/2]	171
7.64 csv_lib/include/cl/to_string.hpp File Reference	172
7.65 csv_lib/include/cl/use_unbuffered_io.hpp File Reference	173
7.66 csv_lib/src/cl/channel.cpp File Reference	173
7.66.1 Macro Definition Documentation	174
7.66.1.1 CL_CHANNEL_X [1/2]	174
7.66.1.2 CL_CHANNEL_X [2/2]	174
7.67 csv_lib/src/cl/data_point.cpp File Reference	175
7.67.1 Function Documentation	175
7.67.1.1 channel()	175
7.67.1.2 fileName()	176
7.67.1.3 sensor()	176
7.67.1.4 time()	177
7.67.1.5 value()	177
7.68 csv_lib/src/cl/data_set.cpp File Reference	178
7.69 csv_lib/src/cl/dos2unix.cpp File Reference	178
7.70 csv_lib/src/cl/error.cpp File Reference	179
7.70.1 Macro Definition Documentation	179
7.70.1.1 CL_ERROR_KIND_X	180
7.71 csv_lib/src/cl/exception.cpp File Reference	180
7.72 csv_lib/src/cl/fs/directory_listing.cpp File Reference	180
7.73 csv_lib/src/cl/fs/file.cpp File Reference	181
7.74 csv_lib/src/cl/fs/file_stream.cpp File Reference	182
7.75 csv_lib/src/cl/fs/path.cpp File Reference	182

7.76 csv_lib/src/cl/fs/windows.cpp File Reference	183
7.77 csv_lib/src/cl/read_csv_file.cpp File Reference	183
7.78 csv_lib/src/cl/sensor.cpp File Reference	184
7.78.1 Macro Definition Documentation	185
7.78.1.1 CL_SENSOR_X	185
7.79 csv_lib/src/cl/use_unbuffered_io.cpp File Reference	185
7.80 csv_lib/test/channel_test.cpp File Reference	186
7.80.1 Function Documentation	186
7.80.1.1 TEST() [1/4]	186
7.80.1.2 TEST() [2/4]	186
7.80.1.3 TEST() [3/4]	187
7.80.1.4 TEST() [4/4]	187
7.81 csv_lib/test/column_test.cpp File Reference	188
7.81.1 Function Documentation	188
7.81.1.1 TEST() [1/2]	188
7.81.1.2 TEST() [2/2]	188
7.82 csv_lib/test/data_point_test.cpp File Reference	189
7.82.1 Function Documentation	189
7.82.1.1 TEST() [1/2]	189
7.82.1.2 TEST() [2/2]	190
7.82.2 Variable Documentation	190
7.82.2.1 dp	190
7.83 csv_lib/test/data_set_test.cpp File Reference	191
7.83.1 Macro Definition Documentation	191
7.83.1.1 EXPECT_LONG_DOUBLE_EQ	191
7.83.2 Function Documentation	191
7.83.2.1 TEST() [1/4]	192
7.83.2.2 TEST() [2/4]	192
7.83.2.3 TEST() [3/4]	193
7.83.2.4 TEST() [4/4]	194
7.84 csv_lib/test/directory_listing_test.cpp File Reference	195
7.84.1 Function Documentation	196
7.84.1.1 TEST() [1/3]	196
7.84.1.2 TEST() [2/3]	196
7.84.1.3 TEST() [3/3]	197
7.85 csv_lib/test/error_test.cpp File Reference	197
7.85.1 Function Documentation	198
7.85.1.1 TEST() [1/4]	198
7.85.1.2 TEST() [2/4]	198
7.85.1.3 TEST() [3/4]	198
7.85.1.4 TEST() [4/4]	198
7.85.2 Variable Documentation	198

7.85.2.1 error	199
7.86 csv_lib/test/exception_test.cpp File Reference	199
7.86.1 Function Documentation	199
7.86.1.1 TEST()	199
7.87 csv_lib/test/read_csv_file_test.cpp File Reference	200
7.87.1 Function Documentation	200
7.87.1.1 TEST() [1/2]	200
7.87.1.2 TEST() [2/2]	201
7.88 csv_lib/test/s2n_test.cpp File Reference	201
7.88.1 Function Documentation	202
7.88.1.1 TEST() [1/3]	202
7.88.1.2 TEST() [2/3]	202
7.88.1.3 TEST() [3/3]	203
7.89 csv_lib/test/sensor_test.cpp File Reference	203
7.89.1 Function Documentation	204
7.89.1.1 TEST() [1/2]	204
7.89.1.2 TEST() [2/2]	204
7.90 csv_lib/test/to_string_test.cpp File Reference	204
7.90.1 Function Documentation	205
7.90.1.1 TEST()	205
7.91 fix_csv/include/adjust_hardware_timestamp.hpp File Reference	205
7.92 fix_csv/include/convert_to_unix_line_endings.hpp File Reference	206
7.93 fix_csv/include/create_backup_file.hpp File Reference	207
7.94 fix_csv/include/delete_non_bosch_sensors.hpp File Reference	208
7.95 fix_csv/include/delete_out_of_bounds_values.hpp File Reference	209
7.96 fix_csv/include/remove_zeros_from_field.hpp File Reference	210
7.97 fix_csv/include/restore_from_backup.hpp File Reference	211
7.98 fix_csv/include/write_file.hpp File Reference	212
7.99 fix_csv/src/adjust_hardware_timestamp.cpp File Reference	213
7.100 fix_csv/src/convert_to_unix_line_endings.cpp File Reference	214
7.101 fix_csv/src/create_backup_file.cpp File Reference	215
7.102 fix_csv/src/delete_non_bosch_sensors.cpp File Reference	215
7.102.1 Macro Definition Documentation	216
7.102.1.1 CL_SENSOR_X	216
7.103 fix_csv/src/delete_out_of_bounds_values.cpp File Reference	216
7.104 fix_csv/src/remove_zeros_from_field.cpp File Reference	217
7.105 fix_csv/src/restore_from_backup.cpp File Reference	217
7.106 fix_csv/src/write_file.cpp File Reference	218
7.107 fix_csv/test/adjust_hardware_timestamp_test.cpp File Reference	219
7.107.1 Function Documentation	219
7.107.1.1 TEST() [1/5]	219
7.107.1.2 TEST() [2/5]	220

7.107.1.3 TEST() [3/5]	220
7.107.1.4 TEST() [4/5]	221
7.107.1.5 TEST() [5/5]	221
7.108 fix_csv/test/remove_zeros_from_field_test.cpp File Reference	221
7.108.1 Function Documentation	222
7.108.1.1 TEST() [1/6]	222
7.108.1.2 TEST() [2/6]	223
7.108.1.3 TEST() [3/6]	223
7.108.1.4 TEST() [4/6]	224
7.108.1.5 TEST() [5/6]	224
7.108.1.6 TEST() [6/6]	225
Index	227

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

cl	11
cl::fs	23
cs	27
ctg	35
fmc	40

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

cl::col_traits< Col >	47
cs::CsvLineBuilder	47
cl::data_set_accessor< Chan >	56
cs::data_set_info< Tag >	57
cl::DataPoint	57
cl::DataSet	61
cl::Error	71
std::exception	
std::runtime_error	
cl::Exception	75
cl::fs::File	78
cl::fs::FileStream	84
cs::LogInfo	87
cs::LogLine	91
cl::fs::Path	94

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

cl::col_traits< Col >	47
cs::CsvLineBuilder	47
cl::data_set_accessor< Chan >	56
cs::data_set_info< Tag >	57
cl::DataPoint	57
cl::DataSet	61
cl::Error	71
cl::Exception	75
cl::fs::File	
Represents a file	78
cl::fs::FileStream	84
cs::LogInfo	87
cs::LogLine	91
cl::fs::Path	94

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

compare_segmentation/include/csv_line.hpp	103
compare_segmentation/include/data_set_info.hpp	103
compare_segmentation/include/filter_kind.hpp	105
compare_segmentation/include/log_files.hpp	106
compare_segmentation/include/log_info.hpp	107
compare_segmentation/include/log_line.hpp	108
compare_segmentation/include/paths.hpp	109
compare_segmentation/include/segmentation_kind.hpp	110
compare_segmentation/src/csv_line.cpp	111
compare_segmentation/src/data_set_info.cpp	111
compare_segmentation/src/filter_kind.cpp	112
compare_segmentation/src/log_files.cpp	112
compare_segmentation/src/log_info.cpp	113
compare_segmentation/src/log_line.cpp	114
compare_segmentation/src/main.cpp	114
compare_segmentation/src/segmentation_kind.cpp	123
compare_segmentation/test/csv_line_test.cpp	124
compare_segmentation/test/data_set_info_test.cpp	124
compare_segmentation/test/log_files_test.cpp	126
compare_segmentation/test/log_info_test.cpp	127
compare_segmentation/test/log_line_test.cpp	137
compare_segmentation/test/main.cpp	116
counting/include/above_threshold.hpp	139
counting/include/average_comparison_value_calculator.hpp	140
counting/include/half_maximum_comparison_value_calculator.hpp	141
counting/include/is_relevant.hpp	142
counting/include/percentage_of.hpp	143
counting/include/run_above_threshold.hpp	144
counting/src/above_threshold.cpp	145
counting/src/average_comparison_value_calculator.cpp	147
counting/src/half_maximum_comparison_value_calculator.cpp	148
counting/src/main.cpp	117
counting/src/run_above_threshold.cpp	148
counting/test/above_threshold_test.cpp	149
counting/test/main.cpp	119

counting/test/percentage_of_test.cpp	151
csv_lib/include/cl/channel.hpp	152
csv_lib/include/cl/column.hpp	155
csv_lib/include/cl/data_point.hpp	157
csv_lib/include/cl/data_set.hpp	158
csv_lib/include/cl/dos2unix.hpp	159
csv_lib/include/cl/error.hpp	160
csv_lib/include/cl/exception.hpp	161
csv_lib/include/cl/read_csv_file.hpp	168
csv_lib/include/cl/s2n.hpp	169
csv_lib/include/cl/sensor.hpp	169
csv_lib/include/cl/to_string.hpp	172
csv_lib/include/cl/use_unbuffered_io.hpp	173
csv_lib/include/cl/fs/directory_listing.hpp	162
csv_lib/include/cl/fs/file.hpp	163
csv_lib/include/cl/fs/file_stream.hpp	164
csv_lib/include/cl/fs/path.hpp	165
csv_lib/include/cl/fs/separator.hpp	166
csv_lib/include/cl/fs/windows.hpp	167
csv_lib/src/cl/channel.cpp	173
csv_lib/src/cl/data_point.cpp	175
csv_lib/src/cl/data_set.cpp	178
csv_lib/src/cl/dos2unix.cpp	178
csv_lib/src/cl/error.cpp	179
csv_lib/src/cl/exception.cpp	180
csv_lib/src/cl/read_csv_file.cpp	183
csv_lib/src/cl/sensor.cpp	184
csv_lib/src/cl/use_unbuffered_io.cpp	185
csv_lib/src/cl/fs/directory_listing.cpp	180
csv_lib/src/cl/fs/file.cpp	181
csv_lib/src/cl/fs/file_stream.cpp	182
csv_lib/src/cl/fs/path.cpp	182
csv_lib/src/cl/fs/windows.cpp	183
csv_lib/test/channel_test.cpp	186
csv_lib/test/column_test.cpp	188
csv_lib/test/data_point_test.cpp	189
csv_lib/test/data_set_test.cpp	191
csv_lib/test/directory_listing_test.cpp	195
csv_lib/test/error_test.cpp	197
csv_lib/test/exception_test.cpp	199
csv_lib/test/main.cpp	120
csv_lib/test/read_csv_file_test.cpp	200
csv_lib/test/s2n_test.cpp	201
csv_lib/test/sensor_test.cpp	203
csv_lib/test/to_string_test.cpp	204
fix_csv/include/adjust_hardware_timestamp.hpp	205
fix_csv/include/convert_to_unix_line_endings.hpp	206
fix_csv/include/create_backup_file.hpp	207
fix_csv/include/delete_non_bosch_sensors.hpp	208
fix_csv/include/delete_out_of_bounds_values.hpp	209
fix_csv/include/remove_zeros_from_field.hpp	210
fix_csv/include/restore_from_backup.hpp	211
fix_csv/include/write_file.hpp	212
fix_csv/src/adjust_hardware_timestamp.cpp	213
fix_csv/src/convert_to_unix_line_endings.cpp	214
fix_csv/src/create_backup_file.cpp	215
fix_csv/src/delete_non_bosch_sensors.cpp	215
fix_csv/src/delete_out_of_bounds_values.cpp	216

fix_csv/src/main.cpp	121
fix_csv/src/remove_zeros_from_field.cpp	217
fix_csv/src/restore_from_backup.cpp	217
fix_csv/src/write_file.cpp	218
fix_csv/test/adjust_hardware_timestamp_test.cpp	219
fix_csv/test/main.cpp	122
fix_csv/test/remove_zeros_from_field_test.cpp	221

Chapter 5

Namespace Documentation

5.1 cl Namespace Reference

Namespaces

- `fs`

Classes

- struct `col_traits`
- struct `data_set_accessor`
- class `DataPoint`
- class `DataSet`
- class `Error`
- class `Exception`

Typedefs

- template<Column Col>
using `column_type` = typename `col_traits`< Col >::type
- template<typename Ty >
using `Expected` = tl::expected< Ty, `Error` >

Enumerations

- enum `Channel` : std::uint64_t { `Channel::CL_CHANNEL_X`, `Channel::CL_CHANNEL` }
- enum `Column` : std::size_t {
`Column::Time`, `Column::HardwareTimestamp`, `Column::ExtractId`, `Column::Trigger`,
`Column::AccelerometerX`, `Column::AccelerometerY`, `Column::AccelerometerZ`, `Column::GyroscopeX`,
`Column::GyroscopeY`, `Column::GyroscopeZ`, `Column::SamplingRate` }
- enum `CsvFileKind` { `CsvFileKind::Raw`, `CsvFileKind::Fixed` }
- enum `Sensor` : std::uint64_t { `Sensor::CL_SENSOR_X`, `Sensor::CL_SENSOR` }

Functions

- `DataSet::ChannelAccessor dataSetAccessor (Channel channel)`
- `std::ostream & operator<< (std::ostream &os, Channel channel)`
- `bool isAccelerometer (Channel channel)`
- `bool isGyroscope (Channel channel)`
- `long double threshold (Channel channel)`
- `CL_SPECIALIZE_COL_TRAITS (Column::Time, long double)`
- `CL_SPECIALIZE_COL_TRAITS (Column::HardwareTimestamp, std::uint64_t)`
- `CL_SPECIALIZE_COL_TRAITS (Column::ExtractId, Sensor)`
- `CL_SPECIALIZE_COL_TRAITS (Column::Trigger, std::uint64_t)`
- `CL_SPECIALIZE_COL_TRAITS (Column::AccelerometerX, long double)`
- `CL_SPECIALIZE_COL_TRAITS (Column::AccelerometerY, long double)`
- `CL_SPECIALIZE_COL_TRAITS (Column::AccelerometerZ, long double)`
- `CL_SPECIALIZE_COL_TRAITS (Column::GyroscopeX, long double)`
- `CL_SPECIALIZE_COL_TRAITS (Column::GyroscopeY, long double)`
- `CL_SPECIALIZE_COL_TRAITS (Column::GyroscopeZ, long double)`
- `CL_SPECIALIZE_COL_TRAITS (Column::SamplingRate, std::uint64_t)`
- `std::vector< pl::byte > dos2unix (const void *p, std::size_t size)`
- `Expected< std::vector< std::vector< std::string > > > readCsvFile (pl::string_view csvFilePath, std::vector< std::string > *columnNames=nullptr, CsvFileKind csvFileKind=CsvFileKind::Fixed) noexcept`
- `template<typename Integer> Expected< Integer > s2n (const std::string &str, std::size_t *pos=nullptr, [[maybe_unused]] int base=10)`
- `std::ostream & operator<< (std::ostream &os, Sensor sensor)`
- `template<typename Ty> std::string to_string (const Ty &ty)`
- `void useUnbufferedIo ()`
- `std::ostream & operator<< (std::ostream &os, const DataPoint &dataPoint)`
- `std::ostream & operator<< (std::ostream &os, const Error &error)`

Variables

- `constexpr std::size_t channelCount`
- `constexpr std::array< Channel, channelCount > channels`
- `template<Channel Chan> constexpr CL_CHANNEL DataSet::ChannelAccessor data_set_accessor_v = data_set_accessor<Chan>::f`
- `constexpr long double accelerometerThreshold {1.99L}`
- `constexpr long double gyroscopeThreshold {1999.99L}`
- `template<Column Col> constexpr std::size_t column_index = col_traits<Col>::index`
- `constexpr std::array< Sensor, 4 > sensors`

5.1.1 Typedef Documentation

5.1.1.1 column_type

```
template<Column Col>
using cl::column_type = typedef typename col_traits<Col>::type
```

Definition at line 49 of file column.hpp.

5.1.1.2 Expected

```
template<typename Ty >
using cl::Expected = typedef tl::expected<Ty, Error>
```

Definition at line 63 of file error.hpp.

5.1.2 Enumeration Type Documentation

5.1.2.1 Channel

```
enum cl::Channel : std::uint64_t [strong]
```

Enumerator

CL_CHANNEL←_X	
CL_CHANNEL	

Definition at line 20 of file channel.hpp.

5.1.2.2 Column

```
enum cl::Column : std::size_t [strong]
```

Enumerator

Time	
HardwareTimestamp	
ExtractId	
Trigger	
AccelerometerX	
AccelerometerY	
AccelerometerZ	
GyroscopeX	
GyroscopeY	
GyroscopeZ	
SamplingRate	

Definition at line 9 of file column.hpp.

5.1.2.3 CsvFileKind

```
enum cl::CsvFileKind [strong]
```

Enumerator

Raw	
Fixed	

Definition at line 11 of file read_csv_file.hpp.

5.1.2.4 Sensor

```
enum cl::Sensor : std::uint64_t [strong]
```

Enumerator

CL_SENSOR_X	
CL_SENSOR	

Definition at line 15 of file sensor.hpp.

5.1.3 Function Documentation

5.1.3.1 CL_SPECIALIZE_COL_TRAITS() [1/11]

```
cl::CL_SPECIALIZE_COL_TRAITS (
    Column::AccelerometerX ,
    long double )
```

5.1.3.2 CL_SPECIALIZE_COL_TRAITS() [2/11]

```
cl::CL_SPECIALIZE_COL_TRAITS (
    Column::AccelerometerY ,
    long double )
```

5.1.3.3 CL_SPECIALIZE_COL_TRAITS() [3/11]

```
cl::CL_SPECIALIZE_COL_TRAITS (
    Column::AccelerometerZ ,
    long double   )
```

5.1.3.4 CL_SPECIALIZE_COL_TRAITS() [4/11]

```
cl::CL_SPECIALIZE_COL_TRAITS (
    Column::ExtractId ,
    Sensor   )
```

5.1.3.5 CL_SPECIALIZE_COL_TRAITS() [5/11]

```
cl::CL_SPECIALIZE_COL_TRAITS (
    Column::GyroscopeX ,
    long double   )
```

5.1.3.6 CL_SPECIALIZE_COL_TRAITS() [6/11]

```
cl::CL_SPECIALIZE_COL_TRAITS (
    Column::GyroscopeY ,
    long double   )
```

5.1.3.7 CL_SPECIALIZE_COL_TRAITS() [7/11]

```
cl::CL_SPECIALIZE_COL_TRAITS (
    Column::GyroscopeZ ,
    long double   )
```

5.1.3.8 CL_SPECIALIZE_COL_TRAITS() [8/11]

```
cl::CL_SPECIALIZE_COL_TRAITS (
    Column::HardwareTimestamp ,
    std::uint64_t   )
```

5.1.3.9 CL_SPECIALIZE_COL_TRAITS() [9/11]

```
cl::CL_SPECIALIZE_COL_TRAITS (
    Column::SamplingRate ,
    std::uint64_t )
```

5.1.3.10 CL_SPECIALIZE_COL_TRAITS() [10/11]

```
cl::CL_SPECIALIZE_COL_TRAITS (
    Column::Time ,
    long double )
```

5.1.3.11 CL_SPECIALIZE_COL_TRAITS() [11/11]

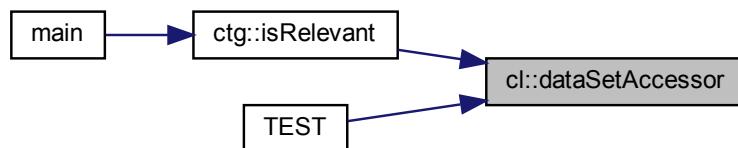
```
cl::CL_SPECIALIZE_COL_TRAITS (
    Column::Trigger ,
    std::uint64_t )
```

5.1.3.12 dataSetAccessor()

```
DataSet::ChannelAccessor cl::dataSetAccessor (
    Channel channel )
```

Definition at line 15 of file channel.cpp.

Here is the caller graph for this function:



5.1.3.13 dos2unix()

```
std::vector< pl::byte > cl::dos2unix (
    const void * p,
    std::size_t size )
```

Definition at line 4 of file dos2unix.cpp.

Here is the caller graph for this function:

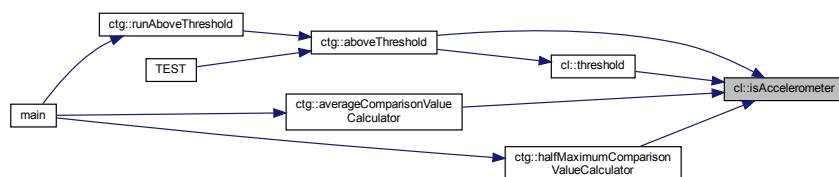


5.1.3.14 isAccelerometer()

```
bool cl::isAccelerometer (
    Channel channel )
```

Definition at line 45 of file channel.cpp.

Here is the caller graph for this function:

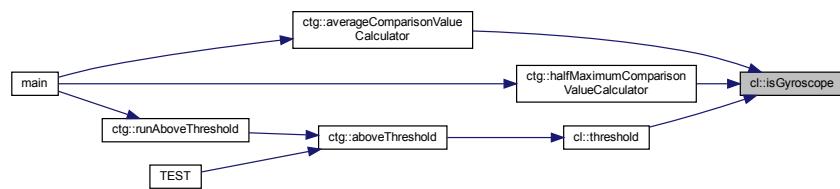


5.1.3.15 isGyroscope()

```
bool cl::isGyroscope (
    Channel channel )
```

Definition at line 50 of file channel.cpp.

Here is the caller graph for this function:



5.1.3.16 operator<<() [1/4]

```
std::ostream & cl::operator<< (
    std::ostream & os,
    Channel channel )
```

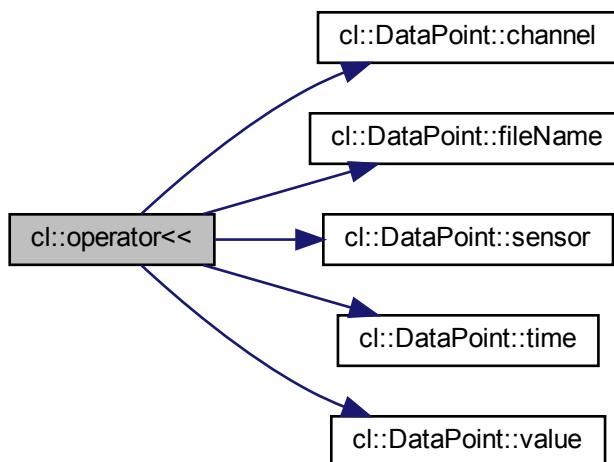
Definition at line 32 of file channel.cpp.

5.1.3.17 operator<<() [2/4]

```
std::ostream& cl::operator<< (
    std::ostream & os,
    const DataPoint & dataPoint )
```

Definition at line 10 of file data_point.cpp.

Here is the call graph for this function:

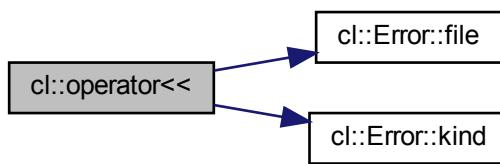


5.1.3.18 operator<<() [3/4]

```
std::ostream& cl::operator<< (
    std::ostream & os,
    const Error & error )
```

Definition at line 30 of file error.cpp.

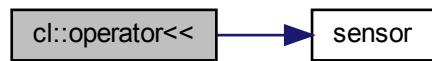
Here is the call graph for this function:

**5.1.3.19 operator<<() [4/4]**

```
std::ostream & cl::operator<< (
    std::ostream & os,
    Sensor sensor )
```

Definition at line 8 of file sensor.cpp.

Here is the call graph for this function:

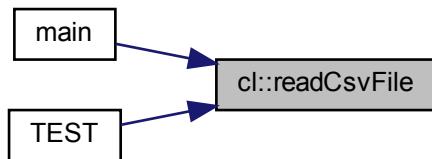


5.1.3.20 `readCsvFile()`

```
Expected< std::vector< std::vector< std::string > > > cl::readCsvFile (
    pl::string_view csvFilePath,
    std::vector< std::string > * columnNames = nullptr,
    CsvFileKind csvFileKind = CsvFileKind::Fixed ) [noexcept]
```

Definition at line 50 of file `read_csv_file.cpp`.

Here is the caller graph for this function:



5.1.3.21 `s2n()`

```
template<typename Integer>
Expected<Integer> cl::s2n(
    const std::string & str,
    std::size_t * pos = nullptr,
    [[maybe_unused]] int base = 10) [inline]
```

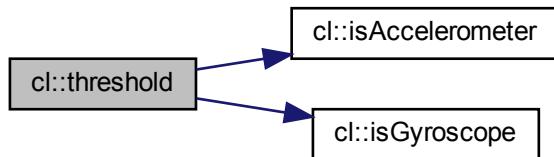
Definition at line 16 of file `s2n.hpp`.

5.1.3.22 `threshold()`

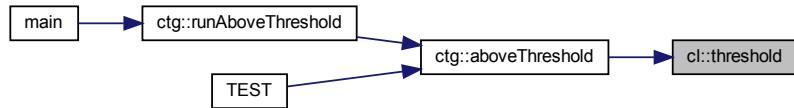
```
long double cl::threshold(
    Channel channel)
```

Definition at line 55 of file `channel.cpp`.

Here is the call graph for this function:



Here is the caller graph for this function:

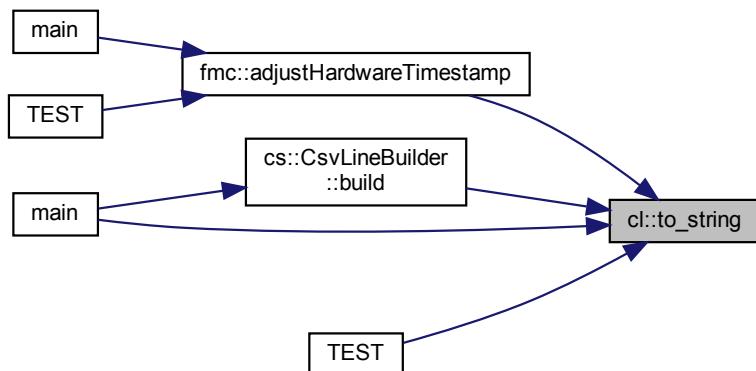


5.1.3.23 `to_string()`

```
template<typename Ty >
std::string cl::to_string (
    const Ty & ty ) [inline]
```

Definition at line 16 of file `to_string.hpp`.

Here is the caller graph for this function:



5.1.3.24 `useUnbufferedIo()`

```
void cl::useUnbufferedIo ( )
```

Definition at line 9 of file `use_unbuffered_io.cpp`.

Here is the caller graph for this function:



5.1.4 Variable Documentation

5.1.4.1 accelerometerThreshold

```
constexpr long double cl::accelerometerThreshold {1.99L} [inline], [constexpr]
```

Definition at line 61 of file channel.hpp.

5.1.4.2 channelCount

```
constexpr std::size_t cl::channelCount [inline], [constexpr]
```

Initial value:

```
{0  
#define CL_CHANNEL_X(enumerator, value, dataSetAccessor)  
    CL_CHANNEL  
}
```

Definition at line 26 of file channel.hpp.

5.1.4.3 channels

```
constexpr std::array<Channel, channelCount> cl::channels [inline], [constexpr]
```

Initial value:

```
{}  
#define CL_CHANNEL_X(enm, v, a)  
    CL_CHANNEL  
}}
```

Definition at line 32 of file channel.hpp.

5.1.4.4 column_index

```
template<Column Col>
constexpr std::size_t cl::column_index = col_traits<Col>::index [inline], [constexpr]
```

Definition at line 46 of file column.hpp.

5.1.4.5 data_set_accessor_v

```
template<Channel Chan>
constexpr CL_CHANNEL DataSet::ChannelAccessor cl::data_set_accessor_v = data_set_accessor<Chan>↔
::f [inline], [constexpr]
```

Definition at line 51 of file channel.hpp.

5.1.4.6 gyroscopeThreshold

```
constexpr long double cl::gyroscopeThreshold {1999.99L} [inline], [constexpr]
```

Definition at line 62 of file channel.hpp.

5.1.4.7 sensors

```
constexpr std::array<Sensor, 4> cl::sensors [inline], [constexpr]
```

Initial value:

```
{}  
#define CL_SENSOR_X(enm, v)  
CL_SENSOR  
{}{}
```

Definition at line 21 of file sensor.hpp.

5.2 cl::fs Namespace Reference

Classes

- class [File](#)
Represents a file.
- class [FileStream](#)
- class [Path](#)

Enumerations

- enum `DirectoryListingOption` { `DirectoryListingOption::None`, `DirectoryListingOption::ExcludeDotAndDotDot` }

Functions

- `Expected< std::vector< Path > > directoryListing (const Path &directoryPath, DirectoryListingOption directoryListingOption=DirectoryListingOption::ExcludeDotAndDotDot)`
- `std::wstring utf8ToUtf16 (pl::string_view utf8)`
- `std::string utf16ToUtf8 (pl::wstring_view utf16)`
- `std::wstring formatError (DWORD errorCode)`
- `std::ostream & operator<< (std::ostream &os, const Path &path)`
- `bool operator< (const Path &lhs, const Path &rhs) noexcept`
- `bool operator== (const Path &lhs, const Path &rhs) noexcept`

5.2.1 Enumeration Type Documentation

5.2.1.1 `DirectoryListingOption`

enum `cl::fs::DirectoryListingOption` [strong]

Enumerator

<code>None</code>	
<code>ExcludeDotAndDotDot</code>	

Definition at line 10 of file `directory_listing.hpp`.

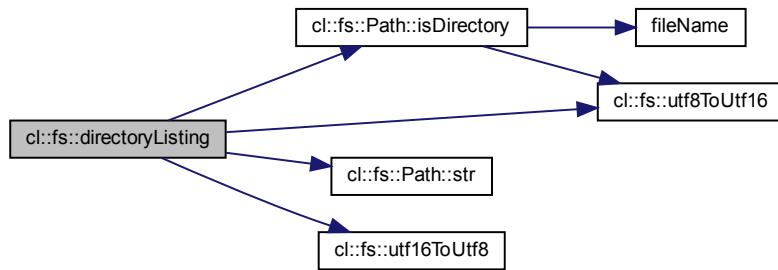
5.2.2 Function Documentation

5.2.2.1 `directoryListing()`

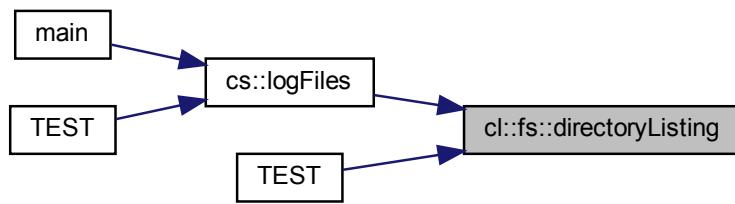
```
Expected< std::vector< Path > > cl::fs::directoryListing (
    const Path & directoryPath,
    DirectoryListingOption directoryListingOption = DirectoryListingOption::ExcludeDotAndDotDot
)
```

Definition at line 24 of file `directory_listing.cpp`.

Here is the call graph for this function:



Here is the caller graph for this function:



5.2.2.2 `formatError()`

```
std::wstring cl::fs::formatError (
    DWORD errorCode )
```

Definition at line 89 of file windows.cpp.

5.2.2.3 `operator<()`

```
bool cl::fs::operator< (
    const Path & lhs,
    const Path & rhs ) [noexcept]
```

Definition at line 27 of file path.cpp.

5.2.2.4 operator<<()

```
std::ostream& cl::fs::operator<< (
    std::ostream & os,
    const Path & path )
```

Definition at line 22 of file path.cpp.

5.2.2.5 operator==(())

```
bool cl::fs::operator== (
    const Path & lhs,
    const Path & rhs ) [noexcept]
```

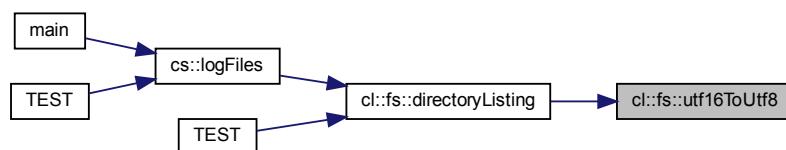
Definition at line 32 of file path.cpp.

5.2.2.6 utf16ToUtf8()

```
std::string cl::fs::utf16ToUtf8 (
    pl::wstring_view utf16 )
```

Definition at line 61 of file windows.cpp.

Here is the caller graph for this function:

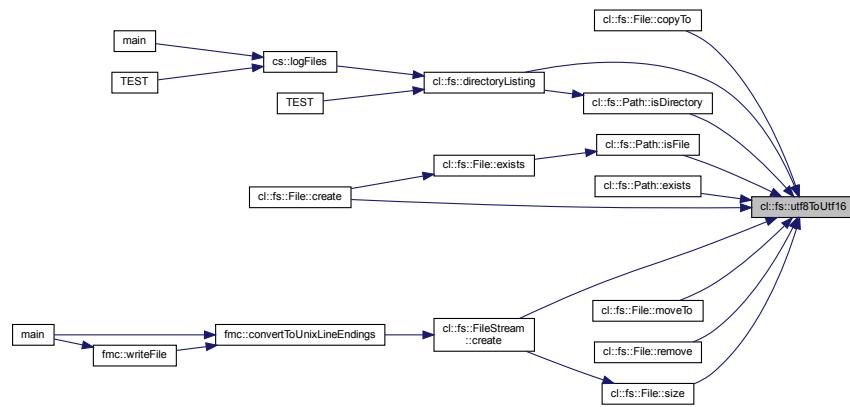


5.2.2.7 utf8ToUtf16()

```
std::wstring cl::fs::utf8ToUtf16 (
    pl::string_view utf8 )
```

Definition at line 35 of file windows.cpp.

Here is the caller graph for this function:



5.3 cs Namespace Reference

Classes

- class [CsvLineBuilder](#)
- struct [data_set_info](#)
- class [LogInfo](#)
- class [LogLine](#)

Enumerations

- enum [FilterKind](#) { `FilterKind::Butterworth`, `FilterKind::MovingAverage` }
- enum [SegmentationKind](#) : `pl::byte` { `SegmentationKind::Minima` = `0b0000'0001`, `SegmentationKind::Maxima` = `0b0000'0010`, `SegmentationKind::Both` = `Minima | Maxima` }

Functions

- [PL_DEFINE_EXCEPTION_TYPE](#) (`NoSuchDataSetException`, `std::logic_error`)
- [CS_SPECIALIZE_DATA_SET_INFO](#) (`Felix1`, "11.17.39", 24)
- [CS_SPECIALIZE_DATA_SET_INFO](#) (`Felix2`, "12.50.00", 20)
- [CS_SPECIALIZE_DATA_SET_INFO](#) (`Felix3`, "13.00.09", 15)
- [CS_SPECIALIZE_DATA_SET_INFO](#) (`Marcelle1`, "14.59.59", 10)
- [CS_SPECIALIZE_DATA_SET_INFO](#) (`Marcelle2`, "15.13.22", 16)
- [CS_SPECIALIZE_DATA_SET_INFO](#) (`Marcelle3`, "15.31.36", 18)
- [CS_SPECIALIZE_DATA_SET_INFO](#) (`Mike1`, "14.07.33", 26)

- `CS_SPECIALIZE_DATA_SET_INFO` (Mike2, "14.14.32", 22)
- `CS_SPECIALIZE_DATA_SET_INFO` (Mike3, "14.20.28", 18)
- `CS_SPECIALIZE_DATA_SET_INFO` (Andre1, "Andre_liegestuetzen1", 27)
- `CS_SPECIALIZE_DATA_SET_INFO` (Andre2, "Andre_liegestuetzen2", 20)
- `CS_SPECIALIZE_DATA_SET_INFO` (Andre3, "Andre_liegestuetzen3", 17)
- `CS_SPECIALIZE_DATA_SET_INFO` (AndreSquats1, "Andre_Squats", 30)
- `CS_SPECIALIZE_DATA_SET_INFO` (AndreSquats2, "Andre_Squats2", 49)
- `CS_SPECIALIZE_DATA_SET_INFO` (Jan1, "Jan_liegestuetzen1", 25)
- `CS_SPECIALIZE_DATA_SET_INFO` (Jan2, "Jan_liegestuetzen2", 19)
- `CS_SPECIALIZE_DATA_SET_INFO` (Jan3, "Jan_liegestuetzen3", 13)
- `CS_SPECIALIZE_DATA_SET_INFO` (Lucas1, "Lucas_liegestuetzen1", 24)
- `CS_SPECIALIZE_DATA_SET_INFO` (Lucas2, "Lucas_liegestuetzen2", 19)
- `CS_SPECIALIZE_DATA_SET_INFO` (Lucas3, "Lucas_liegestuetzen3", 11)
- `std::uint64_t repetitionCount (pl::string_view dataSet)`
- `std::ostream & operator<< (std::ostream &os, FilterKind filterKind)`
- `cl::Expected< std::vector< cl::fs::Path > > logFiles (pl::string_view directoryPath)`
- `std::ostream & operator<< (std::ostream &os, SegmentationKind segmentationKind)`
- `bool operator== (const LogInfo &lhs, const LogInfo &rhs) noexcept`
- `bool operator!= (const LogInfo &lhs, const LogInfo &rhs) noexcept`
- `std::ostream & operator<< (std::ostream &os, const LogInfo &logInfo)`

Variables

- `constexpr pl::string_view logPath {"segmentation_comparison/logs"}`
- `constexpr pl::string_view oldLogPath {"segmentation_comparison/logs/old"}`

5.3.1 Enumeration Type Documentation

5.3.1.1 FilterKind

`enum cs::FilterKind [strong]`

Enumerator

Butterworth	
MovingAverage	

Definition at line 6 of file filter_kind.hpp.

5.3.1.2 SegmentationKind

`enum cs::SegmentationKind : pl::byte [strong]`

Enumerator

Minima	
Maxima	
Both	

Definition at line 9 of file segmentation_kind.hpp.

5.3.2 Function Documentation

5.3.2.1 CS_SPECIALIZE_DATA_SET_INFO() [1/20]

```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    Andre1 ,
    "Andre_liegestuetzen1" ,
    27 )
```

5.3.2.2 CS_SPECIALIZE_DATA_SET_INFO() [2/20]

```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    Andre2 ,
    "Andre_liegestuetzen2" ,
    20 )
```

5.3.2.3 CS_SPECIALIZE_DATA_SET_INFO() [3/20]

```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    Andre3 ,
    "Andre_liegestuetzen3" ,
    17 )
```

5.3.2.4 CS_SPECIALIZE_DATA_SET_INFO() [4/20]

```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    AndreSquats1 ,
    "Andre_Squats" ,
    30 )
```

5.3.2.5 CS_SPECIALIZE_DATA_SET_INFO() [5/20]

```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    AndreSquats2 ,
    "Andre_Squats2" ,
    49   )
```

5.3.2.6 CS_SPECIALIZE_DATA_SET_INFO() [6/20]

```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    Felix1 ,
    "11.17.39" ,
    24   )
```

5.3.2.7 CS_SPECIALIZE_DATA_SET_INFO() [7/20]

```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    Felix2 ,
    "12.50.00" ,
    20   )
```

5.3.2.8 CS_SPECIALIZE_DATA_SET_INFO() [8/20]

```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    Felix3 ,
    "13.00.09" ,
    15   )
```

5.3.2.9 CS_SPECIALIZE_DATA_SET_INFO() [9/20]

```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    Jan1 ,
    "Jan_liegestuetzen1" ,
    25   )
```

5.3.2.10 CS_SPECIALIZE_DATA_SET_INFO() [10/20]

```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    Jan2 ,
    "Jan_liegestuetzen2" ,
    19   )
```

5.3.2.11 CS_SPECIALIZE_DATA_SET_INFO() [11/20]

```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    Jan3 ,
    "Jan_liegestuetzen3" ,
    13  )
```

5.3.2.12 CS_SPECIALIZE_DATA_SET_INFO() [12/20]

```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    Lucas1 ,
    "Lukas_liegestuetzen1" ,
    24  )
```

5.3.2.13 CS_SPECIALIZE_DATA_SET_INFO() [13/20]

```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    Lucas2 ,
    "Lukas_liegestuetzen2" ,
    19  )
```

5.3.2.14 CS_SPECIALIZE_DATA_SET_INFO() [14/20]

```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    Lucas3 ,
    "Lukas_liegestuetzen3" ,
    11  )
```

5.3.2.15 CS_SPECIALIZE_DATA_SET_INFO() [15/20]

```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    Marcellle1 ,
    "14.59.59" ,
    10  )
```

5.3.2.16 CS_SPECIALIZE_DATA_SET_INFO() [16/20]

```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    Marcellle2 ,
    "15.13.22" ,
    16  )
```

5.3.2.17 CS_SPECIALIZE_DATA_SET_INFO() [17/20]

```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    Marcelle3 ,
    "15.31.36" ,
    18   )
```

5.3.2.18 CS_SPECIALIZE_DATA_SET_INFO() [18/20]

```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    Mike1 ,
    "14.07.33" ,
    26   )
```

5.3.2.19 CS_SPECIALIZE_DATA_SET_INFO() [19/20]

```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    Mike2 ,
    "14.14.32" ,
    22   )
```

5.3.2.20 CS_SPECIALIZE_DATA_SET_INFO() [20/20]

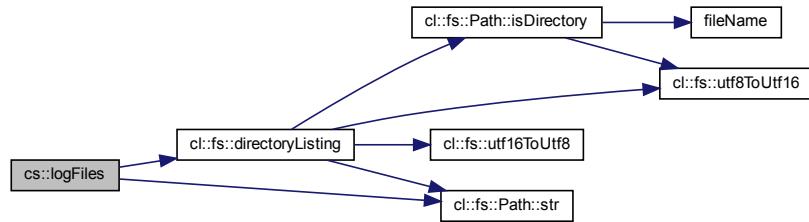
```
cs::CS_SPECIALIZE_DATA_SET_INFO (
    Mike3 ,
    "14.20.28" ,
    18   )
```

5.3.2.21 logFiles()

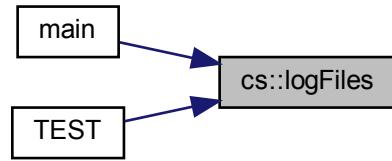
```
cl::Expected< std::vector< cl::fs::Path > > cs::logFiles (
    pl::string_view directoryPath )
```

Definition at line 9 of file log_files.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



5.3.2.22 operator"!=()

```
bool cs::operator!= (
    const LogInfo & lhs,
    const LogInfo & rhs ) [noexcept]
```

Definition at line 269 of file log_info.cpp.

5.3.2.23 operator<<() [1/3]

```
std::ostream& cs::operator<< (
    std::ostream & os,
    const LogInfo & logInfo )
```

Definition at line 274 of file log_info.cpp.

5.3.2.24 operator<<() [2/3]

```
std::ostream & cs::operator<< (
    std::ostream & os,
    FilterKind filterKind )
```

Definition at line 6 of file filter_kind.cpp.

5.3.2.25 operator<<() [3/3]

```
std::ostream & cs::operator<< (
    std::ostream & os,
    SegmentationKind segmentationKind )
```

Definition at line 6 of file segmentation_kind.cpp.

5.3.2.26 operator==()

```
bool cs::operator== (
    const LogInfo & lhs,
    const LogInfo & rhs ) [noexcept]
```

Definition at line 247 of file log_info.cpp.

5.3.2.27 PL_DEFINE_EXCEPTION_TYPE()

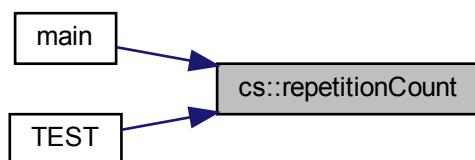
```
cs::PL_DEFINE_EXCEPTION_TYPE (
    NoSuchDataSetException ,
    std::logic_error )
```

5.3.2.28 repetitionCount()

```
std::uint64_t cs::repetitionCount (
    pl::string_view dataSet )
```

Definition at line 10 of file data_set_info.cpp.

Here is the caller graph for this function:



5.3.3 Variable Documentation

5.3.3.1 logPath

```
constexpr pl::string_view cs::logPath {"segmentation_comparison/logs"} [inline], [constexpr]
```

Definition at line 6 of file paths.hpp.

5.3.3.2 oldLogPath

```
constexpr pl::string_view cs::oldLogPath {"segmentation_comparison/logs/old"} [inline], [constexpr]
```

Definition at line 8 of file paths.hpp.

5.4 ctg Namespace Reference

Functions

- std::vector< [cl::DataPoint](#) > [aboveThreshold](#) (const [cl::DataSet](#) &dataSet, long double accelerometerThreshold, long double gyroscopeThreshold)
- long double [averageComparisonValueCalculator](#) ([cl::Sensor](#) sensor, [cl::Channel](#) channel, const [cl::DataSet](#) &dataSet)
- long double [halfMaximumComparisonValueCalculator](#) ([cl::Sensor](#) sensor, [cl::Channel](#) channel, const [cl::DataSet](#) &dataSet)
- template<typename ComparisonValueCalculator>
bool [isRelevant](#) ([cl::Sensor](#) sensor, [cl::Channel](#) channel, const [cl::DataSet](#) &dataSet, ComparisonValueCalculator comparisonValueCalculator)
- constexpr long double [percentageOf](#) (std::size_t amount, std::size_t totalCount) noexcept
- void [runAboveThreshold](#) (std::ostream &aboveThresholdLogFileStream, const [cl::DataSet](#) &dataSet)

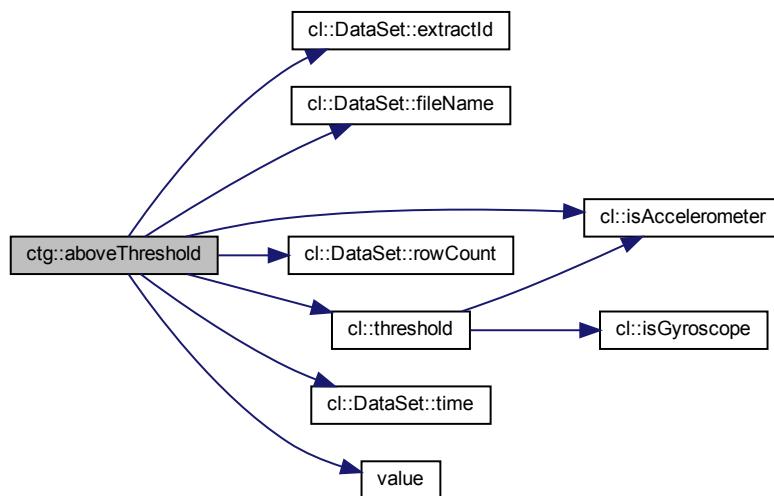
5.4.1 Function Documentation

5.4.1.1 aboveThreshold()

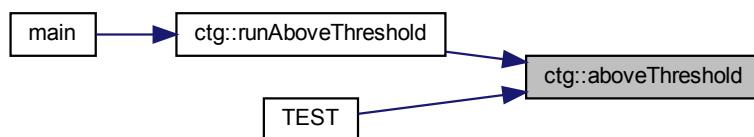
```
std::vector< cl::DataPoint > ctg::aboveThreshold (
    const cl::DataSet & dataSet,
    long double accelerometerThreshold,
    long double gyroscopeThreshold )
```

Definition at line 28 of file above_threshold.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



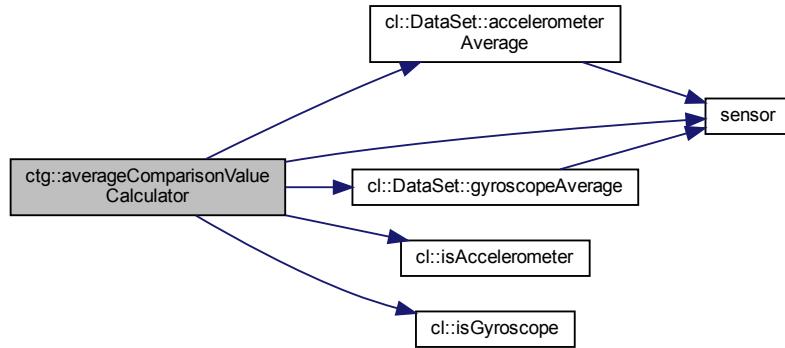
5.4.1.2 averageComparisonValueCalculator()

```
long double ctg::averageComparisonValueCalculator (
    cl::Sensor sensor,
```

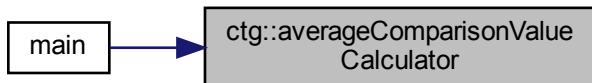
```
cl::Channel channel,
const cl::DataSet & dataSet )
```

Definition at line 10 of file average_comparison_value_calculator.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

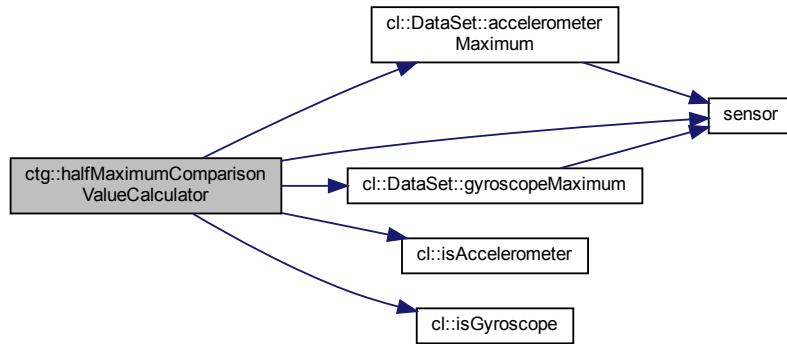


5.4.1.3 halfMaximumComparisonValueCalculator()

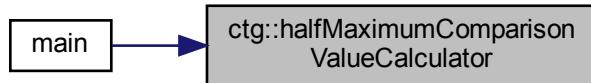
```
long double ctg::halfMaximumComparisonValueCalculator (
    cl::Sensor sensor,
    cl::Channel channel,
    const cl::DataSet & dataSet )
```

Definition at line 10 of file half_maximum_comparison_value_calculator.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



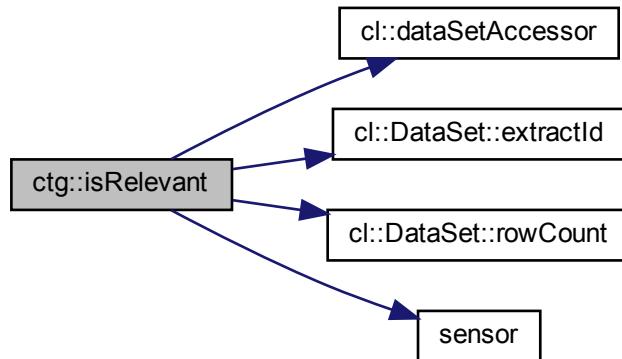
5.4.1.4 `isRelevant()`

```

template<typename ComparisonValueCalculator >
bool ctg::isRelevant (
    cl::Sensor sensor,
    cl::Channel channel,
    const cl::DataSet & dataSet,
    ComparisonValueCalculator comparisonValueCalculator )
  
```

Definition at line 11 of file `is_relevant.hpp`.

Here is the call graph for this function:



Here is the caller graph for this function:



5.4.1.5 percentageOf()

```

constexpr long double ctg::percentageOf (
    std::size_t amount,
    std::size_t totalCount ) [constexpr], [noexcept]
  
```

Definition at line 6 of file percentage_of.hpp.

Here is the caller graph for this function:

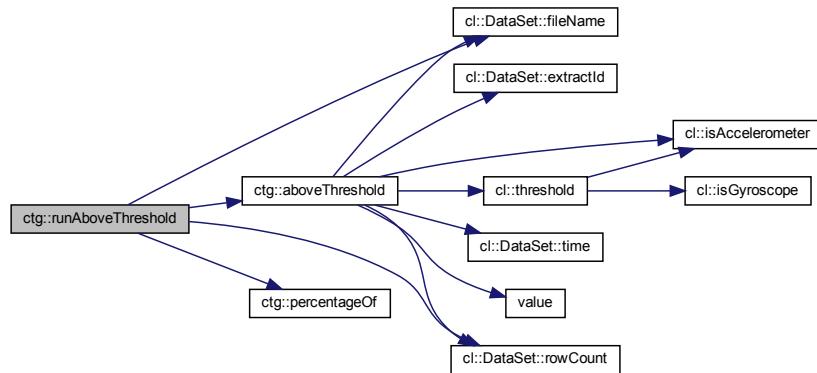


5.4.1.6 runAboveThreshold()

```
void ctg::runAboveThreshold (
    std::ostream & aboveThresholdLogFileStream,
    const cl::DataSet & dataSet )
```

Definition at line 14 of file run_above_threshold.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



5.5 fmc Namespace Reference

Functions

- void [adjustHardwareTimestamp](#) (std::string *cellContent, const std::string &nextRowHardwareTimestamp, std::uint64_t *overflowCount)
- bool [convertToUnixLineEndings](#) (const std::string &csvPath)
- bool [createBackupFile](#) (const std::string &csvFilePath, const std::string &backupFilePath)
- void [deleteNonBoschSensors](#) (std::vector< std::vector< std::string >> *data)
- [cl::Expected< void >](#) [deleteOutOfBoundsValues](#) (std::vector< std::vector< std::string >> *data)
- void [removeZerosFromField](#) (std::string *field)
- bool [restoreFromBackup](#) (const std::string &csvFilePath, const std::string &backupFilePath)
- bool [writeFile](#) (pl::string_view csvPath, pl::string_view csvFileExtension, const std::vector< std::string > &columnNames, const std::vector< std::vector< std::string >> &data)

5.5.1 Function Documentation

5.5.1.1 **adjustHardwareTimestamp()**

```
void fmc::adjustHardwareTimestamp (
    std::string * cellContent,
    const std::string & nextRowHardwareTimestamp,
    std::uint64_t * overflowCount )
```

Definition at line 16 of file `adjust_hardware_timestamp.cpp`.

Here is the call graph for this function:



Here is the caller graph for this function:

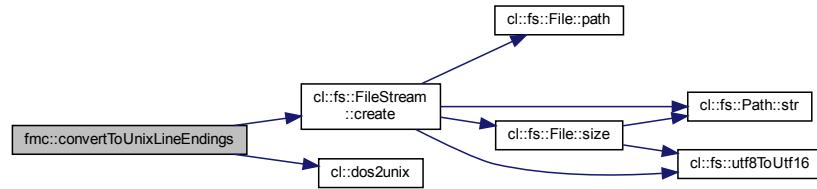


5.5.1.2 **convertToUnixLineEndings()**

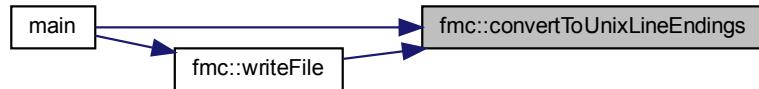
```
bool fmc::convertToUnixLineEndings (
    const std::string & csvPath )
```

Definition at line 18 of file `convert_to_unix_line_endings.cpp`.

Here is the call graph for this function:



Here is the caller graph for this function:



5.5.1.3 `createBackupFile()`

```

bool fmc::createBackupFile (
    const std::string & csvFilePath,
    const std::string & backupFilePath )
  
```

Definition at line 6 of file `create_backup_file.cpp`.

Here is the caller graph for this function:



5.5.1.4 deleteNonBoschSensors()

```
void fmc::deleteNonBoschSensors (
    std::vector< std::vector< std::string >> * data )
```

Definition at line 30 of file `delete_non_bosch_sensors.cpp`.

Here is the caller graph for this function:



5.5.1.5 deleteOutOfBoundsValues()

```
cl::Expected< void > fmc::deleteOutOfBoundsValues (
    std::vector< std::vector< std::string >> * data )
```

Definition at line 29 of file `delete_out_of_bounds_values.cpp`.

Here is the caller graph for this function:

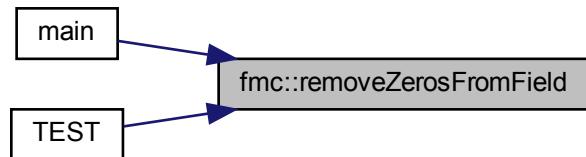


5.5.1.6 removeZerosFromField()

```
void fmc::removeZerosFromField (
    std::string * field )
```

Definition at line 6 of file `remove_zeros_from_field.cpp`.

Here is the caller graph for this function:



5.5.1.7 `restoreFromBackup()`

```
bool fmc::restoreFromBackup (
    const std::string & csvFilePath,
    const std::string & backupFilePath )
```

Definition at line 11 of file `restore_from_backup.cpp`.

Here is the caller graph for this function:

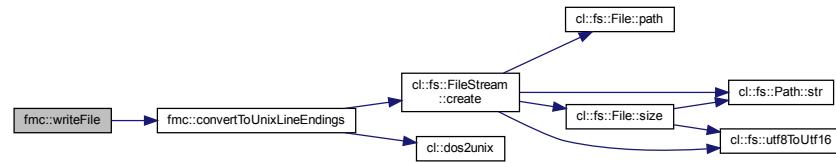


5.5.1.8 `writeFile()`

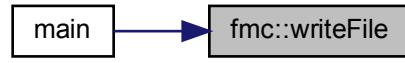
```
bool fmc::writeFile (
    pl::string_view csvPath,
    pl::string_view csvFileExtension,
    const std::vector< std::string > & columnNames,
    const std::vector< std::vector< std::string >> & data )
```

Definition at line 12 of file `write_file.cpp`.

Here is the call graph for this function:



Here is the caller graph for this function:



Chapter 6

Class Documentation

6.1 cl::col_traits< Col > Struct Template Reference

```
#include <column.hpp>
```

6.1.1 Detailed Description

```
template<Column Col>
struct cl::col_traits< Col >
```

Definition at line 24 of file column.hpp.

The documentation for this struct was generated from the following file:

- csv_lib/include/cl/column.hpp

6.2 cs::CsvLineBuilder Class Reference

```
#include <csv_line.hpp>
```

Public Types

- using `this_type = CsvLineBuilder`

Public Member Functions

- `CsvLineBuilder ()`
- `this_type & skipWindow (bool value)`
- `this_type & deleteTooClose (bool value)`
- `this_type & deleteLowVariance (bool value)`
- `this_type & kind (SegmentationKind value)`
- `this_type & windowSize (std::uint64_t value)`
- `this_type & filter (FilterKind value)`
- `this_type & dataSet (std::string value)`
- `this_type & sensor (std::uint64_t value)`
- `this_type & pushUps (std::uint64_t value)`
- `this_type & segmentationPoints (std::uint64_t value)`
- `this_type & isOld (bool value)`
- `std::vector< std::string > build () const`

6.2.1 Detailed Description

Definition at line 14 of file csv_line.hpp.

6.2.2 Member Typedef Documentation

6.2.2.1 `this_type`

```
using cs::CsvLineBuilder::this_type = CsvLineBuilder
```

Definition at line 16 of file csv_line.hpp.

6.2.3 Constructor & Destructor Documentation

6.2.3.1 `CsvLineBuilder()`

```
cs::CsvLineBuilder::CsvLineBuilder( )
```

Definition at line 35 of file csv_line.cpp.

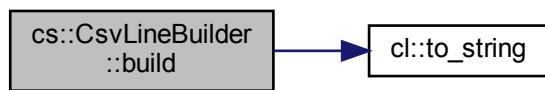
6.2.4 Member Function Documentation

6.2.4.1 `build()`

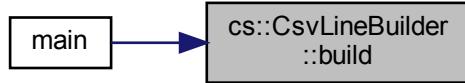
```
std::vector< std::string > cs::CsvLineBuilder::build( ) const
```

Definition at line 115 of file csv_line.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.4.2 dataSet()

```
CsvLineBuilder & cs::CsvLineBuilder::dataSet ( std::string value )
```

Definition at line 85 of file csv_line.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.4.3 deleteLowVariance()

```
CsvLineBuilder & cs::CsvLineBuilder::deleteLowVariance (
    bool value )
```

Definition at line 61 of file csv_line.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

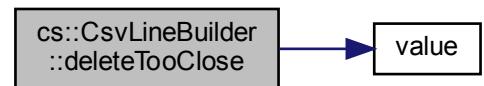


6.2.4.4 deleteTooClose()

```
CsvLineBuilder & cs::CsvLineBuilder::deleteTooClose (
    bool value )
```

Definition at line 55 of file csv_line.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

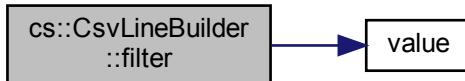


6.2.4.5 filter()

```
CsvLineBuilder & cs::CsvLineBuilder::filter (
    FilterKind value )
```

Definition at line 79 of file csv_line.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

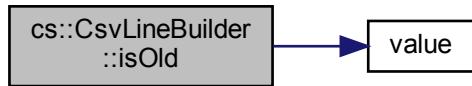


6.2.4.6 `isOld()`

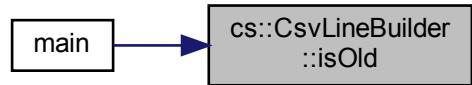
```
CsvLineBuilder & cs::CsvLineBuilder::isOld (
    bool value )
```

Definition at line 109 of file csv_line.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

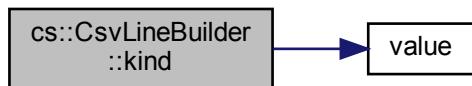


6.2.4.7 `kind()`

```
CsvLineBuilder & cs::CsvLineBuilder::kind (
    SegmentationKind value )
```

Definition at line 67 of file csv_line.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.4.8 pushUps()

```
CsvLineBuilder & cs::CsvLineBuilder::pushUps ( std::uint64_t value )
```

Definition at line 97 of file csv_line.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.4.9 segmentationPoints()

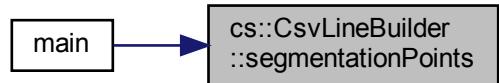
```
CsvLineBuilder & cs::CsvLineBuilder::segmentationPoints (
    std::uint64_t value )
```

Definition at line 103 of file csv_line.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

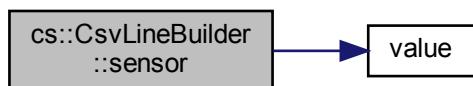


6.2.4.10 sensor()

```
CsvLineBuilder & cs::CsvLineBuilder::sensor (
    std::uint64_t value )
```

Definition at line 91 of file csv_line.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

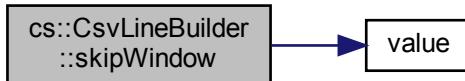


6.2.4.11 skipWindow()

```
CsvLineBuilder & cs::CsvLineBuilder::skipWindow ( bool value )
```

Definition at line 49 of file csv_line.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

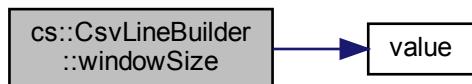


6.2.4.12 `windowSize()`

```
CsvLineBuilder & cs::CsvLineBuilder::windowSize (
    std::uint64_t value )
```

Definition at line 73 of file `csv_line.cpp`.

Here is the call graph for this function:



Here is the caller graph for this function:



The documentation for this class was generated from the following files:

- compare_segmentation/include/[csv_line.hpp](#)
- compare_segmentation/src/[csv_line.cpp](#)

6.3 `cl::data_set_accessor< Chan >` Struct Template Reference

```
#include <channel.hpp>
```

6.3.1 Detailed Description

```
template<Channel Chan>
struct cl::data_set_accessor< Chan >
```

Definition at line 39 of file `channel.hpp`.

The documentation for this struct was generated from the following file:

- `csv_lib/include/cl/channel.hpp`

6.4 cs::data_set_info< Tag > Struct Template Reference

```
#include <data_set_info.hpp>
```

6.4.1 Detailed Description

```
template<typename Tag>
struct cs::data_set_info< Tag >
```

Definition at line 14 of file data_set_info.hpp.

The documentation for this struct was generated from the following file:

- compare_segmentation/include/[data_set_info.hpp](#)

6.5 cl::DataPoint Class Reference

```
#include <data_point.hpp>
```

Public Member Functions

- [DataPoint](#) (std::string [fileName](#), long double [time](#), Sensor [sensor](#), Channel [channel](#), long double [value](#)) noexcept
- const std::string & [fileName](#) () const noexcept
- long double [time](#) () const noexcept
- Sensor [sensor](#) () const noexcept
- Channel [channel](#) () const noexcept
- long double [value](#) () const noexcept

Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [DataPoint](#) &dataPoint)

6.5.1 Detailed Description

Definition at line 10 of file data_point.hpp.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 DataPoint()

```
DataPoint::DataPoint (
    std::string fileName,
    long double time,
    Sensor sensor,
    Channel channel,
    long double value ) [noexcept]
```

Definition at line 21 of file data_point.cpp.

Here is the call graph for this function:



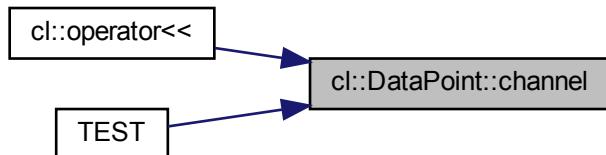
6.5.3 Member Function Documentation

6.5.3.1 channel()

```
Channel DataPoint::channel ( ) const [noexcept]
```

Definition at line 41 of file data_point.cpp.

Here is the caller graph for this function:

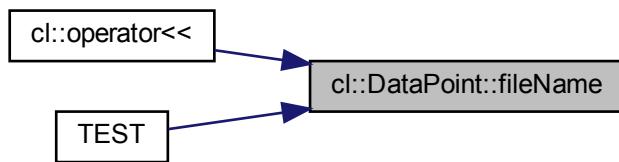


6.5.3.2 fileName()

```
const std::string & DataPoint::fileName ( ) const [noexcept]
```

Definition at line 35 of file data_point.cpp.

Here is the caller graph for this function:

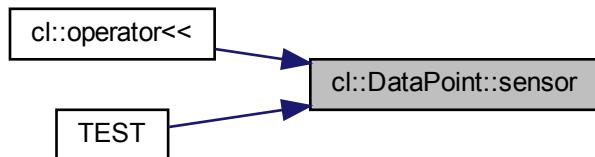


6.5.3.3 sensor()

```
Sensor DataPoint::sensor ( ) const [noexcept]
```

Definition at line 39 of file data_point.cpp.

Here is the caller graph for this function:

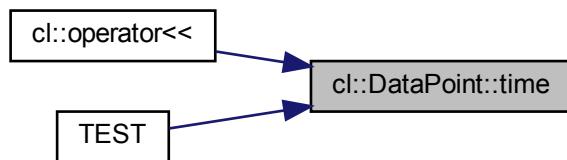


6.5.3.4 time()

```
long double DataPoint::time ( ) const [noexcept]
```

Definition at line 37 of file data_point.cpp.

Here is the caller graph for this function:

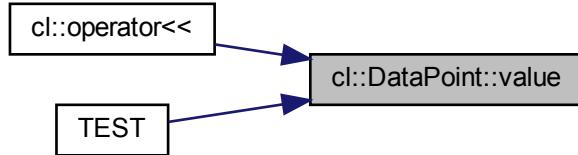


6.5.3.5 value()

```
long double DataPoint::value ( ) const [noexcept]
```

Definition at line 43 of file data_point.cpp.

Here is the caller graph for this function:



6.5.4 Friends And Related Function Documentation

6.5.4.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const DataPoint & dataPoint ) [friend]
```

Definition at line 10 of file data_point.cpp.

The documentation for this class was generated from the following files:

- csv_lib/include/cl/data_point.hpp
- csv_lib/src/cl/data_point.cpp

6.6 cl::DataSet Class Reference

```
#include <data_set.hpp>
```

Public Types

- using `size_type` = `std::size_t`
- using `ChannelAccessor` = `long double(DataSet::*)(size_type)` const

Public Member Functions

- `size_type rowCount () const noexcept`
- `const std::string & fileName () const noexcept`
- `column_type< Column::Time > time (size_type index) const`
- `column_type< Column::HardwareTimestamp > hardwareTimestamp (size_type index) const`
- `column_type< Column::ExtractId > extractId (size_type index) const`
- `column_type< Column::Trigger > trigger (size_type index) const`
- `column_type< Column::AccelerometerX > accelerometerX (size_type index) const`
- `column_type< Column::AccelerometerY > accelerometerY (size_type index) const`
- `column_type< Column::AccelerometerZ > accelerometerZ (size_type index) const`
- `column_type< Column::GyroscopeX > gyroscopeX (size_type index) const`
- `column_type< Column::GyroscopeY > gyroscopeY (size_type index) const`
- `column_type< Column::GyroscopeZ > gyroscopeZ (size_type index) const`
- `long double accelerometerAverage (Sensor sensor) const`
- `long double gyroscopeAverage (Sensor sensor) const`
- `long double accelerometerMaximum (Sensor sensor) const`
- `long double gyroscopeMaximum (Sensor sensor) const`

Static Public Member Functions

- static `Expected< DataSet > create (std::string fileName, const std::vector< std::vector< std::string >> &matrix)`

6.6.1 Detailed Description

Definition at line 14 of file data_set.hpp.

6.6.2 Member Typedef Documentation

6.6.2.1 ChannelAccessor

```
using cl::DataSet::ChannelAccessor = long double (DataSet::*)(size_type) const
```

Definition at line 17 of file data_set.hpp.

6.6.2.2 size_type

```
using cl::DataSet::size_type = std::size_t
```

Definition at line 16 of file data_set.hpp.

6.6.3 Member Function Documentation

6.6.3.1 accelerometerAverage()

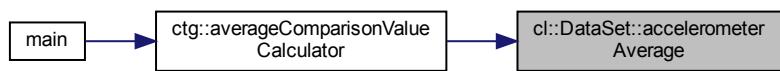
```
long double cl::DataSet::accelerometerAverage (
    Sensor sensor ) const
```

Definition at line 255 of file data_set.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

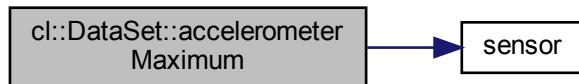


6.6.3.2 accelerometerMaximum()

```
long double cl::DataSet::accelerometerMaximum (
    Sensor sensor ) const
```

Definition at line 265 of file data_set.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

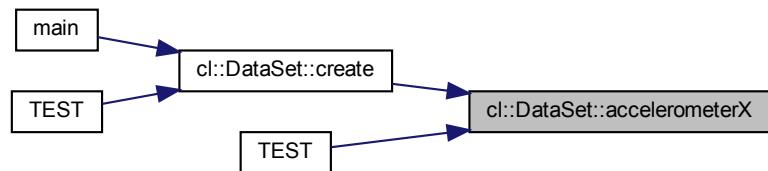


6.6.3.3 accelerometerX()

```
column_type< Column::AccelerometerX > cl::DataSet::accelerometerX (
    size_type index ) const
```

Definition at line 200 of file data_set.cpp.

Here is the caller graph for this function:

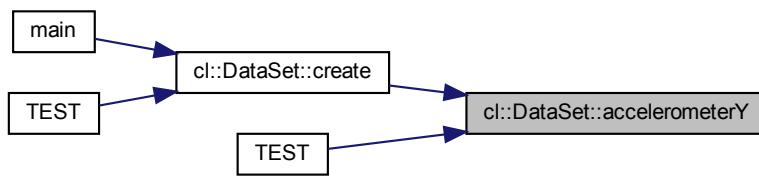


6.6.3.4 accelerometerY()

```
column_type< Column::AccelerometerY > cl::DataSet::accelerometerY ( size_type index ) const
```

Definition at line 208 of file data_set.cpp.

Here is the caller graph for this function:

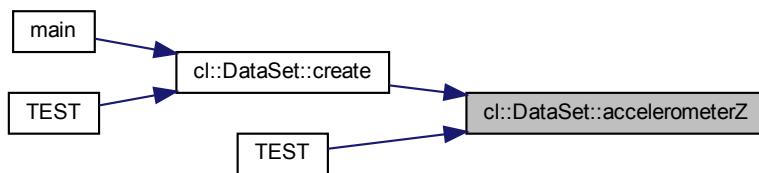


6.6.3.5 accelerometerZ()

```
column_type< Column::AccelerometerZ > cl::DataSet::accelerometerZ ( size_type index ) const
```

Definition at line 216 of file data_set.cpp.

Here is the caller graph for this function:

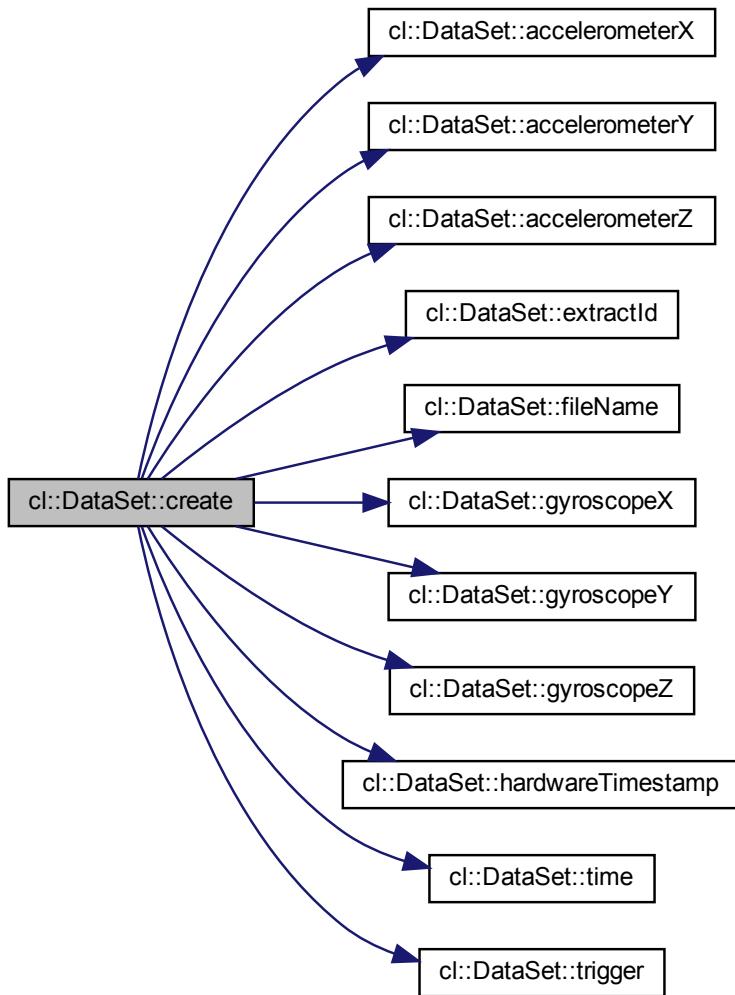


6.6.3.6 create()

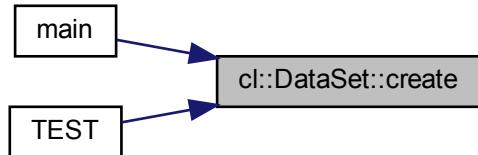
```
Expected< DataSet > cl::DataSet::create (
    std::string fileName,
    const std::vector< std::vector< std::string >> & matrix ) [static]
```

Definition at line 42 of file data_set.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

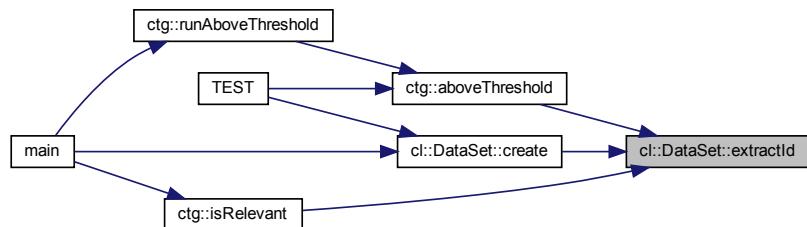


6.6.3.7 extractId()

```
column_type< Column::ExtractId > cl::DataSet::extractId (
    size_type index ) const
```

Definition at line 186 of file data_set.cpp.

Here is the caller graph for this function:

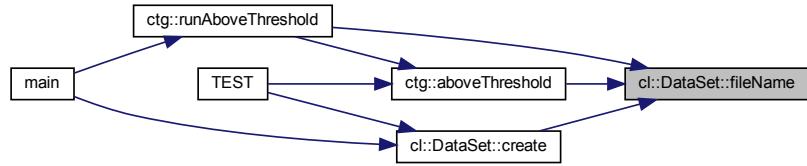


6.6.3.8 fileName()

```
const std::string & cl::DataSet::fileName ( ) const [noexcept]
```

Definition at line 169 of file data_set.cpp.

Here is the caller graph for this function:



6.6.3.9 gyroscopeAverage()

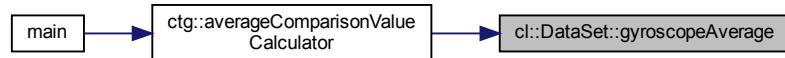
```
long double cl::DataSet::gyroscopeAverage ( Sensor sensor ) const
```

Definition at line 260 of file data_set.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



6.6.3.10 gyroscopeMaximum()

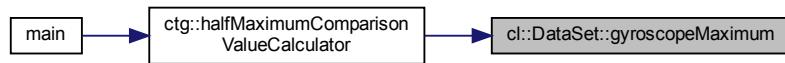
```
long double cl::DataSet::gyroscopeMaximum (
    Sensor sensor ) const
```

Definition at line 270 of file data_set.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

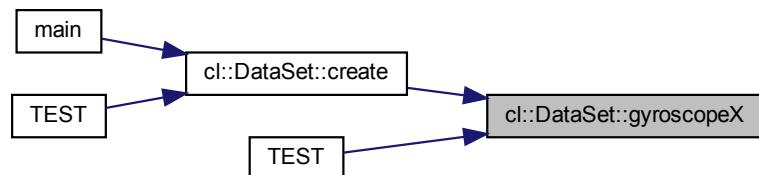


6.6.3.11 gyroscopeX()

```
column_type< Column::GyroscopeX > cl::DataSet::gyroscopeX (
    size_type index ) const
```

Definition at line 224 of file data_set.cpp.

Here is the caller graph for this function:

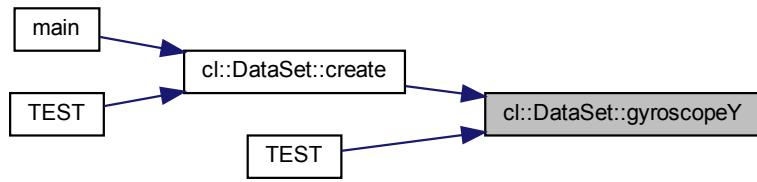


6.6.3.12 gyroscopeY()

```
column_type< Column::GyroscopeY > cl::DataSet::gyroscopeY ( size_type index ) const
```

Definition at line 231 of file data_set.cpp.

Here is the caller graph for this function:

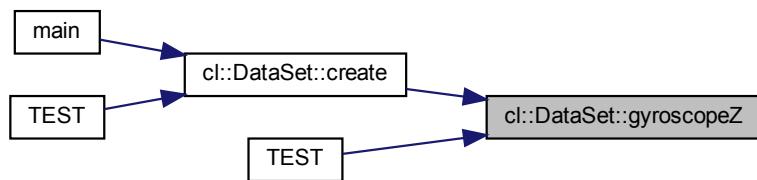


6.6.3.13 gyroscopeZ()

```
column_type< Column::GyroscopeZ > cl::DataSet::gyroscopeZ ( size_type index ) const
```

Definition at line 238 of file data_set.cpp.

Here is the caller graph for this function:

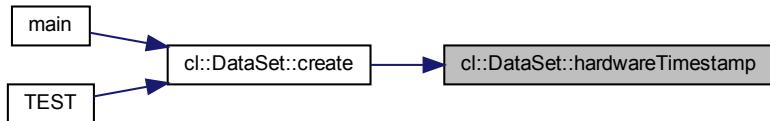


6.6.3.14 hardwareTimestamp()

```
column_type< Column::HardwareTimestamp > cl::DataSet::hardwareTimestamp (
    size_type index ) const
```

Definition at line 178 of file data_set.cpp.

Here is the caller graph for this function:

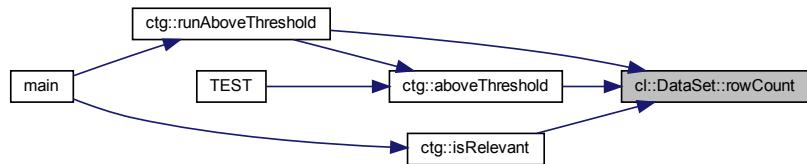


6.6.3.15 rowCount()

```
DataSet::size_type cl::DataSet::rowCount ( ) const [noexcept]
```

Definition at line 152 of file data_set.cpp.

Here is the caller graph for this function:

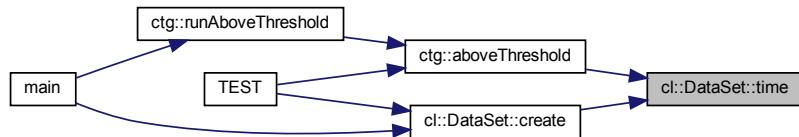


6.6.3.16 time()

```
column_type< Column::Time > cl::DataSet::time (
    size_type index ) const
```

Definition at line 171 of file data_set.cpp.

Here is the caller graph for this function:

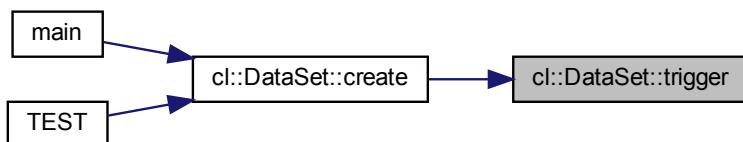


6.6.3.17 trigger()

```
column_type< Column::Trigger > cl::DataSet::trigger (
    size_type index ) const
```

Definition at line 193 of file `data_set.cpp`.

Here is the caller graph for this function:



The documentation for this class was generated from the following files:

- `csv_lib/include/cl/data_set.hpp`
- `csv_lib/src/cl/data_set.cpp`

6.7 cl::Error Class Reference

```
#include <error.hpp>
```

Public Types

- enum `Kind` { `CL_ERROR_KIND` }

Public Member Functions

- `Error (Kind kind, std::string file, std::string function, std::size_t line, std::string message)`
- `Kind kind () const noexcept`
- `const std::string & file () const noexcept`
- `const std::string & function () const noexcept`
- `std::size_t line () const noexcept`
- `const std::string & message () const noexcept`
- `void raise () const`
- `std::string to_string () const`

Friends

- `std::ostream & operator<< (std::ostream &os, const Error &error)`

6.7.1 Detailed Description

Definition at line 22 of file error.hpp.

6.7.2 Member Enumeration Documentation

6.7.2.1 Kind

`enum cl::Error::Kind`

Enumerator

<code>CL_ERROR_KIND</code>	<input type="button" value=""/>
----------------------------	---------------------------------

Definition at line 25 of file error.hpp.

6.7.3 Constructor & Destructor Documentation

6.7.3.1 Error()

```
cl::Error::Error (
    Kind kind,
    std::string file,
    std::string function,
    std::size_t line,
    std::string message )
```

Definition at line 41 of file error.cpp.

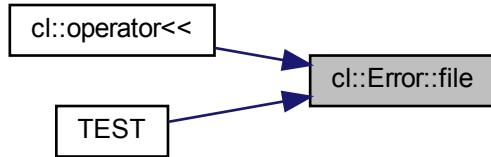
6.7.4 Member Function Documentation

6.7.4.1 file()

```
const std::string & cl::Error::file() const [noexcept]
```

Definition at line 57 of file error.cpp.

Here is the caller graph for this function:

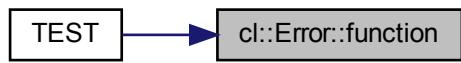


6.7.4.2 function()

```
const std::string & cl::Error::function() const [noexcept]
```

Definition at line 59 of file error.cpp.

Here is the caller graph for this function:

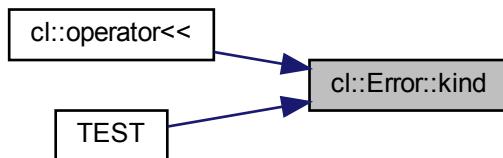


6.7.4.3 kind()

```
Error::Kind cl::Error::kind ( ) const [noexcept]
```

Definition at line 55 of file error.cpp.

Here is the caller graph for this function:



6.7.4.4 line()

```
std::size_t cl::Error::line ( ) const [noexcept]
```

Definition at line 61 of file error.cpp.

6.7.4.5 message()

```
const std::string & cl::Error::message ( ) const [noexcept]
```

Definition at line 63 of file error.cpp.

Here is the caller graph for this function:



6.7.4.6 raise()

```
void cl::Error::raise ( ) const
```

Definition at line 65 of file error.cpp.

6.7.4.7 to_string()

```
std::string cl::Error::to_string ( ) const
```

Definition at line 74 of file error.cpp.

6.7.5 Friends And Related Function Documentation

6.7.5.1 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const Error & error ) [friend]
```

Definition at line 30 of file error.cpp.

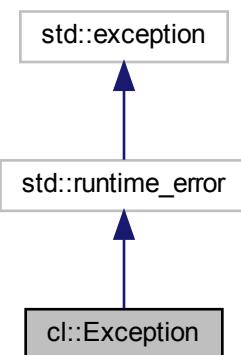
The documentation for this class was generated from the following files:

- csv_lib/include/cl/error.hpp
- csv_lib/src/cl/error.cpp

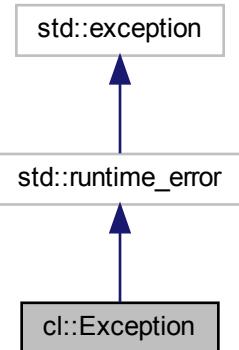
6.8 cl::Exception Class Reference

```
#include <exception.hpp>
```

Inheritance diagram for cl::Exception:



Collaboration diagram for cl::Exception:



Public Types

- using `base_type` = `std::runtime_error`

Public Member Functions

- `Exception (std::string file, std::string function, std::size_t line, const std::string &what_arg)`
- `Exception (std::string file, std::string function, std::size_t line, const char *what_arg)`
- `const std::string & file () const noexcept`
- `const std::string & function () const noexcept`
- `std::size_t line () const noexcept`

6.8.1 Detailed Description

Definition at line 9 of file exception.hpp.

6.8.2 Member Typedef Documentation

6.8.2.1 `base_type`

```
using cl::Exception::base_type = std::runtime_error
```

Definition at line 11 of file exception.hpp.

6.8.3 Constructor & Destructor Documentation

6.8.3.1 Exception() [1/2]

```
cl::Exception::Exception (
    std::string file,
    std::string function,
    std::size_t line,
    const std::string & what_arg )
```

Definition at line 6 of file exception.cpp.

6.8.3.2 Exception() [2/2]

```
cl::Exception::Exception (
    std::string file,
    std::string function,
    std::size_t line,
    const char * what_arg )
```

Definition at line 18 of file exception.cpp.

6.8.4 Member Function Documentation

6.8.4.1 file()

```
const std::string & cl::Exception::file ( ) const [noexcept]
```

Definition at line 30 of file exception.cpp.

6.8.4.2 function()

```
const std::string & cl::Exception::function ( ) const [noexcept]
```

Definition at line 32 of file exception.cpp.

6.8.4.3 line()

```
std::size_t cl::Exception::line() const [noexcept]
```

Definition at line 34 of file exception.cpp.

The documentation for this class was generated from the following files:

- [csv_lib/include/cl/exception.hpp](#)
- [csv_lib/src/cl/exception.cpp](#)

6.9 cl::fs::File Class Reference

Represents a file.

```
#include <file.hpp>
```

Public Member Functions

- [File \(Path path\)](#)
Creates a [File](#) from the given path.
- [bool exists \(\) const noexcept](#)
Determines if this file exists.
- [bool create \(\) const noexcept](#)
Creates this file.
- [bool copyTo \(const Path ©ToPath\) const noexcept](#)
Copies this file in the filesystem.
- [bool moveTo \(const Path &newPath\)](#)
Moves this file in the filesystem.
- [bool remove \(\) noexcept](#)
Deletes this file.
- [std::int64_t size \(\) const noexcept](#)
Determines the size of this file in bytes.
- [const Path & path \(\) const noexcept](#)
Read accessor for the path of this file.

6.9.1 Detailed Description

Represents a file.

Definition at line 11 of file file.hpp.

6.9.2 Constructor & Destructor Documentation

6.9.2.1 File()

```
cl::fs::File::File (
    Path path ) [explicit]
```

Creates a [File](#) from the given path.

Parameters

<i>path</i>	The path to use.
-------------	------------------

Definition at line 21 of file file.cpp.

Here is the call graph for this function:



6.9.3 Member Function Documentation

6.9.3.1 copyTo()

```
bool cl::fs::File::copyTo (const Path & copyToPath) const [noexcept]
```

Copies this file in the filesystem.

Parameters

<i>copyToPath</i>	The path to copy to.
-------------------	----------------------

Returns

true if the file was successfully copied to *copyToPath*; otherwise false.

Warning

There should be no file that already exists at *copyToPath*.

Definition at line 56 of file file.cpp.

Here is the call graph for this function:



6.9.3.2 create()

```
bool cl::fs::File::create( ) const [noexcept]
```

Creates this file.

Returns

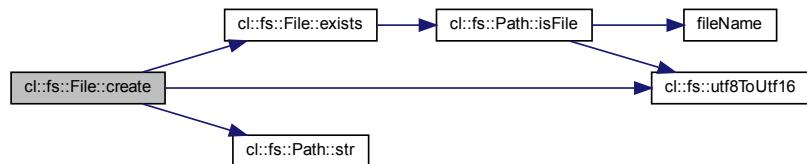
true if the file was successfully created; otherwise false.

Note

Will fail if the file already exists.

Definition at line 25 of file file.cpp.

Here is the call graph for this function:



6.9.3.3 exists()

```
bool cl::fs::File::exists () const [noexcept]
```

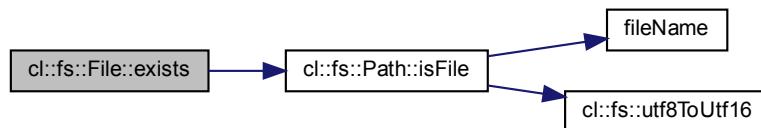
Determines if this file exists.

Returns

true if the file exists; otherwise false.

Definition at line 23 of file file.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



6.9.3.4 moveTo()

```
bool cl::fs::File::moveTo (
    const Path & newPath )
```

Moves this file in the filesystem.

Parameters

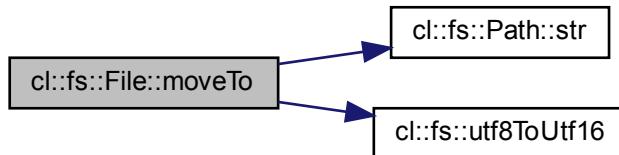
<code>newPath</code>	The path to move this file to.
----------------------	--------------------------------

Returns

true if the file was successfully moved to newPath; otherwise false.

Definition at line 100 of file file.cpp.

Here is the call graph for this function:

**6.9.3.5 path()**

```
const Path & cl::fs::File::path() const [noexcept]
```

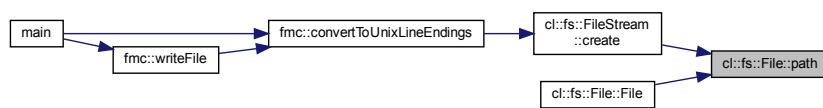
Read accessor for the path of this file.

Returns

The path of this file.

Definition at line 169 of file file.cpp.

Here is the caller graph for this function:



6.9.3.6 remove()

```
bool cl::fs::File::remove( ) [noexcept]
```

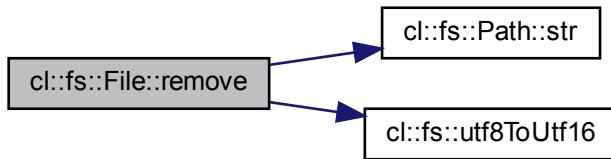
Deletes this file.

Returns

true if deleting succeeded; otherwise false.

Definition at line 117 of file file.cpp.

Here is the call graph for this function:



6.9.3.7 size()

```
std::int64_t cl::fs::File::size( ) const [noexcept]
```

Determines the size of this file in bytes.

Returns

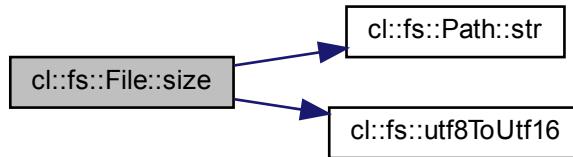
The size of this file in bytes or -1 on error.

Warning

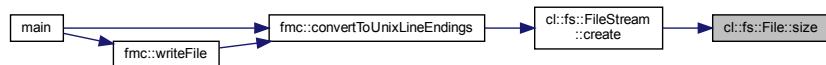
Returns -1 on error.

Definition at line 128 of file `file.cpp`.

Here is the call graph for this function:



Here is the caller graph for this function:



The documentation for this class was generated from the following files:

- `csv_lib/include/cl/fs/file.hpp`
- `csv_lib/src/cl/fs/file.cpp`

6.10 `cl::fs::FileStream` Class Reference

```
#include <file_stream.hpp>
```

Public Types

- enum `OpenMode` : `std::uint8_t` { `Read` = 0b0000'0001, `Write` = 0b0000'0010, `ReadWrite` = `Read` | `Write` }
- using `this_type` = `FileStream`

Public Member Functions

- `PL_NOCOPYABLE (FileStream)`
- `FileStream (this_type &&other) noexcept`
- `this_type & operator= (this_type &&other) noexcept`
- `~FileStream ()`
- bool `write (const void *data, std::size_t byteCount)`
- `std::vector< pl::byte > readAll () const`

Static Public Member Functions

- static Expected< FileStream > create (const File &file, OpenMode openMode)

6.10.1 Detailed Description

Definition at line 16 of file file_stream.hpp.

6.10.2 Member Typedef Documentation

6.10.2.1 this_type

```
using cl::fs::FileStream::this_type = FileStream
```

Definition at line 24 of file file_stream.hpp.

6.10.3 Member Enumeration Documentation

6.10.3.1 OpenMode

```
enum cl::fs::FileStream::OpenMode : std::uint8_t
```

Enumerator

Read	
Write	
ReadWrite	

Definition at line 18 of file file_stream.hpp.

6.10.4 Constructor & Destructor Documentation

6.10.4.1 FileStream()

```
cl::fs::FileStream::FileStream (
    this_type && other ) [noexcept]
```

Definition at line 64 of file file_stream.cpp.

6.10.4.2 ~FileStream()

```
cl::fs::FileStream::~FileStream ( )
```

Definition at line 78 of file file_stream.cpp.

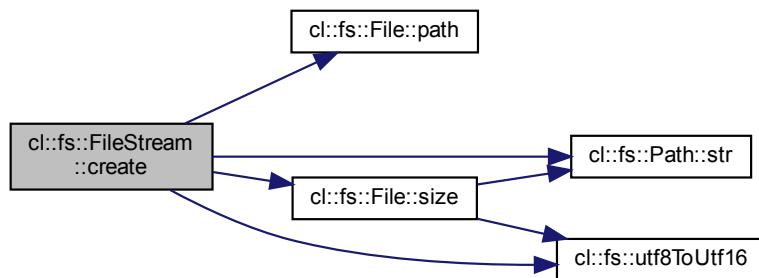
6.10.5 Member Function Documentation

6.10.5.1 create()

```
Expected< FileStream > cl::fs::FileStream::create (
    const File & file,
    OpenMode openMode ) [static]
```

Definition at line 30 of file file_stream.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



6.10.5.2 operator=()

```
FileStream & cl::fs::FileStream::operator= (
    this_type && other ) [noexcept]
```

Definition at line 71 of file file_stream.cpp.

6.10.5.3 PL_NONCOPYABLE()

```
cl::fs::FileStream::PL_NONCOPYABLE (
    FileStream )
```

6.10.5.4 readAll()

```
std::vector< pl::byte > cl::fs::FileStream::readAll ( ) const
```

Definition at line 97 of file file_stream.cpp.

6.10.5.5 write()

```
bool cl::fs::FileStream::write (
    const void * data,
    std::size_t byteCount )
```

Definition at line 90 of file file_stream.cpp.

The documentation for this class was generated from the following files:

- csv_lib/include/cl/fs/file_stream.hpp
- csv_lib/src/cl/fs/file_stream.cpp

6.11 cs::LogInfo Class Reference

```
#include <log_info.hpp>
```

Public Member Functions

- [LogInfo \(\)](#)
- const [cl::fs::Path & logFilePath \(\)](#) const noexcept
- bool [skipWindow \(\)](#) const noexcept
- bool [deleteTooClose \(\)](#) const noexcept
- bool [deleteLowVariance \(\)](#) const noexcept
- [SegmentationKind segmentationKind \(\)](#) const noexcept
- std::uint64_t [windowSize \(\)](#) const noexcept
- [FilterKind filterKind \(\)](#) const noexcept
- std::uint64_t [sensor \(\)](#) const noexcept
- bool [isInitialized \(\)](#) const noexcept

Static Public Member Functions

- static `cl::Expected< LogInfo > create (cl::fs::Path logFilePath) noexcept`

Static Public Attributes

- static const std::uint64_t `invalidSensor = UINT64_C(0xFFFFFFFFFFFFFF)`

Friends

- bool `operator== (const LogInfo &lhs, const LogInfo &rhs) noexcept`
- bool `operator!= (const LogInfo &lhs, const LogInfo &rhs) noexcept`
- std::ostream & `operator<< (std::ostream &os, const LogInfo &logInfo)`

6.11.1 Detailed Description

Definition at line 14 of file log_info.hpp.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 LogInfo()

```
cs::LogInfo::LogInfo ( )
```

Definition at line 286 of file log_info.cpp.

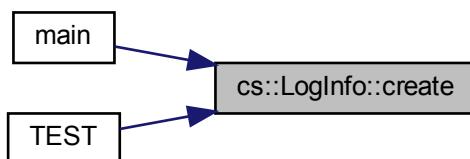
6.11.3 Member Function Documentation

6.11.3.1 create()

```
cl::Expected< LogInfo > cs::LogInfo::create (
    cl::fs::Path logFilePath ) [static], [noexcept]
```

Definition at line 78 of file log_info.cpp.

Here is the caller graph for this function:



6.11.3.2 deleteLowVariance()

```
bool cs::LogInfo::deleteLowVariance ( ) const [noexcept]
```

Definition at line 308 of file log_info.cpp.

6.11.3.3 deleteTooClose()

```
bool cs::LogInfo::deleteTooClose ( ) const [noexcept]
```

Definition at line 306 of file log_info.cpp.

6.11.3.4 filterKind()

```
FilterKind cs::LogInfo::filterKind ( ) const [noexcept]
```

Definition at line 317 of file log_info.cpp.

6.11.3.5 isInitialized()

```
bool cs::LogInfo::isInitialized ( ) const [noexcept]
```

Definition at line 321 of file log_info.cpp.

6.11.3.6 logFilePath()

```
const cl::fs::Path & cs::LogInfo::logFilePath ( ) const [noexcept]
```

Definition at line 299 of file log_info.cpp.

Here is the caller graph for this function:



6.11.3.7 segmentationKind()

```
SegmentationKind cs::LogInfo::segmentationKind () const [noexcept]
```

Definition at line 310 of file log_info.cpp.

6.11.3.8 sensor()

```
std::uint64_t cs::LogInfo::sensor () const [noexcept]
```

Definition at line 319 of file log_info.cpp.

6.11.3.9 skipWindow()

```
bool cs::LogInfo::skipWindow () const [noexcept]
```

Definition at line 304 of file log_info.cpp.

6.11.3.10 windowSize()

```
std::uint64_t cs::LogInfo::windowSize () const [noexcept]
```

Definition at line 315 of file log_info.cpp.

6.11.4 Friends And Related Function Documentation

6.11.4.1 operator"!=

```
bool operator!= (
    const LogInfo & lhs,
    const LogInfo & rhs ) [friend]
```

Definition at line 269 of file log_info.cpp.

6.11.4.2 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const LogInfo & logInfo ) [friend]
```

Definition at line 274 of file log_info.cpp.

6.11.4.3 operator==

```
bool operator== (
    const LogInfo & lhs,
    const LogInfo & rhs ) [friend]
```

Definition at line 247 of file log_info.cpp.

6.11.5 Member Data Documentation

6.11.5.1 invalidSensor

```
const std::uint64_t cs::LogInfo::invalidSensor = UINT64_C(0xFFFFFFFFFFFFFF) [static]
```

Definition at line 16 of file log_info.hpp.

The documentation for this class was generated from the following files:

- compare_segmentation/include/[log_info.hpp](#)
- compare_segmentation/src/[log_info.cpp](#)

6.12 cs::LogLine Class Reference

```
#include <log_line.hpp>
```

Public Member Functions

- std::uint64_t [segmentationPointCount \(\) const noexcept](#)
- const [cl::fs::Path & filePath \(\) const noexcept](#)
- [cl::Expected< std::string > fileName \(\) const](#)
- std::uint64_t [sensor \(\) const noexcept](#)

Static Public Member Functions

- static [cl::Expected< LogLine > parse \(const std::string &line\)](#)

Static Public Attributes

- static const std::uint64_t `invalidSensor` = `UINT64_C(0xFFFFFFFFFFFFFF)`

6.12.1 Detailed Description

Definition at line 11 of file `log_line.hpp`.

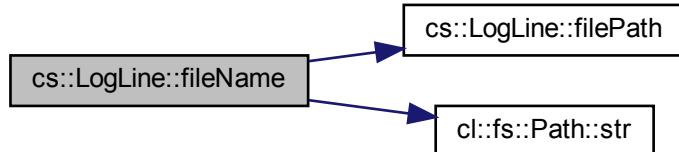
6.12.2 Member Function Documentation

6.12.2.1 `fileName()`

```
cl::Expected< std::string > cs::LogLine::fileName ( ) const
```

Definition at line 120 of file `log_line.cpp`.

Here is the call graph for this function:

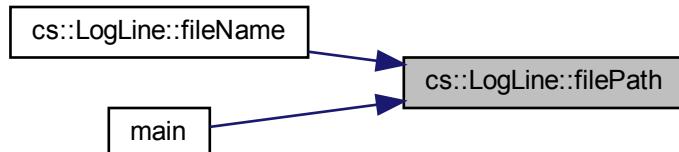


6.12.2.2 `filePath()`

```
const cl::fs::Path & cs::LogLine::filePath ( ) const [noexcept]
```

Definition at line 118 of file `log_line.cpp`.

Here is the caller graph for this function:

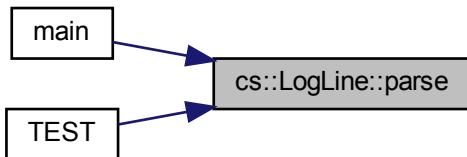


6.12.2.3 parse()

```
cl::Expected< LogLine > cs::LogLine::parse (
    const std::string & line ) [static]
```

Definition at line 27 of file log_line.cpp.

Here is the caller graph for this function:



6.12.2.4 segmentationPointCount()

```
std::uint64_t cs::LogLine::segmentationPointCount( ) const [noexcept]
```

Definition at line 113 of file log_line.cpp.

6.12.2.5 sensor()

```
std::uint64_t cs::LogLine::sensor( ) const [noexcept]
```

Definition at line 154 of file log_line.cpp.

6.12.3 Member Data Documentation

6.12.3.1 invalidSensor

```
const std::uint64_t cs::LogLine::invalidSensor = UINT64_C(0xFFFFFFFFFFFFFF) [static]
```

Definition at line 13 of file log_line.hpp.

The documentation for this class was generated from the following files:

- compare_segmentation/include/[log_line.hpp](#)
- compare_segmentation/src/[log_line.cpp](#)

6.13 cl::fs::Path Class Reference

```
#include <path.hpp>
```

Public Member Functions

- PL_IMPLICIT `Path` (`std::string path`)
- `bool exists () const noexcept`
- `bool isFile () const noexcept`
- `bool isDirectory () const noexcept`
- `const std::string & str () const noexcept`

Friends

- `std::ostream & operator<< (std::ostream &os, const Path &path)`
- `bool operator< (const Path &lhs, const Path &rhs) noexcept`
- `bool operator== (const Path &lhs, const Path &rhs) noexcept`

6.13.1 Detailed Description

Definition at line 10 of file path.hpp.

6.13.2 Constructor & Destructor Documentation

6.13.2.1 Path()

```
cl::fs::Path::Path (
    std::string path )
```

Definition at line 37 of file path.cpp.

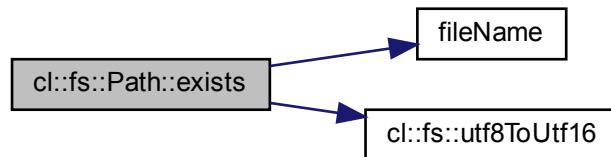
6.13.3 Member Function Documentation

6.13.3.1 exists()

```
bool cl::fs::Path::exists ( ) const [noexcept]
```

Definition at line 44 of file path.cpp.

Here is the call graph for this function:

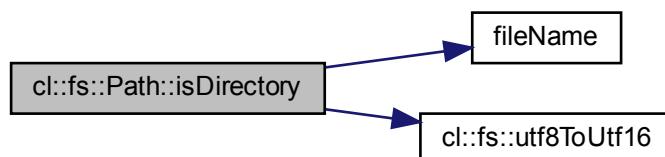


6.13.3.2 isDirectory()

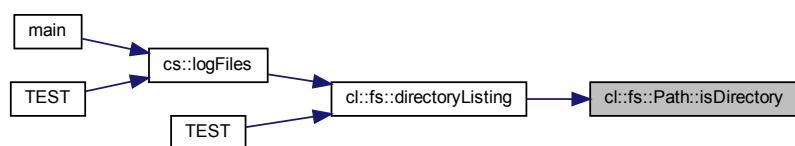
```
bool cl::fs::Path::isDirectory ( ) const [noexcept]
```

Definition at line 100 of file path.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

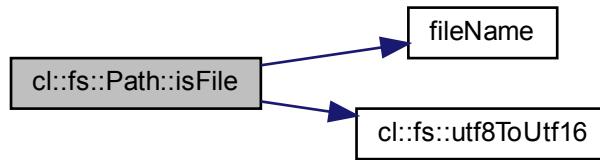


6.13.3.3 `isFile()`

```
bool cl::fs::Path::isFile() const [noexcept]
```

Definition at line 73 of file path.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

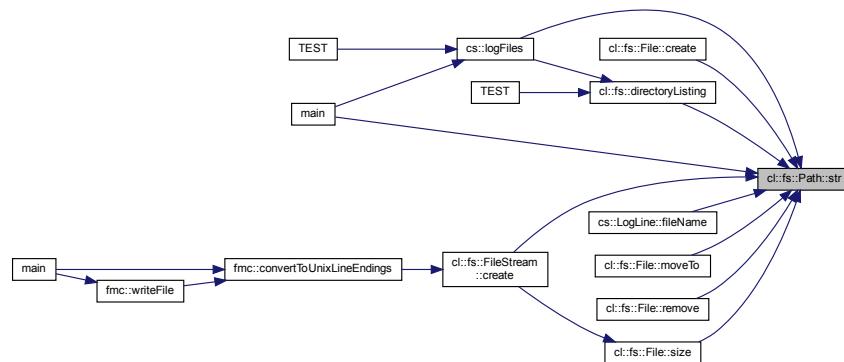


6.13.3.4 `str()`

```
const std::string & cl::fs::Path::str() const [noexcept]
```

Definition at line 123 of file path.cpp.

Here is the caller graph for this function:



6.13.4 Friends And Related Function Documentation

6.13.4.1 operator<

```
bool operator< (
    const Path & lhs,
    const Path & rhs ) [friend]
```

Definition at line 27 of file path.cpp.

6.13.4.2 operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const Path & path ) [friend]
```

Definition at line 22 of file path.cpp.

6.13.4.3 operator==

```
bool operator== (
    const Path & lhs,
    const Path & rhs ) [friend]
```

Definition at line 32 of file path.cpp.

The documentation for this class was generated from the following files:

- csv_lib/include/cl/fs/path.hpp
- csv_lib/src/cl/fs/path.cpp

Chapter 7

File Documentation

7.1 compare_segmentation/CMakeLists.txt File Reference

Functions

- `set (LIB_NAME compare_segmentation_lib) set(LIB_HEADERS include/csv_line.hpp include/data_set_info.hpp include/filter_kind.hpp include/log_files.hpp include/log_info.hpp include/log_line.hpp include/paths.hpp include/segmentation_kind.hpp) set(LIB_SOURCES src/csv_line.cpp src/data_set_info.cpp src/filter_kind.cpp src/log_files.cpp src/log_info.cpp src/log_line.cpp src/segmentation_kind.cpp) add_library($`

7.1.1 Function Documentation

7.1.1.1 set()

```
set (  
    LIB_NAME compare_segmentation_lib )
```

Definition at line 2 of file CMakeLists.txt.

7.2 compare_segmentation/test/CMakeLists.txt File Reference

Functions

- `include (GoogleTest) set(TEST_NAME compare_segmentation_test) set(TEST_SOURCES csv_line_test.cpp data_set_info_test.cpp log_files_test.cpp log_info_test.cpp log_line_test.cpp main.cpp) add_executable($`

7.2.1 Function Documentation

7.2.1.1 include()

```
include (
    GoogleTest    )
```

Definition at line 1 of file CMakeLists.txt.

7.3 counting/CMakeLists.txt File Reference

Functions

- `set (LIB_NAME counting_lib) set(LIB_HEADERS include/above_threshold.hpp include/average_← comparison_value_calculator.hpp include/half_maximum_comparison_value_calculator.hpp include/is_← relevant.hpp include/percentage_of.hpp include/run_above_threshold.hpp) set(LIB_SOURCES src/above← _threshold.cpp src/average_comparison_value_calculator.cpp src/half_maximum_comparison_value← calculator.cpp src/run_above_threshold.cpp) add_library($`

7.3.1 Function Documentation

7.3.1.1 set()

```
set (
    LIB_NAME counting_lib )
```

Definition at line 2 of file CMakeLists.txt.

7.4 counting/test/CMakeLists.txt File Reference

Functions

- `include (GoogleTest) set(TEST_NAME counting_test) set(TEST_SOURCES above_threshold_test.cpp main.cpp percentage_of_test.cpp) add_executable($`

7.4.1 Function Documentation

7.4.1.1 include()

```
include (
    GoogleTest    )
```

Definition at line 1 of file CMakeLists.txt.

7.5 csv_lib/CMakeLists.txt File Reference

Functions

- `set (LIB_NAME csv_lib) set(LIB_HEADERS include/cl/fs/directory_listing.hpp include/cl/fs/file.hpp include/cl/fs/file_stream.hpp include/cl/fs/path.hpp include/cl/fs/sePARATOR.hpp include/cl/fs/windows.hpp include/cl/channel.hpp include/cl/column.hpp include/cl/data_point.hpp include/cl/data_set.hpp include/cl/dos2unix.hpp include/cl/error.hpp include/cl/exception.hpp include/cl/read_csv_file.hpp include/cl/s2n.hpp include/cl/sensor.hpp include/cl/to_string.hpp include/cl/use_unbuffered_io.hpp) set(LIB_SOURCES src/cl/fs/directory_listing.cpp src/cl/fs/file.cpp src/cl/fs/file_stream.cpp src/cl/fs/path.cpp src/cl/fs/windows.hpp src/cl/channel.cpp src/cl/data_point.cpp src/cl/data_set.cpp src/cl/dos2unix.cpp src/cl/error.cpp src/cl/exception.cpp src/cl/read_csv_file.cpp src/cl/sensor.cpp src/cl/use_unbuffered_io.cpp) add_library($`

7.5.1 Function Documentation

7.5.1.1 set()

```
set (  
    LIB_NAME csv_lib )
```

Definition at line 2 of file CMakeLists.txt.

7.6 csv_lib/test/CMakeLists.txt File Reference

Functions

- `include (GoogleTest) set(TEST_NAME csv_lib_test) set(TEST_SOURCES channel_test.cpp column_test.cpp data_point_test.cpp directory_listing_test.cpp error_test.cpp exception_test.cpp main.cpp sensor_test.cpp to_string_test.cpp read_csv_file_test.cpp data_set_test.cpp s2n_test.cpp) add_executable($`

7.6.1 Function Documentation

7.6.1.1 include()

```
include (  
    GoogleTest )
```

Definition at line 1 of file CMakeLists.txt.

7.7 fix_csv/CMakeLists.txt File Reference

Functions

- `set (LIB_NAME fix_mogasens_csv_lib) set(LIB_HEADERS include/adjust_hardware_timestamp.hpp include/convert_to_unix_line_endings.hpp include/create_backup_file.hpp include/delete_non_bosch_sensors.hpp include/delete_out_of_bounds_values.hpp include/remove_zeros_from_field.hpp include/restore_from_backup.hpp include/write_file.hpp) set(LIB_SOURCES src/adjust_hardware_timestamp.cpp src/convert_to_unix_line_endings.cpp src/create_backup_file.cpp src/delete_non_bosch_sensors.cpp src/delete_out_of_bounds_values.cpp src/remove_zeros_from_field.cpp src/restore_from_backup.cpp src/write_file.cpp) add_library($`

7.7.1 Function Documentation

7.7.1.1 `set()`

```
set (
    LIB_NAME fix_mogasens_csv_lib )
```

Definition at line 2 of file CMakeLists.txt.

7.8 fix_csv/test/CMakeLists.txt File Reference

Functions

- `include (GoogleTest) set(TEST_NAME fmc_test) set(TEST_SOURCES main.cpp remove_zeros_from_field_test.cpp adjust_hardware_timestamp_test.cpp) add_executable($`

7.8.1 Function Documentation

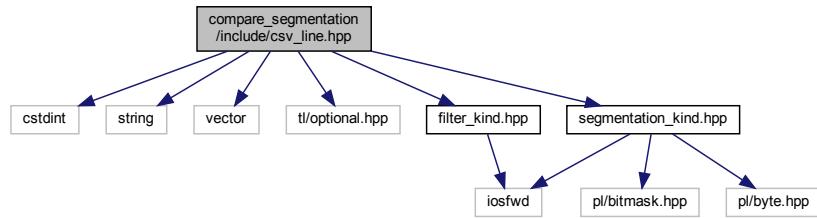
7.8.1.1 `include()`

```
include (
    GoogleTest )
```

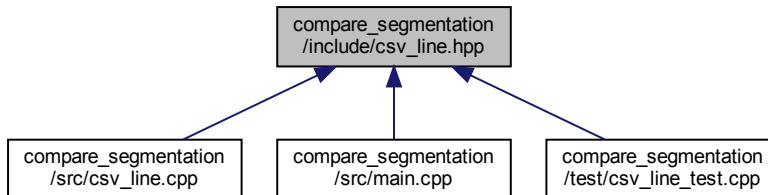
Definition at line 1 of file CMakeLists.txt.

7.9 compare_segmentation/include/csv_line.hpp File Reference

```
#include <cstdint>
#include <string>
#include <vector>
#include <t1/optional.hpp>
#include "filter_kind.hpp"
#include "segmentation_kind.hpp"
Include dependency graph for csv_line.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [cs::CsvLineBuilder](#)

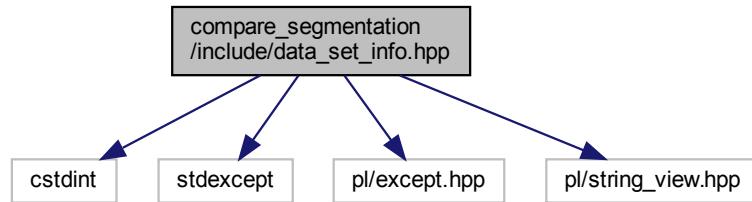
Namespaces

- [cs](#)

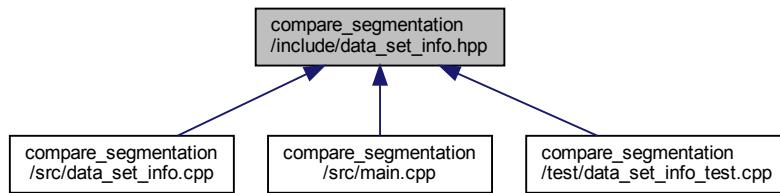
7.10 compare_segmentation/include/data_set_info.hpp File Reference

```
#include <cstdint>
#include <stdexcept>
#include <pl/except.hpp>
```

```
#include <pl/string_view.hpp>
Include dependency graph for data_set_info.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct `cs::data_set_info< Tag >`

Namespaces

- `cs`

Macros

- `#define CS_SPECIALIZE_DATA_SET_INFO(tag, string, repetitionCount)`

Functions

- `cs::PL_DEFINE_EXCEPTION_TYPE` (`NoSuchDataSetException`, `std::logic_error`)
- `cs::CS_SPECIALIZE_DATA_SET_INFO` (`Felix1`, "11.17.39", 24)
- `cs::CS_SPECIALIZE_DATA_SET_INFO` (`Felix2`, "12.50.00", 20)
- `cs::CS_SPECIALIZE_DATA_SET_INFO` (`Felix3`, "13.00.09", 15)
- `cs::CS_SPECIALIZE_DATA_SET_INFO` (`Marcelle1`, "14.59.59", 10)
- `cs::CS_SPECIALIZE_DATA_SET_INFO` (`Marcelle2`, "15.13.22", 16)

- `cs::CS_SPECIALIZE_DATA_SET_INFO` (Marcelle3, "15.31.36", 18)
- `cs::CS_SPECIALIZE_DATA_SET_INFO` (Mike1, "14.07.33", 26)
- `cs::CS_SPECIALIZE_DATA_SET_INFO` (Mike2, "14.14.32", 22)
- `cs::CS_SPECIALIZE_DATA_SET_INFO` (Mike3, "14.20.28", 18)
- `cs::CS_SPECIALIZE_DATA_SET_INFO` (Andre1, "Andre_liegestuetzen1", 27)
- `cs::CS_SPECIALIZE_DATA_SET_INFO` (Andre2, "Andre_liegestuetzen2", 20)
- `cs::CS_SPECIALIZE_DATA_SET_INFO` (Andre3, "Andre_liegestuetzen3", 17)
- `cs::CS_SPECIALIZE_DATA_SET_INFO` (AndreSquats1, "Andre_Squats", 30)
- `cs::CS_SPECIALIZE_DATA_SET_INFO` (AndreSquats2, "Andre_Squats2", 49)
- `cs::CS_SPECIALIZE_DATA_SET_INFO` (Jan1, "Jan_liegestuetzen1", 25)
- `cs::CS_SPECIALIZE_DATA_SET_INFO` (Jan2, "Jan_liegestuetzen2", 19)
- `cs::CS_SPECIALIZE_DATA_SET_INFO` (Jan3, "Jan_liegestuetzen3", 13)
- `cs::CS_SPECIALIZE_DATA_SET_INFO` (Lucas1, "Lukas_liegestuetzen1", 24)
- `cs::CS_SPECIALIZE_DATA_SET_INFO` (Lucas2, "Lukas_liegestuetzen2", 19)
- `cs::CS_SPECIALIZE_DATA_SET_INFO` (Lucas3, "Lukas_liegestuetzen3", 11)
- `std::uint64_t cs::repetitionCount` (`pl::string_view dataSet`)

7.10.1 Macro Definition Documentation

7.10.1.1 CS_SPECIALIZE_DATA_SET_INFO

```
#define CS_SPECIALIZE_DATA_SET_INFO( \
    tag, \
    string, \
    repetitionCount )
```

Value:

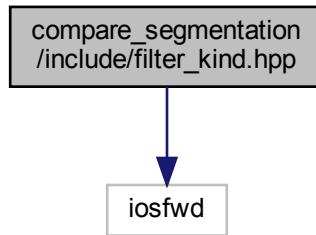
```
struct tag { \
}; \
constexpr bool contains##tag(pl::string_view other) \
{ \
    return other.contains(string); \
} \
template<> \
struct data_set_info<tag> { \
    static constexpr pl::string_view text      = string; \
    static constexpr std::uint64_t repetitions = UINT64_C(repetitionCount); \
}
```

Definition at line 16 of file `data_set_info.hpp`.

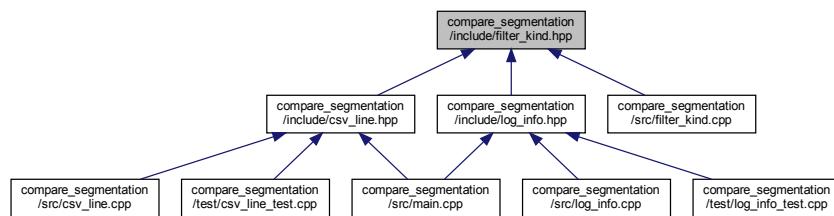
7.11 compare_segmentation/include/filter_kind.hpp File Reference

```
#include <iostream>
```

Include dependency graph for filter_kind.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- `cs`

Enumerations

- enum `cs::FilterKind { cs::FilterKind::Butterworth, cs::FilterKind::MovingAverage }`

Functions

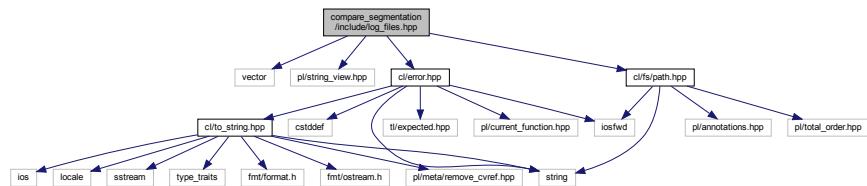
- `std::ostream & cs::operator<< (std::ostream &os, FilterKind filterKind)`

7.12 compare_segmentation/include/log_files.hpp File Reference

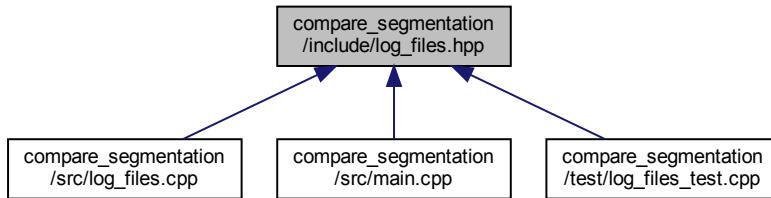
```
#include <vector>
#include <pl/string_view.hpp>
#include <c1/error.hpp>
```

```
#include <cl/fs/path.hpp>
```

Include dependency graph for log_files.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

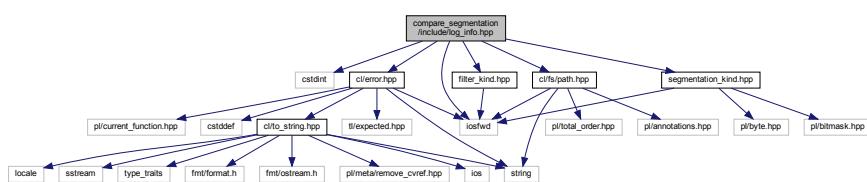
- `cs`

Functions

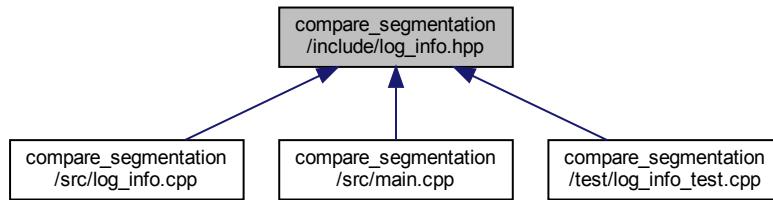
- `cl::Expected< std::vector< cl::fs::Path > > cs::logFiles (pl::string_view directoryPath)`

7.13 compare_segmentation/include/log_info.hpp File Reference

```
#include <cstdint>
#include <iiosfwd>
#include <cl/error.hpp>
#include <cl/fs/path.hpp>
#include "filter_kind.hpp"
#include "segmentation_kind.hpp"
Include dependency graph for log_info.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

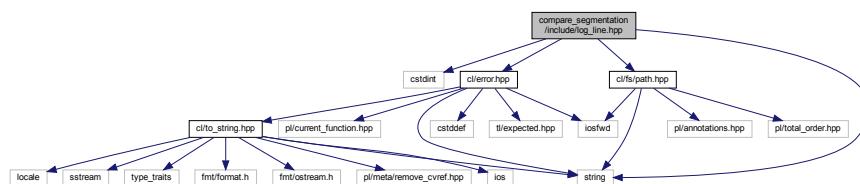
- class [cs::LogInfo](#)

Namespaces

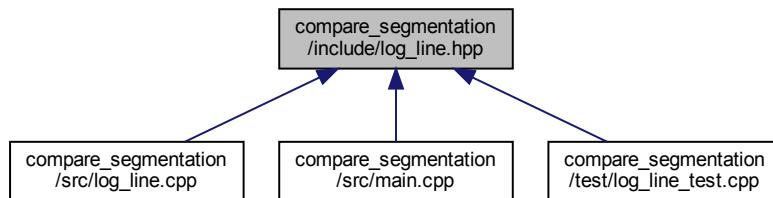
- [cs](#)

7.14 compare_segmentation/include/log_line.hpp File Reference

```
#include <cstdint>
#include <string>
#include "cl/error.hpp"
#include "cl/fs/path.hpp"
Include dependency graph for log_line.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

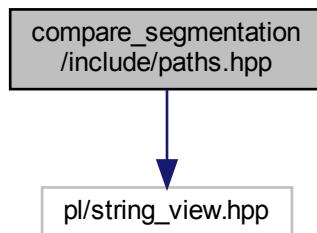
- class [cs::LogLine](#)

Namespaces

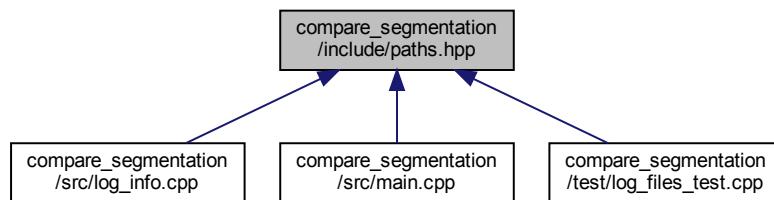
- [cs](#)

7.15 compare_segmentation/include/paths.hpp File Reference

```
#include <pl/string_view.hpp>
Include dependency graph for paths.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

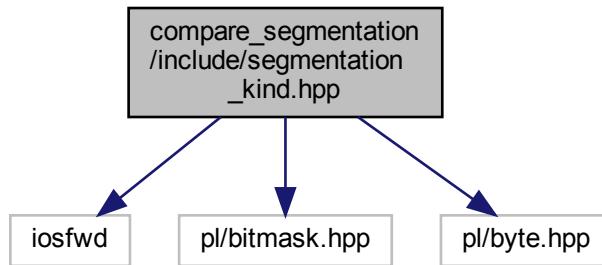
- [cs](#)

Variables

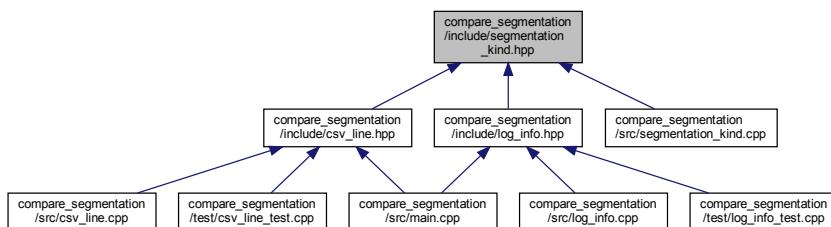
- `constexpr pl::string_view cs::logPath {"segmentation_comparison/logs"}`
- `constexpr pl::string_view cs::oldLogPath {"segmentation_comparison/logs/old"}`

7.16 compare_segmentation/include/segmentation_kind.hpp File Reference

```
#include <iostream>
#include <pl/bitmask.hpp>
#include <pl/byte.hpp>
Include dependency graph for segmentation_kind.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- `cs`

Enumerations

- enum `cs::SegmentationKind : pl::byte` { `cs::SegmentationKind::Minima` = 0b0000'0001, `cs::SegmentationKind::Maxima` = 0b0000'0010, `cs::SegmentationKind::Both` = Minima | Maxima }

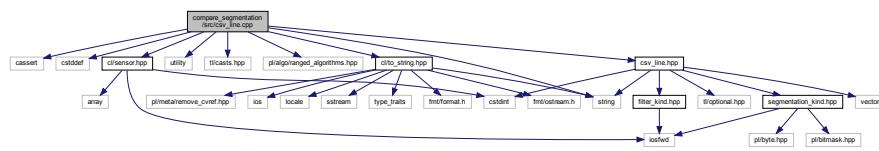
Functions

- `std::ostream & cs::operator<< (std::ostream &os, SegmentationKind segmentationKind)`

7.17 compare_segmentation/src/csv_line.cpp File Reference

```
#include <cassert>
#include <cstddef>
#include <string>
#include <utility>
#include <tl/casts.hpp>
#include <pl/algo/ranged_algorithms.hpp>
#include "cl/sensor.hpp"
#include "cl/to_string.hpp"
#include "csv_line.hpp"

Include dependency graph for csv_line.cpp:
```



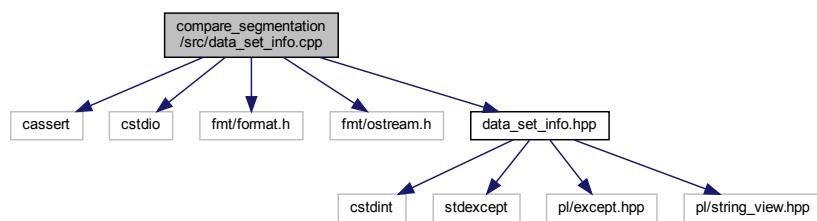
Namespaces

- CS

7.18 compare_segmentation/src/data_set_info.cpp File Reference

```
#include <cassert>
#include <cstdio>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include "data_set_info.hpp"

Include dependency graph for data_set_info.cpp:
```



Namespaces

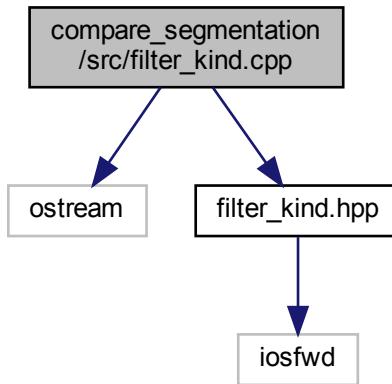
- CS

Functions

- std::uint64_t [cs::repetitionCount](#) (pl::string_view dataSet)

7.19 compare_segmentation/src/filter_kind.cpp File Reference

```
#include <iostream>
#include "filter_kind.hpp"
Include dependency graph for filter.cpp:
```



Namespaces

- [cs](#)

Functions

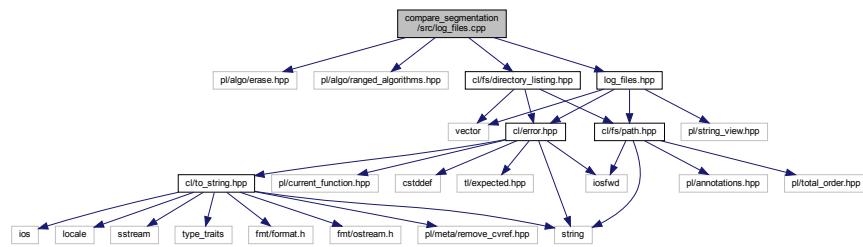
- std::ostream & [cs::operator<<](#) (std::ostream &os, FilterKind filterKind)

7.20 compare_segmentation/src/log_files.cpp File Reference

```
#include <pl/algo/erase.hpp>
#include <pl/algo/ranged_algorithms.hpp>
#include <cl/fs/directory_listing.hpp>
```

```
#include "log_files.hpp"
```

Include dependency graph for log_files.cpp:



Namespaces

- [CS](#)

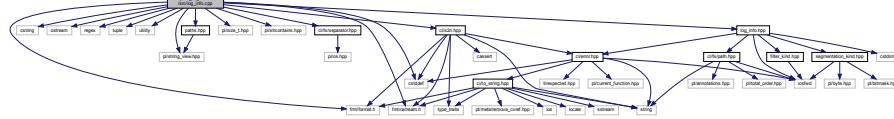
Functions

- `cl::Expected< std::vector< cl::fs::Path > > cs::logFiles (pl::string_view directoryPath)`

7.21 compare_segmentation/src/log_info.cpp File Reference

```
#include <cstddef>
#include <cstring>
#include <iostream>
#include <regex>
#include <tuple>
#include <utility>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include <pl/size_t.hpp>
#include <pl/strcontains.hpp>
#include <pl/string_view.hpp>
#include "cl/fs/separators.hpp"
#include "cl/s2n.hpp"
#include "log_info.hpp"
#include "paths.hpp"
```

Include dependency graph for log_info.cpp:



Namespaces

- [CS](#)

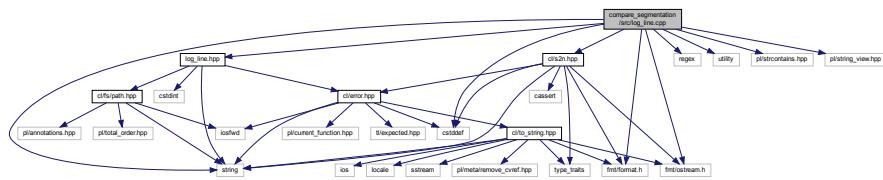
Functions

- bool `cs::operator==` (const LogInfo &lhs, const LogInfo &rhs) noexcept
 - bool `cs::operator!=` (const LogInfo &lhs, const LogInfo &rhs) noexcept
 - std::ostream & `cs::operator<<` (std::ostream &os, const LogInfo &logInfo)

7.22 compare_segmentation/src/log_line.cpp File Reference

```
#include <cstddef>
#include <regex>
#include <string>
#include <utility>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include <pl/strcontains.hpp>
#include <pl/string_view.hpp>
#include "cl/s2n.hpp"
#include "log_line.hpp"
```

Include dependency graph for log_line.cpp:



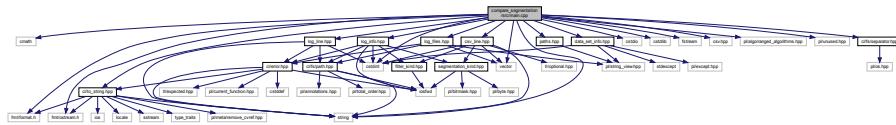
Namespaces

- CS

7.23 compare_segmentation/src/main.cpp File Reference

```
#include <cmath>
#include <cstdint>
#include <cstdio>
#include <cstdlib>
#include <fstream>
#include <string>
#include <vector>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include <csv.hpp>
#include <pl/algo/ranged_algorithms.hpp>
#include <pl/unused.hpp>
#include "cl/fs/separator.hpp"
#include "cl/to_string.hpp"
#include "csv_line.hpp"
#include "data_set_info.hpp"
#include "log_files.hpp"
```

```
#include "log_info.hpp"
#include "log_line.hpp"
#include "paths.hpp"
Include dependency graph for main.cpp:
```



Functions

- int main (int argc, char *argv[])

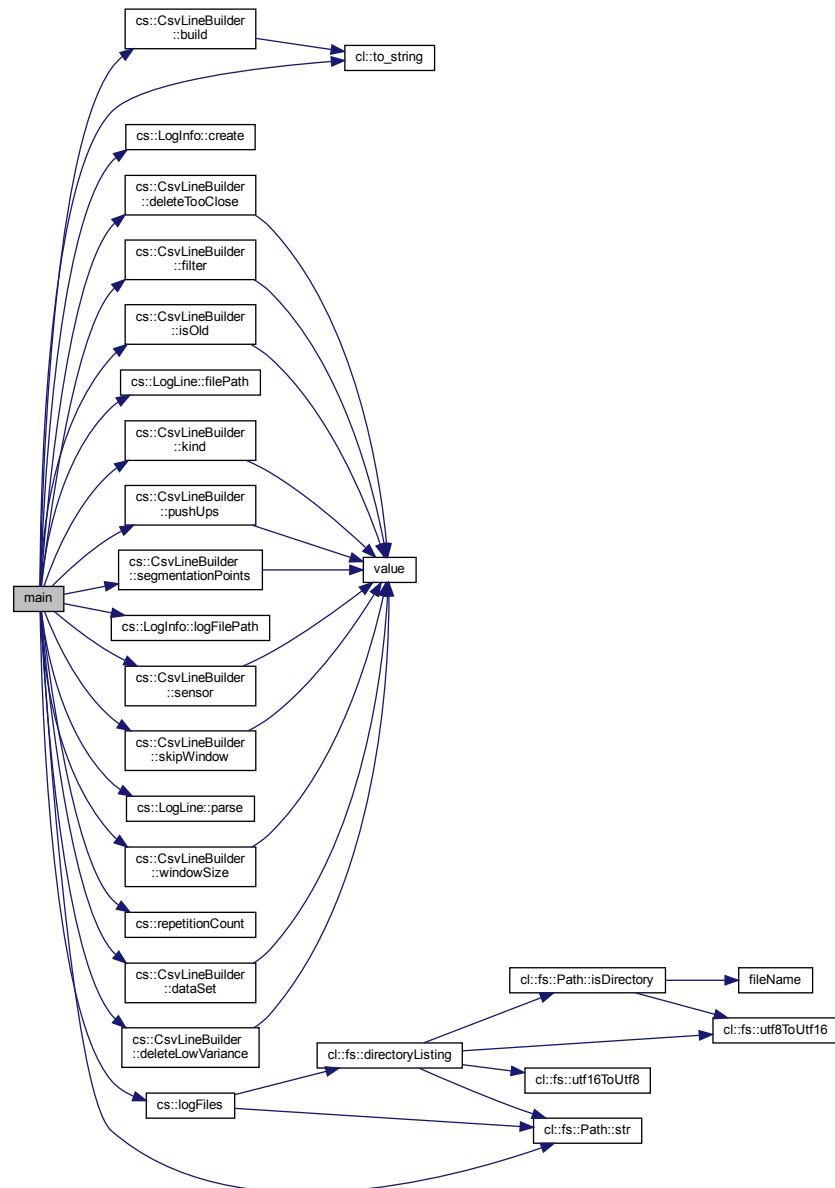
7.23.1 Function Documentation

7.23.1.1 main()

```
int main ( int argc,  
           char * argv[ ] )
```

Definition at line 28 of file main.cpp.

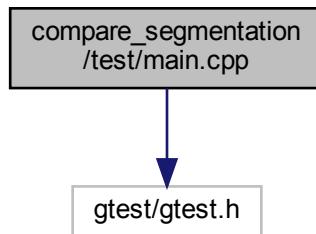
Here is the call graph for this function:



7.24 compare_segmentation/test/main.cpp File Reference

```
#include "gtest/gtest.h"
```

Include dependency graph for main.cpp:



Functions

- int `main` (int argc, char *argv[])

7.24.1 Function Documentation

7.24.1.1 `main()`

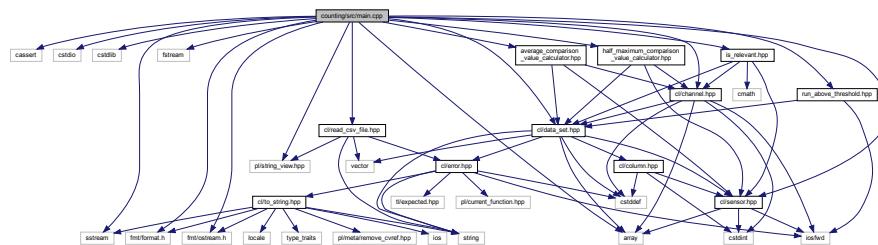
```
int main (
    int argc,
    char * argv[] )
```

Definition at line 3 of file main.cpp.

7.25 counting/src/main.cpp File Reference

```
#include <cassert>
#include <cstdio>
#include <cstdlib>
#include <array>
#include <fstream>
#include <sstream>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include <pl/string_view.hpp>
#include "cl/channel.hpp"
#include "cl/data_set.hpp"
#include "cl/read_csv_file.hpp"
#include "cl/sensor.hpp"
#include "average_comparison_value_calculator.hpp"
#include "half_maximum_comparison_value_calculator.hpp"
```

```
#include "is_relevant.hpp"
#include "run_above_threshold.hpp"
Include dependency graph for main.cpp:
```



Functions

- int `main` (int argc, char *argv[])

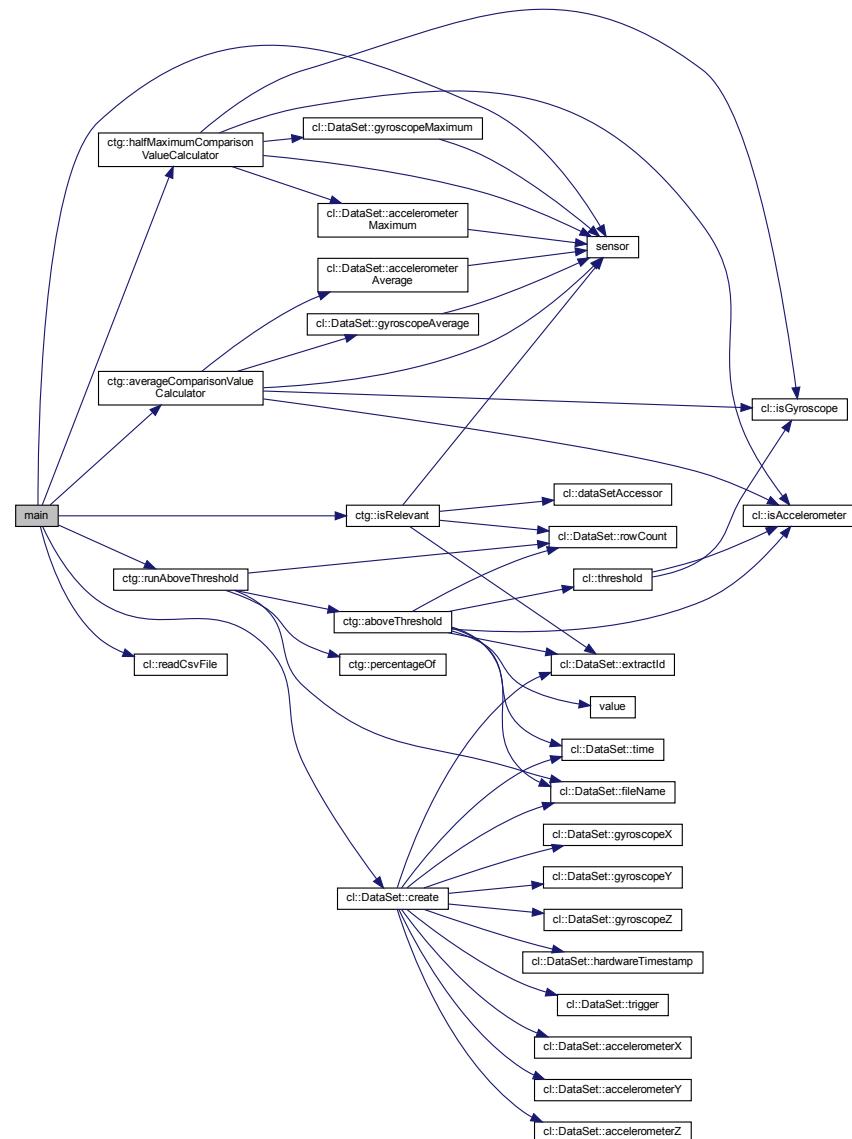
7.25.1 Function Documentation

7.25.1.1 `main()`

```
int main (
    int argc,
    char * argv[ ] )
```

Definition at line 24 of file `main.cpp`.

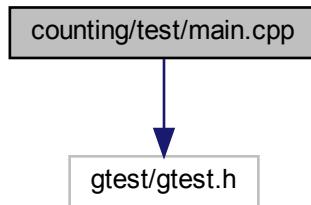
Here is the call graph for this function:



7.26 counting/test/main.cpp File Reference

```
#include "gtest/gtest.h"
```

Include dependency graph for main.cpp:



Functions

- int `main` (int argc, char *argv[])

7.26.1 Function Documentation

7.26.1.1 `main()`

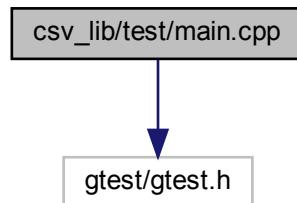
```
int main (
    int argc,
    char * argv[] )
```

Definition at line 3 of file main.cpp.

7.27 csv_lib/test/main.cpp File Reference

```
#include "gtest/gtest.h"
```

Include dependency graph for main.cpp:



Functions

- int main (int argc, char *argv[])

7.27.1 Function Documentation

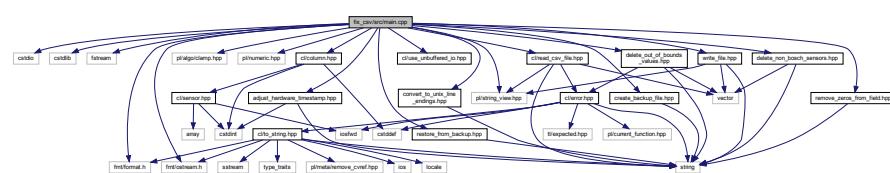
7.27.1.1 main()

```
int main ( int argc,  
           char * argv[ ] )
```

Definition at line 3 of file main.cpp.

7.28 fix_csv/src/main.cpp File Reference

```
#include <cstdio>
#include <cstdlib>
#include <fstream>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include <pl/algo/clamp.hpp>
#include <pl/numeric.hpp>
#include <pl/string_view.hpp>
#include "cl/column.hpp"
#include "cl/read_csv_file.hpp"
#include "cl/use_unbuffered_io.hpp"
#include "adjust_hardware_timestamp.hpp"
#include "convert_to_unix_line_endings.hpp"
#include "create_backup_file.hpp"
#include "delete_non_bosch_sensors.hpp"
#include "delete_out_of_bounds_values.hpp"
#include "remove_zeros_from_field.hpp"
#include "restore_from_backup.hpp"
#include "write_file.hpp"
Include dependency graph for main.cpp:
```



Functions

- int main (int argc, char *argv[])

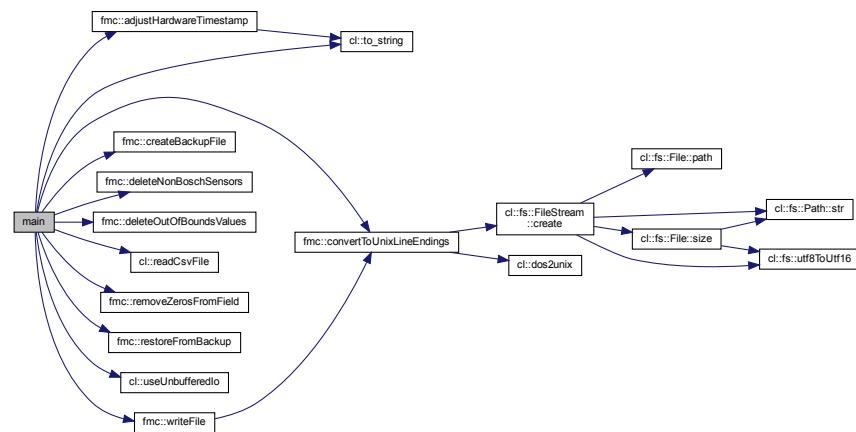
7.28.1 Function Documentation

7.28.1.1 main()

```
int main (
    int argc,
    char * argv[] )
```

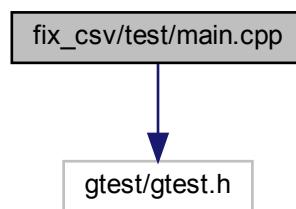
Definition at line 26 of file main.cpp.

Here is the call graph for this function:



7.29 fix_csv/test/main.cpp File Reference

```
#include "gtest/gtest.h"
Include dependency graph for main.cpp:
```



Functions

- int `main` (int argc, char *argv[])

7.29.1 Function Documentation

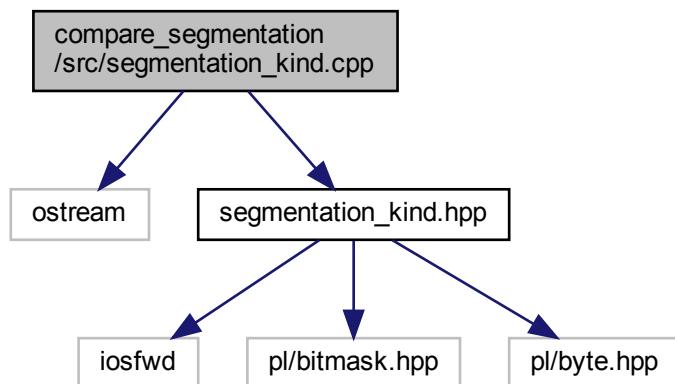
7.29.1.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Definition at line 3 of file main.cpp.

7.30 compare_segmentation/src/segmentation_kind.cpp File Reference

```
#include <iostream>
#include "segmentation_kind.hpp"
Include dependency graph for segmentation_kind.cpp:
```



Namespaces

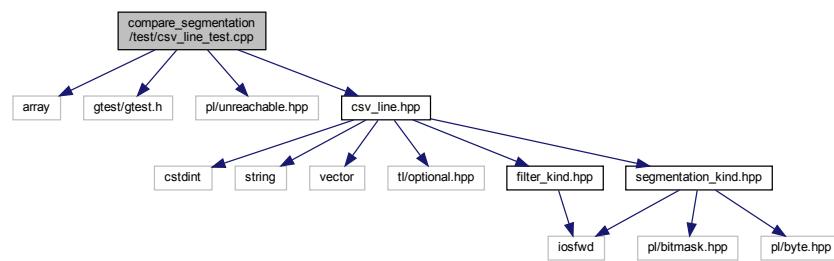
- `cs`

Functions

- `std::ostream & cs::operator<< (std::ostream &os, SegmentationKind segmentationKind)`

7.31 compare_segmentation/test/csv_line_test.cpp File Reference

```
#include <array>
#include "gtest/gtest.h"
#include <pl/unreachable.hpp>
#include "csv_line.hpp"
Include dependency graph for csv_line_test.cpp:
```



Functions

- [TEST](#) (CsvLine, shouldWork)

7.31.1 Function Documentation

7.31.1.1 TEST()

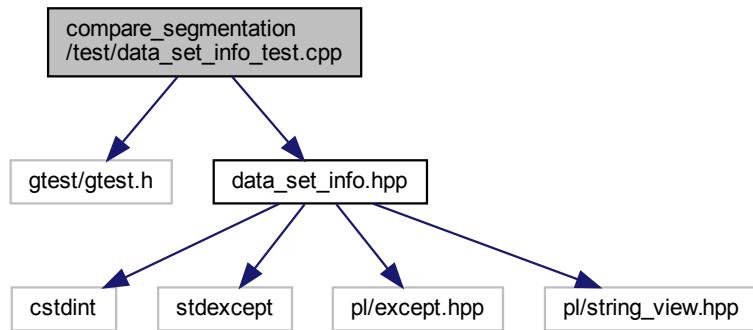
```
TEST (
    CsvLine ,
    shouldWork )
```

Definition at line 30 of file csv_line_test.cpp.

7.32 compare_segmentation/test/data_set_info_test.cpp File Reference

```
#include "gtest/gtest.h"
#include "data_set_info.hpp"
```

Include dependency graph for data_set_info_test.cpp:



Functions

- [TEST](#) (dataSetInfo, repetitionCount)

7.32.1 Function Documentation

7.32.1.1 TEST()

```
TEST (
    dataSetInfo ,
    repetitionCount )
```

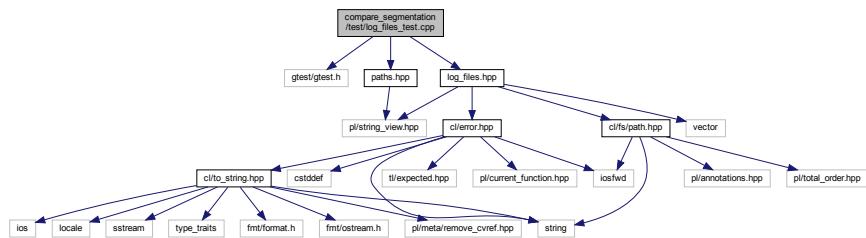
Definition at line 5 of file `data_set_info_test.cpp`.

Here is the call graph for this function:



7.33 compare_segmentation/test/log_files_test.cpp File Reference

```
#include "gtest/gtest.h"
#include <log_files.hpp>
#include <paths.hpp>
Include dependency graph for log_files_test.cpp:
```



Functions

- **TEST** (logFiles, shouldFindLogFiles)
 - **TEST** (logFiles, shouldFindOldLogFile)
 - **TEST** (logFiles, shouldNotFindGarbage)

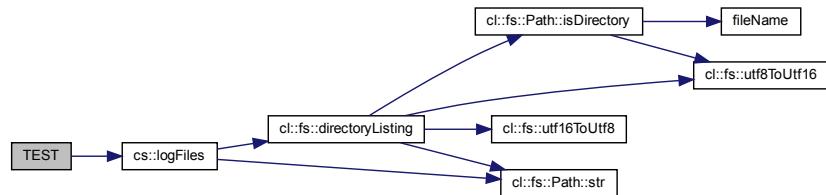
7.33.1 Function Documentation

7.33.1.1 TEST() [1/3]

```
TEST ( logFiles ,  
       shouldFindLogFiles )
```

Definition at line 6 of file log_files_test.cpp.

Here is the call graph for this function:

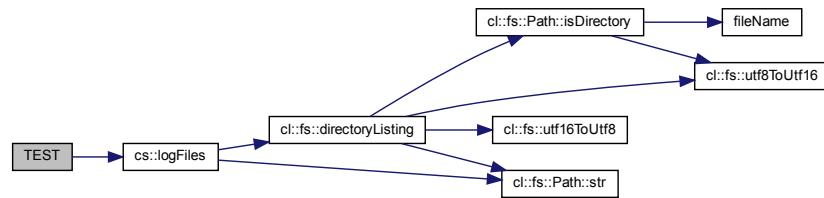


7.33.1.2 TEST() [2/3]

```
TEST (
    logFiles ,
    shouldFindOldLogFiles )
```

Definition at line 23 of file log_files_test.cpp.

Here is the call graph for this function:

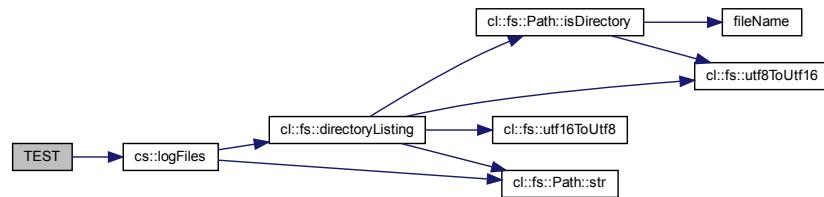


7.33.1.3 TEST() [3/3]

```
TEST (
    logFiles ,
    shouldNotFindGarbage )
```

Definition at line 40 of file log_files_test.cpp.

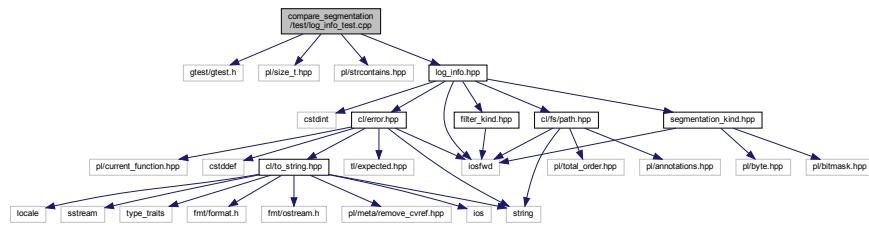
Here is the call graph for this function:



7.34 compare_segmentation/test/log_info_test.cpp File Reference

```
#include "gtest/gtest.h"
#include <pl/size_t.hpp>
#include <pl/strcontains.hpp>
```

```
#include "log_info.hpp"
Include dependency graph for log_info_test.cpp:
```



Functions

- `TEST` (`LogInfo`, `shouldWork`)
- `TEST` (`LogInfo`, `shouldWork2`)
- `TEST` (`LogInfo`, `shouldWork3`)
- `TEST` (`LogInfo`, `shouldWork4`)
- `TEST` (`LogInfo`, `shouldWork5`)
- `TEST` (`LogInfo`, `shouldWork6`)
- `TEST` (`LogInfo`, `shouldWork7`)
- `TEST` (`LogInfo`, `shouldWork8`)
- `TEST` (`LogInfo`, `shouldWork9`)
- `TEST` (`LogInfo`, `shouldWorkWithPathOldPath`)
- `TEST` (`LogInfo`, `shouldWorkWithPathOldPath2`)
- `TEST` (`LogInfo`, `shouldResultInErrorIfLogFilepathIsTooShort`)
- `TEST` (`LogInfo`, `shouldFailIfSkipWindowIsInvalid`)
- `TEST` (`LogInfo`, `shouldFailIfDeleteTooClosesIsInvalid`)
- `TEST` (`LogInfo`, `shouldFailIfDeleteTooLowVarianceIsInvalid`)
- `TEST` (`LogInfo`, `shouldFailIfSegmentationKindIsInvalid`)
- `TEST` (`LogInfo`, `shouldFailIfWindowSizeIsInvalid`)
- `TEST` (`LogInfo`, `shouldFailIfFilterIsInvalid`)
- `TEST` (`LogInfo`, `shouldCreateUninitializedObjectWhenDefaultConstructorIsCalled`)

7.34.1 Function Documentation

7.34.1.1 TEST() [1/19]

```
TEST (
    LogInfo ,
    shouldCreateUninitializedObjectWhenDefaultConstructorIsCalled )
```

Definition at line 388 of file `log_info_test.cpp`.

7.34.1.2 TEST() [2/19]

```
TEST (
    LogInfo ,
    shouldFailIfDeleteTooCloseIsInvalid )
```

Definition at line 341 of file log_info_test.cpp.

Here is the call graph for this function:



7.34.1.3 TEST() [3/19]

```
TEST (
    LogInfo ,
    shouldFailIfDeleteTooLowVarianceIsInvalid )
```

Definition at line 350 of file log_info_test.cpp.

Here is the call graph for this function:



7.34.1.4 TEST() [4/19]

```
TEST (
    LogInfo ,
    shouldFailIfFilterIsInvalid )
```

Definition at line 379 of file log_info_test.cpp.

Here is the call graph for this function:



7.34.1.5 TEST() [5/19]

```
TEST (
    LogInfo ,
    shouldFailIfSegmentationKindIsInvalid )
```

Definition at line 359 of file log_info_test.cpp.

Here is the call graph for this function:

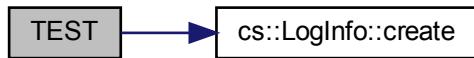


7.34.1.6 TEST() [6/19]

```
TEST (
    LogInfo ,
    shouldFailIfSkipWindowIsValid )
```

Definition at line 332 of file log_info_test.cpp.

Here is the call graph for this function:

**7.34.1.7 TEST() [7/19]**

```
TEST (
    LogInfo ,
    shouldFailIfWindowSizeIsValid )
```

Definition at line 368 of file log_info_test.cpp.

Here is the call graph for this function:



7.34.1.8 TEST() [8/19]

```
TEST (
    LogInfo ,
    shouldResultInErrorIfLogFilePathIsTooShort )
```

Definition at line 325 of file log_info_test.cpp.

Here is the call graph for this function:



7.34.1.9 TEST() [9/19]

```
TEST (
    LogInfo ,
    shouldWork )
```

Definition at line 8 of file log_info_test.cpp.

Here is the call graph for this function:



7.34.1.10 TEST() [10/19]

```
TEST (
    LogInfo ,
    shouldWork2 )
```

Definition at line 37 of file log_info_test.cpp.

Here is the call graph for this function:

**7.34.1.11 TEST() [11/19]**

```
TEST (
    LogInfo ,
    shouldWork3 )
```

Definition at line 66 of file log_info_test.cpp.

Here is the call graph for this function:



7.34.1.12 TEST() [12/19]

```
TEST (
    LogInfo ,
    shouldWork4 )
```

Definition at line 95 of file log_info_test.cpp.

Here is the call graph for this function:

**7.34.1.13 TEST() [13/19]**

```
TEST (
    LogInfo ,
    shouldWork5 )
```

Definition at line 124 of file log_info_test.cpp.

Here is the call graph for this function:



7.34.1.14 TEST() [14/19]

```
TEST (
    LogInfo ,
    shouldWork6 )
```

Definition at line 153 of file log_info_test.cpp.

Here is the call graph for this function:

**7.34.1.15 TEST() [15/19]**

```
TEST (
    LogInfo ,
    shouldWork7 )
```

Definition at line 182 of file log_info_test.cpp.

Here is the call graph for this function:



7.34.1.16 TEST() [16/19]

```
TEST (
    LogInfo ,
    shouldWork8 )
```

Definition at line 211 of file log_info_test.cpp.

Here is the call graph for this function:

**7.34.1.17 TEST() [17/19]**

```
TEST (
    LogInfo ,
    shouldWork9 )
```

Definition at line 240 of file log_info_test.cpp.

Here is the call graph for this function:



7.34.1.18 TEST() [18/19]

```
TEST (
    LogInfo ,
    shouldWorkWithOldPath )
```

Definition at line 269 of file log_info_test.cpp.

Here is the call graph for this function:



7.34.1.19 TEST() [19/19]

```
TEST (
    LogInfo ,
    shouldWorkWithOldPath2 )
```

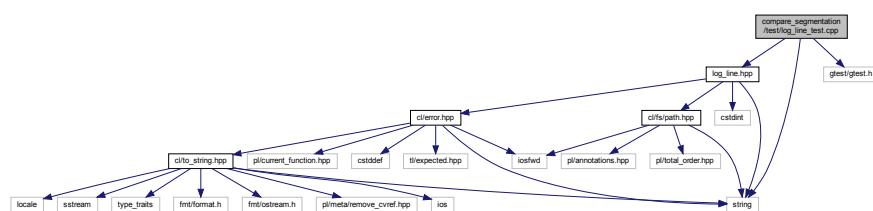
Definition at line 297 of file log_info_test.cpp.

Here is the call graph for this function:



7.35 compare_segmentation/test/log_line_test.cpp File Reference

```
#include <string>
#include "gtest/gtest.h"
#include "log_line.hpp"
Include dependency graph for log_line_test.cpp:
```



Functions

- [TEST](#) (LogLine, shouldWorkWithPreprocessedLine)
- [TEST](#) (LogLine, shouldWorkWithOldLine)
- [TEST](#) (LogLine, shouldNotMatchGarbage)
- [TEST](#) (LogLine, shouldNotParseGarbageSensor)

7.35.1 Function Documentation

7.35.1.1 TEST() [1/4]

```
TEST (
    LogLine ,
    shouldNotMatchGarbage )
```

Definition at line 41 of file log_line_test.cpp.

Here is the call graph for this function:



7.35.1.2 TEST() [2/4]

```
TEST (
    LogLine ,
    shouldNotParseGarbageSensor )
```

Definition at line 48 of file log_line_test.cpp.

Here is the call graph for this function:



7.35.1.3 TEST() [3/4]

```
TEST (
    LogLine ,
    shouldWorkWithOldLine )
```

Definition at line 25 of file log_line_test.cpp.

Here is the call graph for this function:



7.35.1.4 TEST() [4/4]

```
TEST (
    LogLine ,
    shouldWorkWithPreprocessedLine )
```

Definition at line 9 of file log_line_test.cpp.

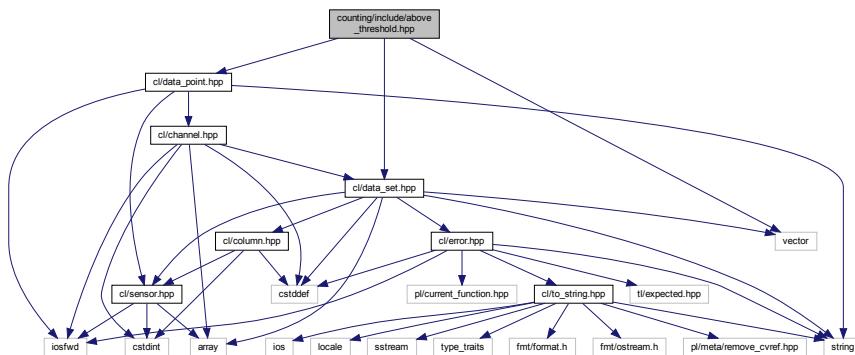
Here is the call graph for this function:



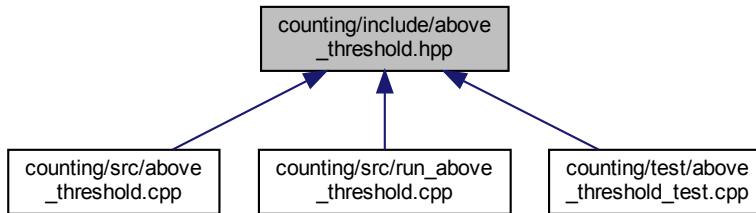
7.36 counting/include/above_threshold.hpp File Reference

```
#include <vector>
#include "cl/data_point.hpp"
```

```
#include "cl/data_set.hpp"
Include dependency graph for above_threshold.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- `ctg`

Functions

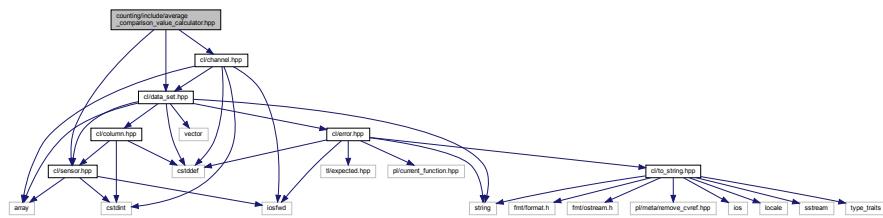
- `std::vector< cl::DataPoint > ctg::aboveThreshold (const cl::DataSet &dataSet, long double accelerometerThreshold, long double gyroscopeThreshold)`

7.37 counting/include/average_comparison_value_calculator.hpp File Reference

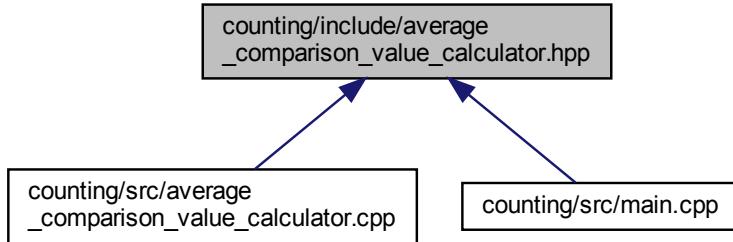
```
#include "cl/channel.hpp"
#include "cl/data_set.hpp"
```

```
#include "cl/sensor.hpp"
```

Include dependency graph for average_comparison_value_calculator.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- `ctg`

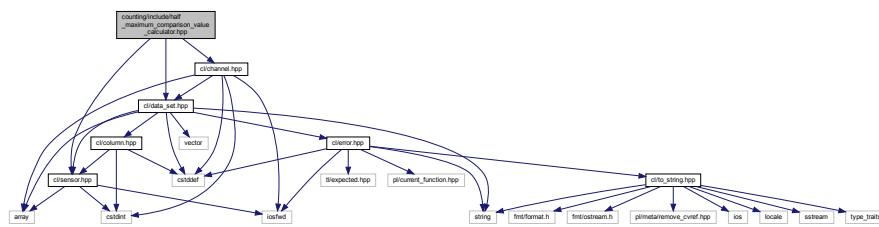
Functions

- long double `ctg::averageComparisonValueCalculator (cl::Sensor sensor, cl::Channel channel, const cl::DataSet &dataSet)`

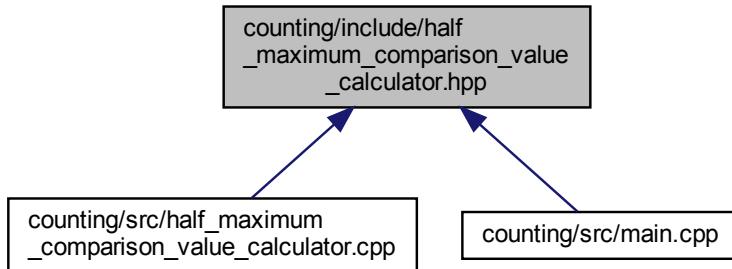
7.38 counting/include/half_maximum_comparison_value_calculator.hpp File Reference

```
#include "cl/channel.hpp"
#include "cl/data_set.hpp"
```

```
#include "cl/sensor.hpp"
Include dependency graph for half_maximum_comparison_value_calculator.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- `ctg`

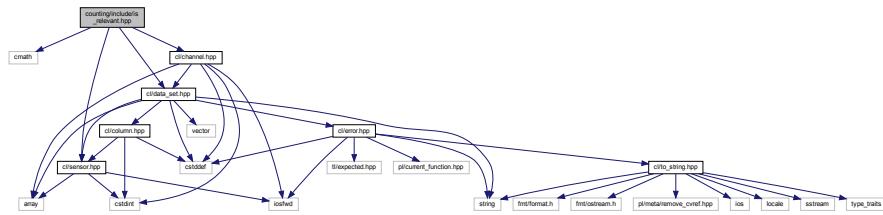
Functions

- long double `ctg::halfMaximumComparisonValueCalculator (cl::Sensor sensor, cl::Channel channel, const cl::DataSet &dataSet)`

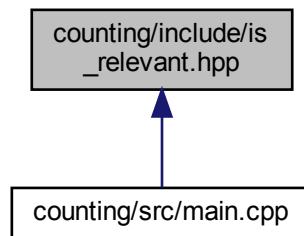
7.39 counting/include/is_relevant.hpp File Reference

```
#include <cmath>
#include "cl/channel.hpp"
#include "cl/data_set.hpp"
```

```
#include "cl/sensor.hpp"
Include dependency graph for is_relevant.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- ctg

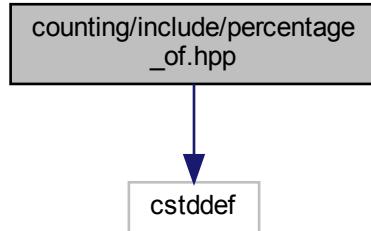
Functions

- template<typename ComparisonValueCalculator >
bool [ctg::isRelevant](#)([cl::Sensor sensor](#), [cl::Channel channel](#), const [cl::DataSet &dataSet](#), [ComparisonValueCalculator comparisonValueCalculator](#))

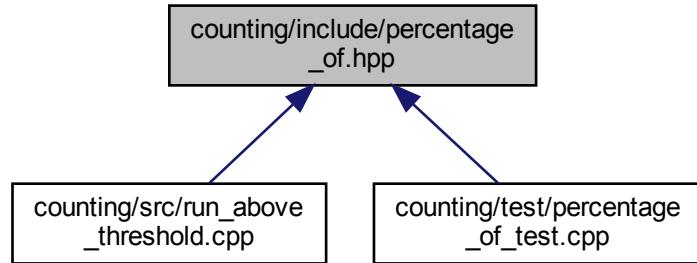
7.40 counting/include/percentage_of.hpp File Reference

```
#include <cstddef>
```

Include dependency graph for percentage_of.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- [ctg](#)

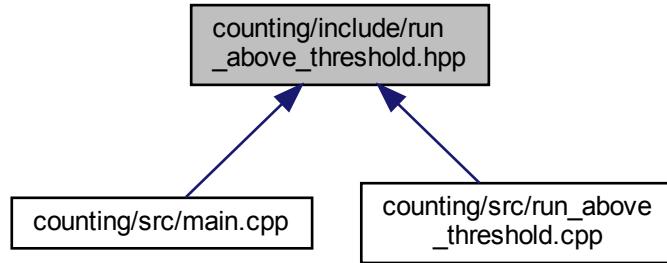
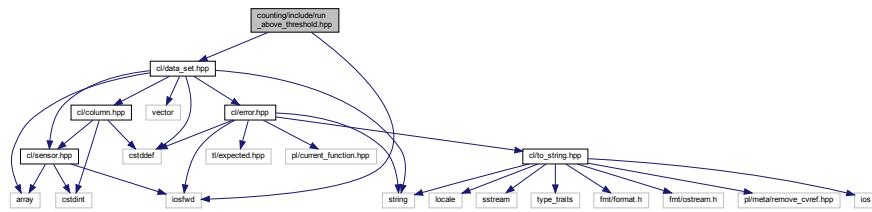
Functions

- `constexpr long double ctg::percentageOf (std::size_t amount, std::size_t totalCount) noexcept`

7.41 counting/include/run_above_threshold.hpp File Reference

```
#include <iostream>
#include "cl/data_set.hpp"
```

Include dependency graph for run_above_threshold.hpp:



Namespaces

- `ctg`

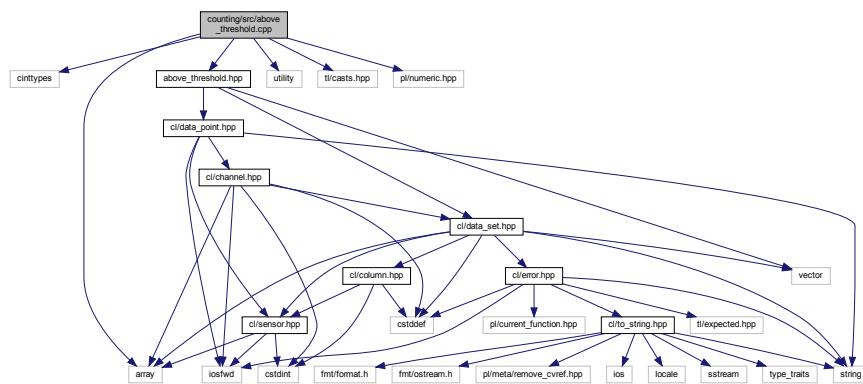
Functions

- void `ctg::runAboveThreshold` (`std::ostream &aboveThresholdLogFileStream, const cl::DataSet &dataSet)`

7.42 counting/src/above_threshold.cpp File Reference

```
#include <cinttypes>
#include <array>
#include <utility>
#include <tl/casts.hpp>
#include <pl/numeric.hpp>
```

```
#include "above_threshold.hpp"
Include dependency graph for above_threshold.cpp:
```



Namespaces

- `ctg`

Macros

- `#define CL_CHANNEL_X(enm, v, accessor) {accessor, cl::Channel::enm},`

Functions

- `std::vector< cl::DataPoint > ctg::aboveThreshold (const cl::DataSet &dataSet, long double accelerometerThreshold, long double gyroscopeThreshold)`

7.42.1 Macro Definition Documentation

7.42.1.1 CL_CHANNEL_X

```
#define CL_CHANNEL_X(
    enm,
    v,
    accessor ) {accessor, cl::Channel::enm},
```

7.42.2 Variable Documentation

7.42.2.1 channel

```
cl::Channel channel
```

Definition at line 18 of file above_threshold.cpp.

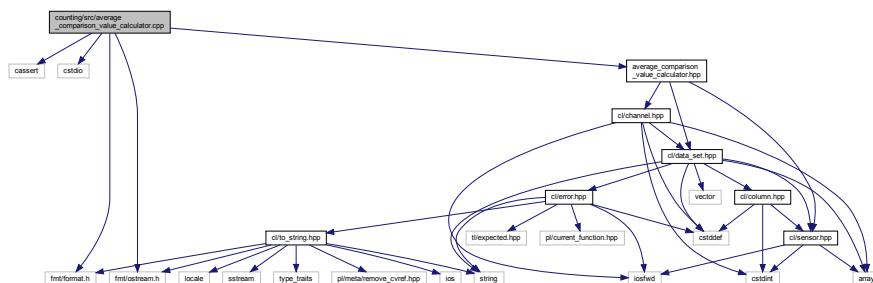
7.42.2.2 channelAccessor

```
cl::DataSet::ChannelAccessor channelAccessor
```

Definition at line 17 of file above_threshold.cpp.

7.43 counting/src/average_comparison_value_calculator.cpp File Reference

```
#include <cassert>
#include <cstdio>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include "average_comparison_value_calculator.hpp"
Include dependency graph for average_comparison_value_calculator.cpp:
```



Namespaces

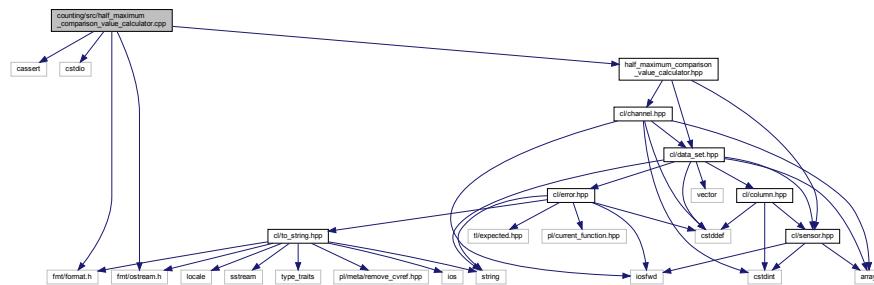
- `ctg`

Functions

- long double `ctg::averageComparisonValueCalculator (cl::Sensor sensor, cl::Channel channel, const cl::DataSet &dataSet)`

7.44 counting/src/half_maximum_comparison_value_calculator.cpp File Reference

```
#include <cassert>
#include <cstdio>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include "half_maximum_comparison_value_calculator.hpp"
Include dependency graph for half_maximum_comparison_value_calculator.cpp:
```



Namespaces

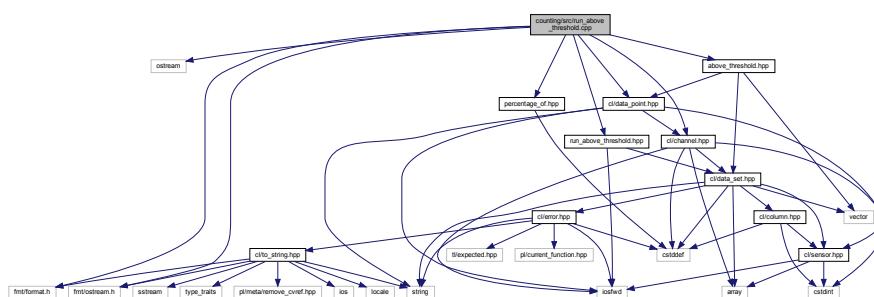
- [ctg](#)

Functions

- long double [ctg::halfMaximumComparisonValueCalculator](#) (cl::Sensor sensor, cl::Channel channel, const cl::DataSet &dataSet)

7.45 counting/src/run_above_threshold.cpp File Reference

```
#include <ostream>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include "cl/channel.hpp"
#include "cl/data_point.hpp"
#include "above_threshold.hpp"
#include "percentage_of.hpp"
#include "run_above_threshold.hpp"
Include dependency graph for run_above_threshold.cpp:
```



Namespaces

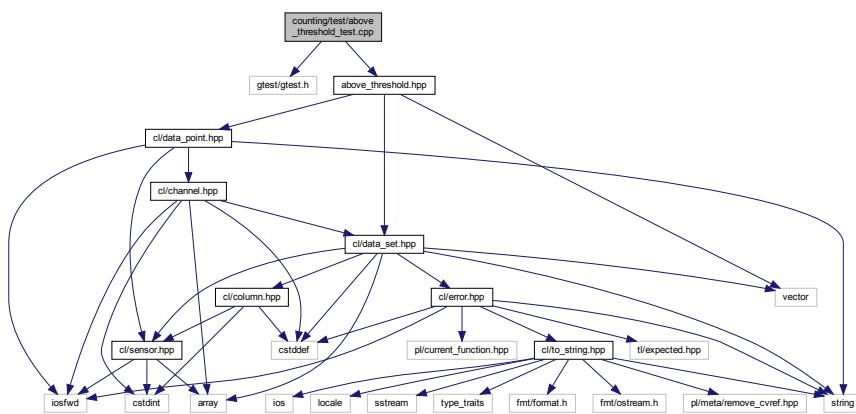
- `ctg`

Functions

- void `ctg::runAboveThreshold` (`std::ostream &aboveThresholdLogFileStream, const cl::DataSet &dataSet)`

7.46 counting/test/above_threshold_test.cpp File Reference

```
#include "gtest/gtest.h"
#include "above_threshold.hpp"
Include dependency graph for above_threshold_test.cpp:
```



Macros

- `#define EXPECT_LONG_DOUBLE_EQ(a, b) EXPECT_DOUBLE_EQ(static_cast<double>(a), static_cast<double>(b))`

Functions

- `TEST` (`aboveThreshold, shouldFindDataPointsIfThereAreAny`)

7.46.1 Macro Definition Documentation

7.46.1.1 EXPECT_LONG_DOUBLE_EQ

```
#define EXPECT_LONG_DOUBLE_EQ( a, b ) EXPECT_DOUBLE_EQ( static_cast<double>(a), static_cast<double>(b) )
```

Definition at line 6 of file above_threshold_test.cpp.

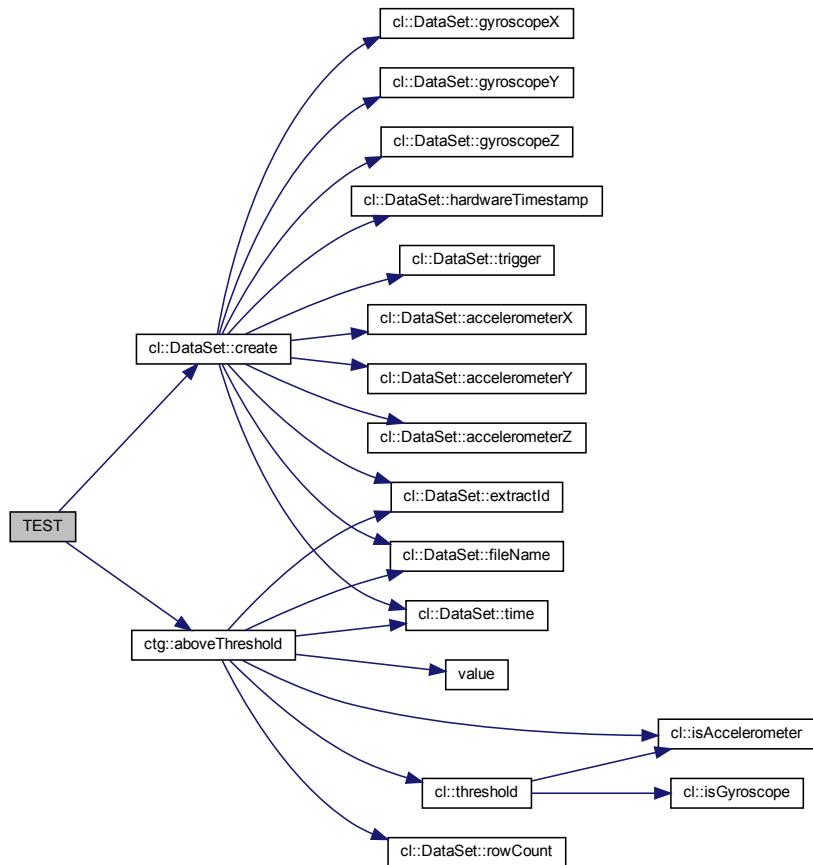
7.46.2 Function Documentation

7.46.2.1 TEST()

```
TEST( aboveThreshold , shouldFindDataPointsIfThereAreAny )
```

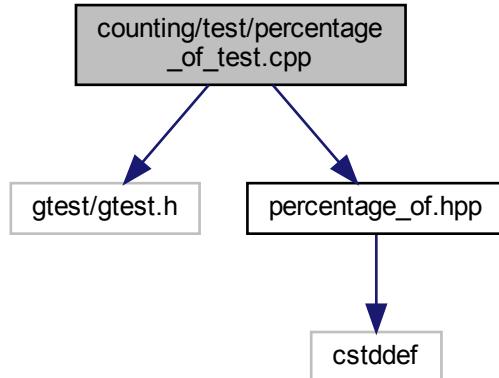
Definition at line 10 of file above_threshold_test.cpp.

Here is the call graph for this function:



7.47 counting/test/percentage_of_test.cpp File Reference

```
#include "gtest/gtest.h"
#include "percentage_of.hpp"
Include dependency graph for percentage_of_test.cpp:
```



Macros

- `#define EXPECT_LONG_DOUBLE_EQ(a, b) EXPECT_DOUBLE_EQ(static_cast<double>(a), static_cast<double>(b))`

Functions

- `TEST(percentageOf, shouldWork)`

7.47.1 Macro Definition Documentation

7.47.1.1 EXPECT_LONG_DOUBLE_EQ

```
#define EXPECT_LONG_DOUBLE_EQ(
    a,
    b ) EXPECT_DOUBLE_EQ(static_cast<double>(a), static_cast<double>(b))
```

Definition at line 6 of file percentage_of_test.cpp.

7.47.2 Function Documentation

7.47.2.1 TEST()

```
TEST (
    percentageOf ,
    shouldWork )
```

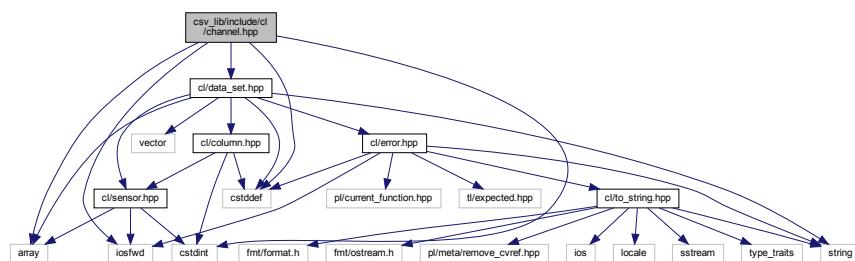
Definition at line 10 of file percentage_of_test.cpp.

Here is the call graph for this function:

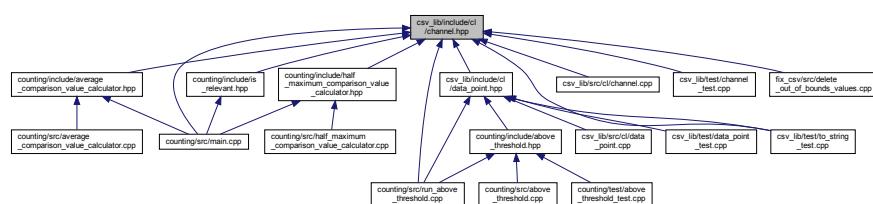


7.48 csv_lib/include/cl/channel.hpp File Reference

```
#include <cstdint>
#include <array>
#include <iostream>
#include "cl/data_set.hpp"
Include dependency graph for channel.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct `cl::data_set_accessor< Chan >`

Namespaces

- `cl`

Macros

- `#define CL_CHANNEL`
- `#define CL_CHANNEL_X(enumerator, value, dataSetAccessor) enumerator = value,`
- `#define CL_CHANNEL_X(enumerator, value, dataSetAccessor) +1`
- `#define CL_CHANNEL_X(enm, v, a) ::cl::Channel::enm,`
- `#define CL_CHANNEL_X(enumerator, value, dataSetAccessor)`

Enumerations

- enum `cl::Channel : std::uint64_t { CL_CHANNEL, CL_CHANNEL }`

Functions

- `DataSet::ChannelAccessor cl::dataSetAccessor (Channel channel)`
- `std::ostream & cl::operator<< (std::ostream &os, Channel channel)`
- `bool cl::isAccelerometer (Channel channel)`
- `bool cl::isGyroscope (Channel channel)`
- `long double cl::threshold (Channel channel)`

Variables

- `constexpr std::size_t cl::channelCount`
- `constexpr std::array< Channel, channelCount > cl::channels`
- `template<Channel Chan>`
`constexpr CL_CHANNEL DataSet::ChannelAccessor cl::data_set_accessor_v = data_set_accessor<Chan>::f`
- `constexpr long double cl::accelerometerThreshold {1.99L}`
- `constexpr long double cl::gyroscopeThreshold {1999.99L}`

7.48.1 Macro Definition Documentation

7.48.1.1 CL_CHANNEL

```
#define CL_CHANNEL
```

Value:

```
CL_CHANNEL_X(AccelerometerX, 1, &::cl::DataSet::accelerometerX) \
CL_CHANNEL_X(AccelerometerY, 2, &::cl::DataSet::accelerometerY) \
CL_CHANNEL_X(AccelerometerZ, 3, &::cl::DataSet::accelerometerZ) \
CL_CHANNEL_X(GyroscopeX, 4, &::cl::DataSet::gyroscopeX) \
CL_CHANNEL_X(GyroscopeY, 5, &::cl::DataSet::gyroscopeY) \
CL_CHANNEL_X(GyroscopeZ, 6, &::cl::DataSet::gyroscopeZ)
```

Definition at line 11 of file channel.hpp.

7.48.1.2 CL_CHANNEL_X [1/4]

```
#define CL_CHANNEL_X(
    enm,
    v,
    a ) ::cl::Channel::enm,
```

Definition at line 41 of file channel.hpp.

7.48.1.3 CL_CHANNEL_X [2/4]

```
#define CL_CHANNEL_X(
    enumerator,
    value,
    dataSetAccessor ) enumerator = value,
```

Definition at line 41 of file channel.hpp.

7.48.1.4 CL_CHANNEL_X [3/4]

```
#define CL_CHANNEL_X(
    enumerator,
    value,
    dataSetAccessor ) +1
```

Definition at line 41 of file channel.hpp.

7.48.1.5 CL_CHANNEL_X [4/4]

```
#define CL_CHANNEL_X(enumerator, value, dataSetAccessor )
```

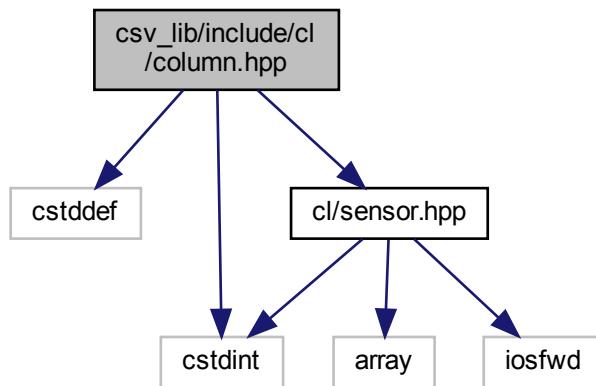
Value:

```
template<>
struct data_set_accessor<Channel::enumerator> {
    static constexpr ::cl::DataSet::ChannelAccessor f = dataSetAccessor; \
};
```

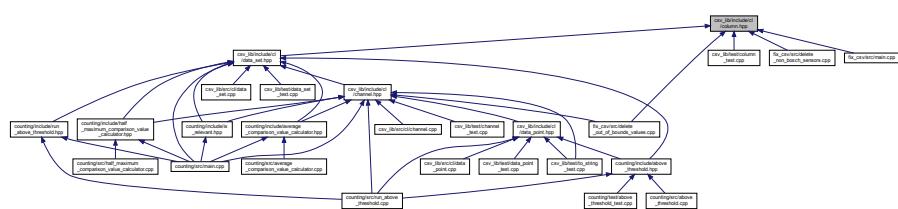
Definition at line 41 of file channel.hpp.

7.49 csv_lib/include/cl/column.hpp File Reference

```
#include <cstddef>
#include <cstdint>
#include "cl/sensor.hpp"
Include dependency graph for column.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct `cl::col_traits< Col >`

Namespaces

- `cl`

Macros

- `#define CL_SPECIALIZE_COL_TRAITS(column, columnType)`

Typedefs

- template<Column Col>
using `cl::column_type` = typename `col_traits< Col >::type`

Enumerations

- enum `cl::Column` : `std::size_t` {
`cl::Column::Time, cl::Column::HardwareTimestamp, cl::Column::ExtractId, cl::Column::Trigger,`
`cl::Column::AccelerometerX, cl::Column::AccelerometerY, cl::Column::AccelerometerZ, cl::Column::GyroscopeX,`
`cl::Column::GyroscopeY, cl::Column::GyroscopeZ, cl::Column::SamplingRate }`

Functions

- `cl::CL_SPECIALIZE_COL_TRAITS (Column::Time, long double)`
- `cl::CL_SPECIALIZE_COL_TRAITS (Column::HardwareTimestamp, std::uint64_t)`
- `cl::CL_SPECIALIZE_COL_TRAITS (Column::ExtractId, Sensor)`
- `cl::CL_SPECIALIZE_COL_TRAITS (Column::Trigger, std::uint64_t)`
- `cl::CL_SPECIALIZE_COL_TRAITS (Column::AccelerometerX, long double)`
- `cl::CL_SPECIALIZE_COL_TRAITS (Column::AccelerometerY, long double)`
- `cl::CL_SPECIALIZE_COL_TRAITS (Column::AccelerometerZ, long double)`
- `cl::CL_SPECIALIZE_COL_TRAITS (Column::GyroscopeX, long double)`
- `cl::CL_SPECIALIZE_COL_TRAITS (Column::GyroscopeY, long double)`
- `cl::CL_SPECIALIZE_COL_TRAITS (Column::GyroscopeZ, long double)`
- `cl::CL_SPECIALIZE_COL_TRAITS (Column::SamplingRate, std::uint64_t)`

Variables

- template<Column Col>
constexpr `std::size_t cl::column_index` = `col_traits<Col>::index`

7.49.1 Macro Definition Documentation

7.49.1.1 CL_SPECIALIZE_COL_TRAITS

```
#define CL_SPECIALIZE_COL_TRAITS(
    column,
    columnType )
```

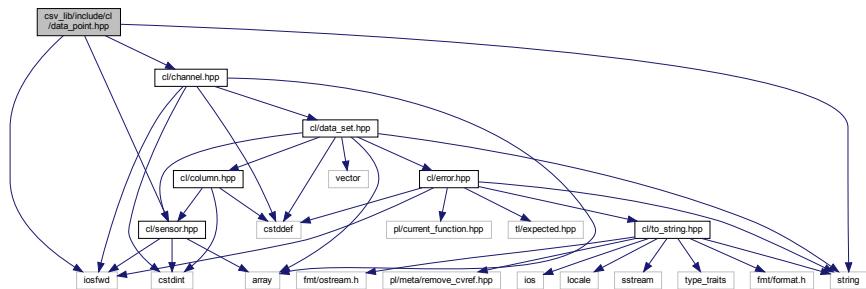
Value:

```
template<>
struct col_traits<column> {
    static constexpr std::size_t index = static_cast<std::size_t>(column);
    using type = columnType;
}
```

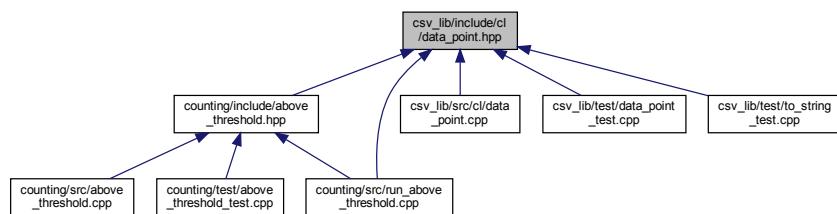
Definition at line 26 of file column.hpp.

7.50 csv_lib/include/cl/data_point.hpp File Reference

```
#include <iostream>
#include <string>
#include "cl/channel.hpp"
#include "cl/sensor.hpp"
Include dependency graph for data_point.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

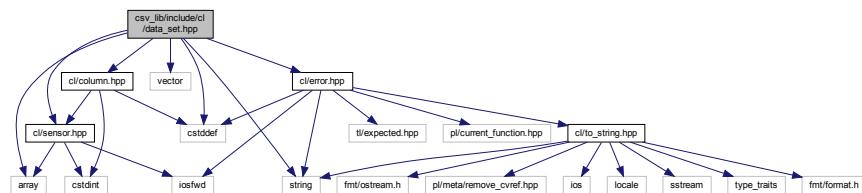
- class [cl::DataPoint](#)

Namespaces

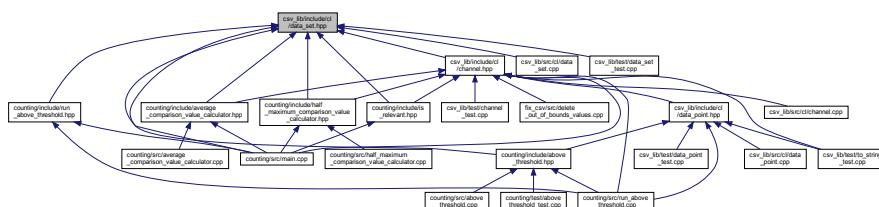
- cl

7.51 csv_lib/include/cl/data_set.hpp File Reference

```
#include <cstddef>
#include <array>
#include <string>
#include <vector>
#include "cl/column.hpp"
#include "cl/error.hpp"
#include "cl/sensor.hpp"
Include dependency graph for data_set.hpp
```



This graph shows which files directly or indirectly include this file:



Classes

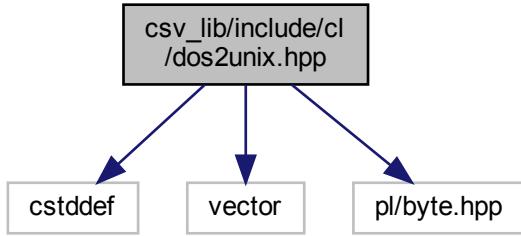
- class cl::DataSet

Namespaces

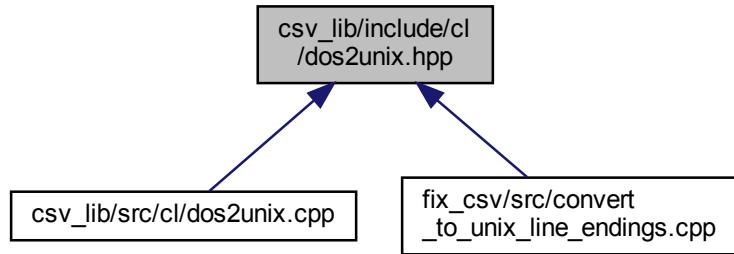
- C

7.52 csv_lib/include/cl/dos2unix.hpp File Reference

```
#include <cstddef>
#include <vector>
#include <pl/byte.hpp>
Include dependency graph for dos2unix.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

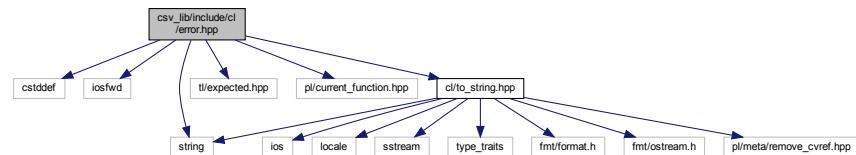
- `cl`

Functions

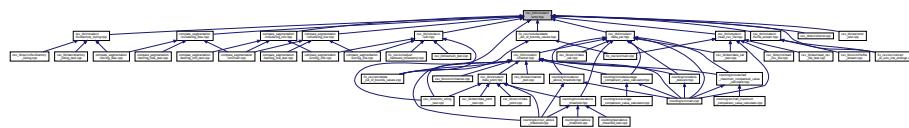
- `std::vector< pl::byte > cl::dos2unix (const void *p, std::size_t size)`

7.53 csv_lib/include/cl/error.hpp File Reference

```
#include <cstddef>
#include <iostream>
#include <string>
#include <t1/expected.hpp>
#include <pl/current_function.hpp>
#include "cl/to_string.hpp"
Include dependency graph for error.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [cl::Error](#)

Namespaces

- [cl](#)

Macros

- `#define CL_ERROR_KIND`
- `#define CL_ERROR_KIND_X(kind) kind,`
- `#define CL_UNEXPECTED(kind, message)`

Typedefs

- template<typename Ty >
using [cl::Expected](#) = t1::expected< Ty, Error >

7.53.1 Macro Definition Documentation

7.53.1.1 CL_ERROR_KIND

```
#define CL_ERROR_KIND
```

Value:

```
CL_ERROR_KIND_X(Filesystem) \
CL_ERROR_KIND_X(InvalidArgument) \
CL_ERROR_KIND_X(OutOfRange) \
CL_ERROR_KIND_X(Parsing) \
CL_ERROR_KIND_X(Logic)
```

Definition at line 14 of file error.hpp.

7.53.1.2 CL_ERROR_KIND_X

```
#define CL_ERROR_KIND_X(
    kind ) kind,
```

Definition at line 26 of file error.hpp.

7.53.1.3 CL_UNEXPECTED

```
#define CL_UNEXPECTED (
    kind,
    message )
```

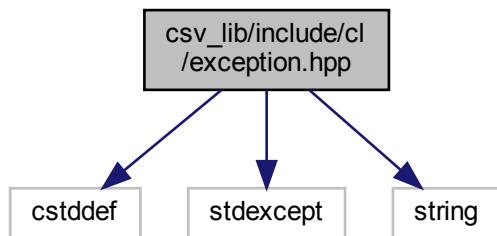
Value:

```
::tl::make_unexpected(
    ::cl::Error(kind, __FILE__, __CURRENT_FUNCTION__, __LINE__, message))
```

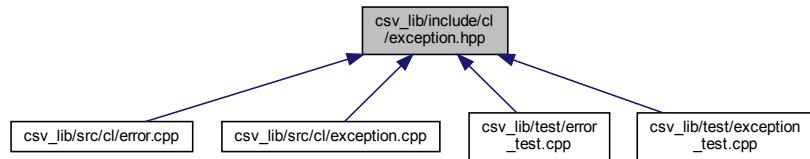
Definition at line 66 of file error.hpp.

7.54 csv_lib/include/cl/exception.hpp File Reference

```
#include <cstddef>
#include <stdexcept>
#include <string>
Include dependency graph for exception.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

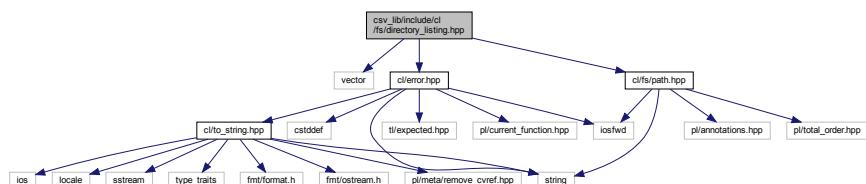
- class [cl::Exception](#)

Namespaces

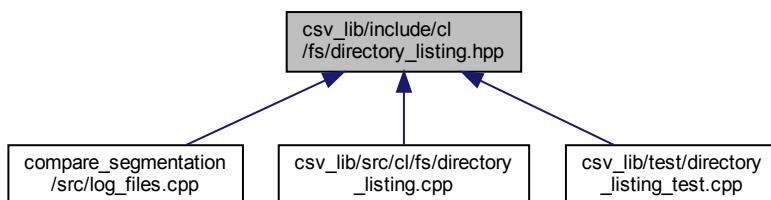
- [cl](#)

7.55 csv_lib/include/cl/fs/directory_listing.hpp File Reference

```
#include <vector>
#include <cl/error.hpp>
#include <cl/fs/path.hpp>
Include dependency graph for directory_listing.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- `cl`
- `cl::fs`

Enumerations

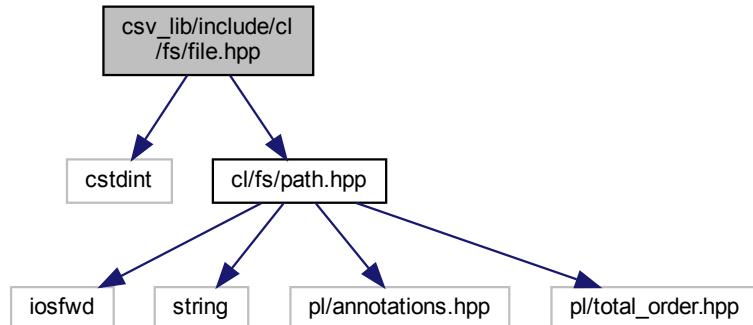
- enum `cl::fs::DirectoryListingOption` { `cl::fs::DirectoryListingOption::None`, `cl::fs::DirectoryListingOption::ExcludeDotAndDotDot` }

Functions

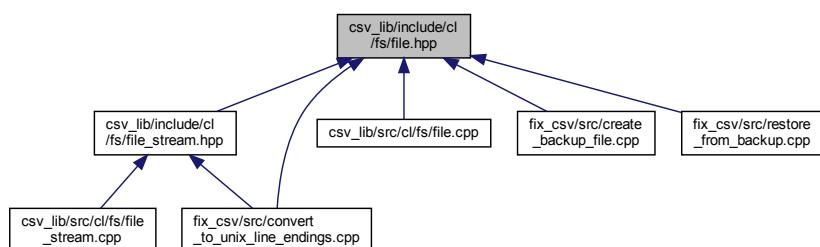
- Expected< std::vector< Path > > `cl::fs::directoryListing` (const Path &directoryPath, DirectoryListingOption directoryListingOption=DirectoryListingOption::ExcludeDotAndDotDot)

7.56 csv_lib/include/cl/fs/file.hpp File Reference

```
#include <cstdint>
#include "cl/fs/path.hpp"
Include dependency graph for file.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [cl::fs::File](#)

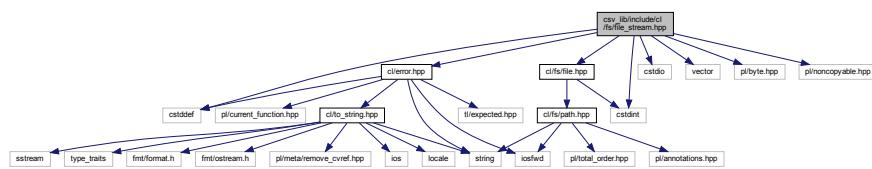
Represents a file.

Namespaces

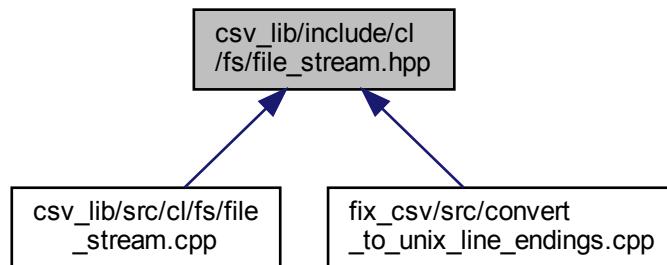
- [cl](#)
- [cl::fs](#)

7.57 csv_lib/include/cl/fs/file_stream.hpp File Reference

```
#include <cstddef>
#include <cstdint>
#include <cstdio>
#include <vector>
#include <pl/byte.hpp>
#include <pl/noncopyable.hpp>
#include "cl/error.hpp"
#include "cl/fs/file.hpp"
Include dependency graph for file_stream.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

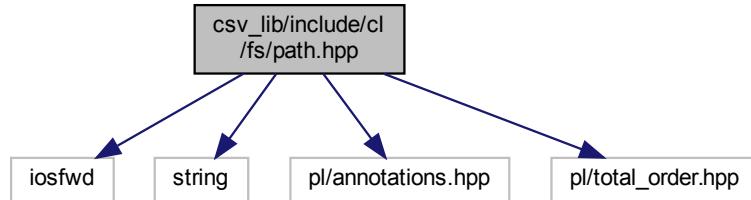
- class [cl::fs::FileStream](#)

Namespaces

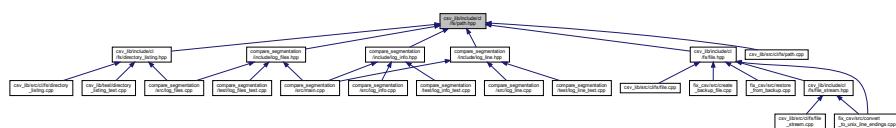
- [cl](#)
- [cl::fs](#)

7.58 csv_lib/include/cl/fs/path.hpp File Reference

```
#include <iostream>
#include <string>
#include <pl/annotations.hpp>
#include <pl/total_order.hpp>
Include dependency graph for path.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [cl::fs::Path](#)

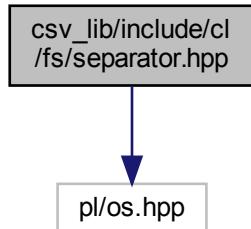
Namespaces

- [cl](#)
- [cl::fs](#)

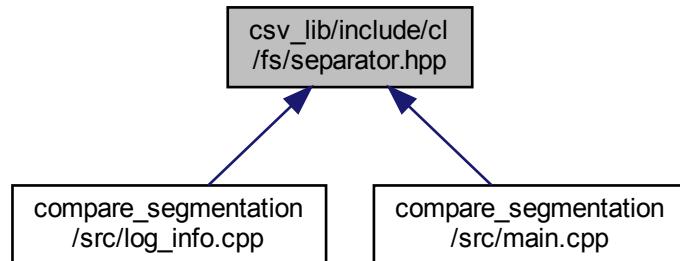
7.59 csv_lib/include/cl/fs/separator.hpp File Reference

```
#include <pl/os.hpp>
```

Include dependency graph for separator.hpp:



This graph shows which files directly or indirectly include this file:



Macros

- `#define CL_FS_SEPARATOR "\\\"`

7.59.1 Macro Definition Documentation

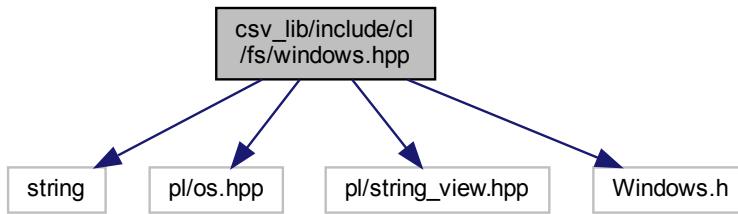
7.59.1.1 CL_FS_SEPARATOR

```
#define CL_FS_SEPARATOR "\\\"
```

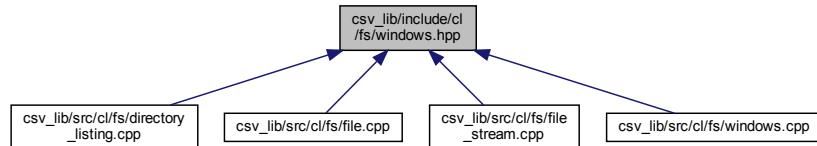
Definition at line 6 of file separator.hpp.

7.60 csv_lib/include/cl/fs/windows.hpp File Reference

```
#include <string>
#include <pl/os.hpp>
#include <pl/string_view.hpp>
#include <Windows.h>
Include dependency graph for windows.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

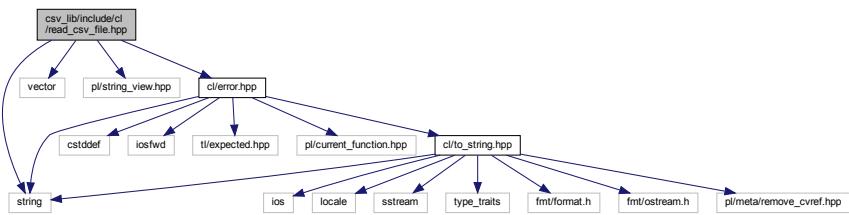
- `cl`
- `cl::fs`

Functions

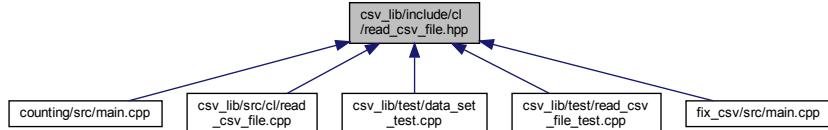
- `std::wstring cl::fs::utf8ToUtf16 (pl::string_view utf8)`
- `std::string cl::fs::utf16ToUtf8 (pl::wstring_view utf16)`
- `std::wstring cl::fs::formatError (DWORD errorCode)`

7.61 csv_lib/include/cl/read_csv_file.hpp File Reference

```
#include <string>
#include <vector>
#include <pl/string_view.hpp>
#include "cl/error.hpp"
Include dependency graph for read_csv_file.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- `cl`

Enumerations

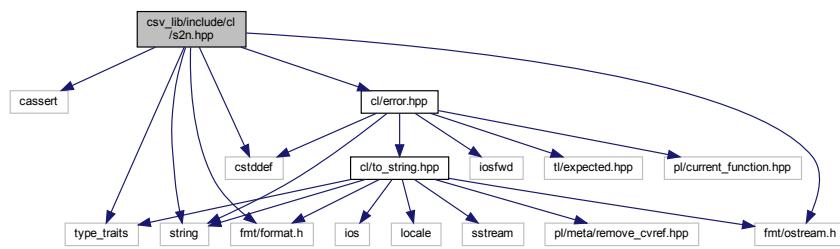
- enum `cl::CsvFileKind` { `cl::CsvFileKind::Raw`, `cl::CsvFileKind::Fixed` }

Functions

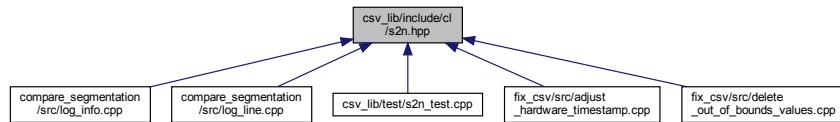
- `Expected< std::vector< std::vector< std::string > >> cl::readCsvFile (ppl::string_view csvFilePath, std::vector< std::string > *columnNames=nullptr, CsvFileKind csvFileKind=CsvFileKind::Fixed) noexcept`

7.62 csv_lib/include/cl/s2n.hpp File Reference

```
#include <cassert>
#include <cstddef>
#include <string>
#include <type_traits>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include "cl/error.hpp"
Include dependency graph for s2n.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- `cl`

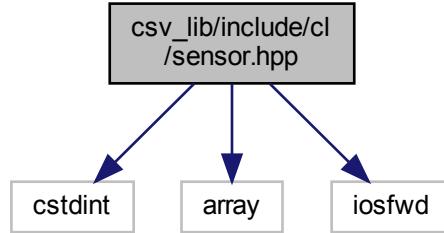
Functions

- template<typename Integer >
`Expected< Integer > cl::s2n (const std::string &str, std::size_t *pos=nullptr, [[maybe_unused]] int base=10)`

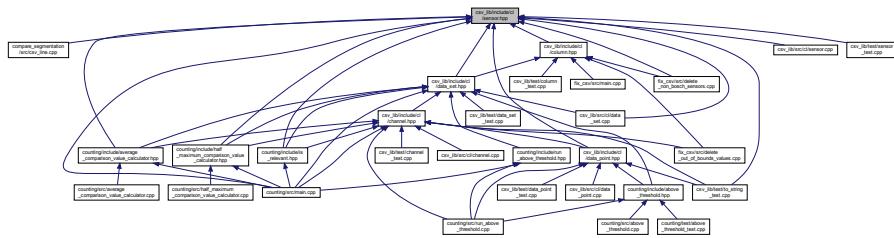
7.63 csv_lib/include/cl/sensor.hpp File Reference

```
#include <cstdint>
#include <array>
```

```
#include <iostream>  
Include dependency graph for sensor.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- cl

Macros

- `#define CL_SENSOR`
 - `#define CL_SENSOR_X(enumerator, value) enumerator = value,`
 - `#define CL_SENSOR_X(enm, v) ::cl::Sensor::enm,`

Enumerations

- enum cl::Sensor : std::uint64_t { cl::Sensor::CL_SENSOR_X, cl::Sensor::CL_SENSOR_Y }

Functions

- std::ostream & `cl::operator<<` (std::ostream &os, Sensor `sensor`)

Variables

- `constexpr std::array< Sensor, 4 > cl::sensors`

7.63.1 Macro Definition Documentation

7.63.1.1 CL_SENSOR

```
#define CL_SENSOR
```

Value:

```
CL_SENSOR_X(LeftArm, 769) \
CL_SENSOR_X(Belly, 770) \
CL_SENSOR_X(RightArm, 771) \
CL_SENSOR_X(Chest, 772)
```

Definition at line 9 of file sensor.hpp.

7.63.1.2 CL_SENSOR_X [1/2]

```
#define CL_SENSOR_X(
    enm,
    v ) ::cl::Sensor::enm,
```

Definition at line 16 of file sensor.hpp.

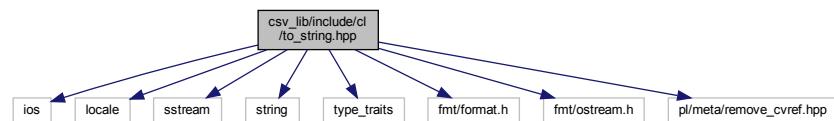
7.63.1.3 CL_SENSOR_X [2/2]

```
#define CL_SENSOR_X(
    enumerator,
    value ) enumerator = value,
```

Definition at line 16 of file sensor.hpp.

7.64 csv_lib/include/cl/to_string.hpp File Reference

```
#include <iostream>
#include <locale>
#include <sstream>
#include <string>
#include <type_traits>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include <pl/meta/remove_cvref.hpp>
Include dependency graph for to_string.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

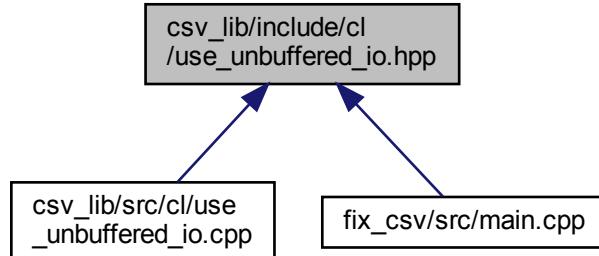
- [cl](#)

Functions

- template<typename Ty >
`std::string cl::to_string (const Ty &ty)`

7.65 csv_lib/include/cl/use_unbuffered_io.hpp File Reference

This graph shows which files directly or indirectly include this file:



Namespaces

- `cl`

Functions

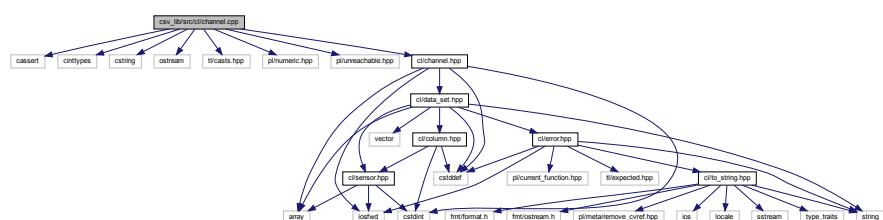
- void `cl::useUnbufferedIo ()`

7.66 csv_lib/src/cl/channel.cpp File Reference

```

#include <cassert>
#include <cinttypes>
#include <cstring>
#include <iostream>
#include <tl/casts.hpp>
#include <pl/numeric.hpp>
#include <pl/unreachable.hpp>
#include "cl/channel.hpp"
  
```

Include dependency graph for `channel.cpp`:



Namespaces

- `cl`

Macros

- `#define CL_CHANNEL_X(enm, v, acc) case Channel::enm: return data_set_accessor_v<Channel::enm>;`
- `#define CL_CHANNEL_X(enumerator, value, dataSetAccessor) case Channel::enumerator: return os << #enumerator;`

Functions

- `DataSet::ChannelAccessor cl::dataSetAccessor (Channel channel)`
- `std::ostream & cl::operator<< (std::ostream &os, Channel channel)`
- `bool cl::isAccelerometer (Channel channel)`
- `bool cl::isGyroscope (Channel channel)`
- `long double cl::threshold (Channel channel)`

7.66.1 Macro Definition Documentation

7.66.1.1 CL_CHANNEL_X [1/2]

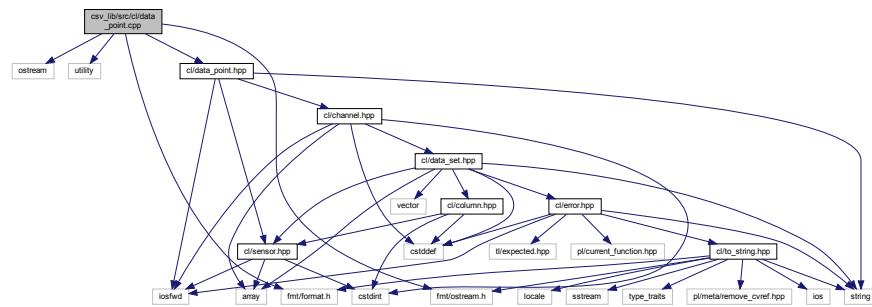
```
#define CL_CHANNEL_X(
    enm,
    v,
    acc ) case Channel::enm: return data_set_accessor_v<Channel::enm>;
```

7.66.1.2 CL_CHANNEL_X [2/2]

```
#define CL_CHANNEL_X(
    enumerator,
    value,
    dataSetAccessor ) case Channel::enumerator: return os << #enumerator;
```

7.67 csv_lib/src/cl/data_point.cpp File Reference

```
#include <iostream>
#include <utility>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include "cl/data_point.hpp"
Include dependency graph for data_point.cpp:
```



Namespaces

- `cl`

Functions

- `std::ostream & cl::operator<< (std::ostream &os, const DataPoint &dataPoint)`
- `dataPoint fileName ()`
- `dataPoint dataPoint time ()`
- `dataPoint dataPoint dataPoint sensor ()`
- `dataPoint dataPoint dataPoint dataPoint channel ()`
- `dataPoint dataPoint dataPoint dataPoint value ()`

7.67.1 Function Documentation

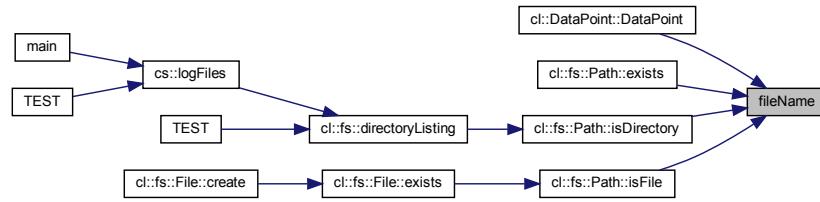
7.67.1.1 channel()

```
dataPoint dataPoint dataPoint dataPoint channel ( )
```

7.67.1.2 fileName()

dataPoint fileName ()

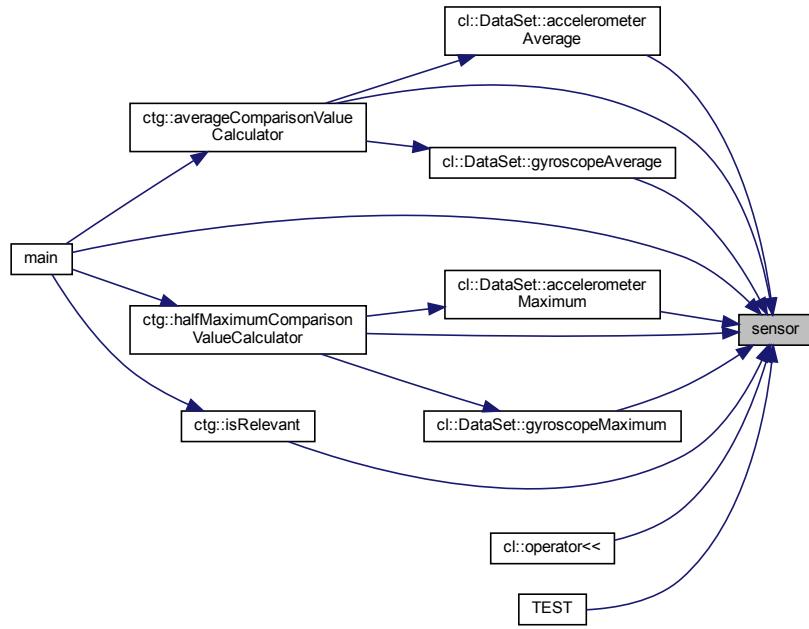
Here is the caller graph for this function:



7.67.1.3 sensor()

```
dataPoint dataPoint dataPoint sensor ( )
```

Here is the caller graph for this function:



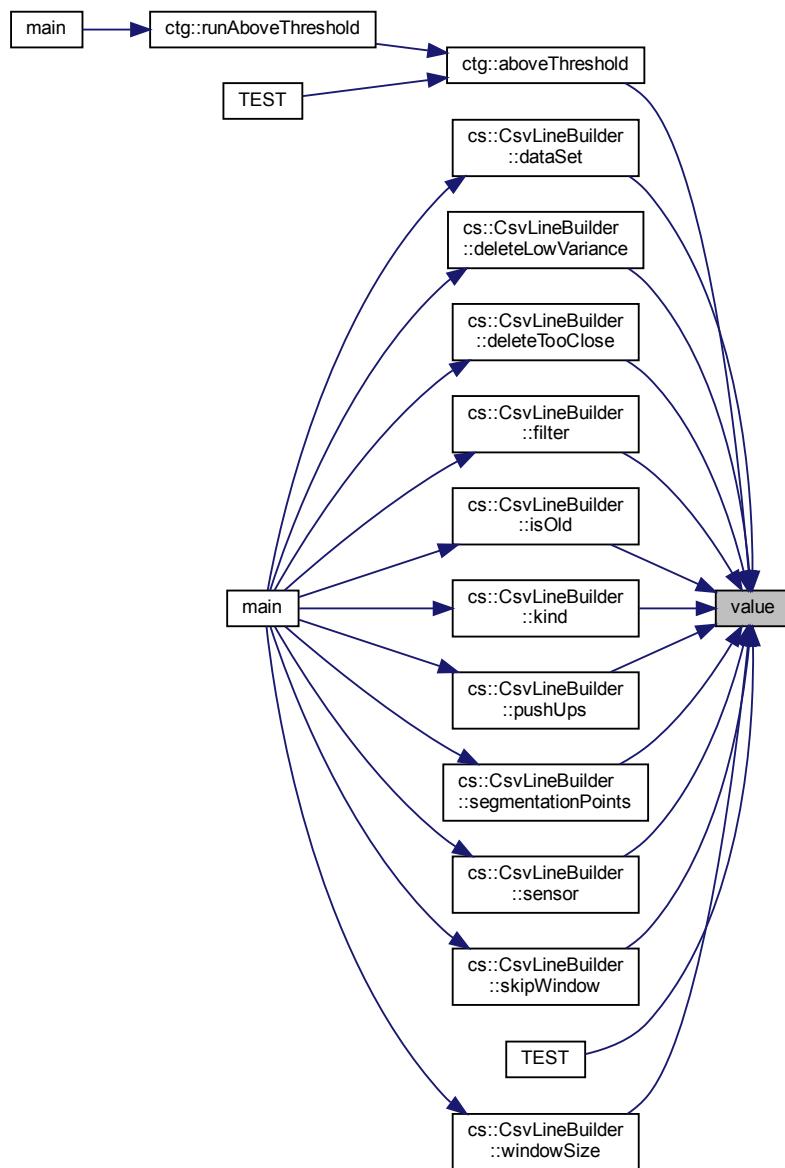
7.67.1.4 time()

```
dataPoint dataPoint time ( )
```

7.67.1.5 value()

```
dataPoint dataPoint dataPoint dataPoint value ( )
```

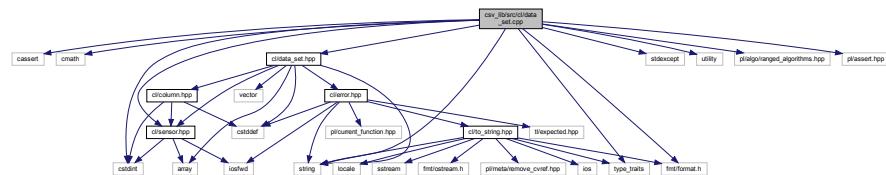
Here is the caller graph for this function:



7.68 csv_lib/src/cl/data_set.cpp File Reference

```
#include <cassert>
#include <cmath>
#include <cstdint>
#include <stdexcept>
#include <string>
#include <type_traits>
#include <utility>
#include <fmt/format.h>
#include <pl/algo/ranged_algorithms.hpp>
#include <pl/assert.hpp>
#include "cl/data_set.hpp"
#include "cl/sensor.hpp"

Include dependency graph for data_set.cpp:
```



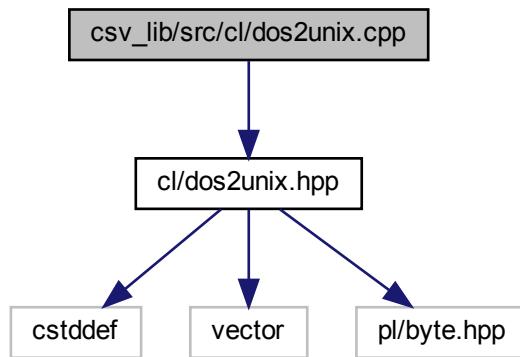
Namespaces

- `cl`

7.69 csv_lib/src/cl/dos2unix.cpp File Reference

```
#include "cl/dos2unix.hpp"

Include dependency graph for dos2unix.cpp:
```



Namespaces

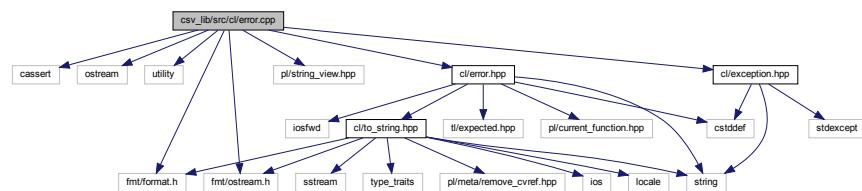
- [cl](#)

Functions

- std::vector< pl::byte > [cl::dos2unix](#) (const void *p, std::size_t size)

7.70 csv_lib/src/cl/error.cpp File Reference

```
#include <cassert>
#include <iostream>
#include <utility>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include <pl/string_view.hpp>
#include "cl/error.hpp"
#include "cl/exception.hpp"
Include dependency graph for error.cpp:
```



Namespaces

- [cl](#)

Macros

- #define [CL_ERROR_KIND_X](#)(kind) case Error::kind: return #kind;

Functions

- std::ostream & [cl::operator<<](#) (std::ostream &os, const Error &[error](#))

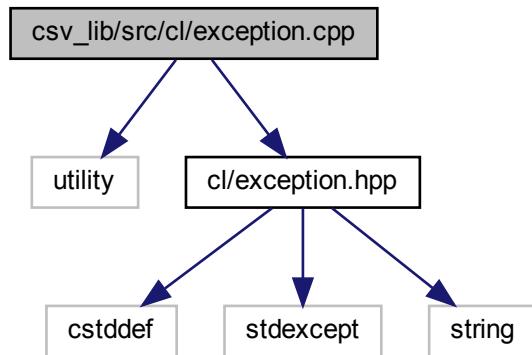
7.70.1 Macro Definition Documentation

7.70.1.1 CL_ERROR_KIND_X

```
#define CL_ERROR_KIND_X(
    kind ) case Error::kind: return #kind;
```

7.71 csv_lib/src/cl/exception.cpp File Reference

```
#include <utility>
#include "cl/exception.hpp"
Include dependency graph for exception.cpp:
```



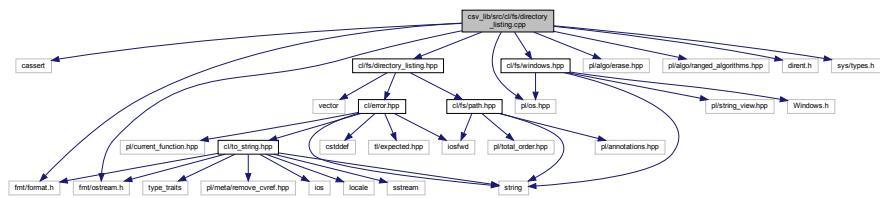
Namespaces

- `cl`

7.72 csv_lib/src/cl/fs/directory_listing.cpp File Reference

```
#include <cassert>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include <pl/algo/erase.hpp>
#include <pl/algo/ranged_algorithms.hpp>
#include <pl/os.hpp>
#include <cl/fs/windows.hpp>
#include <dirent.h>
#include <sys/types.h>
```

```
#include <cl/fs/directory_listing.hpp>
Include dependency graph for directory_listing.cpp:
```



Namespaces

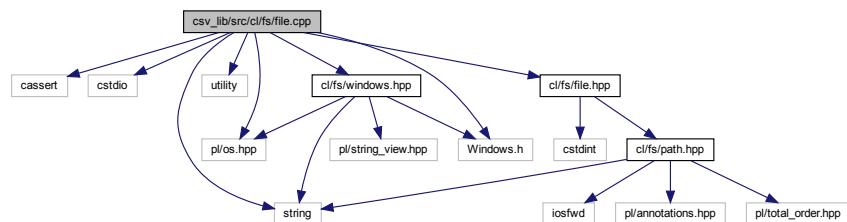
- `cl`
- `cl::fs`

Functions

- Expected< std::vector< Path > > `cl::fs::directoryListing` (const Path &directoryPath, DirectoryListingOption directoryListingOption=DirectoryListingOption::ExcludeDotAndDotDot)

7.73 csv_lib/src/cl/fs/file.cpp File Reference

```
#include <cassert>
#include <cstdio>
#include <string>
#include <utility>
#include <pl/os.hpp>
#include "cl/fs/windows.hpp"
#include <Windows.h>
#include "cl/fs/file.hpp"
Include dependency graph for file.cpp:
```

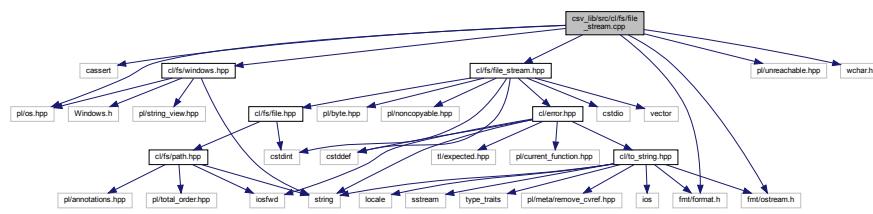


Namespaces

- `cl`
- `cl::fs`

7.74 csv_lib/src/cl/fs/file_stream.cpp File Reference

```
#include <cassert>
#include <pl/os.hpp>
#include <pl/unreachable.hpp>
#include "cl/fs/windows.hpp"
#include <wchar.h>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include "cl/fs/file_stream.hpp"
Include dependency graph for file_stream.cpp:
```

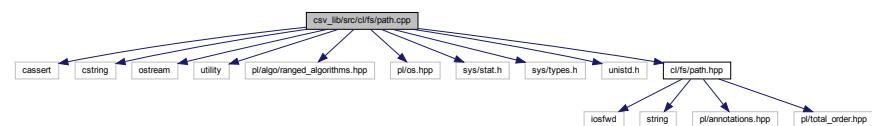


Namespaces

- `cl`
- `cl::fs`

7.75 csv_lib/src/cl/fs/path.cpp File Reference

```
#include <cassert>
#include <cstring>
#include <iostream>
#include <utility>
#include <pl/algo/ranged_algorithms.hpp>
#include <pl/os.hpp>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
#include "cl/fs/path.hpp"
Include dependency graph for path.cpp:
```



Namespaces

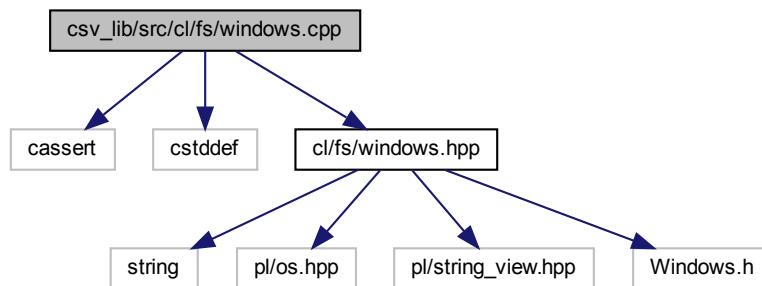
- `cl`
- `cl::fs`

Functions

- std::ostream & `cl::fs::operator<<` (std::ostream &os, const Path &path)
- bool `cl::fs::operator<` (const Path &lhs, const Path &rhs) noexcept
- bool `cl::fs::operator==` (const Path &lhs, const Path &rhs) noexcept

7.76 csv_lib/src/cl/fs/windows.cpp File Reference

```
#include <cassert>
#include <cstddef>
#include "cl/fs/windows.hpp"
Include dependency graph for windows.cpp:
```



Namespaces

- `cl`
- `cl::fs`

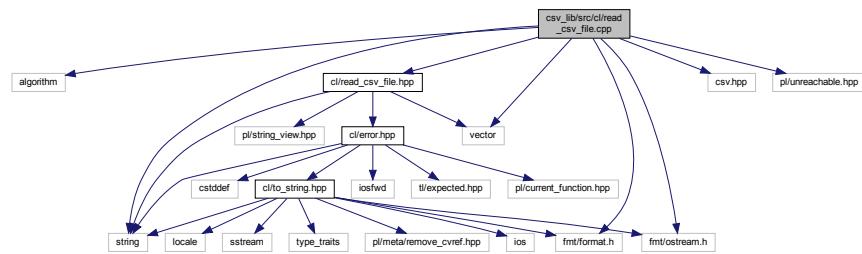
Functions

- std::wstring `cl::fs::utf8ToUtf16` (pl::string_view utf8)
- std::string `cl::fs::utf16ToUtf8` (pl::wstring_view utf16)
- std::wstring `cl::fs::formatError` (DWORD errorCode)

7.77 csv_lib/src/cl/read_csv_file.cpp File Reference

```
#include <algorithm>
#include <string>
#include <vector>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include <csv.hpp>
#include <pl/unreachable.hpp>
```

```
#include "cl/read_csv_file.hpp"
Include dependency graph for read_csv_file.cpp:
```



Namespaces

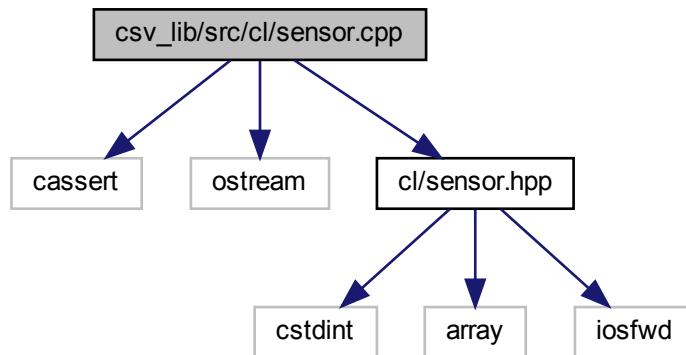
- `cl`

Functions

- `Expected< std::vector< std::vector< std::string > > > cl::readCsvFile (pl::string_view csvFilePath, std::vector< std::string > *columnNames=nullptr, CsvFileKind csvFileKind=CsvFileKind::Fixed) noexcept`

7.78 csv_lib/src/cl/sensor.cpp File Reference

```
#include <cassert>
#include <ostream>
#include "cl/sensor.hpp"
Include dependency graph for sensor.cpp:
```



Namespaces

- `cl`

Macros

- `#define CL_SENSOR_X(enum, value) case Sensor::enum: return os << #enum;`

Functions

- `std::ostream & cl::operator<< (std::ostream &os, Sensor sensor)`

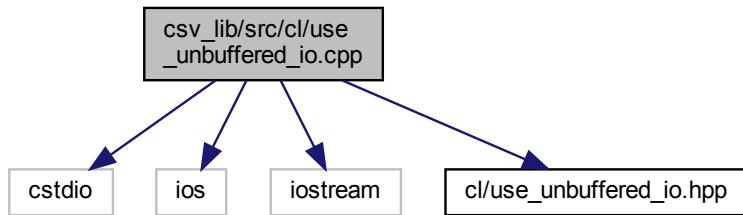
7.78.1 Macro Definition Documentation

7.78.1.1 CL_SENSOR_X

```
#define CL_SENSOR_X(
    enum,
    value ) case Sensor::enum: return os << #enum;
```

7.79 csv_lib/src/cl/use_unbuffered_io.cpp File Reference

```
#include <cstdio>
#include <iostream>
#include <iostream>
#include "cl/use_unbuffered_io.hpp"
Include dependency graph for use_unbuffered_io.cpp:
```



Namespaces

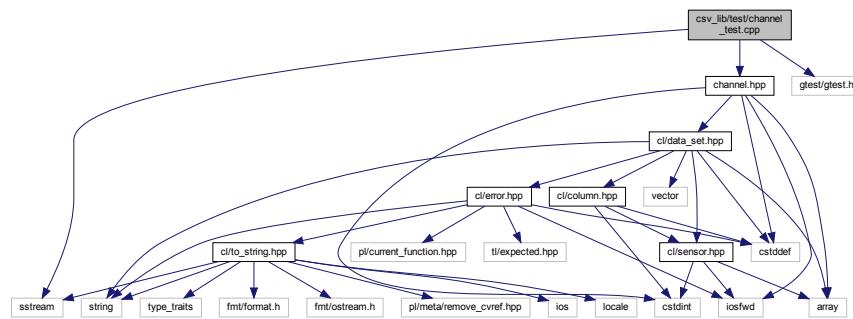
- `cl`

Functions

- `void cl::useUnbufferedIo ()`

7.80 csv_lib/test/channel_test.cpp File Reference

```
#include <iostream>
#include "gtest/gtest.h"
#include "channel.hpp"
Include dependency graph for channel_test.cpp:
```



Functions

- [TEST \(channel, shouldHaveCorrectCount\)](#)
- [TEST \(channel, shouldHaveCorrectValues\)](#)
- [TEST \(channel, shouldPrintCorrectly\)](#)
- [TEST \(channel, shouldMapToCorrectDataSetAccessors\)](#)

7.80.1 Function Documentation

7.80.1.1 TEST() [1/4]

```
TEST (
    channel ,
    shouldHaveCorrectCount )
```

Definition at line 7 of file channel_test.cpp.

7.80.1.2 TEST() [2/4]

```
TEST (
    channel ,
    shouldHaveCorrectValues )
```

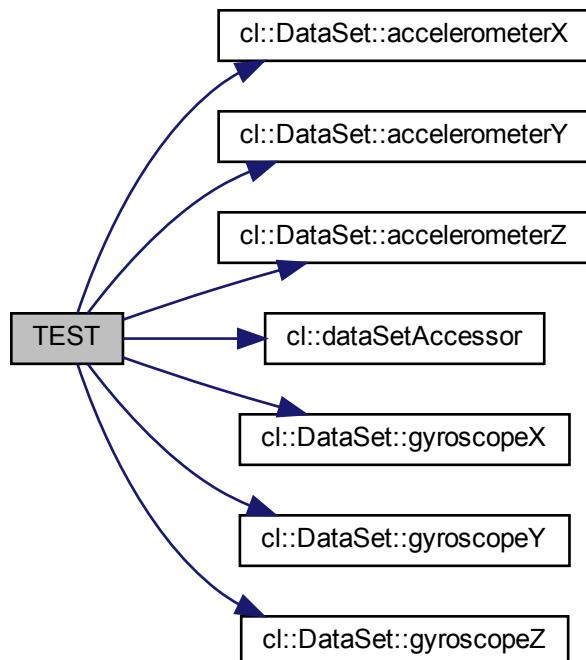
Definition at line 9 of file channel_test.cpp.

7.80.1.3 TEST() [3/4]

```
TEST (
    channel ,
    shouldMapToCorrectDataSetAccessors )
```

Definition at line 35 of file channel_test.cpp.

Here is the call graph for this function:



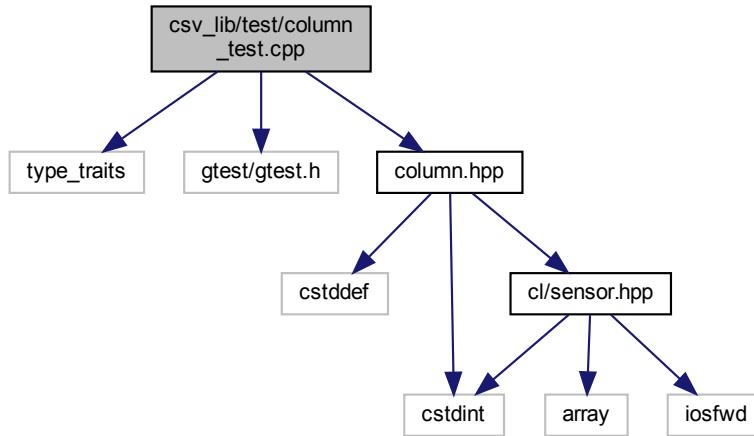
7.80.1.4 TEST() [4/4]

```
TEST (
    channel ,
    shouldPrintCorrectly )
```

Definition at line 19 of file channel_test.cpp.

7.81 csv_lib/test/column_test.cpp File Reference

```
#include <type_traits>
#include "gtest/gtest.h"
#include "column.hpp"
Include dependency graph for column_test.cpp:
```



Functions

- [TEST](#) (`column`, `shouldHaveCorrectIndex`)
- [TEST](#) (`column`, `shouldHaveCorrectColumnType`)

7.81.1 Function Documentation

7.81.1.1 TEST() [1/2]

```
TEST (
    column ,
    shouldHaveCorrectColumnType )
```

Definition at line 22 of file `column_test.cpp`.

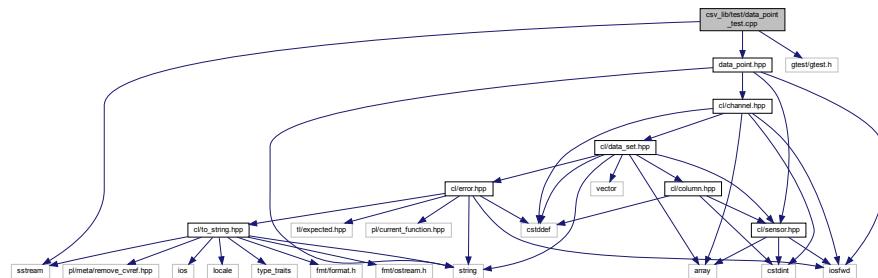
7.81.1.2 TEST() [2/2]

```
TEST (
    column ,
    shouldHaveCorrectIndex )
```

Definition at line 7 of file `column_test.cpp`.

7.82 csv_lib/test/data_point_test.cpp File Reference

```
#include <sstream>
#include "gtest/gtest.h"
#include "data_point.hpp"
Include dependency graph for data_point_test.cpp:
```



Functions

- **TEST** (DataPoint, shouldPrintCorrectly)
 - **TEST** (DataPoint, shouldGetValuesCorrectly)

Variables

- const cl::DataPoint dp

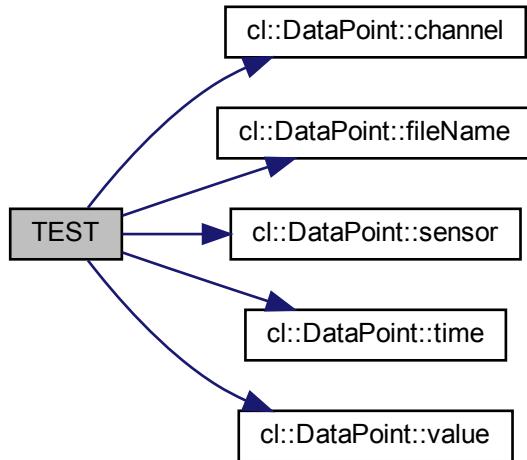
7.82.1 Function Documentation

7.82.1.1 TEST() [1/2]

```
TEST ( DataPoint ,  
       shouldGetValuesCorrectly )
```

Definition at line 23 of file data_point_test.cpp.

Here is the call graph for this function:



7.82.1.2 TEST() [2/2]

```
TEST (
    DataPoint ,
    shouldPrintCorrectly )
```

Definition at line 14 of file data_point_test.cpp.

7.82.2 Variable Documentation

7.82.2.1 dp

```
const cl::DataPoint dp
```

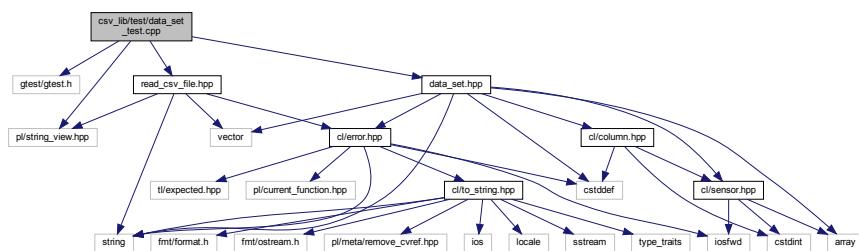
Initial value:

```
{
    "file.csv",
    0.01,
    cl::Sensor::Chest,
    cl::Channel::AccelerometerX,
    50.01}
```

Definition at line 7 of file data_point_test.cpp.

7.83 csv_lib/test/data_set_test.cpp File Reference

```
#include "gtest/gtest.h"
#include <pl/string_view.hpp>
#include "data_set.hpp"
#include "read_csv_file.hpp"
Include dependency graph for data_set_test.cpp:
```



Macros

- #define `EXPECT_LONG_DOUBLE_EQ(a, b)` `EXPECT_DOUBLE_EQ(static_cast<double>(a), static_cast<double>(b))`

Functions

- `TEST` (`DataSet`, `shouldBeAbleToCreateFromValidData`)
- `TEST` (`DataSet`, `shouldNotBeAbleToCreateFromEmtyMatrix`)
- `TEST` (`DataSet`, `shouldNotBeAbleToCreateFromJaggedMatrix`)
- `TEST` (`DataSet`, `shouldNotBeAbleToCreateFromInvalidData`)

7.83.1 Macro Definition Documentation

7.83.1.1 EXPECT_LONG_DOUBLE_EQ

```
#define EXPECT_LONG_DOUBLE_EQ(
    a,
    b ) EXPECT_DOUBLE_EQ(static_cast<double>(a), static_cast<double>(b))
```

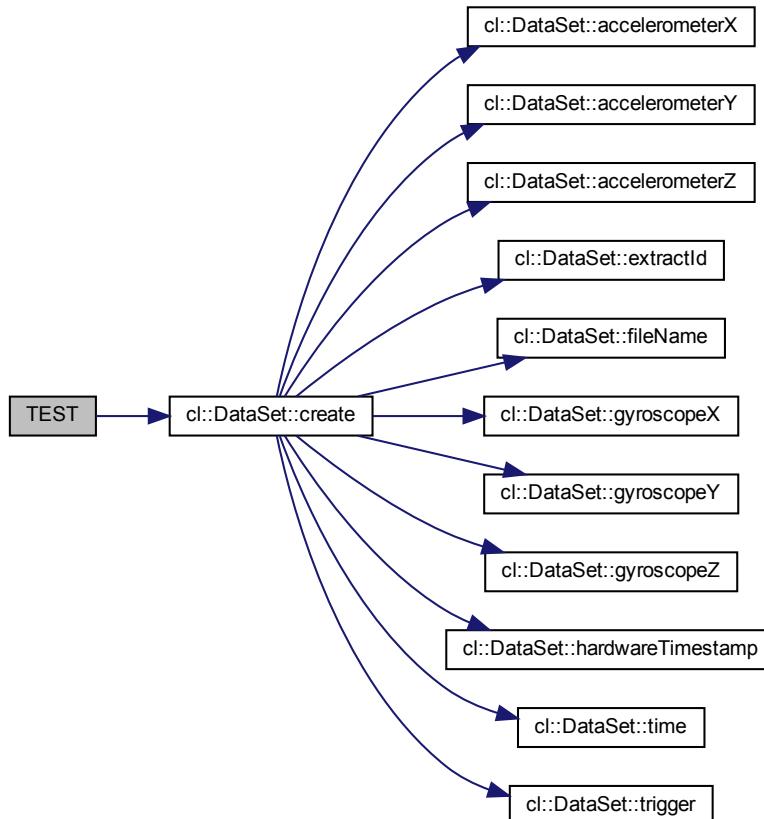
7.83.2 Function Documentation

7.83.2.1 TEST() [1/4]

```
TEST (
    DataSet ,
    shouldBeAbleToCreateFromValidData )
```

Definition at line 17 of file `data_set_test.cpp`.

Here is the call graph for this function:

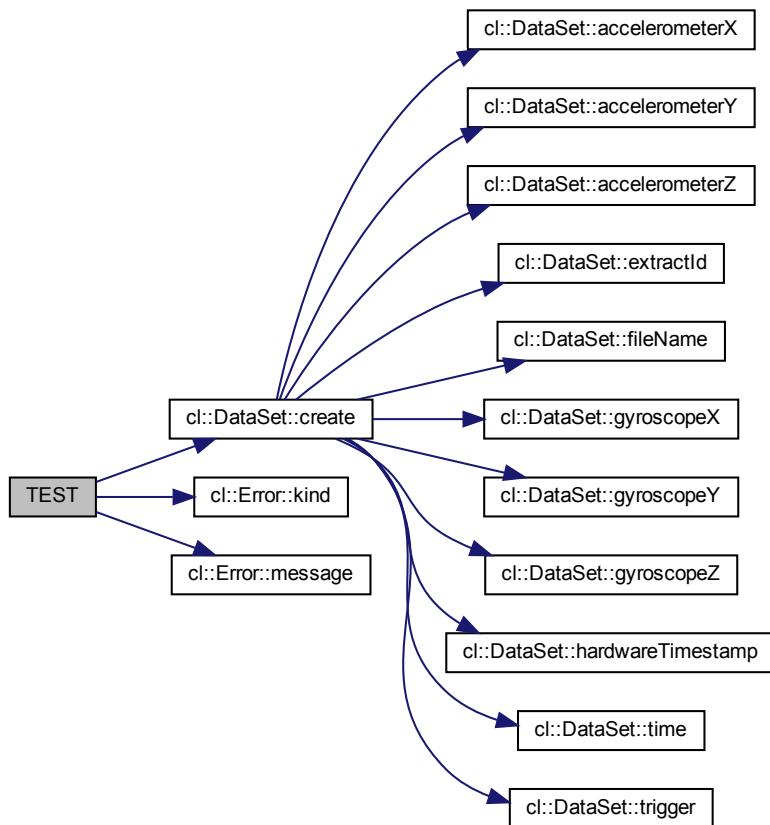


7.83.2.2 TEST() [2/4]

```
TEST (
    DataSet ,
    shouldNotBeAbleToCreateFromEmptyMatrix )
```

Definition at line 68 of file `data_set_test.cpp`.

Here is the call graph for this function:

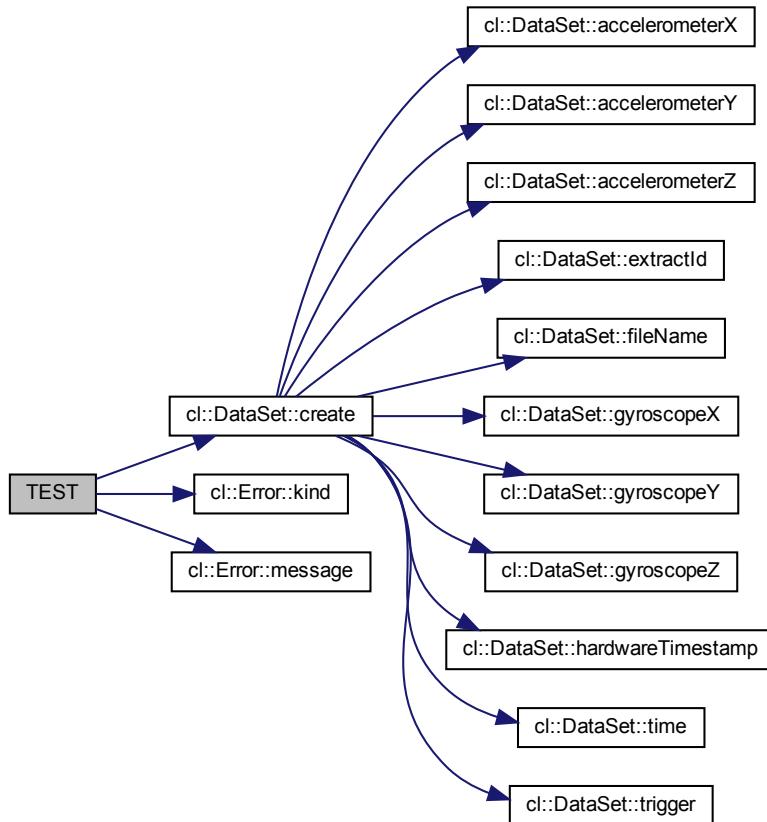


7.83.2.3 TEST() [3/4]

```
TEST (
    DataSet ,
    shouldNotBeAbleToCreateFromInvalidData )
```

Definition at line 108 of file `data_set_test.cpp`.

Here is the call graph for this function:

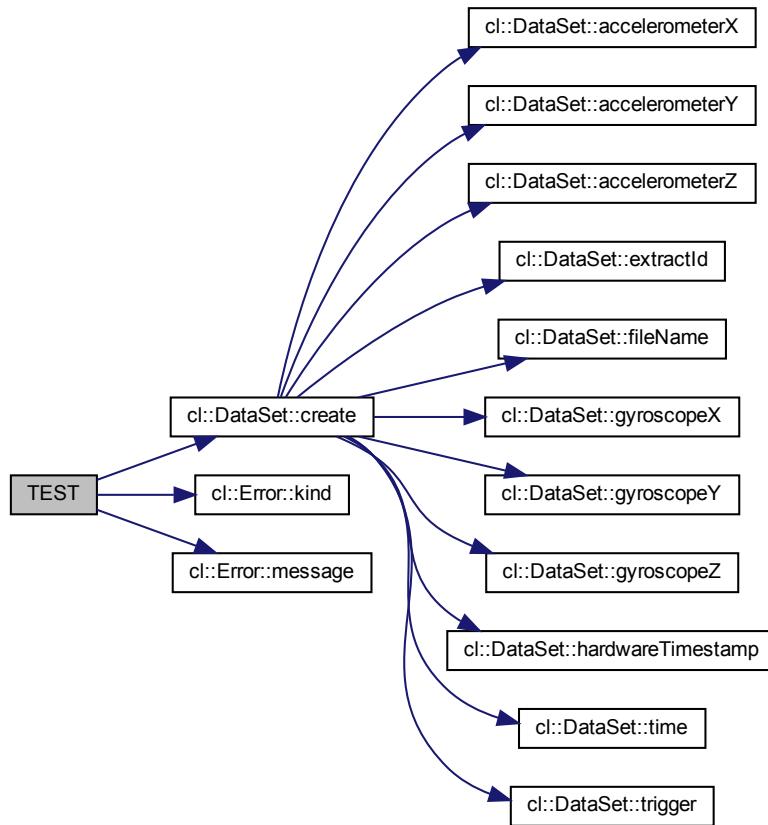


7.83.2.4 TEST() [4/4]

```
TEST (
    DataSet ,
    shouldNotBeAbleToCreateFromJaggedMatrix )
```

Definition at line 80 of file `data_set_test.cpp`.

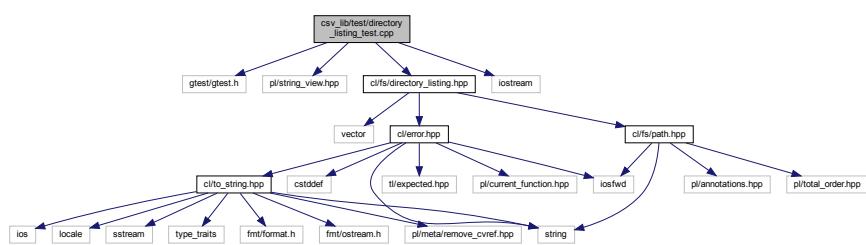
Here is the call graph for this function:



7.84 csv_lib/test/directory_listing_test.cpp File Reference

```

#include "gtest/gtest.h"
#include <pl/string_view.hpp>
#include <cl/fs/directory_listing.hpp>
#include <iostream>
Include dependency graph for directory_listing_test.cpp:
  
```



Functions

- `TEST` (`directoryListing`, `shouldFindFiles`)
- `TEST` (`directoryListing`, `shouldFindFilesWithDotAndDotDot`)
- `TEST` (`directoryListing`, `shouldReturnErrorWhenPathDoesNotExist`)

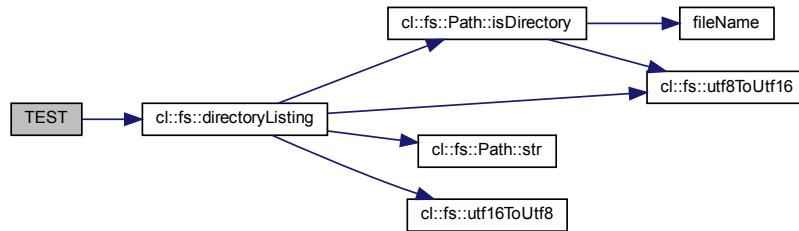
7.84.1 Function Documentation

7.84.1.1 TEST() [1/3]

```
TEST (
    directoryListing ,
    shouldFindFiles )
```

Definition at line 13 of file `directory_listing_test.cpp`.

Here is the call graph for this function:

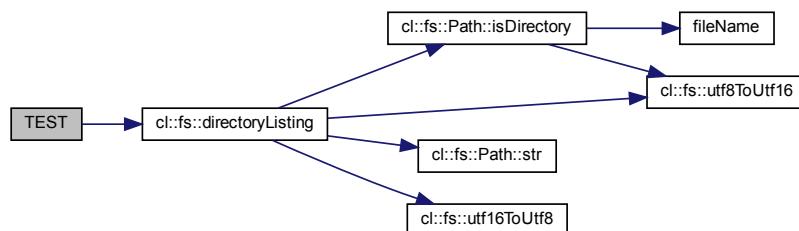


7.84.1.2 TEST() [2/3]

```
TEST (
    directoryListing ,
    shouldFindFilesWithDotAndDotDot )
```

Definition at line 28 of file `directory_listing_test.cpp`.

Here is the call graph for this function:

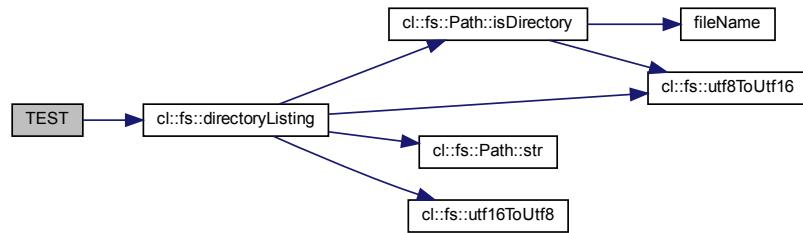


7.84.1.3 TEST() [3/3]

```
TEST ( directoryListing ,  
      shouldReturnErrorWhenPathDoesNotExist )
```

Definition at line 46 of file directory_listing_test.cpp.

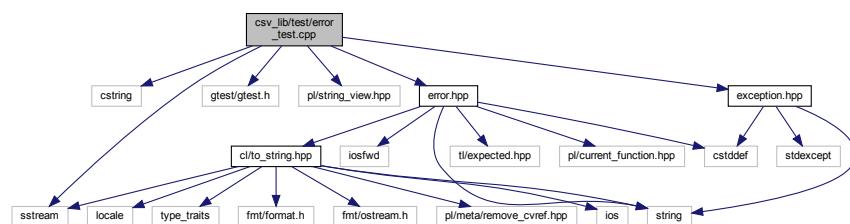
Here is the call graph for this function:



7.85 csv_lib/test/error_test.cpp File Reference

```
#include <cstring>
#include <sstream>
#include "gtest/gtest.h"
#include <pl/string_view.hpp>
#include "error.hpp"
#include "exception.hpp"
Include dependency graph for error_test.cpp:
```

Include dependency graph for error_test.cpp:



Functions

- **TEST** (`error`, `shouldPrint`)
 - **TEST** (`error`, `shouldReturnValues`)
 - **TEST** (`error`, `shouldThrowExceptionWhenRaisesCalled`)
 - **TEST** (`error`, `shouldCreateExpectedWithUnexpected`)

Variables

- const `cl::Error error`

7.85.1 Function Documentation

7.85.1.1 TEST() [1/4]

```
TEST (
    error ,
    shouldCreateExpectedWithUnexpected )
```

Definition at line 59 of file `error_test.cpp`.

7.85.1.2 TEST() [2/4]

```
TEST (
    error ,
    shouldPrint )
```

Definition at line 19 of file `error_test.cpp`.

7.85.1.3 TEST() [3/4]

```
TEST (
    error ,
    shouldReturnValues )
```

Definition at line 29 of file `error_test.cpp`.

7.85.1.4 TEST() [4/4]

```
TEST (
    error ,
    shouldThrowExceptionWhenRaiseIsCalled )
```

Definition at line 37 of file `error_test.cpp`.

7.85.2 Variable Documentation

7.85.2.1 error

```
const cl::Error error
```

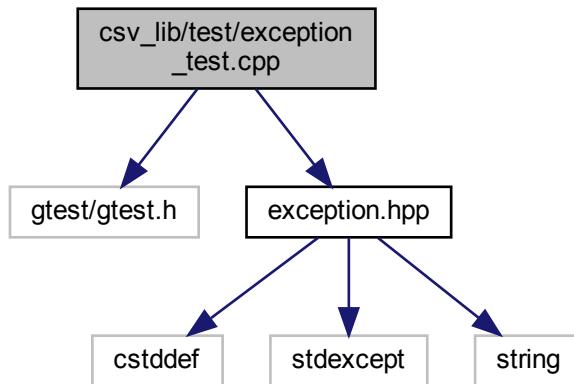
Initial value:

```
{
    cl::Error::Filesystem,
    "test_file.cpp",
    "bad_function",
    48,
    "Couldn't initialize the flux capacitor."}
```

Definition at line 12 of file error_test.cpp.

7.86 csv_lib/test/exception_test.cpp File Reference

```
#include "gtest/gtest.h"
#include "exception.hpp"
Include dependency graph for exception_test.cpp:
```



Functions

- **TEST** (exception, shouldWork)

7.86.1 Function Documentation

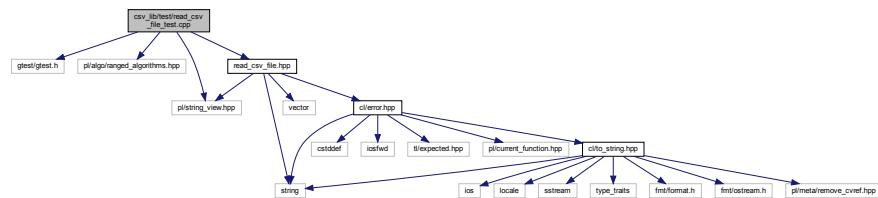
7.86.1.1 TEST()

```
TEST (
    exception ,
    shouldWork )
```

Definition at line 5 of file exception_test.cpp.

7.87 csv_lib/test/read_csv_file_test.cpp File Reference

```
#include "gtest/gtest.h"
#include <pl/algo/ranged_algorithms.hpp>
#include <pl/string_view.hpp>
#include "read_csv_file.hpp"
Include dependency graph for read_csv_file_test.cpp:
```



Functions

- [TEST](#) (`readCsvFile`, `shouldReadCsvFile`)
- [TEST](#) (`readCsvFile`, `shouldNotReadNonexistantCsvFile`)

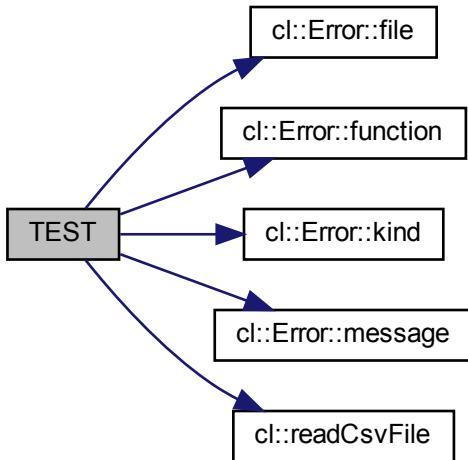
7.87.1 Function Documentation

7.87.1.1 TEST() [1/2]

```
TEST (
    readCsvFile ,
    shouldNotReadNonexistantCsvFile )
```

Definition at line 30 of file `read_csv_file_test.cpp`.

Here is the call graph for this function:

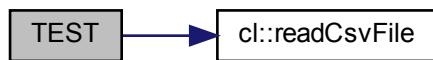


7.87.1.2 TEST() [2/2]

```
TEST (
    readCsvFile ,
    shouldReadCsvFile )
```

Definition at line 8 of file read_csv_file_test.cpp.

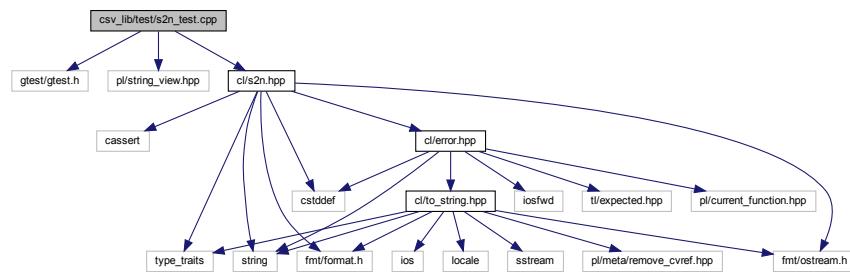
Here is the call graph for this function:



7.88 csv_lib/test/s2n_test.cpp File Reference

```
#include "gtest/gtest.h"
#include <pl/string_view.hpp>
```

```
#include "cl/s2n.hpp"
Include dependency graph for s2n_test.cpp:
```



Functions

- `TEST` (`s2n`, `shouldWork`)
- `TEST` (`s2n`, `shouldReturnInvalidArgumentErrorIfInputIsInvalid`)
- `TEST` (`s2n`, `shouldReturnOutOfRangeErrorIfInputIsOutOfRange`)

7.88.1 Function Documentation

7.88.1.1 TEST() [1/3]

```
TEST (
    s2n ,
    shouldReturnInvalidArgumentErrorIfInputIsInvalid )
```

Definition at line 21 of file `s2n_test.cpp`.

7.88.1.2 TEST() [2/3]

```
TEST (
    s2n ,
    shouldReturnOutOfRangeErrorIfInputIsOutOfRange )
```

Definition at line 29 of file `s2n_test.cpp`.

7.88.1.3 TEST() [3/3]

```
TEST (
    s2n ,
    shouldWork )
```

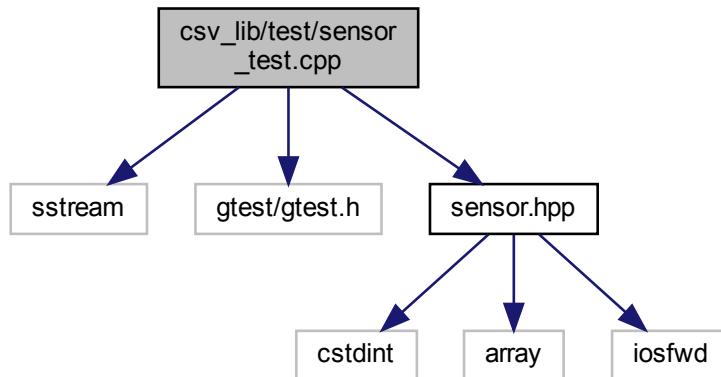
Definition at line 7 of file s2n_test.cpp.

Here is the call graph for this function:



7.89 csv_lib/test/sensor_test.cpp File Reference

```
#include <sstream>
#include "gtest/gtest.h"
#include "sensor.hpp"
Include dependency graph for sensor_test.cpp:
```



Functions

- [TEST \(`sensor`, `shouldHaveCorrectValues`\)](#)
- [TEST \(`sensor`, `shouldPrintCorrely`\)](#)

7.89.1 Function Documentation

7.89.1.1 TEST() [1/2]

```
TEST (
    sensor ,
    shouldHaveCorrectValues )
```

Definition at line 7 of file sensor_test.cpp.

7.89.1.2 TEST() [2/2]

```
TEST (
    sensor ,
    shouldPrintCorrectly )
```

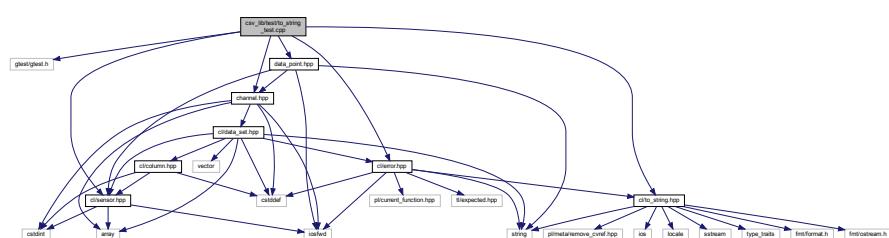
Definition at line 15 of file sensor_test.cpp.

Here is the call graph for this function:



7.90 csv_lib/test/to_string_test.cpp File Reference

```
#include "gtest/gtest.h"
#include "channel.hpp"
#include "data_point.hpp"
#include "error.hpp"
#include "sensor.hpp"
#include "to_string.hpp"
Include dependency graph for to_string_test.cpp:
```



Functions

- [TEST](#) (to_string, test)

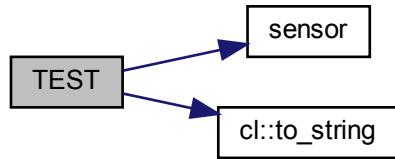
7.90.1 Function Documentation

7.90.1.1 TEST()

```
TEST (
    to_string ,
    test )
```

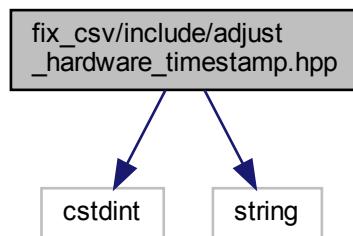
Definition at line 9 of file to_string_test.cpp.

Here is the call graph for this function:

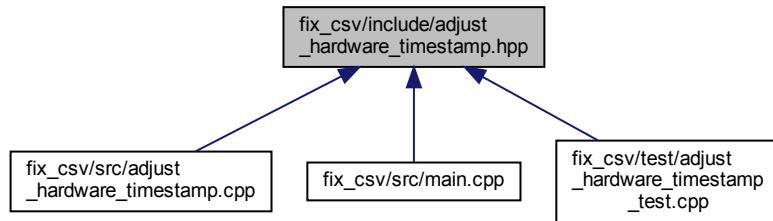


7.91 fix_csv/include/adjust_hardware_timestamp.hpp File Reference

```
#include <cstdint>
#include <string>
Include dependency graph for adjust_hardware_timestamp.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

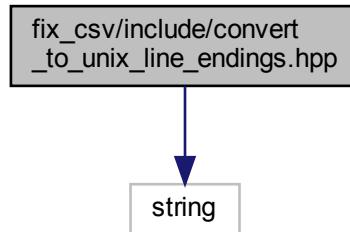
- `fmc`

Functions

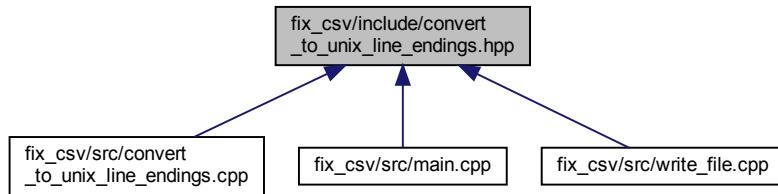
- void `fmc::adjustHardwareTimestamp` (`std::string *cellContent, const std::string &nextRowHardwareTimestamp, std::uint64_t *overflowCount)`

7.92 fix_csv/include/convert_to_unix_line_endings.hpp File Reference

```
#include <string>
Include dependency graph for convert_to_unix_line_endings.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

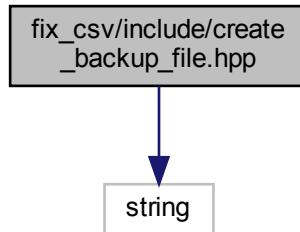
- `fmc`

Functions

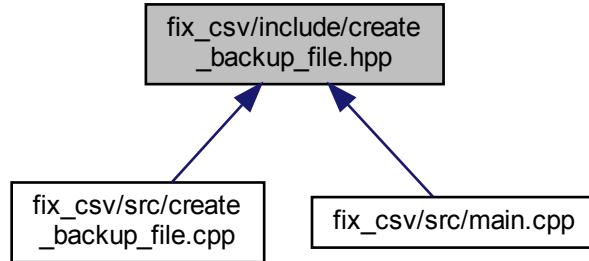
- `bool fmc::convertToUnixLineEndings (const std::string &csvPath)`

7.93 fix_csv/include/create_backup_file.hpp File Reference

```
#include <string>
Include dependency graph for create_backup_file.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

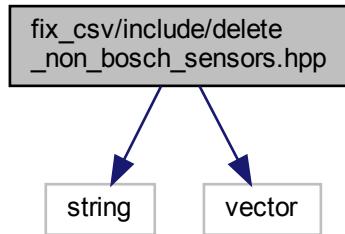
- [fmc](#)

Functions

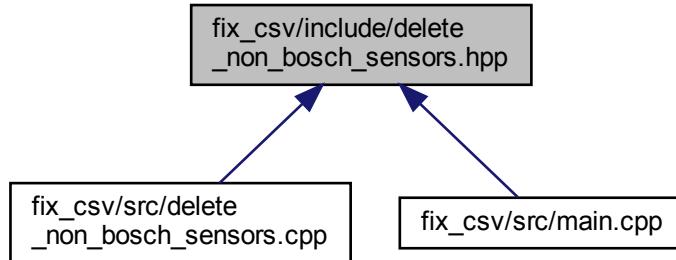
- `bool fmc::createBackupFile (const std::string &csvFilePath, const std::string &backupFilePath)`

7.94 fix_csv/include/delete_non_bosch_sensors.hpp File Reference

```
#include <string>
#include <vector>
Include dependency graph for delete_non_bosch_sensors.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

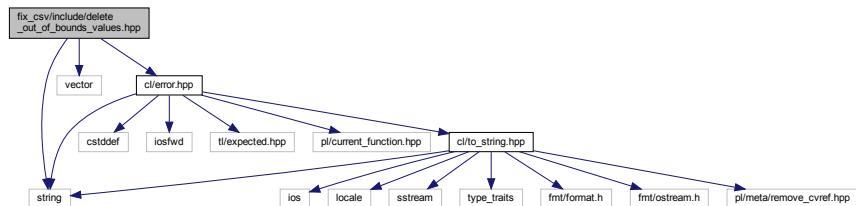
- [fmc](#)

Functions

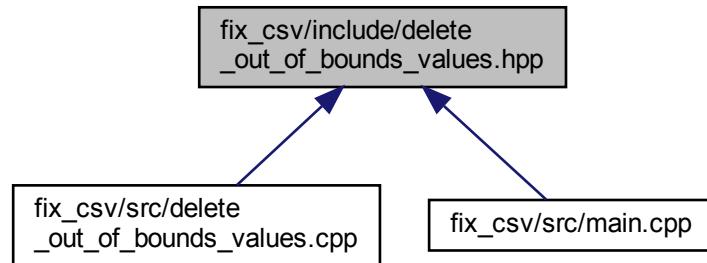
- void [fmc::deleteNonBoschSensors](#) (std::vector< std::vector< std::string >> *data)

7.95 fix_csv/include/delete_out_of_bounds_values.hpp File Reference

```
#include <string>
#include <vector>
#include "cl/error.hpp"
Include dependency graph for delete_out_of_bounds_values.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

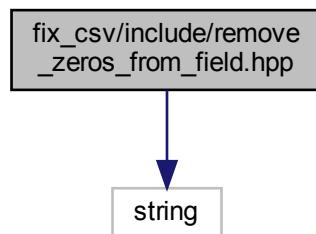
- [fmc](#)

Functions

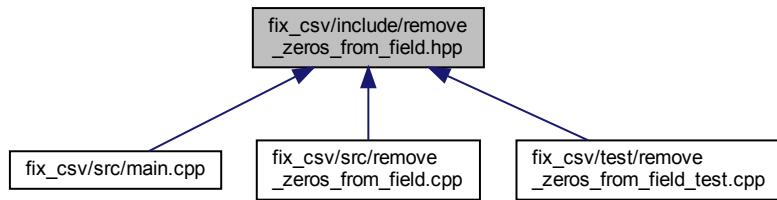
- [cl::Expected< void > fmc::deleteOutOfBoundsValues \(std::vector< std::vector< std::string >> *data\)](#)

7.96 fix_csv/include/remove_zeros_from_field.hpp File Reference

```
#include <string>
Include dependency graph for remove_zeros_from_field.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

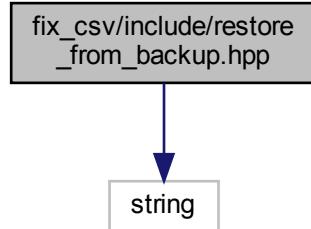
- `fmc`

Functions

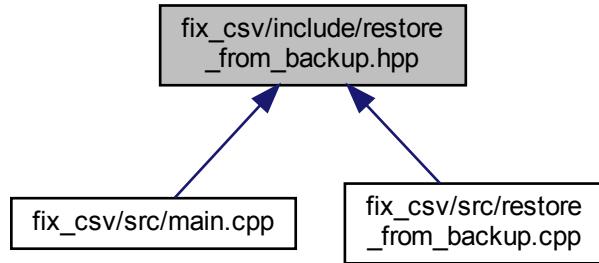
- void `fmc::removeZerosFromField` (`std::string *field`)

7.97 fix_csv/include/restore_from_backup.hpp File Reference

```
#include <string>
Include dependency graph for restore_from_backup.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

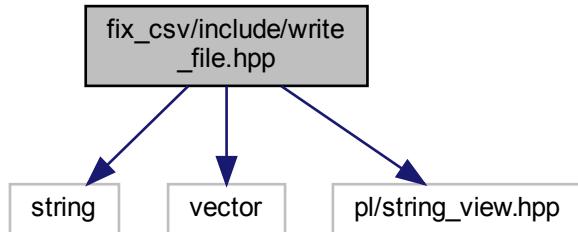
- [fmc](#)

Functions

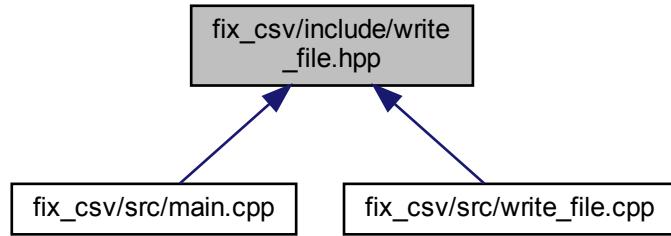
- bool [fmc::restoreFromBackup](#) (const std::string &csvFilePath, const std::string &backupFilePath)

7.98 fix_csv/include/write_file.hpp File Reference

```
#include <string>
#include <vector>
#include <pl/string_view.hpp>
Include dependency graph for write_file.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

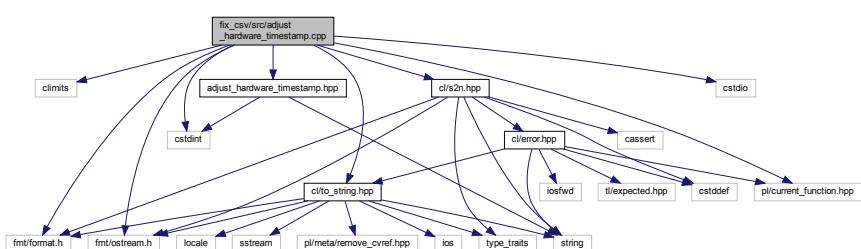
- `fmc`

Functions

- `bool fmc::writeFile (pl::string_view csvPath, pl::string_view csvFileExtension, const std::vector< std::string > &columnNames, const std::vector< std::vector< std::string >> &data)`

7.99 fix_csv/src/adjust_hw timestamp.cpp File Reference

```
#include <climits>
#include <cstdint>
#include <cstdio>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include <pl/current_function.hpp>
#include "cl/s2n.hpp"
#include "cl/to_string.hpp"
#include "adjust_hw timestamp.hpp"
Include dependency graph for adjust_hw timestamp.hpp:
```



Namespaces

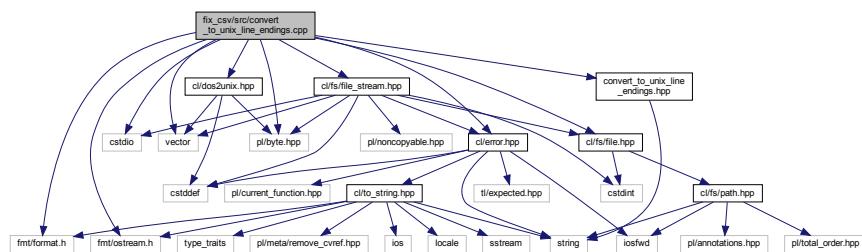
- [fmc](#)

Functions

- void [fmc::adjustHardwareTimestamp](#) (std::string *cellContent, const std::string &nextRowHardwareTimestamp, std::uint64_t *overflowCount)

7.100 fix_csv/src/convert_to_unix_line_endings.cpp File Reference

```
#include <cstdio>
#include <vector>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include <pl/byte.hpp>
#include "cl/dos2unix.hpp"
#include "cl/error.hpp"
#include "cl/fs/file.hpp"
#include "cl/fs/file_stream.hpp"
#include "convert_to_unix_line_endings.hpp"
Include dependency graph for convert_to_unix_line_endings.cpp:
```



Namespaces

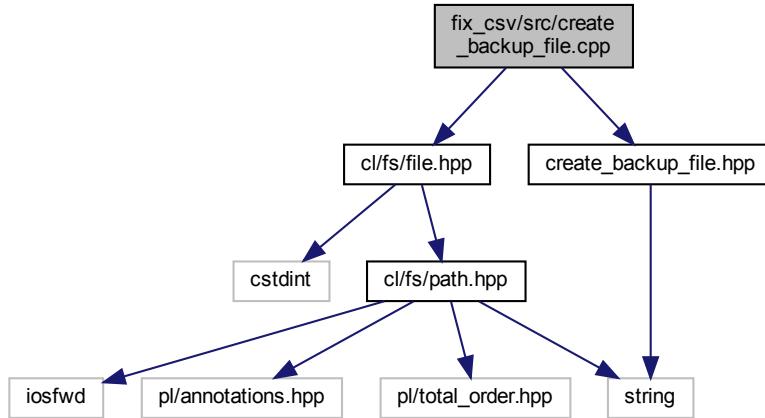
- [fmc](#)

Functions

- bool [fmc::convertToUnixLineEndings](#) (const std::string &csvPath)

7.101 fix_csv/src/create_backup_file.cpp File Reference

```
#include "cl/fs/file.hpp"
#include "create_backup_file.hpp"
Include dependency graph for create_backup_file.cpp:
```



Namespaces

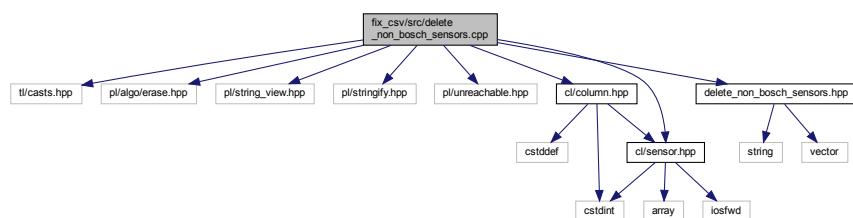
- `fmc`

Functions

- `bool fmc::createBackupFile (const std::string &csvFilePath, const std::string &backupFilePath)`

7.102 fix_csv/src/delete_non_bosch_sensors.cpp File Reference

```
#include <t1/casts.hpp>
#include <pl/algo/erase.hpp>
#include <pl/string_view.hpp>
#include <pl/stringify.hpp>
#include <pl/unreachable.hpp>
#include "cl/column.hpp"
#include "cl/sensor.hpp"
#include "delete_non_bosch_sensors.hpp"
Include dependency graph for delete_non_bosch_sensors.cpp:
```



Namespaces

- fmc

Macros

- ```
• #define CL_SENSOR_X(enm, value) case cl::Sensor::enm: return PL_STRINGIFY(value);
```

# Functions

- void **fmc::deleteNonBoschSensors** (std::vector< std::vector< std::string >> \*data)

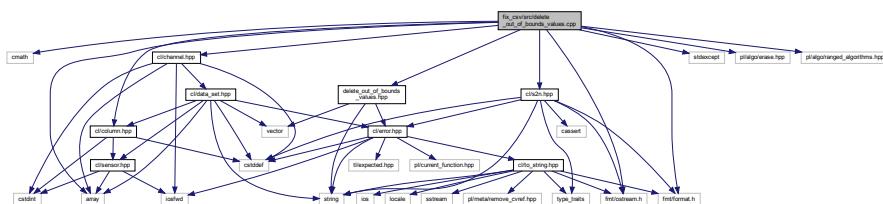
### **7.102.1 Macro Definition Documentation**

#### **7.102.1.1 CL\_SENSOR\_X**

```
#define CL_SENSOR_X(enm, value) case cl::Sensor::enm: return PL_STRINGIFY(value);
```

## 7.103 fix\_csv/src/delete\_out\_of\_bounds\_values.cpp File Reference

```
#include <cmath>
#include <array>
#include <stdexcept>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include <pl/algo/erase.hpp>
#include <pl/algo/ranged_algorithms.hpp>
#include "cl/channel.hpp"
#include "cl/column.hpp"
#include "cl/s2n.hpp"
#include "delete_out_of_bounds_values.hpp"
Include dependency graph for delete_out_of_bounds_values.cpp
```



## Namespaces

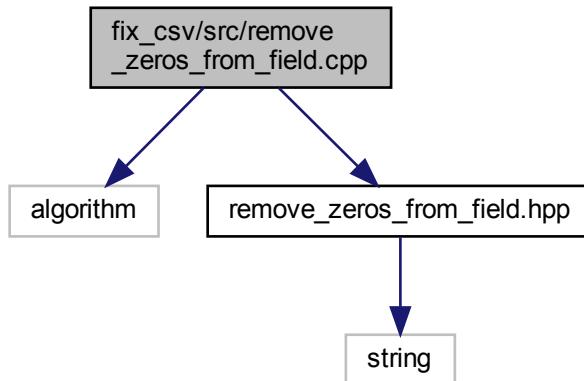
- fmc

## Functions

- `cl::Expected< void > fmc::deleteOutOfBoundsValues (std::vector< std::vector< std::string >> *data)`

## 7.104 fix\_csv/src/remove\_zeros\_from\_field.cpp File Reference

```
#include <algorithm>
#include "remove_zeros_from_field.hpp"
Include dependency graph for remove_zeros_from_field.cpp:
```



## Namespaces

- `fmc`

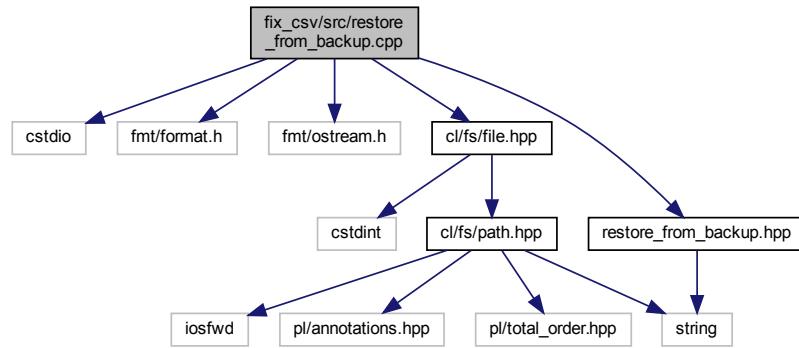
## Functions

- `void fmc::removeZerosFromField (std::string *field)`

## 7.105 fix\_csv/src/restore\_from\_backup.cpp File Reference

```
#include <cstdio>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include "cl/fs/file.hpp"
```

```
#include "restore_from_backup.hpp"
Include dependency graph for restore_from_backup.cpp:
```



## Namespaces

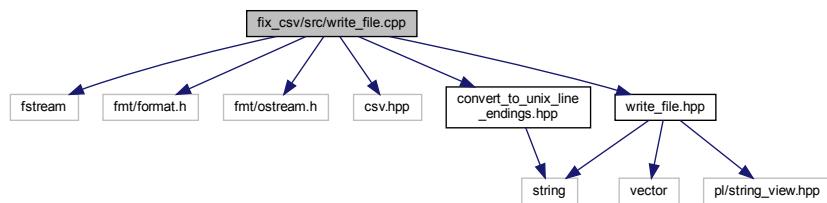
- [fmc](#)

## Functions

- bool [fmc::restoreFromBackup](#) (const std::string &csvFilePath, const std::string &backupFilePath)

## 7.106 fix\_csv/src/write\_file.cpp File Reference

```
#include <fstream>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include <csv.hpp>
#include "convert_to_unix_line_endings.hpp"
#include "write_file.hpp"
Include dependency graph for write_file.cpp:
```



## Namespaces

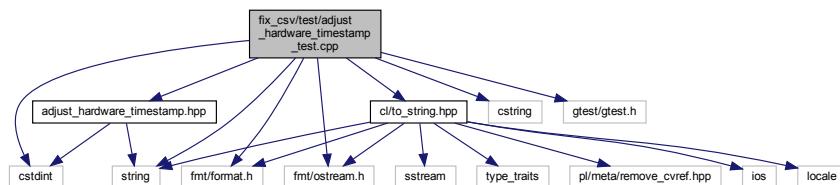
- [fmc](#)

## Functions

- bool `fmc::writeFile` (pl::string\_view csvPath, pl::string\_view csvFileExtension, const std::vector< std::string > &columnNames, const std::vector< std::vector< std::string >> &data)

## 7.107 fix\_csv/test/adjust\_hw\_timestamp\_test.cpp File Reference

```
#include <cstdint>
#include <cstring>
#include <string>
#include <fmt/format.h>
#include <fmt/ostream.h>
#include "gtest/gtest.h"
#include "cl/to_string.hpp"
#include "adjust_hw_timestamp.hpp"
Include dependency graph for adjust_hw_timestamp_test.cpp:
```



## Functions

- `TEST` (`adjustHardwareTimestamp`, `shouldDoNothingForNonOverflowedValue`)
- `TEST` (`adjustHardwareTimestamp`, `shouldIncrementOverflowCount`)
- `TEST` (`adjustHardwareTimestamp`, `shouldWorkForOneRoundOfOverflow`)
- `TEST` (`adjustHardwareTimestamp`, `shouldWorkForTwoRoundsOfOverflow`)
- `TEST` (`adjustHardwareTimestamp`, `shouldWork`)

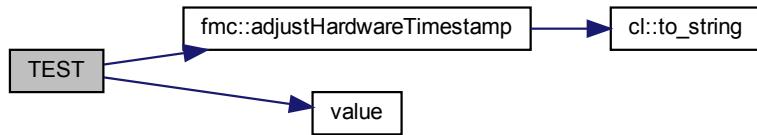
### 7.107.1 Function Documentation

#### 7.107.1.1 TEST() [1/5]

```
TEST (
 adjustHardwareTimestamp ,
 shouldDoNothingForNonOverflowedValue)
```

Definition at line 15 of file `adjust_hw_timestamp_test.cpp`.

Here is the call graph for this function:



### 7.107.1.2 TEST() [2/5]

```
TEST (
 adjustHardwareTimestamp ,
 shouldIncrementOverflowCount)
```

Definition at line 26 of file `adjust_hardware_timestamp_test.cpp`.

Here is the call graph for this function:

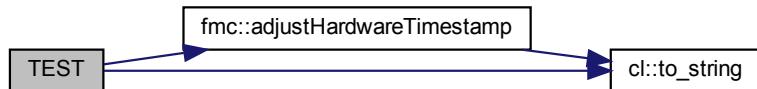


### 7.107.1.3 TEST() [3/5]

```
TEST (
 adjustHardwareTimestamp ,
 shouldWork)
```

Definition at line 132 of file `adjust_hardware_timestamp_test.cpp`.

Here is the call graph for this function:



**7.107.1.4 TEST() [4/5]**

```
TEST (
 adjustHardwareTimestamp ,
 shouldWorkForOneRoundOfOverflow)
```

Definition at line 48 of file adjust\_hardware\_timestamp\_test.cpp.

Here is the call graph for this function:

**7.107.1.5 TEST() [5/5]**

```
TEST (
 adjustHardwareTimestamp ,
 shouldWorkForTwoRoundsOfOverflow)
```

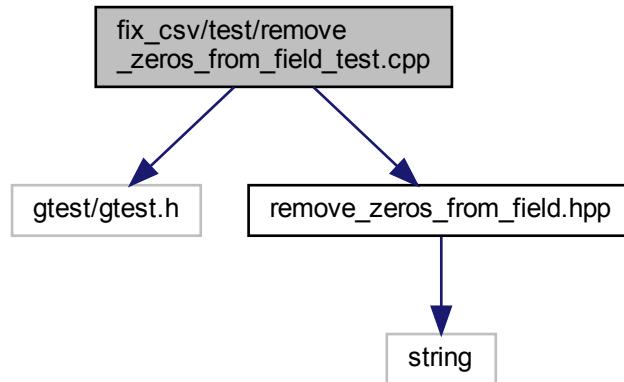
Definition at line 96 of file adjust\_hardware\_timestamp\_test.cpp.

Here is the call graph for this function:

**7.108 fix\_csv/test/remove\_zeros\_from\_field\_test.cpp File Reference**

```
#include "gtest/gtest.h"
#include "remove_zeros_from_field.hpp"
```

Include dependency graph for remove\_zeros\_from\_field\_test.cpp:



## Functions

- `TEST (removeZerosFromField, shouldRemoveDotAndZeros)`
- `TEST (removeZerosFromField, shouldNotRemovelfNonZerosFollow)`
- `TEST (removeZerosFromField, shouldNotRemovelfNoDot)`
- `TEST (removeZerosFromField, shouldDoNothingIfStringIsEmpty)`
- `TEST (removeZerosFromField, shouldDeleteStringIfStringIsSingleDot)`
- `TEST (removeZerosFromField, shouldDeleteStringIfStringIsDotAndZero)`

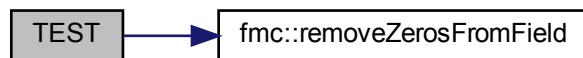
### 7.108.1 Function Documentation

#### 7.108.1.1 TEST() [1/6]

```
TEST (
 removeZerosFromField ,
 shouldDeleteStringIfStringIsDotAndZero)
```

Definition at line 53 of file `remove_zeros_from_field_test.cpp`.

Here is the call graph for this function:



**7.108.1.2 TEST() [2/6]**

```
TEST (
 removeZerosFromField ,
 shouldDeleteStringIfStringIsSingleDot)
```

Definition at line 44 of file remove\_zeros\_from\_field\_test.cpp.

Here is the call graph for this function:

**7.108.1.3 TEST() [3/6]**

```
TEST (
 removeZerosFromField ,
 shouldDoNothingIfStringIsEmpty)
```

Definition at line 35 of file remove\_zeros\_from\_field\_test.cpp.

Here is the call graph for this function:

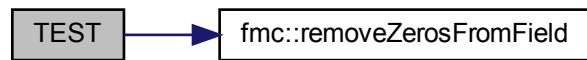


#### 7.108.1.4 TEST() [4/6]

```
TEST (
 removeZerosFromField ,
 shouldNotRemoveIfNoDot)
```

Definition at line 25 of file remove\_zeros\_from\_field\_test.cpp.

Here is the call graph for this function:



#### 7.108.1.5 TEST() [5/6]

```
TEST (
 removeZerosFromField ,
 shouldNotRemoveIfNonZerosFollow)
```

Definition at line 15 of file remove\_zeros\_from\_field\_test.cpp.

Here is the call graph for this function:



### 7.108.1.6 TEST() [6/6]

```
TEST (
 removeZerosFromField ,
 shouldRemoveDotAndZeros)
```

Definition at line 5 of file remove\_zeros\_from\_field\_test.cpp.

Here is the call graph for this function:





# Index

~FileStream  
    cl::fs::FileStream, 85

above\_threshold.cpp  
    channel, 146  
    channelAccessor, 147  
    CL\_CHANNEL\_X, 146

above\_threshold\_test.cpp  
    EXPECT\_LONG\_DOUBLE\_EQ, 149  
    TEST, 150

aboveThreshold  
    ctg, 35

accelerometerAverage  
    cl::DataSet, 62

accelerometerMaximum  
    cl::DataSet, 62

accelerometerThreshold  
    cl, 22

AccelerometerX  
    cl, 13

accelerometerX  
    cl::DataSet, 63

AccelerometerY  
    cl, 13

accelerometerY  
    cl::DataSet, 63

AccelerometerZ  
    cl, 13

accelerometerZ  
    cl::DataSet, 64

adjust\_hardware\_timestamp\_test.cpp  
    TEST, 219–221

adjustHardwareTimestamp  
    fmc, 41

averageComparisonValueCalculator  
    ctg, 36

base\_type  
    cl::Exception, 76

Both  
    cs, 29

build  
    cs::CsvLineBuilder, 48

Butterworth  
    cs, 28

Channel  
    cl, 13

channel  
    above\_threshold.cpp, 146

    cl::DataPoint, 58  
    data\_point.cpp, 175

channel.cpp  
    CL\_CHANNEL\_X, 174

channel.hpp  
    CL\_CHANNEL, 153  
    CL\_CHANNEL\_X, 154

channel\_test.cpp  
    TEST, 186, 187

ChannelAccessor  
    cl::DataSet, 62

channelAccessor  
    above\_threshold.cpp, 147

channelCount  
    cl, 22

channels  
    cl, 22

cl, 11  
    accelerometerThreshold, 22  
    AccelerometerX, 13  
    AccelerometerY, 13  
    AccelerometerZ, 13  
    Channel, 13  
    channelCount, 22  
    channels, 22  
    CL\_CHANNEL, 13  
    CL\_CHANNEL\_X, 13  
    CL\_SENSOR, 14  
    CL\_SENSOR\_X, 14  
    CL\_SPECIALIZE\_COL\_TRAITS, 14–16  
    Column, 13  
    column\_index, 22  
    column\_type, 12  
    CsvFileKind, 13  
    data\_set\_accessor\_v, 23  
    dataSetAccessor, 16  
    dos2unix, 16  
    Expected, 12  
    ExtractId, 13  
    Fixed, 14  
    gyroscopeThreshold, 23  
    GyroscopeX, 13  
    GyroscopeY, 13  
    GyroscopeZ, 13  
    HardwareTimestamp, 13  
    isAccelerometer, 17  
    isGyroscope, 17  
    operator<<, 18, 19  
    Raw, 14

readCsvFile, 19  
 s2n, 20  
 SamplingRate, 13  
 Sensor, 14  
 sensors, 23  
 threshold, 20  
 Time, 13  
 to\_string, 21  
 Trigger, 13  
 useUnbufferedIo, 21  
 cl::col\_traits< Col >, 47  
 cl::data\_set\_accessor< Chan >, 56  
 cl::DataPoint, 57  
     channel, 58  
     DataPoint, 57  
     fileName, 58  
     operator<<, 60  
     sensor, 59  
     time, 59  
     value, 60  
 cl::DataSet, 61  
     accelerometerAverage, 62  
     accelerometerMaximum, 62  
     accelerometerX, 63  
     accelerometerY, 63  
     accelerometerZ, 64  
     ChannelAccessor, 62  
     create, 64  
     extractId, 66  
     fileName, 66  
     gyroscopeAverage, 67  
     gyroscopeMaximum, 67  
     gyroscopeX, 68  
     gyroscopeY, 68  
     gyroscopeZ, 69  
     hardwareTimestamp, 69  
     rowCount, 70  
     size\_type, 62  
     time, 70  
     trigger, 71  
 cl::Error, 71  
     CL\_ERROR\_KIND, 72  
     Error, 72  
     file, 73  
     function, 73  
     Kind, 72  
     kind, 73  
     line, 74  
     message, 74  
     operator<<, 75  
     raise, 74  
     to\_string, 75  
 cl::Exception, 75  
     base\_type, 76  
     Exception, 77  
     file, 77  
     function, 77  
     line, 77  
 cl::fs, 23  
     directoryListing, 24  
     DirectoryListingOption, 24  
     ExcludeDotAndDotDot, 24  
     formatError, 25  
     None, 24  
     operator<, 25  
     operator<<, 25  
     operator==, 26  
     utf16ToUtf8, 26  
     utf8ToUtf16, 26  
 cl::fs::File, 78  
     copyTo, 79  
     create, 80  
     exists, 80  
     File, 78  
     moveTo, 81  
     path, 82  
     remove, 82  
     size, 83  
 cl::fs::FileStream, 84  
     ~FileStream, 85  
     create, 86  
     FileStream, 85  
     OpenMode, 85  
     operator=, 86  
     PL\_NONCOPYABLE, 87  
     Read, 85  
     readAll, 87  
     ReadWrite, 85  
     this\_type, 85  
     Write, 85  
     write, 87  
 cl::fs::Path, 94  
     exists, 94  
     isDirectory, 95  
     isFile, 95  
     operator<, 97  
     operator<<, 97  
     operator==, 97  
     Path, 94  
     str, 96  
 CL\_CHANNEL  
     channel.hpp, 153  
     cl, 13  
 CL\_CHANNEL\_X  
     above\_threshold.cpp, 146  
     channel.cpp, 174  
     channel.hpp, 154  
     cl, 13  
 CL\_ERROR\_KIND  
     cl::Error, 72  
     error.hpp, 160  
 CL\_ERROR\_KIND\_X  
     error.cpp, 179  
     error.hpp, 161  
 CL\_FS\_SEPARATOR  
     separator.hpp, 166

CL\_SENSOR  
  cl, 14  
  sensor.hpp, 171

CL\_SENSOR\_X  
  cl, 14  
  delete\_non\_bosch\_sensors.cpp, 216  
  sensor.cpp, 185  
  sensor.hpp, 171

CL\_SPECIALIZE\_COL\_TRAITS  
  cl, 14–16  
  column.hpp, 156

CL\_UNEXPECTED  
  error.hpp, 161

CMakeLists.txt  
  include, 99–102  
  set, 99–102

Column  
  cl, 13

column.hpp  
  CL\_SPECIALIZE\_COL\_TRAITS, 156

column\_index  
  cl, 22

column\_test.cpp  
  TEST, 188

column\_type  
  cl, 12

compare\_segmentation/CMakeLists.txt, 99

compare\_segmentation/include/csv\_line.hpp, 103

compare\_segmentation/include/data\_set\_info.hpp, 103

compare\_segmentation/include/filter\_kind.hpp, 105

compare\_segmentation/include/log\_files.hpp, 106

compare\_segmentation/include/log\_info.hpp, 107

compare\_segmentation/include/log\_line.hpp, 108

compare\_segmentation/include/paths.hpp, 109

compare\_segmentation/include/segmentation\_kind.hpp, 110

compare\_segmentation/src/csv\_line.cpp, 111

compare\_segmentation/src/data\_set\_info.cpp, 111

compare\_segmentation/src/filter\_kind.cpp, 112

compare\_segmentation/src/log\_files.cpp, 112

compare\_segmentation/src/log\_info.cpp, 113

compare\_segmentation/src/log\_line.cpp, 114

compare\_segmentation/src/main.cpp, 114

compare\_segmentation/src/segmentation\_kind.cpp, 123

compare\_segmentation/test/CMakeLists.txt, 99

compare\_segmentation/test/csv\_line\_test.cpp, 124

compare\_segmentation/test/data\_set\_info\_test.cpp, 124

compare\_segmentation/test/log\_files\_test.cpp, 126

compare\_segmentation/test/log\_info\_test.cpp, 127

compare\_segmentation/test/log\_line\_test.cpp, 137

compare\_segmentation/test/main.cpp, 116

convertToUnixLineEndings  
  fmc, 41

copyTo  
  cl::fs::File, 79

counting/CMakeLists.txt, 100

  counting/include/above\_threshold.hpp, 139  
  counting/include/average\_comparison\_value\_calculator.hpp, 140  
  counting/include/half\_maximum\_comparison\_value\_calculator.hpp, 141  
  counting/include/is\_relevant.hpp, 142  
  counting/include/percentage\_of.hpp, 143  
  counting/include/run\_above\_threshold.hpp, 144  
  counting/src/above\_threshold.cpp, 145  
  counting/src/average\_comparison\_value\_calculator.cpp, 147  
  counting/src/half\_maximum\_comparison\_value\_calculator.cpp, 148  
  counting/src/main.cpp, 117  
  counting/src/run\_above\_threshold.cpp, 148  
  counting/test/above\_threshold\_test.cpp, 149  
  counting/test/CMakeLists.txt, 100  
  counting/test/main.cpp, 119  
  counting/test/percentage\_of\_test.cpp, 151

create  
  cl::DataSet, 64  
  cl::fs::File, 80  
  cl::fs::FileStream, 86  
  cs::LogInfo, 88

createBackupFile  
  fmc, 42

cs, 27  
  Both, 29  
  Butterworth, 28  
  CS\_SPECIALIZE\_DATA\_SET\_INFO, 29–32  
  FilterKind, 28  
  logFiles, 32  
  logPath, 35  
  Maxima, 29  
  Minima, 29  
  MovingAverage, 28  
  oldLogPath, 35  
  operator!=, 33  
  operator<<, 33, 34  
  operator==, 34  
  PL\_DEFINE\_EXCEPTION\_TYPE, 34  
  repetitionCount, 34  
  SegmentationKind, 28

cs::CsvLineBuilder, 47  
  build, 48  
  CsvLineBuilder, 48  
  dataSet, 49  
  deleteLowVariance, 49  
  deleteTooClose, 50  
  filter, 51  
  isOld, 51  
  kind, 52  
  pushUps, 53  
  segmentationPoints, 53  
  sensor, 54  
  skipWindow, 55  
  this\_type, 48  
  windowSize, 55

cs::data\_set\_info< Tag >, 57  
 cs::LogInfo, 87  
     create, 88  
     deleteLowVariance, 88  
     deleteTooClose, 89  
     filterKind, 89  
     invalidSensor, 91  
     isInitialized, 89  
     logFilePath, 89  
     LogInfo, 88  
     operator!=, 90  
     operator<<, 90  
     operator==, 91  
     segmentationKind, 89  
     sensor, 90  
     skipWindow, 90  
     windowSize, 90  
 cs::LogLine, 91  
     fileName, 92  
     filePath, 92  
     invalidSensor, 93  
     parse, 92  
     segmentationPointCount, 93  
     sensor, 93  
**CS\_SPECIALIZE\_DATA\_SET\_INFO**  
     cs, 29–32  
         data\_set\_info.hpp, 105  
 csv\_lib/CMakeLists.txt, 101  
 csv\_lib/include/cl/channel.hpp, 152  
 csv\_lib/include/cl/column.hpp, 155  
 csv\_lib/include/cl/data\_point.hpp, 157  
 csv\_lib/include/cl/data\_set.hpp, 158  
 csv\_lib/include/cl/dos2unix.hpp, 159  
 csv\_lib/include/cl/error.hpp, 160  
 csv\_lib/include/cl/exception.hpp, 161  
 csv\_lib/include/cl/fs/directory\_listing.hpp, 162  
 csv\_lib/include/cl/fs/file.hpp, 163  
 csv\_lib/include/cl/fs/file\_stream.hpp, 164  
 csv\_lib/include/cl/fs/path.hpp, 165  
 csv\_lib/include/cl/fs/separator.hpp, 166  
 csv\_lib/include/cl/fs/windows.hpp, 167  
 csv\_lib/include/cl/read\_csv\_file.hpp, 168  
 csv\_lib/include/cl/s2n.hpp, 169  
 csv\_lib/include/cl/sensor.hpp, 169  
 csv\_lib/include/cl/to\_string.hpp, 172  
 csv\_lib/include/cl/use\_unbuffered\_io.hpp, 173  
 csv\_lib/src/cl/channel.cpp, 173  
 csv\_lib/src/cl/data\_point.cpp, 175  
 csv\_lib/src/cl/data\_set.cpp, 178  
 csv\_lib/src/cl/dos2unix.cpp, 178  
 csv\_lib/src/cl/error.cpp, 179  
 csv\_lib/src/cl/exception.cpp, 180  
 csv\_lib/src/cl/fs/directory\_listing.cpp, 180  
 csv\_lib/src/cl/fs/file.cpp, 181  
 csv\_lib/src/cl/fs/file\_stream.cpp, 182  
 csv\_lib/src/cl/fs/path.cpp, 182  
 csv\_lib/src/cl/fs/windows.cpp, 183  
 csv\_lib/src/cl/read\_csv\_file.cpp, 183  
     csv\_lib/src/cl/sensor.cpp, 184  
     csv\_lib/src/cl/use\_unbuffered\_io.cpp, 185  
     csv\_lib/test/channel\_test.cpp, 186  
     csv\_lib/test/CMakeLists.txt, 101  
     csv\_lib/test/column\_test.cpp, 188  
     csv\_lib/test/data\_point\_test.cpp, 189  
     csv\_lib/test/data\_set\_test.cpp, 191  
     csv\_lib/test/directory\_listing\_test.cpp, 195  
     csv\_lib/test/error\_test.cpp, 197  
     csv\_lib/test/exception\_test.cpp, 199  
     csv\_lib/test/main.cpp, 120  
     csv\_lib/test/read\_csv\_file\_test.cpp, 200  
     csv\_lib/test/s2n\_test.cpp, 201  
     csv\_lib/test/sensor\_test.cpp, 203  
     csv\_lib/test/to\_string\_test.cpp, 204  
     csv\_line\_test.cpp  
         TEST, 124  
 CsvFileKind  
     cl, 13  
 CsvLineBuilder  
     cs::CsvLineBuilder, 48  
 ctg, 35  
     aboveThreshold, 35  
     averageComparisonValueCalculator, 36  
     halfMaximumComparisonValueCalculator, 37  
     isRelevant, 38  
     percentageOf, 39  
     runAboveThreshold, 39  
 data\_point.cpp  
     channel, 175  
     fileName, 175  
     sensor, 176  
     time, 176  
     value, 177  
 data\_point\_test.cpp  
     dp, 190  
     TEST, 189, 190  
 data\_set\_accessor\_v  
     cl, 23  
 data\_set\_info.hpp  
     CS\_SPECIALIZE\_DATA\_SET\_INFO, 105  
 data\_set\_info\_test.cpp  
     TEST, 125  
 data\_set\_test.cpp  
     EXPECT\_LONG\_DOUBLE\_EQ, 191  
     TEST, 191–194  
 DataPoint  
     cl::DataPoint, 57  
 dataSet  
     cs::CsvLineBuilder, 49  
 dataSetAccessor  
     cl, 16  
 delete\_non\_bosch\_sensors.cpp  
     CL\_SENSOR\_X, 216  
 deleteLowVariance  
     cs::CsvLineBuilder, 49  
     cs::LogInfo, 88  
 deleteNonBoschSensors

fmc, 42  
deleteOutOfBoundsValues  
    fmc, 43  
deleteTooClose  
    cs::CsvLineBuilder, 50  
    cs::LogInfo, 89  
directory\_listing\_test.cpp  
    TEST, 196  
directoryListing  
    cl::fs, 24  
DirectoryListingOption  
    cl::fs, 24  
dos2unix  
    cl, 16  
dp  
    data\_point\_test.cpp, 190  
  
Error  
    cl::Error, 72  
error  
    error\_test.cpp, 198  
error.cpp  
    CL\_ERROR\_KIND\_X, 179  
error.hpp  
    CL\_ERROR\_KIND, 160  
    CL\_ERROR\_KIND\_X, 161  
    CL\_UNEXPECTED, 161  
error\_test.cpp  
    error, 198  
    TEST, 198  
Exception  
    cl::Exception, 77  
exception\_test.cpp  
    TEST, 199  
ExcludeDotAndDotDot  
    cl::fs, 24  
exists  
    cl::fs::File, 80  
    cl::fs::Path, 94  
EXPECT\_LONG\_DOUBLE\_EQ  
    above\_threshold\_test.cpp, 149  
    data\_set\_test.cpp, 191  
    percentage\_of\_test.cpp, 151  
Expected  
    cl, 12  
ExtractId  
    cl, 13  
extractId  
    cl::DataSet, 66  
  
File  
    cl::fs::File, 78  
file  
    cl::Error, 73  
    cl::Exception, 77  
fileName  
    cl::DataPoint, 58  
    cl::DataSet, 66  
    cs::LogLine, 92  
    data\_point.cpp, 175  
filePath  
    cs::LogLine, 92  
FileStream  
    cl::fs::FileStream, 85  
filter  
    cs::CsvLineBuilder, 51  
FilterKind  
    cs, 28  
filterKind  
    cs::LogInfo, 89  
fix\_csv/CMakeLists.txt, 102  
fix\_csv/include/adjust\_hardware\_timestamp.hpp, 205  
fix\_csv/include/convert\_to\_unix\_line\_endings.hpp, 206  
fix\_csv/include/create\_backup\_file.hpp, 207  
fix\_csv/include/delete\_non\_bosch\_sensors.hpp, 208  
fix\_csv/include/delete\_out\_of\_bounds\_values.hpp, 209  
fix\_csv/include/remove\_zeros\_from\_field.hpp, 210  
fix\_csv/include/restore\_from\_backup.hpp, 211  
fix\_csv/include/write\_file.hpp, 212  
fix\_csv/src/adjust\_hardware\_timestamp.cpp, 213  
fix\_csv/src/convert\_to\_unix\_line\_endings.cpp, 214  
fix\_csv/src/create\_backup\_file.cpp, 215  
fix\_csv/src/delete\_non\_bosch\_sensors.cpp, 215  
fix\_csv/src/delete\_out\_of\_bounds\_values.cpp, 216  
fix\_csv/src/main.cpp, 121  
fix\_csv/src/remove\_zeros\_from\_field.cpp, 217  
fix\_csv/src/restore\_from\_backup.cpp, 217  
fix\_csv/src/write\_file.cpp, 218  
fix\_csv/test/adjust\_hardware\_timestamp\_test.cpp, 219  
fix\_csv/test/CMakeLists.txt, 102  
fix\_csv/test/main.cpp, 122  
fix\_csv/test/remove\_zeros\_from\_field\_test.cpp, 221  
Fixed  
    cl, 14  
fmc, 40  
    adjustHardwareTimestamp, 41  
    convertToUnixLineEndings, 41  
    createBackupFile, 42  
    deleteNonBoschSensors, 42  
    deleteOutOfBoundsValues, 43  
    removeZerosFromField, 43  
    restoreFromBackup, 44  
    writeFile, 44  
formatError  
    cl::fs, 25  
function  
    cl::Error, 73  
    cl::Exception, 77  
gyroscopeAverage  
    cl::DataSet, 67  
gyroscopeMaximum  
    cl::DataSet, 67  
gyroscopeThreshold  
    cl, 23  
GyroscopeX  
    cl, 13  
gyroscopeX

cl::DataSet, 68  
 GyroscopeY  
     cl, 13  
 gyroscopeY  
     cl::DataSet, 68  
 GyroscopeZ  
     cl, 13  
 gyroscopeZ  
     cl::DataSet, 69  
  
 halfMaximumComparisonValueCalculator  
     ctg, 37  
 HardwareTimestamp  
     cl, 13  
 hardwareTimestamp  
     cl::DataSet, 69  
  
 include  
     CMakeLists.txt, 99–102  
 invalidSensor  
     cs::LogInfo, 91  
     cs::LogLine, 93  
 isAccelerometer  
     cl, 17  
 isDirectory  
     cl::fs::Path, 95  
 isFile  
     cl::fs::Path, 95  
 isGyroscope  
     cl, 17  
 isInitialized  
     cs::LogInfo, 89  
 isOld  
     cs::CsvLineBuilder, 51  
 isRelevant  
     ctg, 38  
  
 Kind  
     cl::Error, 72  
 kind  
     cl::Error, 73  
     cs::CsvLineBuilder, 52  
  
 line  
     cl::Error, 74  
     cl::Exception, 77  
 log\_files\_test.cpp  
     TEST, 126, 127  
 log\_info\_test.cpp  
     TEST, 128–137  
 log\_line\_test.cpp  
     TEST, 138, 139  
 logFilePath  
     cs::LogInfo, 89  
 logFiles  
     cs, 32  
 LogInfo  
     cs::LogInfo, 88  
 logPath

cs, 35  
  
 main  
     main.cpp, 115, 117, 118, 120–123  
 main.cpp  
     main, 115, 117, 118, 120–123  
 Maxima  
     cs, 29  
 message  
     cl::Error, 74  
 Minima  
     cs, 29  
 moveTo  
     cl::fs::File, 81  
 MovingAverage  
     cs, 28  
  
 None  
     cl::fs, 24  
  
 oldLogPath  
     cs, 35  
 OpenMode  
     cl::fs::FileStream, 85  
 operator!=  
     cs, 33  
     cs::LogInfo, 90  
 operator<  
     cl::fs, 25  
     cl::fs::Path, 97  
 operator<<  
     cl, 18, 19  
     cl::DataPoint, 60  
     cl::Error, 75  
     cl::fs, 25  
     cl::fs::Path, 97  
     cs, 33, 34  
     cs::LogInfo, 90  
 operator=  
     cl::fs::FileStream, 86  
 operator==  
     cl::fs, 26  
     cl::fs::Path, 97  
     cs, 34  
     cs::LogInfo, 91  
  
 parse  
     cs::LogLine, 92  
 Path  
     cl::fs::Path, 94  
 path  
     cl::fs::File, 82  
 percentage\_of\_test.cpp  
     EXPECT\_LONG\_DOUBLE\_EQ, 151  
     TEST, 151  
 percentageOf  
     ctg, 39  
 PL\_DEFINE\_EXCEPTION\_TYPE  
     cs, 34

PL\_NONCOPYABLE  
    cl::fs::FileStream, 87

pushUps  
    cs::CsvLineBuilder, 53

raise  
    cl::Error, 74

Raw  
    cl, 14

Read  
    cl::fs::FileStream, 85

read\_csv\_file\_test.cpp  
    TEST, 200, 201

readAll  
    cl::fs::FileStream, 87

readCsvFile  
    cl, 19

ReadWrite  
    cl::fs::FileStream, 85

remove  
    cl::fs::File, 82

remove\_zeros\_from\_field\_test.cpp  
    TEST, 222–224

removeZerosFromField  
    fmc, 43

repetitionCount  
    cs, 34

restoreFromBackup  
    fmc, 44

rowCount  
    cl::DataSet, 70

runAboveThreshold  
    ctg, 39

s2n  
    cl, 20

s2n\_test.cpp  
    TEST, 202

SamplingRate  
    cl, 13

SegmentationKind  
    cs, 28

segmentationKind  
    cs::LogInfo, 89

segmentationPointCount  
    cs::LogLine, 93

segmentationPoints  
    cs::CsvLineBuilder, 53

Sensor  
    cl, 14

sensor  
    cl::DataPoint, 59  
    cs::CsvLineBuilder, 54  
    cs::LogInfo, 90  
    cs::LogLine, 93  
    data\_point.cpp, 176

sensor.cpp  
    CL\_SENSOR\_X, 185

sensor.hpp

CL\_SENSOR, 171

CL\_SENSOR\_X, 171

sensor\_test.cpp  
    TEST, 204

sensors  
    cl, 23

separator.hpp  
    CL\_FS\_SEPARATOR, 166

set  
    CMakeLists.txt, 99–102

size  
    cl::fs::File, 83

size\_type  
    cl::DataSet, 62

skipWindow  
    cs::CsvLineBuilder, 55  
    cs::LogInfo, 90

str  
    cl::fs::Path, 96

TEST  
    above\_threshold\_test.cpp, 150  
    adjust\_hardware\_timestamp\_test.cpp, 219–221  
    channel\_test.cpp, 186, 187  
    column\_test.cpp, 188  
    csv\_line\_test.cpp, 124  
    data\_point\_test.cpp, 189, 190  
    data\_set\_info\_test.cpp, 125  
    data\_set\_test.cpp, 191–194  
    directory\_listing\_test.cpp, 196  
    error\_test.cpp, 198  
    exception\_test.cpp, 199  
    log\_files\_test.cpp, 126, 127  
    log\_info\_test.cpp, 128–137  
    log\_line\_test.cpp, 138, 139  
    percentage\_of\_test.cpp, 151  
    read\_csv\_file\_test.cpp, 200, 201  
    remove\_zeros\_from\_field\_test.cpp, 222–224  
    s2n\_test.cpp, 202  
    sensor\_test.cpp, 204  
    to\_string\_test.cpp, 205

this\_type  
    cl::fs::FileStream, 85  
    cs::CsvLineBuilder, 48

threshold  
    cl, 20

Time  
    cl, 13

time  
    cl::DataPoint, 59  
    cl::DataSet, 70  
    data\_point.cpp, 176

to\_string  
    cl, 21  
    cl::Error, 75

to\_string\_test.cpp  
    TEST, 205

Trigger  
    cl, 13

trigger  
  cl::DataSet, [71](#)

useUnbufferedIo  
  cl, [21](#)

utf16ToUtf8  
  cl::fs, [26](#)

utf8ToUtf16  
  cl::fs, [26](#)

value  
  cl::DataPoint, [60](#)  
  data\_point.cpp, [177](#)

windowSize  
  cs::CsvLineBuilder, [55](#)  
  cs::LogInfo, [90](#)

Write  
  cl::fs::FileStream, [85](#)

write  
  cl::fs::FileStream, [87](#)

writeFile  
  fmc, [44](#)