

FManC

1.0.0

Generated on Fri Feb 24 2023 19:36:06 for FManC by Doxygen 1.9.6

Fri Feb 24 2023 19:36:06

1 Welcome to the FManC documentation website !	1
2 File Index	3
2.1 File List	3
3 File Documentation	5
3.1 docs/documentation_pages/main_page.dox File Reference	5
3.2 src/code_utils/FMC_code_utils.h File Reference	5
3.2.1 Macro Definition Documentation	6
3.2.1.1 FMC_CODE_UTILS_H	6
3.3 FMC_code_utils.h	6
3.4 src/code_utils/FMC_codeUtils.c File Reference	6
3.5 FMC_codeUtils.c	6
3.6 src/cpp/FMC_dir/FMC_dir.cpp File Reference	7
3.6.1 Function Documentation	8
3.6.1.1 FMC_dirExists_()	8
3.6.1.2 FMC_getAbsolutePath_()	8
3.6.1.3 FMC_getCurrentPath_()	9
3.6.1.4 FMC_isBlock_()	9
3.6.1.5 FMC_isCharFile_()	10
3.6.1.6 FMC_isDir_()	10
3.6.1.7 FMC_isEmpty_()	11
3.6.1.8 FMC_isFIFO_()	11
3.6.1.9 FMC_isOther_()	12
3.6.1.10 FMC_isRegFile_()	12
3.6.1.11 FMC_isSocket_()	13
3.6.1.12 FMC_isSymLink_()	13
3.7 FMC_dir.cpp	14
3.8 src/cpp/FMC_dir/FMC_dir.hpp File Reference	16
3.8.1 Function Documentation	17
3.8.1.1 FMC_dirExists_()	17
3.8.1.2 FMC_getAbsolutePath_()	17
3.8.1.3 FMC_getCurrentPath_()	18
3.8.1.4 FMC_isBlock_()	18
3.8.1.5 FMC_isCharFile_()	19
3.8.1.6 FMC_isDir_()	19
3.8.1.7 FMC_isEmpty_()	20
3.8.1.8 FMC_isFIFO_()	20
3.8.1.9 FMC_isOther_()	21
3.8.1.10 FMC_isRegFile_()	21
3.8.1.11 FMC_isSocket_()	22
3.8.1.12 FMC_isSymLink_()	22
3.9 FMC_dir.hpp	23

3.10 src/cpp/FMC_dir/FMC_dir_wrapper.cpp File Reference	24
3.10.1 Function Documentation	25
3.10.1.1 FMC_dirExists()	25
3.10.1.2 FMC_getAbsolutePath()	26
3.10.1.3 FMC_getCurrentPath()	27
3.10.1.4 FMC_isBlock()	28
3.10.1.5 FMC_isCharFile()	29
3.10.1.6 FMC_isDir()	30
3.10.1.7 FMC_isEmpty()	31
3.10.1.8 FMC_isFIFO()	32
3.10.1.9 FMC_isOther()	33
3.10.1.10 FMC_isRegFile()	34
3.10.1.11 FMC_isSocket()	35
3.10.1.12 FMC_isSymLink()	36
3.11 FMC_dir_wrapper.cpp	36
3.12 src/cpp/FMC_wrapper.h File Reference	38
3.12.1 Function Documentation	39
3.12.1.1 FMC_dirExists()	39
3.12.1.2 FMC_getAbsolutePath()	40
3.12.1.3 FMC_getCurrentPath()	41
3.12.1.4 FMC_isBlock()	42
3.12.1.5 FMC_isCharFile()	43
3.12.1.6 FMC_isDir()	44
3.12.1.7 FMC_isEmpty()	45
3.12.1.8 FMC_isFIFO()	46
3.12.1.9 FMC_isOther()	47
3.12.1.10 FMC_isRegFile()	48
3.12.1.11 FMC_isSocket()	49
3.12.1.12 FMC_isSymLink()	50
3.13 FMC_wrapper.h	51
3.14 src/data_analyze/encodings/FMC_encodings.c File Reference	52
3.14.1 Macro Definition Documentation	52
3.14.1.1 __STDC_WANT_LIB_EXT1__	53
3.14.2 Function Documentation	53
3.14.2.1 FMC_checkEncodingFlag()	53
3.14.2.2 FMC_getEncoding()	53
3.15 FMC_encodings.c	54
3.16 src/data_analyze/encodings/FMC_encodings.h File Reference	57
3.16.1 Macro Definition Documentation	58
3.16.1.1 FMC_ENCODINGS	58
3.16.2 Function Documentation	58
3.16.2.1 FMC_checkEncodingFlag()	58

3.16.2.2 FMC_getEncoding()	58
3.17 FMC_encodings.h	59
3.18 src/data_analyze/FMC_data_analyze.h File Reference	59
3.18.1 Macro Definition Documentation	60
3.18.1.1 FMC_DATA_ANALYZE_H	60
3.19 FMC_data_analyze.h	60
3.20 src/data_analyze/strings/FMC_chars.c File Reference	61
3.21 FMC_chars.c	61
3.22 src/data_analyze/strings/FMC_strings.c File Reference	62
3.23 FMC_strings.c	62
3.24 src/data_analyze/strings/FMC_strings.h File Reference	62
3.24.1 Macro Definition Documentation	63
3.24.1.1 FMC_STRINGS_H	63
3.25 FMC_strings.h	63
3.26 src/files/FMC_file_management.h File Reference	64
3.26.1 Macro Definition Documentation	65
3.26.1.1 FMC_FILE_MANAGEMENT_H	65
3.26.2 Function Documentation	65
3.26.2.1 FMC_cutFilename()	65
3.26.2.2 FMC_extractFilename()	66
3.26.2.3 FMC_getExtension()	67
3.27 FMC_file_management.h	68
3.28 src/files/FMC_fileMan.c File Reference	68
3.29 FMC_fileMan.c	68
3.30 src/files/FMC_files.c File Reference	69
3.31 FMC_files.c	69
3.32 src/files/FMC_paths.c File Reference	70
3.32.1 Function Documentation	70
3.32.1.1 FMC_cutFilename()	70
3.32.1.2 FMC_extractFilename()	71
3.32.1.3 FMC_getExtension()	72
3.33 FMC_paths.c	73
3.34 src/FMC.h File Reference	76
3.34.1 Macro Definition Documentation	76
3.34.1.1 FMC_H	76
3.35 FMC.h	77
3.36 src/general/FMC_general.h File Reference	77
3.36.1 Macro Definition Documentation	78
3.36.1.1 FMC_DATA_H	78
3.37 FMC_general.h	78
3.38 src/general/preprocessor/FMC_attributes.h File Reference	79
3.39 FMC_attributes.h	80

3.40 src/general/preprocessor/FMC_consts.h File Reference	84
3.40.1 Macro Definition Documentation	85
3.40.1.1 BG_BLACK	85
3.40.1.2 BG_BLUE	85
3.40.1.3 BG_BRIGHT_BLACK	85
3.40.1.4 BG_BRIGHT_BLUE	86
3.40.1.5 BG_BRIGHT_CYAN	86
3.40.1.6 BG_BRIGHT_GREEN	86
3.40.1.7 BG_BRIGHT_MAGENTA	86
3.40.1.8 BG_BRIGHT_RED	86
3.40.1.9 BG_BRIGHT_WHITE	86
3.40.1.10 BG_BRIGHT_YELLOW	87
3.40.1.11 BG_CYAN	87
3.40.1.12 BG_GREEN	87
3.40.1.13 BG_MAGENTA	87
3.40.1.14 BG_RED	87
3.40.1.15 BG_WHITE	87
3.40.1.16 BG_YELLOW	88
3.40.1.17 False	88
3.40.1.18 FG_BLACK	88
3.40.1.19 FG_BLUE	88
3.40.1.20 FG_BRIGHT_BLACK	88
3.40.1.21 FG_BRIGHT_BLUE	88
3.40.1.22 FG_BRIGHT_CYAN	89
3.40.1.23 FG_BRIGHT_GREEN	89
3.40.1.24 FG_BRIGHT_MAGENTA	89
3.40.1.25 FG_BRIGHT_RED	89
3.40.1.26 FG_BRIGHT_WHITE	89
3.40.1.27 FG_BRIGHT_YELLOW	89
3.40.1.28 FG_CYAN	90
3.40.1.29 FG_GREEN	90
3.40.1.30 FG_MAGENTA	90
3.40.1.31 FG_RED	90
3.40.1.32 FG_WHITE	90
3.40.1.33 FG_YELLOW	90
3.40.1.34 FMC_BOOLEANS	91
3.40.1.35 FMC_CONSTS_H	91
3.40.1.36 FMC_MAX_PATH_COMPONENTS_SIZE	91
3.40.1.37 FMC_STYLES	91
3.40.1.38 MAX_FEXT_SIZE	91
3.40.1.39 MAX_FNAME_SIZE	91
3.40.1.40 MAX_FPATH_SIZE	92

3.40.1.41 RESET	92
3.40.1.42 True	92
3.40.1.43 TXT_BLINK	92
3.40.1.44 TXT_BOLD	92
3.40.1.45 TXT_DIM	92
3.40.1.46 TXT_HIDDEN	93
3.40.1.47 TXT_REVERSE	93
3.40.1.48 TXT_UNDERLINED	93
3.41 FMC_consts.h	93
3.42 src/general/preprocessor/FMC_flags.h File Reference	94
3.42.1 Macro Definition Documentation	95
3.42.1.1 ASCII	95
3.42.1.2 FMC_ENCODING_FLAGS	95
3.42.1.3 FMC_FLAGS	95
3.42.1.4 UNKNOWN	95
3.42.1.5 UTF16_BE	96
3.42.1.6 UTF16_LE	96
3.42.1.7 UTF32_BE	96
3.42.1.8 UTF32_LE	96
3.42.1.9 UTF8	96
3.42.1.10 UTF8_BOM	96
3.43 FMC_flags.h	97
3.44 src/general/preprocessor/FMC_macros.h File Reference	97
3.44.1 Macro Definition Documentation	98
3.44.1.1 FMC_COMPILE_TIME_ERROR	98
3.44.1.2 FMC_DEFER	98
3.44.1.3 FMC_ERROR_CHECK	99
3.44.1.4 FMC_MACROS_H	99
3.44.1.5 FMC_MAJOR_VERSION	99
3.44.1.6 FMC_MINOR_VERSION	99
3.44.1.7 FMC_PATCH_VERSION	99
3.44.1.8 FMC_VERSION	100
3.44.1.9 FMC_VERSION_NUMBER	100
3.44.1.10 FMC_VERSION_STRING	100
3.45 FMC_macros.h	100
3.46 src/general/preprocessor/FMC_platform.h File Reference	103
3.47 FMC_platform.h	103
3.48 src/general/types/FMC_enums.h File Reference	104
3.48.1 Macro Definition Documentation	105
3.48.1.1 FMC_ENUMS_H	105
3.48.2 Typedef Documentation	105
3.48.2.1 FMC_Encodings	106

3.48.3 Enumeration Type Documentation	106
3.48.3.1 FManC_Encodings	106
3.49 FMC_enums.h	106
3.50 src/general/types/FMC_structs.h File Reference	107
3.50.1 Data Structure Documentation	108
3.50.1.1 struct FManC_Char	108
3.50.1.2 struct FManC_CharComp	109
3.50.1.3 struct FManC_CStrView	109
3.50.1.4 struct FManC_File	110
3.50.1.5 struct FManC_String	111
3.50.1.6 struct FManC_StrOcc	112
3.50.2 Macro Definition Documentation	112
3.50.2.1 FMC_STRUCTS_H	112
3.50.3 Typedef Documentation	113
3.50.3.1 FMC_Char	113
3.50.3.2 FMC_CharComp	113
3.50.3.3 FMC_CStrView	113
3.50.3.4 FMC_File	113
3.50.3.5 FMC_String	113
3.50.3.6 FMC_StrOcc	113
3.51 FMC_structs.h	114
3.52 src/general/types/FMC_typedefs.h File Reference	115
3.52.1 Macro Definition Documentation	116
3.52.1.1 FMC_TYPEDEFS_H	116
3.52.2 Typedef Documentation	116
3.52.2.1 FMC_Bool	117
3.52.2.2 FMC_CharControl	117
3.52.2.3 FMC_FileState	117
3.52.2.4 found_bs_n	117
3.52.2.5 found_bs_r_bs_n	117
3.52.2.6 found_bs_t	117
3.53 FMC_typedefs.h	118
3.54 src/general/utls/FMC_deprecated.h File Reference	119
3.54.1 Function Documentation	120
3.54.1.1 FMC_FUNC_UNAVAILABLE() [1/4]	120
3.54.1.2 FMC_FUNC_UNAVAILABLE() [2/4]	120
3.54.1.3 FMC_FUNC_UNAVAILABLE() [3/4]	120
3.54.1.4 FMC_FUNC_UNAVAILABLE() [4/4]	120
3.54.1.5 FMC_TYPE_UNAVAILABLE()	120
3.54.2 Variable Documentation	121
3.54.2.1 extension	121
3.54.2.2 fileName	121

3.54.2.3 filePath	121
3.54.2.4 pathToCopy	121
3.54.2.5 toSearch	121
3.55 FMC_deprecated.h	122
3.56 src/general/utils/FMC_errors.c File Reference	123
3.56.1 Function Documentation	124
3.56.1.1 FMC_changeStreamTextColorToBlue()	124
3.56.1.2 FMC_changeStreamTextColorToBrightBlue()	125
3.56.1.3 FMC_changeStreamTextColorToBrightCyan()	125
3.56.1.4 FMC_changeStreamTextColorToBrightGreen()	126
3.56.1.5 FMC_changeStreamTextColorToBrightMagenta()	126
3.56.1.6 FMC_changeStreamTextColorToBrightRed()	127
3.56.1.7 FMC_changeStreamTextColorToBrightWhite()	127
3.56.1.8 FMC_changeStreamTextColorToBrightYellow()	128
3.56.1.9 FMC_changeStreamTextColorToCyan()	128
3.56.1.10 FMC_changeStreamTextColorToGreen()	129
3.56.1.11 FMC_changeStreamTextColorToMagenta()	129
3.56.1.12 FMC_changeStreamTextColorToRed()	130
3.56.1.13 FMC_changeStreamTextColorToWhite()	130
3.56.1.14 FMC_changeStreamTextColorToYellow()	131
3.56.1.15 FMC_makeMsg_f()	131
3.56.1.16 FMC_printBlueError()	131
3.56.1.17 FMC_printBlueText()	132
3.56.1.18 FMC_printBrightBlueError()	133
3.56.1.19 FMC_printBrightBlueText()	133
3.56.1.20 FMC_printBrightCyanError()	134
3.56.1.21 FMC_printBrightCyanText()	134
3.56.1.22 FMC_printBrightGreenError()	135
3.56.1.23 FMC_printBrightGreenText()	135
3.56.1.24 FMC_printBrightMagentaError()	136
3.56.1.25 FMC_printBrightMagentaText()	136
3.56.1.26 FMC_printBrightRedError()	137
3.56.1.27 FMC_printBrightRedText()	137
3.56.1.28 FMC_printBrightWhiteError()	138
3.56.1.29 FMC_printBrightWhiteText()	139
3.56.1.30 FMC_printBrightYellowError()	139
3.56.1.31 FMC_printBrightYellowText()	140
3.56.1.32 FMC_printCyanError()	140
3.56.1.33 FMC_printCyanText()	141
3.56.1.34 FMC_printGreenError()	142
3.56.1.35 FMC_printGreenText()	142
3.56.1.36 FMC_printMagentaError()	143

3.56.1.37 FMC_printMagentaText()	143
3.56.1.38 FMC_printRedError()	144
3.56.1.39 FMC_printRedText()	144
3.56.1.40 FMC_printWhiteError()	145
3.56.1.41 FMC_printWhiteText()	145
3.56.1.42 FMC_printYellowError()	146
3.56.1.43 FMC_printYellowText()	146
3.56.1.44 FMC_resetStreamOutputStyle()	147
3.57 FMC_errors.c	148
3.58 src/general/utils/FMC_errors.h File Reference	150
3.58.1 Macro Definition Documentation	152
3.58.1.1 FMC_ERRORS	152
3.58.1.2 FMC_makeMsg	152
3.58.2 Function Documentation	152
3.58.2.1 FMC_changeStreamTextColorToBlue()	152
3.58.2.2 FMC_changeStreamTextColorToBrightBlue()	153
3.58.2.3 FMC_changeStreamTextColorToBrightCyan()	153
3.58.2.4 FMC_changeStreamTextColorToBrightGreen()	154
3.58.2.5 FMC_changeStreamTextColorToBrightMagenta()	154
3.58.2.6 FMC_changeStreamTextColorToBrightRed()	155
3.58.2.7 FMC_changeStreamTextColorToBrightWhite()	155
3.58.2.8 FMC_changeStreamTextColorToBrightYellow()	156
3.58.2.9 FMC_changeStreamTextColorToCyan()	156
3.58.2.10 FMC_changeStreamTextColorToGreen()	157
3.58.2.11 FMC_changeStreamTextColorToMagenta()	157
3.58.2.12 FMC_changeStreamTextColorToRed()	158
3.58.2.13 FMC_changeStreamTextColorToWhite()	158
3.58.2.14 FMC_changeStreamTextColorToYellow()	159
3.58.2.15 FMC_makeMsg_f()	159
3.58.2.16 FMC_printBlueError()	159
3.58.2.17 FMC_printBlueText()	160
3.58.2.18 FMC_printBrightBlueError()	161
3.58.2.19 FMC_printBrightBlueText()	161
3.58.2.20 FMC_printBrightCyanError()	162
3.58.2.21 FMC_printBrightCyanText()	162
3.58.2.22 FMC_printBrightGreenError()	163
3.58.2.23 FMC_printBrightGreenText()	163
3.58.2.24 FMC_printBrightMagentaError()	164
3.58.2.25 FMC_printBrightMagentaText()	164
3.58.2.26 FMC_printBrightRedError()	165
3.58.2.27 FMC_printBrightRedText()	165
3.58.2.28 FMC_printBrightWhiteError()	166

3.58.2.29 FMC_printBrightWhiteText()	167
3.58.2.30 FMC_printBrightYellowError()	167
3.58.2.31 FMC_printBrightYellowText()	168
3.58.2.32 FMC_printCyanError()	168
3.58.2.33 FMC_printCyanText()	169
3.58.2.34 FMC_printGreenError()	170
3.58.2.35 FMC_printGreenText()	170
3.58.2.36 FMC_printMagentaError()	171
3.58.2.37 FMC_printMagentaText()	171
3.58.2.38 FMC_printRedError()	172
3.58.2.39 FMC_printRedText()	172
3.58.2.40 FMC_printWhiteError()	173
3.58.2.41 FMC_printWhiteText()	173
3.58.2.42 FMC_printYellowError()	174
3.58.2.43 FMC_printYellowText()	174
3.58.2.44 FMC_resetStreamOutputStyle()	175
3.58.2.45 FMC_setBGStreamColorToBlue()	176
3.58.2.46 FMC_setBGStreamColorToBrightBlue()	177
3.58.2.47 FMC_setBGStreamColorToBrightCyan()	177
3.58.2.48 FMC_setBGStreamColorToBrightGreen()	177
3.58.2.49 FMC_setBGStreamColorToBrightMagenta()	177
3.58.2.50 FMC_setBGStreamColorToBrightRed()	177
3.58.2.51 FMC_setBGStreamColorToBrightWhite()	178
3.58.2.52 FMC_setBGStreamColorToBrightYellow()	178
3.58.2.53 FMC_setBGStreamColorToCyan()	178
3.58.2.54 FMC_setBGStreamColorToGreen()	178
3.58.2.55 FMC_setBGStreamColorToMagenta()	178
3.58.2.56 FMC_setBGStreamColorToRed()	179
3.58.2.57 FMC_setBGStreamColorToWhite()	179
3.58.2.58 FMC_setBGStreamColorToYellow()	179
3.58.2.59 FMC_setTextStyleToBlink()	179
3.58.2.60 FMC_setTextStyleToBold()	180
3.58.2.61 FMC_setTextStyleToDim()	180
3.58.2.62 FMC_setTextStyleToHidden()	181
3.58.2.63 FMC_setTextStyleToReverse()	181
3.58.2.64 FMC_setTextStyleToUnderlined()	181
3.59 FMC_errors.h	181
3.60 src/general/utils/FMC_globals.h File Reference	187
3.60.1 Variable Documentation	187
3.60.1.1 FMC_VAR_COMMON	188
3.61 FMC_globals.h	188

Chapter 1

Welcome to the FManC documentation website !

Copyright

This C library is licenced under the MIT license terms

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

src/ FMC.h	76
src/code_utils/ FMC_code_utils.h	5
src/code_utils/ FMC_codeUtils.c	6
src/cpp/ FMC_wrapper.h	38
src/cpp/FMC_dir/ FMC_dir.cpp	7
src/cpp/FMC_dir/ FMC_dir.hpp	16
src/cpp/FMC_dir/ FMC_dir_wrapper.cpp	24
src/data_analyze/ FMC_data_analyze.h	59
src/data_analyze/encodings/ FMC_encodings.c	52
src/data_analyze/encodings/ FMC_encodings.h	57
src/data_analyze/strings/ FMC_chars.c	61
src/data_analyze/strings/ FMC_strings.c	62
src/data_analyze/strings/ FMC_strings.h	62
src/files/ FMC_file_management.h	64
src/files/ FMC_fileMan.c	68
src/files/ FMC_files.c	69
src/files/ FMC_paths.c	70
src/general/ FMC_general.h	77
src/general/preprocessor/ FMC_attributes.h	79
src/general/preprocessor/ FMC_consts.h	84
src/general/preprocessor/ FMC_flags.h	94
src/general/preprocessor/ FMC_macros.h	97
src/general/preprocessor/ FMC_platform.h	103
src/general/types/ FMC_enums.h	104
src/general/types/ FMC_structs.h	107
src/general/types/ FMC_typedefs.h	115
src/general/utils/ FMC_deprecated.h	119
src/general/utils/ FMC_errors.c	123
src/general/utils/ FMC_errors.h	150
src/general/utils/ FMC_globals.h	187

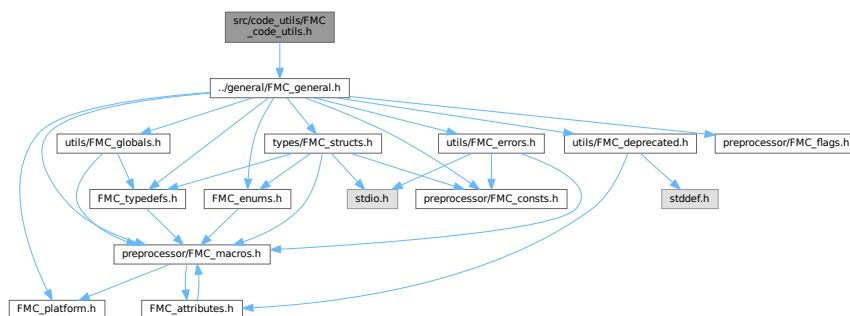
Chapter 3

File Documentation

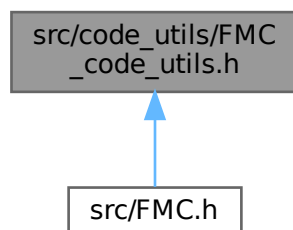
3.1 docs/documentation_pages/main_page.dox File Reference

3.2 src/code_utils/FMC_code_utils.h File Reference

Include dependency graph for FMC_code_utils.h:



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



Macros

- `#define FMC_CODE_UTILS_H`

3.2.1 Macro Definition Documentation

3.2.1.1 FMC_CODE_UTILS_H

`#define FMC_CODE_UTILS_H`

Definition at line 30 of file [FMC_code_utils.h](#).

3.3 FMC_code_utils.h

[Go to the documentation of this file.](#)

```
00001 /*
00002
00003 MIT License
00004
00005 Copyright (c) 2022 Axel PASCON
00006
00007 Permission is hereby granted, free of charge, to any person obtaining a copy
00008 of this software and associated documentation files (the "Software"), to deal
00009 in the Software without restriction, including without limitation the rights
00010 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011 copies of the Software, and to permit persons to whom the Software is
00012 furnished to do so, subject to the following conditions:
00013
00014 The above copyright notice and this permission notice shall be included in all
00015 copies or substantial portions of the Software.
00016
00017 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023 SOFTWARE.
00024
00025 */
00026
00027 #pragma once
00028
00029 #ifndef FMC_CODE_UTILS_H
00030 #define FMC_CODE_UTILS_H
00031
00032 #include "../general/FMC_general.h"
00033
00034 #endif // FMC_CODE_UTILS_H
```

3.4 src/code_utils/FMC_codeUtils.c File Reference

3.5 FMC_codeUtils.c

[Go to the documentation of this file.](#)

```
00001 /*
00002
00003 MIT License
00004
```

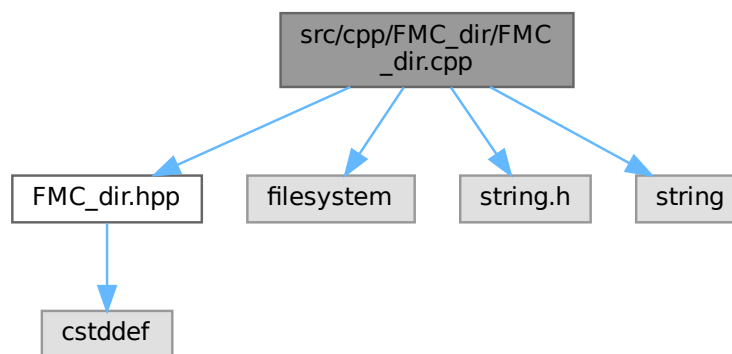
```

00005 Copyright (c) 2022 Axel PASCON
00006
00007 Permission is hereby granted, free of charge, to any person obtaining a copy
00008 of this software and associated documentation files (the "Software"), to deal
00009 in the Software without restriction, including without limitation the rights
00010 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011 copies of the Software, and to permit persons to whom the Software is
00012 furnished to do so, subject to the following conditions:
00013
00014 The above copyright notice and this permission notice shall be included in all
00015 copies or substantial portions of the Software.
00016
00017 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023 SOFTWARE.
00024
00025 */

```

3.6 src/cpp/FMC_dir/FMC_dir.cpp File Reference

Include dependency graph for FMC_dir.cpp:



Functions

- int [FMC_dirExists_](#) (const char *path)
- char * [FMC_getAbsolutePath_](#) (char *path, char *buffer, const size_t size)
- char * [FMC_getCurrentPath_](#) (char *path, const size_t size)
- int [FMC_isBlock_](#) (const char *path)
- int [FMC_isCharFile_](#) (const char *path)
- int [FMC_isDir_](#) (const char *path)
- int [FMC_isEmpty_](#) (const char *path)
- int [FMC_isFIFO_](#) (const char *path)
- int [FMC_isOther_](#) (const char *path)
- int [FMC_isRegFile_](#) (const char *path)
- int [FMC_isSocket_](#) (const char *path)
- int [FMC_isSymLink_](#) (const char *path)

3.6.1 Function Documentation

3.6.1.1 FMC_dirExists_()

```
int FMC_dirExists_ (
    const char * path )
```

Definition at line 38 of file [FMC_dir.cpp](#).

Referenced by [FMC_dirExists\(\)](#).

Here is the caller graph for this function:



3.6.1.2 FMC_getAbsolutePath_()

```
char * FMC_getAbsolutePath_ (
    char * path,
    char * buffer,
    const size_t size )
```

Definition at line 157 of file [FMC_dir.cpp](#).

Referenced by [FMC_getAbsolutePath\(\)](#).

Here is the caller graph for this function:



3.6.1.3 FMC_getCurrentPath_()

```
char * FMC_getCurrentPath_ (  
    char * path,  
    const size_t size )
```

Definition at line 142 of file [FMC_dir.cpp](#).

Referenced by [FMC_getCurrentPath\(\)](#).

Here is the caller graph for this function:



3.6.1.4 FMC_isBlock_()

```
int FMC_isBlock_ (  
    const char * path )
```

Definition at line 70 of file [FMC_dir.cpp](#).

Referenced by [FMC_isBlock\(\)](#).

Here is the caller graph for this function:



3.6.1.5 FMC_isCharFile_()

```
int FMC_isCharFile_ (  
    const char * path )
```

Definition at line 79 of file [FMC_dir.cpp](#).

Referenced by [FMC_isCharFile\(\)](#).

Here is the caller graph for this function:



3.6.1.6 FMC_isDir_()

```
int FMC_isDir_ (  
    const char * path )
```

Definition at line 43 of file [FMC_dir.cpp](#).

Referenced by [FMC_isDir\(\)](#).

Here is the caller graph for this function:



3.6.1.7 FMC_isEmpty_()

```
int FMC_isEmpty_ (  
    const char * path )
```

Definition at line 128 of file [FMC_dir.cpp](#).

Referenced by [FMC_isEmpty\(\)](#).

Here is the caller graph for this function:



3.6.1.8 FMC_isFIFO_()

```
int FMC_isFIFO_ (  
    const char * path )
```

Definition at line 97 of file [FMC_dir.cpp](#).

Referenced by [FMC_isFIFO\(\)](#).

Here is the caller graph for this function:



3.6.1.9 FMC_isOther_()

```
int FMC_isOther_ (
    const char * path )
```

Definition at line 106 of file [FMC_dir.cpp](#).

Referenced by [FMC_isOther\(\)](#).

Here is the caller graph for this function:



3.6.1.10 FMC_isRegFile_()

```
int FMC_isRegFile_ (
    const char * path )
```

Definition at line 52 of file [FMC_dir.cpp](#).

Referenced by [FMC_isRegFile\(\)](#).

Here is the caller graph for this function:



3.6.1.11 FMC_isSocket_()

```
int FMC_isSocket_ (  
    const char * path )
```

Definition at line 88 of file [FMC_dir.cpp](#).

Referenced by [FMC_isSocket\(\)](#).

Here is the caller graph for this function:



3.6.1.12 FMC_isSymLink_()

```
int FMC_isSymLink_ (  
    const char * path )
```

Definition at line 61 of file [FMC_dir.cpp](#).

Referenced by [FMC_isSymLink\(\)](#).

Here is the caller graph for this function:



3.7 FMC_dir.cpp

[Go to the documentation of this file.](#)

```

00001  /*
00002
00003  MIT License
00004
00005  Copyright (c) 2022-2023 Axel PASCON
00006
00007  Permission is hereby granted, free of charge, to any person obtaining a copy
00008  of this software and associated documentation files (the "Software"), to deal
00009  in the Software without restriction, including without limitation the rights
00010  to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011  copies of the Software, and to permit persons to whom the Software is
00012  furnished to do so, subject to the following conditions:
00013
00014  The above copyright notice and this permission notice shall be included in all
00015  copies or substantial portions of the Software.
00016
00017  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018  IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019  FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020  AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021  LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022  OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023  SOFTWARE.
00024
00025  */
00026
00027  #if __cplusplus < 201703L
00028      #error "The contents of <filesystem> are available only with C++17 or later."
00029  #endif
00030
00031  #include "FMC_dir.hpp"
00032  #include <filesystem>
00033  #include <string.h>
00034  #include <string>
00035
00036  namespace fs = std::filesystem;
00037
00038  int FMC_dirExists_(const char *path)
00039  {
00040      return fs::exists(path);
00041  }
00042
00043  int FMC_isDir_(const char *path)
00044  {
00045      if(fs::exists(path))
00046      {
00047          return fs::is_directory(path);
00048      }
00049      else return -1;
00050  }
00051
00052  int FMC_isRegFile_(const char *path)
00053  {
00054      if(fs::exists(path))
00055      {
00056          return fs::is_regular_file(path);
00057      }
00058      else return -1;
00059  }
00060
00061  int FMC_isSymLink_(const char *path)
00062  {
00063      if (fs::exists(path))
00064      {
00065          return fs::is_symlink(path);
00066      }
00067      else return -1;
00068  }
00069
00070  int FMC_isBlock_(const char* path)
00071  {
00072      if (fs::exists(path))
00073      {
00074          return fs::is_block_file(path);
00075      }
00076      else return -1;
00077  }
00078
00079  int FMC_isCharFile_(const char* path)
00080  {
00081      if (fs::exists(path))
00082      {

```

```

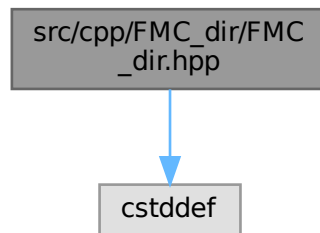
00083         return fs::is_character_file(path);
00084     }
00085     else return -1;
00086 }
00087
00088 int FMC_isSocket_(const char* path)
00089 {
00090     if (fs::exists(path))
00091     {
00092         return fs::is_socket(path);
00093     }
00094     else return -1;
00095 }
00096
00097 int FMC_isFIFO_(const char* path)
00098 {
00099     if (fs::exists(path))
00100     {
00101         return fs::is_fifo(path);
00102     }
00103     else return -1;
00104 }
00105
00106 int FMC_isOther_(const char* path)
00107 {
00108     if (fs::exists(path))
00109     {
00110         return fs::is_other(path);
00111     }
00112     else return -1;
00113 }
00114
00115 /*
00116 char *FMC_readSymlink_(char *path_sym, const char * path, const int size)
00117 {
00118     memset(path_sym, 0, size);
00119     fs::path p(path);
00120     if (is_symlink(p) && exists(p) && size >= fs::read_symlink(p).string().size()) // to be changed
00121     {
00122         fs::path target = fs::read_symlink(p);
00123         strcpy(path_sym, target.c_str());
00124     }
00125     return path_sym;
00126 } */
00127
00128 int FMC_isEmpty_(const char *path)
00129 {
00130     if (fs::exists(path))
00131     {
00132         return fs::is_empty(path);
00133     }
00134     else return -1;
00135 }
00136
00137 /*int FMC_createDir_(const char *path)
00138 {
00139     return fs::create_directory(path);
00140 } */
00141
00142 char *FMC_getCurrentPath_(char *path, const size_t size)
00143 {
00144     std::string s = fs::current_path().string();
00145     if (size >= s.length()+1)
00146     {
00147         memset(path, 0, size);
00148         strcpy(path, fs::current_path().string().c_str(), fs::current_path().string().length());
00149         if (strchr(path, '/') != NULL) strcat(path, "/");
00150         else if (strchr(path, '\\') != NULL) strcat(path, "\\");
00151         else return NULL;
00152         return path;
00153     }
00154     else return NULL;
00155 }
00156
00157 char *FMC_getAbsolutePath_(char *path, char *buffer, const size_t size)
00158 {
00159     if (fs::exists(path) && size > fs::absolute(path).string().length())
00160     {
00161         memset(buffer, 0, size);
00162         strcpy(buffer, fs::absolute(path).string().c_str(), fs::absolute(path).string().length());
00163         if (strchr(path, '/') != NULL) strcat(buffer, "/");
00164         else if (strchr(path, '\\') != NULL) strcat(buffer, "\\");
00165         else return NULL;
00166         return buffer;
00167     }
00168     else return NULL;

```

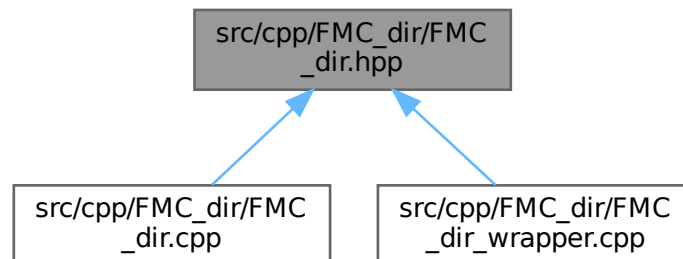
```
00170 }
```

3.8 src/cpp/FMC_dir/FMC_dir.hpp File Reference

Include dependency graph for FMC_dir.hpp:



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



Functions

- int [FMC_dirExists_](#) (const char *path)
- char * [FMC_getAbsolutePath_](#) (char *path, char *buffer, const size_t size)
- char * [FMC_getCurrentPath_](#) (char *path, const size_t size)
- int [FMC_isBlock_](#) (const char *path)
- int [FMC_isCharFile_](#) (const char *path)
- int [FMC_isDir_](#) (const char *path)
- int [FMC_isEmpty_](#) (const char *path)
- int [FMC_isFIFO_](#) (const char *path)
- int [FMC_isOther_](#) (const char *path)
- int [FMC_isRegFile_](#) (const char *path)
- int [FMC_isSocket_](#) (const char *path)
- int [FMC_isSymLink_](#) (const char *path)

3.8.1 Function Documentation

3.8.1.1 FMC_dirExists_()

```
int FMC_dirExists_ (  
    const char * path )
```

Definition at line 38 of file [FMC_dir.cpp](#).

Referenced by [FMC_dirExists\(\)](#).

Here is the caller graph for this function:



3.8.1.2 FMC_getAbsolutePath_()

```
char * FMC_getAbsolutePath_ (  
    char * path,  
    char * buffer,  
    const size_t size )
```

Definition at line 157 of file [FMC_dir.cpp](#).

Referenced by [FMC_getAbsolutePath\(\)](#).

Here is the caller graph for this function:



3.8.1.3 FMC_getCurrentPath_()

```
char * FMC_getCurrentPath_ (  
    char * path,  
    const size_t size )
```

Definition at line 142 of file [FMC_dir.cpp](#).

Referenced by [FMC_getCurrentPath\(\)](#).

Here is the caller graph for this function:



3.8.1.4 FMC_isBlock_()

```
int FMC_isBlock_ (  
    const char * path )
```

Definition at line 70 of file [FMC_dir.cpp](#).

Referenced by [FMC_isBlock\(\)](#).

Here is the caller graph for this function:



3.8.1.5 FMC_isCharFile_()

```
int FMC_isCharFile_ (
    const char * path )
```

Definition at line 79 of file [FMC_dir.cpp](#).

Referenced by [FMC_isCharFile\(\)](#).

Here is the caller graph for this function:



3.8.1.6 FMC_isDir_()

```
int FMC_isDir_ (
    const char * path )
```

Definition at line 43 of file [FMC_dir.cpp](#).

Referenced by [FMC_isDir\(\)](#).

Here is the caller graph for this function:



3.8.1.7 FMC_isEmpty_()

```
int FMC_isEmpty_ (
    const char * path )
```

Definition at line 128 of file [FMC_dir.cpp](#).

Referenced by [FMC_isEmpty\(\)](#).

Here is the caller graph for this function:



3.8.1.8 FMC_isFIFO_()

```
int FMC_isFIFO_ (
    const char * path )
```

Definition at line 97 of file [FMC_dir.cpp](#).

Referenced by [FMC_isFIFO\(\)](#).

Here is the caller graph for this function:



3.8.1.9 FMC_isOther_()

```
int FMC_isOther_ (
    const char * path )
```

Definition at line 106 of file [FMC_dir.cpp](#).

Referenced by [FMC_isOther\(\)](#).

Here is the caller graph for this function:



3.8.1.10 FMC_isRegFile_()

```
int FMC_isRegFile_ (
    const char * path )
```

Definition at line 52 of file [FMC_dir.cpp](#).

Referenced by [FMC_isRegFile\(\)](#).

Here is the caller graph for this function:



3.8.1.11 FMC_isSocket_()

```
int FMC_isSocket_ (  
    const char * path )
```

Definition at line 88 of file [FMC_dir.cpp](#).

Referenced by [FMC_isSocket\(\)](#).

Here is the caller graph for this function:



3.8.1.12 FMC_isSymLink_()

```
int FMC_isSymLink_ (  
    const char * path )
```

Definition at line 61 of file [FMC_dir.cpp](#).

Referenced by [FMC_isSymLink\(\)](#).

Here is the caller graph for this function:



3.9 FMC_dir.hpp

[Go to the documentation of this file.](#)

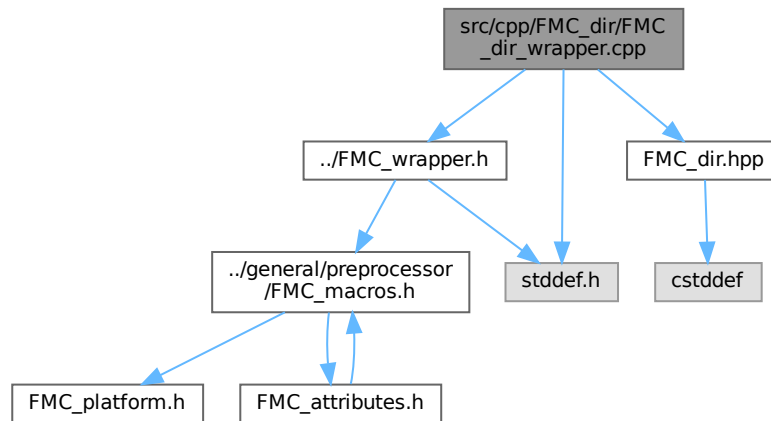
```

00001  /*
00002
00003  MIT License
00004
00005  Copyright (c) 2022-2023 Axel PASCON
00006
00007  Permission is hereby granted, free of charge, to any person obtaining a copy
00008  of this software and associated documentation files (the "Software"), to deal
00009  in the Software without restriction, including without limitation the rights
00010  to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011  copies of the Software, and to permit persons to whom the Software is
00012  furnished to do so, subject to the following conditions:
00013
00014  The above copyright notice and this permission notice shall be included in all
00015  copies or substantial portions of the Software.
00016
00017  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018  IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019  FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020  AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021  LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022  OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023  SOFTWARE.
00024
00025  */
00026
00027  #ifndef FMC_DIR_HPP
00028  #define FMC_DIR_HPP
00029
00030  #include <cstdint>
00031
00032  int FMC_dirExists_(const char *path);
00033  int FMC_isDir_(const char *path);
00034  int FMC_isRegFile_(const char *path);
00035  int FMC_isSymLink_(const char *path);
00036  int FMC_isBlock_(const char* path);
00037  int FMC_isCharFile_(const char* path);
00038  int FMC_isSocket_(const char* path);
00039  int FMC_isFIFO_(const char* path);
00040  int FMC_isOther_(const char* path);
00041  //char *FMC_readSymLink_(char *path_sym, const char * path);
00042  int FMC_isEmpty_(const char *path);
00043  //int FMC_createDir_(const char *path);
00044  char *FMC_getCurrentPath_(char *path, const size_t size);
00045  char *FMC_getAbsolutePath_(char *path, char *buffer, const size_t size);
00046
00047  #endif // FMC_DIR_HPP

```

3.10 src/cpp/FMC_dir/FMC_dir_wrapper.cpp File Reference

Include dependency graph for FMC_dir_wrapper.cpp:



Functions

- int [FMC_dirExists](#) (const char *path)
Checks if a directory exists.
- char * [FMC_getAbsolutePath](#) (char *path, char *buffer, const size_t size)
This function converts a relative path into an absolute one.
- char * [FMC_getCurrentPath](#) (char *path, const size_t size)
This function is equivalent to \$PWD in bash.
- int [FMC_isBlock](#) (const char *path)
Checks if a path is a block device.
- int [FMC_isCharFile](#) (const char *path)
Checks if a path is a character device.
- int [FMC_isDir](#) (const char *path)
Checks if a path is a directory.
- int [FMC_isEmpty](#) (const char *path)
Checks if a directory is empty.
- int [FMC_isFIFO](#) (const char *path)
Checks if a path is a FIFO.
- int [FMC_isOther](#) (const char *path)
Checks if a path is of an unknown type.
- int [FMC_isRegFile](#) (const char *path)
Checks if a path is a regular file.
- int [FMC_isSocket](#) (const char *path)
Checks if a path is a socket.
- int [FMC_isSymLink](#) (const char *path)
Checks if a path is a symbolic link.

3.10.1 Function Documentation

3.10.1.1 FMC_dirExists()

```
int FMC_dirExists (
    const char * path )
```

Checks if a directory exists.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

in	<i>path</i>	The path whose existence is to be checked.
----	-------------	--

Returns

An integer value.

Return values

1	if the directory exists.
0	if the directory does not exist.

Definition at line 35 of file [FMC_dir_wrapper.cpp](#).

References [FMC_dirExists_\(\)](#).

Here is the call graph for this function:



3.10.1.2 FMC_getAbsolutePath()

```
char * FMC_getAbsolutePath (
    char * path,
    char * buffer,
    const size_t size )
```

This function converts a relative path into an absolute one.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

in	<i>path</i>	The path to convert.
out	<i>buffer</i>	The memory buffer to store the absolute path.
in	<i>size</i>	The size of the memory buffer.

Returns

A pointer to the memory buffer.

Return values

<i>NULL</i>	if an error occurred.
<i>buffer</i>	The pointer to the buffer after the call if the function succeeded.

Definition at line 92 of file [FMC_dir_wrapper.cpp](#).

References [FMC_getAbsolutePath_\(\)](#).

Here is the call graph for this function:



3.10.1.3 FMC_getCurrentPath()

```
char * FMC_getCurrentPath (
    char * path,
    const size_t size )
```

This function is equivalent to \$PWD in bash.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

out	<i>path</i>	The memory buffer to store the current path.
in	<i>size</i>	The size of the memory buffer.

Returns

A pointer to the memory buffer.

Return values

<i>NULL</i>	if an error occurred.
<i>path</i>	The pointer to path after the call if the function succeeded.

Definition at line 87 of file [FMC_dir_wrapper.cpp](#).

References [FMC_getCurrentPath_\(\)](#).

Here is the call graph for this function:



3.10.1.4 FMC_isBlock()

```
int FMC_isBlock (
    const char * path )
```

Checks if a path is a block device.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

in	<i>path</i>	The path to check.
----	-------------	--------------------

Returns

An integer value.

Return values

1	if the path is a block device.
0	if the path is not a block device.
-1	if an error occurred.

Definition at line 55 of file [FMC_dir_wrapper.cpp](#).

References [FMC_isBlock_\(\)](#).

Here is the call graph for this function:



3.10.1.5 FMC_isCharFile()

```
int FMC_isCharFile (
    const char * path )
```

Checks if a path is a character device.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

in	<i>path</i>	The path to check.
----	-------------	--------------------

Returns

An integer value.

Return values

1	if the path is a character device.
0	if the path is not a character device.
-1	if an error occurred.

Definition at line 60 of file [FMC_dir_wrapper.cpp](#).

References [FMC_isCharFile_\(\)](#).

Here is the call graph for this function:



3.10.1.6 FMC_isDir()

```
int FMC_isDir (
    const char * path )
```

Checks if a path is a directory.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

in	<i>path</i>	The path to check.
----	-------------	--------------------

Returns

An integer value.

Return values

1	if the path is a directory.
0	if the path is not a directory.
-1	if an error occurred.

Definition at line 40 of file [FMC_dir_wrapper.cpp](#).

References [FMC_isDir_\(\)](#).

Here is the call graph for this function:



3.10.1.7 FMC_isEmpty()

```
int FMC_isEmpty (
    const char * path )
```

Checks if a directory is empty.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

in	<i>path</i>	The path to check.
----	-------------	--------------------

Returns

An integer value.

Return values

1	if the directory is empty.
0	if the directory is not empty.
-1	if an error occurred.

Definition at line 80 of file [FMC_dir_wrapper.cpp](#).

References [FMC_isEmpty_\(\)](#).

Here is the call graph for this function:



3.10.1.8 FMC_isFIFO()

```
int FMC_isFIFO (
    const char * path )
```

Checks if a path is a FIFO.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

in	<i>path</i>	The path to check.
----	-------------	--------------------

Returns

An integer value.

Return values

1	if the path is a FIFO.
0	if the path is not a FIFO.
-1	if an error occurred.

Definition at line 70 of file [FMC_dir_wrapper.cpp](#).

References [FMC_isFIFO_\(\)](#).

Here is the call graph for this function:



3.10.1.9 FMC_isOther()

```
int FMC_isOther (
    const char * path )
```

Checks if a path is of an unknown type.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

in	<i>path</i>	The path to check.
----	-------------	--------------------

Returns

An integer value.

Return values

1	if the path is of an unknown type.
0	if the path is not of an unknown type.
-1	if an error occurred.

Definition at line 75 of file [FMC_dir_wrapper.cpp](#).

References [FMC_isOther_\(\)](#).

Here is the call graph for this function:



3.10.1.10 FMC_isRegFile()

```
int FMC_isRegFile (
    const char * path )
```

Checks if a path is a regular file.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

in	<i>path</i>	The path to check.
----	-------------	--------------------

Returns

An integer value.

Return values

1	if the path is a regular file.
0	if the path is not a regular file.
-1	if an error occurred.

Definition at line 45 of file [FMC_dir_wrapper.cpp](#).

References [FMC_isRegFile_\(\)](#).

Here is the call graph for this function:



3.10.1.11 FMC_isSocket()

```
int FMC_isSocket (
    const char * path )
```

Checks if a path is a socket.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

in	<i>path</i>	The path to check.
----	-------------	--------------------

Returns

An integer value.

Return values

1	if the path is a socket.
0	if the path is not a socket.
-1	if an error occurred.

Definition at line 65 of file [FMC_dir_wrapper.cpp](#).

References [FMC_isSocket_\(\)](#).

Here is the call graph for this function:



3.10.1.12 FMC_isSymLink()

```
int FMC_isSymLink (
    const char * path )
```

Checks if a path is a symbolic link.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

in	<i>path</i>	The path to check.
----	-------------	--------------------

Returns

An integer value.

Return values

1	if the path is a symbolic link.
0	if the path is not a symbolic link.
-1	if an error occurred.

Definition at line 50 of file [FMC_dir_wrapper.cpp](#).

References [FMC_isSymLink_\(\)](#).

Here is the call graph for this function:



3.11 FMC_dir_wrapper.cpp

[Go to the documentation of this file.](#)


```

00001  /*
00002
00003  MIT License
00004
00005  Copyright (c) 2022-2023 Axel PASCON
00006
00007  Permission is hereby granted, free of charge, to any person obtaining a copy
00008  of this software and associated documentation files (the "Software"), to deal
00009  in the Software without restriction, including without limitation the rights
00010  to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011  copies of the Software, and to permit persons to whom the Software is
00012  furnished to do so, subject to the following conditions:
00013
00014  The above copyright notice and this permission notice shall be included in all
00015  copies or substantial portions of the Software.
00016
00017  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018  IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019  FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020  AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021  LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022  OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023  SOFTWARE.
00024
00025  */
00026
00027  #include "../FMC_wrapper.h"
00028  #include "FMC_dir.hpp"
00029  #include <stddef.h>
00030
00031  #ifdef __cplusplus
00032  extern "C" {
00033  #endif
00034
00035  FMC_SHARED int FMC_dirExists(const char *path)
00036  {
00037      return FMC_dirExists_(path);
00038  }
00039
00040  FMC_SHARED int FMC_isDir(const char *path)
00041  {
00042      return FMC_isDir_(path);
00043  }
00044
00045  FMC_SHARED int FMC_isRegFile(const char *path)
00046  {
00047      return FMC_isRegFile_(path);
00048  }
00049
00050  FMC_SHARED int FMC_isSymLink(const char *path)
00051  {
00052      return FMC_isSymLink_(path);
00053  }
00054
00055  FMC_SHARED int FMC_isBlock(const char* path)
00056  {
00057      return FMC_isBlock_(path);
00058  }
00059
00060  FMC_SHARED int FMC_isCharFile(const char* path)
00061  {
00062      return FMC_isCharFile_(path);
00063  }
00064
00065  FMC_SHARED int FMC_isSocket(const char* path)
00066  {
00067      return FMC_isSocket_(path);
00068  }
00069
00070  FMC_SHARED int FMC_isFIFO(const char* path)
00071  {
00072      return FMC_isFIFO_(path);
00073  }
00074
00075  FMC_SHARED int FMC_isOther(const char* path)
00076  {
00077      return FMC_isOther_(path);
00078  }
00079
00080  FMC_SHARED int FMC_isEmpty(const char *path)
00081  {
00082      return FMC_isEmpty_(path);
00083  }
00084
00085  //FMC_SHARED int FMC_createDir_(const char *path);
00086
00087  FMC_SHARED char *FMC_getCurrentPath(char *path, const size_t size)

```

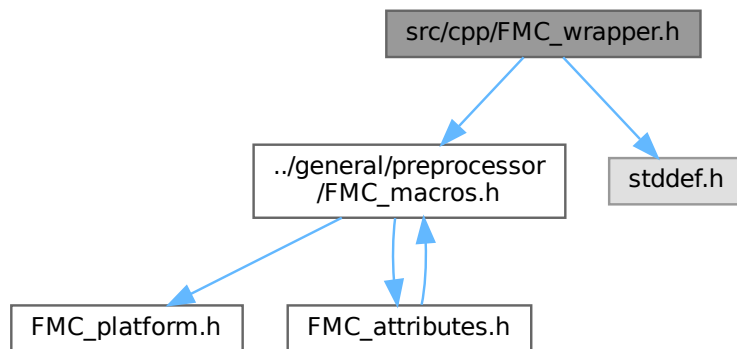
```

00088 {
00089     return FMC_getCurrentPath_(path, size);
00090 }
00091
00092 FMC_SHARED char *FMC_getAbsolutePath(char *path, char *buffer, const size_t size)
00093 {
00094     return FMC_getAbsolutePath_(path, buffer, size);
00095 }
00096
00097 #ifdef __cplusplus
00098 }
00099 #endif

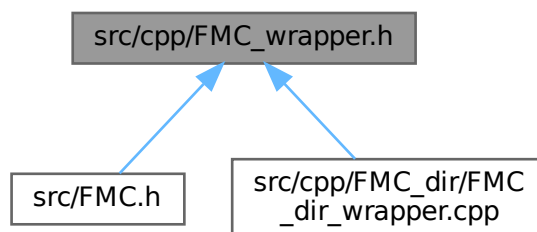
```

3.12 src/cpp/FMC_wrapper.h File Reference

Include dependency graph for FMC_wrapper.h:



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



Functions

- int `FMC_dirExists` (const char *path)

- Checks if a directory exists.*

 - char * [FMC_getAbsolutePath](#) (char *path, char *buffer, const size_t size)

This function converts a relative path into an absolute one.
- char * [FMC_getCurrentPath](#) (char *path, const size_t size)

This function is equivalent to \$PWD in bash.
- int [FMC_isBlock](#) (const char *path)

Checks if a path is a block device.
- int [FMC_isCharFile](#) (const char *path)

Checks if a path is a character device.
- int [FMC_isDir](#) (const char *path)

Checks if a path is a directory.
- int [FMC_isEmpty](#) (const char *path)

Checks if a directory is empty.
- int [FMC_isFIFO](#) (const char *path)

Checks if a path is a FIFO.
- int [FMC_isOther](#) (const char *path)

Checks if a path is of an unknown type.
- int [FMC_isRegFile](#) (const char *path)

Checks if a path is a regular file.
- int [FMC_isSocket](#) (const char *path)

Checks if a path is a socket.
- int [FMC_isSymLink](#) (const char *path)

Checks if a path is a symbolic link.

3.12.1 Function Documentation

3.12.1.1 FMC_dirExists()

```
int FMC_dirExists (
    const char * path )
```

Checks if a directory exists.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

in	<i>path</i>	The path whose existence is to be checked.
----	-------------	--

Returns

An integer value.

Return values

1	if the directory exists.
0	if the directory does not exist.

Definition at line 35 of file [FMC_dir_wrapper.cpp](#).

References [FMC_dirExists_\(\)](#).

Here is the call graph for this function:

**3.12.1.2 FMC_getAbsolutePath()**

```

char * FMC_getAbsolutePath (
    char * path,
    char * buffer,
    const size_t size )
  
```

This function converts a relative path into an absolute one.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

in	<i>path</i>	The path to convert.
out	<i>buffer</i>	The memory buffer to store the absolute path.
in	<i>size</i>	The size of the memory buffer.

Returns

A pointer to the memory buffer.

Return values

<i>NULL</i>	if an error occurred.
<i>buffer</i>	The pointer to the buffer after the call if the function succeeded.

Definition at line 92 of file [FMC_dir_wrapper.cpp](#).

References [FMC_getAbsolutePath_\(\)](#).

Here is the call graph for this function:

**3.12.1.3 FMC_getCurrentPath()**

```
char * FMC_getCurrentPath (
    char * path,
    const size_t size )
```

This function is equivalent to \$PWD in bash.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

out	<i>path</i>	The memory buffer to store the current path.
in	<i>size</i>	The size of the memory buffer.

Returns

A pointer to the memory buffer.

Return values

<i>NULL</i>	if an error occurred.
<i>path</i>	The pointer to path after the call if the function succeeded.

Definition at line 87 of file [FMC_dir_wrapper.cpp](#).

References [FMC_getCurrentPath_\(\)](#).

Here is the call graph for this function:

**3.12.1.4 FMC_isBlock()**

```
int FMC_isBlock (
    const char * path )
```

Checks if a path is a block device.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

in	<i>path</i>	The path to check.
----	-------------	--------------------

Returns

An integer value.

Return values

<i>1</i>	if the path is a block device.
<i>0</i>	if the path is not a block device.
<i>-1</i>	if an error occurred.

Definition at line 55 of file [FMC_dir_wrapper.cpp](#).

References [FMC_isBlock_\(\)](#).

Here is the call graph for this function:

**3.12.1.5 FMC_isCharFile()**

```
int FMC_isCharFile (
    const char * path )
```

Checks if a path is a character device.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

<i>in</i>	<i>path</i>	The path to check.
-----------	-------------	--------------------

Returns

An integer value.

Return values

<i>1</i>	if the path is a character device.
<i>0</i>	if the path is not a character device.
<i>-1</i>	if an error occurred.

Definition at line 60 of file [FMC_dir_wrapper.cpp](#).

References [FMC_isCharFile_\(\)](#).

Here is the call graph for this function:

**3.12.1.6 FMC_isDir()**

```
int FMC_isDir (  
    const char * path )
```

Checks if a path is a directory.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

<i>in</i>	<i>path</i>	The path to check.
-----------	-------------	--------------------

Returns

An integer value.

Return values

<i>1</i>	if the path is a directory.
<i>0</i>	if the path is not a directory.
<i>-1</i>	if an error occurred.

Definition at line 40 of file [FMC_dir_wrapper.cpp](#).

References [FMC_isDir_\(\)](#).

Here is the call graph for this function:

**3.12.1.7 FMC_isEmpty()**

```
int FMC_isEmpty (
    const char * path )
```

Checks if a directory is empty.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

<i>in</i>	<i>path</i>	The path to check.
-----------	-------------	--------------------

Returns

An integer value.

Return values

<i>1</i>	if the directory is empty.
<i>0</i>	if the directory is not empty.
<i>-1</i>	if an error occurred.

Definition at line 80 of file [FMC_dir_wrapper.cpp](#).

References [FMC_isEmpty_\(\)](#).

Here is the call graph for this function:

**3.12.1.8 FMC_isFIFO()**

```
int FMC_isFIFO (
    const char * path )
```

Checks if a path is a FIFO.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

<i>in</i>	<i>path</i>	The path to check.
-----------	-------------	--------------------

Returns

An integer value.

Return values

<i>1</i>	if the path is a FIFO.
<i>0</i>	if the path is not a FIFO.
<i>-1</i>	if an error occurred.

Definition at line 70 of file [FMC_dir_wrapper.cpp](#).

References [FMC_isFIFO_\(\)](#).

Here is the call graph for this function:

**3.12.1.9 FMC_isOther()**

```
int FMC_isOther (
    const char * path )
```

Checks if a path is of an unknown type.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

<i>in</i>	<i>path</i>	The path to check.
-----------	-------------	--------------------

Returns

An integer value.

Return values

<i>1</i>	if the path is of an unknown type.
<i>0</i>	if the path is not of an unknown type.
<i>-1</i>	if an error occurred.

Definition at line 75 of file [FMC_dir_wrapper.cpp](#).

References [FMC_isOther_\(\)](#).

Here is the call graph for this function:

**3.12.1.10 FMC_isRegFile()**

```
int FMC_isRegFile (  
    const char * path )
```

Checks if a path is a regular file.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

<i>in</i>	<i>path</i>	The path to check.
-----------	-------------	--------------------

Returns

An integer value.

Return values

<i>1</i>	if the path is a regular file.
<i>0</i>	if the path is not a regular file.
<i>-1</i>	if an error occurred.

Definition at line 45 of file [FMC_dir_wrapper.cpp](#).

References [FMC_isRegFile_\(\)](#).

Here is the call graph for this function:

**3.12.1.11 FMC_isSocket()**

```
int FMC_isSocket (
    const char * path )
```

Checks if a path is a socket.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

<i>in</i>	<i>path</i>	The path to check.
-----------	-------------	--------------------

Returns

An integer value.

Return values

<i>1</i>	if the path is a socket.
<i>0</i>	if the path is not a socket.
<i>-1</i>	if an error occurred.

Definition at line 65 of file [FMC_dir_wrapper.cpp](#).

References [FMC_isSocket_\(\)](#).

Here is the call graph for this function:

**3.12.1.12 FMC_isSymLink()**

```
int FMC_isSymLink (  
    const char * path )
```

Checks if a path is a symbolic link.

Author

Axel PASCON

Date

2023

This function is a wrapper around the C++ filesystem library associated function.

Parameters

<i>in</i>	<i>path</i>	The path to check.
-----------	-------------	--------------------

Returns

An integer value.

Return values

1	if the path is a symbolic link.
0	if the path is not a symbolic link.
-1	if an error occurred.

Definition at line 50 of file [FMC_dir_wrapper.cpp](#).

References [FMC_isSymLink_\(\)](#).

Here is the call graph for this function:



3.13 FMC_wrapper.h

[Go to the documentation of this file.](#)

```

00001  /*
00002
00003  MIT License
00004
00005  Copyright (c) 2022-2023 Axel PASCON
00006
00007  Permission is hereby granted, free of charge, to any person obtaining a copy
00008  of this software and associated documentation files (the "Software"), to deal
00009  in the Software without restriction, including without limitation the rights
00010  to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011  copies of the Software, and to permit persons to whom the Software is
00012  furnished to do so, subject to the following conditions:
00013
00014  The above copyright notice and this permission notice shall be included in all
00015  copies or substantial portions of the Software.
00016
00017  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018  IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019  FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020  AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021  LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022  OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023  SOFTWARE.
00024
00025  */
00026
00027  #ifndef FMC_WRAPPER_H
00028  #define FMC_WRAPPER_H
00029
00030  #ifdef __cplusplus
00031  extern "C" {
00032  #endif
00033
00034  #include "../general/preprocessor/FMC_macros.h"
00035
00036  #include <stddef.h>
00037

```

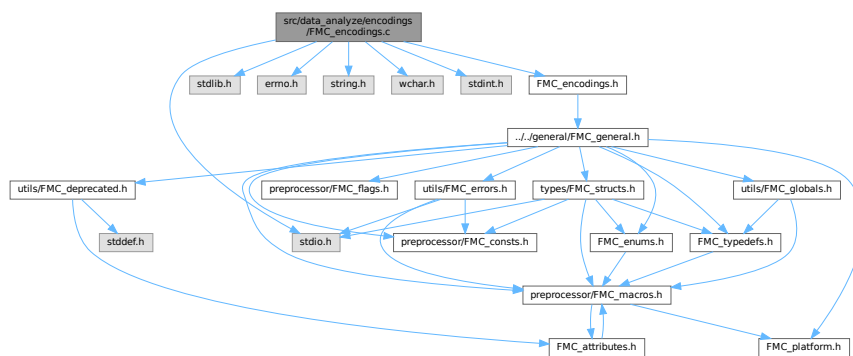
```

00038 // FMC_dir
00052 FMC_SHARED int FMC_dirExists(const char *path);
00066 FMC_SHARED int FMC_isDir(const char *path);
00080 FMC_SHARED int FMC_isRegFile(const char *path);
00094 FMC_SHARED int FMC_isSymLink(const char *path);
00108 FMC_SHARED int FMC_isBlock(const char* path);
00122 FMC_SHARED int FMC_isCharFile(const char* path);
00136 FMC_SHARED int FMC_isSocket(const char* path);
00150 FMC_SHARED int FMC_isFIFO(const char* path);
00164 FMC_SHARED int FMC_isOther(const char* path);
00178 FMC_SHARED int FMC_isEmpty(const char *path);
00192 FMC_SHARED char *FMC_getCurrentPath(char *path, const size_t size);
00207 FMC_SHARED char *FMC_getAbsolutePath(char *path, char *buffer, const size_t size);
00208 // !FMC_dir
00209
00210 #ifdef __cplusplus
00211 }
00212 #endif
00213
00214
00215 #endif // FMC_WRAPPER_H

```

3.14 src/data_analyze/encodings/FMC_encodings.c File Reference

Include dependency graph for FMC_encodings.c:



Macros

- `#define __STDC_WANT_LIB_EXT1__ 1`

Functions

- `FMC_FUNC_CONST FMC_Encodings FMC_checkEncodingFlag` (int encoding)
- `FMC_Encodings FMC_getEncoding` (FILE *file)

3.14.1 Macro Definition Documentation

3.14.1.1 __STDC_WANT_LIB_EXT1__

```
#define __STDC_WANT_LIB_EXT1__ 1
```

Definition at line 26 of file [FMC_encodings.c](#).

3.14.2 Function Documentation

3.14.2.1 FMC_checkEncodingFlag()

```
FMC_FUNC_CONST FMC_Encodings FMC_checkEncodingFlag (
    int encoding )
```

Definition at line 204 of file [FMC_encodings.c](#).

References [ASCII](#), [ascii](#), [error](#), [unknown](#), [UTF16_BE](#), [utf16_be](#), [UTF16_LE](#), [utf16_le](#), [UTF32_BE](#), [utf32_be](#), [UTF32_LE](#), [utf32_le](#), [UTF8](#), [utf8](#), [UTF8_BOM](#), and [utf8_bom](#).

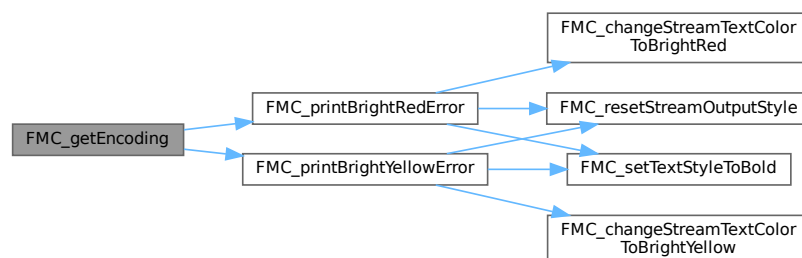
3.14.2.2 FMC_getEncoding()

```
FMC_Encodings FMC_getEncoding (
    FILE * file )
```

Definition at line 36 of file [FMC_encodings.c](#).

References [ascii](#), [error](#), [FMC_makeMsg](#), [FMC_printBrightRedError\(\)](#), [FMC_printBrightYellowError\(\)](#), [unknown](#), [utf16_be](#), [utf16_le](#), [utf32_be](#), [utf32_le](#), [utf8](#), and [utf8_bom](#).

Here is the call graph for this function:



3.15 FMC_encodings.c

[Go to the documentation of this file.](#)

```

00001  /*
00002
00003  MIT License
00004
00005  Copyright (c) 2023 Axel PASCON
00006
00007  Permission is hereby granted, free of charge, to any person obtaining a copy
00008  of this software and associated documentation files (the "Software"), to deal
00009  in the Software without restriction, including without limitation the rights
00010  to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011  copies of the Software, and to permit persons to whom the Software is
00012  furnished to do so, subject to the following conditions:
00013
00014  The above copyright notice and this permission notice shall be included in all
00015  copies or substantial portions of the Software.
00016
00017  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018  IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019  FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020  AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021  LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022  OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023  SOFTWARE.
00024
00025  */
00026  #define __STDC_WANT_LIB_EXT1__ 1
00027  #include <stdio.h>
00028  #include <stdlib.h>
00029  #include <errno.h>
00030  #include <string.h>
00031  #include <wchar.h> // fwide
00032  #include <stdint.h>
00033
00034  #include "FMC_encodings.h"
00035
00036  FMC_SHARED FMC_FUNC_WARN_UNUSED_RESULT FMC_FUNC_NONNULL(1) FMC_Encodings FMC_getEncoding(FILE *file)
00037  {
00038      #pragma GCC diagnostic ignored "-Wnonnull-compare" // get an error at compile time without this
00039      (because of attribute nonnull)
00040      if (file == NULL)
00041      {
00042          if (FMC_ENABLE_DEBUG)
00043          {
00044              FMC_makeMsg(err_null, 4, "ERROR : ", "In function : ", __func__, ". The provided file must
not be NULL.");
00045              FMC_printBrightRedError(stderr, err_null);
00046          }
00047          return error;
00048      }
00049      #pragma GCC diagnostic pop
00050      // check orientation
00051      if (fwide(file, -1) >= 0)
00052      {
00053          if (FMC_ENABLE_DEBUG)
00054          {
00055              FMC_makeMsg(err_wide, 4, "ERROR : ", "In function : ", __func__, ". The provided file must
be opened with by orientation.");
00056              FMC_printBrightRedError(stderr, err_wide);
00057          }
00058          return error;
00059      }
00060      long long sizeOfFile = 0;
00061      if (fseek(file, 0, SEEK_END))
00062      {
00063          FMC_makeMsg(err_seek_1, 4, "FMC INTERNAL ERROR : ", "In function : ", __func__, ". fseek
failure.");
00064          FMC_printBrightRedError(stderr, err_seek_1);
00065          return error;
00066      }
00067      errno = 0;
00068      sizeOfFile = ftell(file);
00069      if (errno || sizeOfFile == -1L)
00070      {
00071          FMC_makeMsg(err_tell, 5, "FMC INTERNAL ERROR : ", "In function : ", __func__, ". ftell
failure.", strerror(errno));
00072          FMC_printBrightRedError(stderr, err_tell);
00073          return error;
00074      }
00075      rewind(file);

```

```

00078     char buff[4] = {0};
00079     // 1st if
00080     if(sizeofFile < 0) // no error, must have overflowed
00081     {
00082         sizeofFile = SIZE_MAX;
00083         size_t ret = fread(buff, 1, 4, file);
00084         if(ret != 4) goto check_error_type_1;
00085         else if (ret == 4) goto end_check_1;
00086         else return error;
00087     }
00088
00089     // 2nd if
00090     else if (sizeofFile <= 4 && sizeofFile >= 0)
00091     {
00092         size_t ret = fread(buff, 1, sizeofFile, file);
00093         if(ret != (size_t) sizeofFile) goto check_error_type_1;
00094         else if (ret == (size_t) sizeofFile) goto end_check_1;
00095         else return error;
00096
00097         check_error_type_1 :
00098         FMC_LABEL_COLD;
00099         if (feof(file))
00100         {
00101             FMC_makeMsg(err_eof, 4, "FMC INTERNAL ERROR : ", "In function : ", __func__, ". EOF
indicator set.");
00102             FMC_printBrightRedError(stderr, err_eof);
00103             return error;
00104         }
00105         else if (ferror(file))
00106         {
00107             FMC_makeMsg(err_ferror, 5, "FMC INTERNAL ERROR : ", "In function : ", __func__, ". Error
indicator set.", strerror(errno));
00108             FMC_printBrightRedError(stderr, err_ferror);
00109             return error;
00110         }
00111         else
00112         {
00113             FMC_makeMsg(err_fread, 4, "FMC INTERNAL ERROR : ", "In function : ", __func__, ". fread
failure.");
00114             FMC_printBrightRedError(stderr, err_fread);
00115             return error;
00116         }
00117     }
00118 }
00119
00120 // 3rd if
00121 else if(fread(buff, 1, 4, file) != 4)
00122 {
00123     if (feof(file))
00124     {
00125         FMC_makeMsg(err_eof, 4, "FMC INTERNAL ERROR : ", "In function : ", __func__, ". EOF
indicator set.");
00126         FMC_printBrightRedError(stderr, err_eof);
00127         return error;
00128     }
00129     else if (ferror(file))
00130     {
00131         FMC_makeMsg(err_ferror, 5, "FMC INTERNAL ERROR : ", "In function : ", __func__, ". Error
indicator set.", strerror(errno));
00132         FMC_printBrightRedError(stderr, err_ferror);
00133         return error;
00134     }
00135     else
00136     {
00137         FMC_makeMsg(err_fread, 4, "FMC INTERNAL ERROR : ", "In function : ", __func__, ". fread
failure.");
00138         FMC_printBrightRedError(stderr, err_fread);
00139         return error;
00140     }
00141 }
00142
00143 end_check_1 :
00144 FMC_LABEL_HOT;
00145 if (sizeofFile >= 3 && (unsigned char) buff[0] == 0xEF && (unsigned char) buff[1] == 0xBB &&
(unsigned char) buff[2] == 0xBF)
00146 {
00147     rewind(file);
00148     return utf8_bom;
00149 }
00150 else if (sizeofFile >= 2 && (unsigned char) buff[0] == 0xFF && (unsigned char) buff[1] == 0xFE)
00151 {
00152     rewind(file);
00153     return utf16_le;
00154 }
00155 else if (sizeofFile >= 2 && (unsigned char) buff[0] == 0xFE && (unsigned char) buff[1] == 0xFF)
00156 {
00157     rewind(file);

```

```

00158         return utf16_be;
00159     }
00160     else if (sizeofFile >= 4 && (unsigned char) buff[0] == 0x00 && (unsigned char) buff[1] == 0x00 &&
(unsigned char) buff[2] == 0xFE && (unsigned char) buff[3] == 0xFF)
00161     {
00162         rewind(file);
00163         return utf32_be;
00164     }
00165     else if (sizeofFile >= 4 && (unsigned char) buff[0] == 0xFF && (unsigned char) buff[1] == 0xFE &&
(unsigned char) buff[2] == 0x00 && (unsigned char) buff[3] == 0x00)
00166     {
00167         rewind(file);
00168         return utf32_le;
00169     }
00170     else
00171     {
00172         rewind(file);
00173         if (sizeofFile == 0)
00174         {
00175             rewind(file);
00176             if (FMC_ENABLE_DEBUG)
00177             {
00178                 FMC_makeMsg(err_empty, 4, "WARNING : ", "In function : ", __func__, ". The provided
file is empty.");
00179                 FMC_printBrightYellowError(stderr, err_empty);
00180             }
00181             return unknown;
00182         }
00183
00184         char currentChar = 0;
00185         size_t cpt = 0;
00186         while((currentChar = fgetc(file)) != EOF)
00187         {
00188             if (currentChar != EOF && (unsigned char) currentChar > 127)
00189             {
00190                 rewind(file);
00191                 return utf8;
00192             }
00193             cpt++;
00194             if ((long long) cpt >= sizeofFile)
00195             {
00196                 break;
00197             }
00198         }
00199         rewind(file);
00200         return ascii;
00201     }
00202 }
00203
00204 FMC_SHARED FMC_FUNC_CONST FMC_Encodings FMC_checkEncodingFlag(int encoding)
00205 {
00206     switch (encoding)
00207     {
00208     case ASCII:
00209         return ascii;
00210         break;
00211     case UTF8:
00212         return utf8;
00213         break;
00214     case UTF8_BOM:
00215         return utf8_bom;
00216         break;
00217     case UTF16_LE:
00218         return utf16_le;
00219         break;
00220     case UTF16_BE:
00221         return utf16_be;
00222         break;
00223     case UTF32_LE:
00224         return utf32_le;
00225         break;
00226     case UTF32_BE:
00227         return utf32_be;
00228         break;
00229     default: // TODO : add error in case of unknown encoding
00230         return unknown;
00231         break;
00232     }
00233     return error;
00234 }
00235
00236 /*FMC_SHARED FMC_Char FMC_getc(FMC_File file)
00237 {
00238     FMC_Char c = {.encoding = file.encoding, .comp = {.mostLeft = 0, .middleLeft = 0, .middleRight =
0, .mostRight = 0}, .isNull = 0};
00239     if(file.file == NULL || file.encoding == error || file.encoding == unknown)
00240     {

```

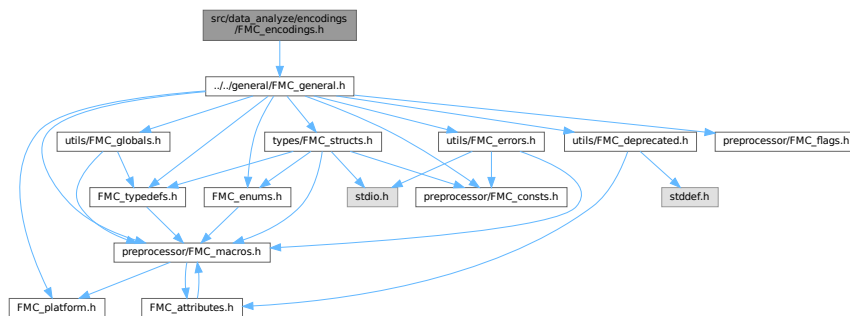
```

00241         c.isNull = 1;
00242         return c;
00243     }
00244     else if (fwide(file.file, 0) > 0)
00245     {
00246         fprintf(stderr, "Error: file is wide oriented when trying to read with byte orientation\n");
00247         c.isNull = 1;
00248         return c;
00249     }
00250     else if (file.encoding == ascii)
00251     {
00252         if (!feof(file.file))
00253         {
00254             }
00255         }
00256     }
00257 }
00258 */

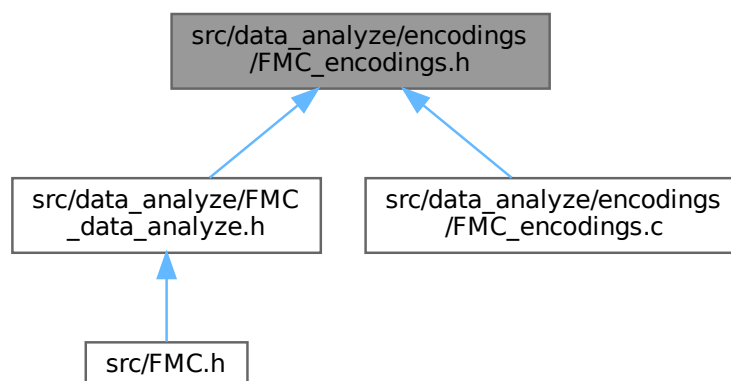
```

3.16 src/data_analyze/encodings/FMC_encodings.h File Reference

Include dependency graph for FMC_encodings.h:



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



Macros

- `#define FMC_ENCODINGS`

Functions

- `FMC_FUNC_CONST FMC_Encodings FMC_checkEncodingFlag` (int encoding)
- `FMC_Encodings FMC_getEncoding` (FILE *file)

3.16.1 Macro Definition Documentation

3.16.1.1 FMC_ENCODINGS

```
#define FMC_ENCODINGS
```

Definition at line 30 of file [FMC_encodings.h](#).

3.16.2 Function Documentation

3.16.2.1 FMC_checkEncodingFlag()

```
FMC_FUNC_CONST FMC_Encodings FMC_checkEncodingFlag (
    int encoding )
```

Definition at line 204 of file [FMC_encodings.c](#).

References [ASCII](#), [ascii](#), [error](#), [unknown](#), [UTF16_BE](#), [utf16_be](#), [UTF16_LE](#), [utf16_le](#), [UTF32_BE](#), [utf32_be](#), [UTF32_LE](#), [utf32_le](#), [UTF8](#), [utf8](#), [UTF8_BOM](#), and [utf8_bom](#).

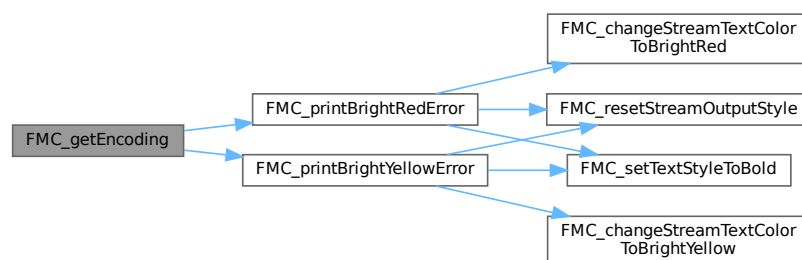
3.16.2.2 FMC_getEncoding()

```
FMC_Encodings FMC_getEncoding (
    FILE * file )
```

Definition at line 36 of file [FMC_encodings.c](#).

References [ascii](#), [error](#), [FMC_makeMsg](#), [FMC_printBrightRedError\(\)](#), [FMC_printBrightYellowError\(\)](#), [unknown](#), [utf16_be](#), [utf16_le](#), [utf32_be](#), [utf32_le](#), [utf8](#), and [utf8_bom](#).

Here is the call graph for this function:



3.17 FMC_encodings.h

[Go to the documentation of this file.](#)

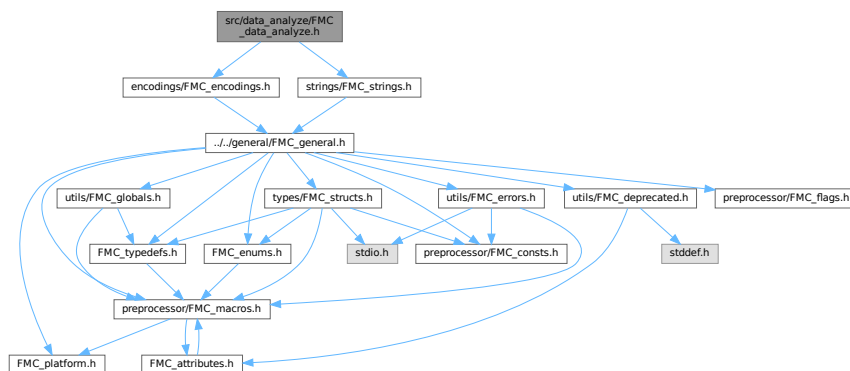
```

00001  /*
00002
00003  MIT License
00004
00005  Copyright (c) 2023 Axel PASCON
00006
00007  Permission is hereby granted, free of charge, to any person obtaining a copy
00008  of this software and associated documentation files (the "Software"), to deal
00009  in the Software without restriction, including without limitation the rights
00010  to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011  copies of the Software, and to permit persons to whom the Software is
00012  furnished to do so, subject to the following conditions:
00013
00014  The above copyright notice and this permission notice shall be included in all
00015  copies or substantial portions of the Software.
00016
00017  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018  IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019  FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020  AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021  LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022  OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023  SOFTWARE.
00024
00025  */
00026
00027  #pragma once
00028
00029  #ifndef FMC_ENCODINGS
00030  #define FMC_ENCODINGS
00031
00032  #include "../general/FMC_general.h"
00033
00034  FMC_SHARED FMC_FUNC_WARN_UNUSED_RESULT FMC_FUNC_NONNULL(1) FMC_Encodings FMC_getEncoding(FILE *file);
00035  FMC_SHARED FMC_FUNC_CONST FMC_Encodings FMC_checkEncodingFlag(int encoding);
00036
00037  #endif // FMC_ENCODINGS

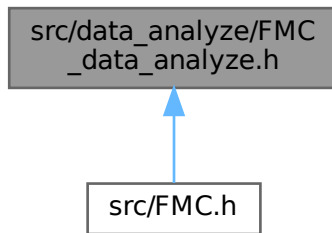
```

3.18 src/data_analyze/FMC_data_analyze.h File Reference

Include dependency graph for FMC_data_analyze.h:



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



Macros

- `#define FMC_DATA_ANALYZE_H`

3.18.1 Macro Definition Documentation

3.18.1.1 FMC_DATA_ANALYZE_H

```
#define FMC_DATA_ANALYZE_H
```

Definition at line 30 of file [FMC_data_analyze.h](#).

3.19 FMC_data_analyze.h

[Go to the documentation of this file.](#)

```

00001  /*
00002
00003  MIT License
00004
00005  Copyright (c) 2022 Axel PASCON
00006
00007  Permission is hereby granted, free of charge, to any person obtaining a copy
00008  of this software and associated documentation files (the "Software"), to deal
00009  in the Software without restriction, including without limitation the rights
00010  to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011  copies of the Software, and to permit persons to whom the Software is
00012  furnished to do so, subject to the following conditions:
00013
00014  The above copyright notice and this permission notice shall be included in all
00015  copies or substantial portions of the Software.
00016
00017  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018  IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019  FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020  AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021  LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022  OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023  SOFTWARE.
00024

```



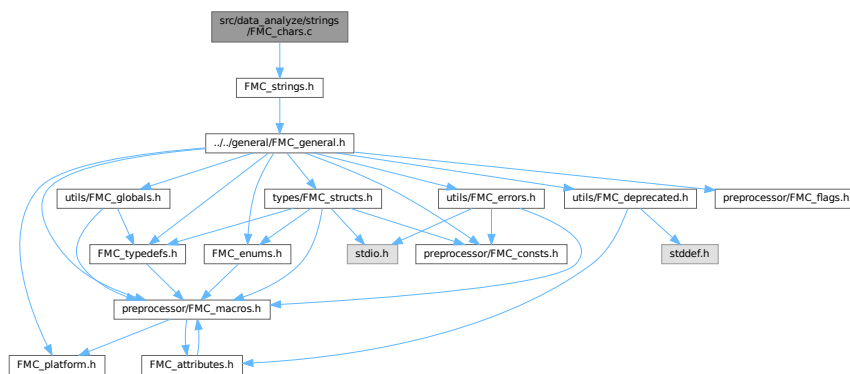
```

00025 */
00026
00027 #pragma once
00028
00029 #ifndef FMC_DATA_ANALYZE_H
00030 #define FMC_DATA_ANALYZE_H
00031
00032
00033 #include "encodings/FMC_encodings.h"
00034 #include "strings/FMC_strings.h"
00035
00036 #endif // FMC_DATA_ANALYZE_H

```

3.20 src/data_analyze/strings/FMC_chars.c File Reference

Include dependency graph for FMC_chars.c:



3.21 FMC_chars.c

[Go to the documentation of this file.](#)

```

00001 /*
00002
00003 MIT License
00004
00005 Copyright (c) 2022-2023 Axel PASCON
00006
00007 Permission is hereby granted, free of charge, to any person obtaining a copy
00008 of this software and associated documentation files (the "Software"), to deal
00009 in the Software without restriction, including without limitation the rights
00010 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011 copies of the Software, and to permit persons to whom the Software is
00012 furnished to do so, subject to the following conditions:
00013
00014 The above copyright notice and this permission notice shall be included in all
00015 copies or substantial portions of the Software.
00016
00017 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023 SOFTWARE.
00024
00025 */
00026
00027 #include "FMC_strings.h"
00028
00029 /* FMC_SHARED FMC_FUNC_NONNULL(1) FMC_FUNC_HOT FMC_Char FMC_getChar(FMC_File *file)
00030 {
00031     #pragma GCC diagnostic ignored "-Wnonnull-compare"

```

```

00032     if (file == NULL)
00033     {
00034         FMC_makeMsg(err_nullarg, 3, "ERROR : In function : ", __func__, " : the provided file pointer
is NULL");
00035         FMC_printRedError(stderr, err_nullarg);
00036     }
00037 } */

```

3.22 src/data_analyze/strings/FMC_strings.c File Reference

3.23 FMC_strings.c

[Go to the documentation of this file.](#)

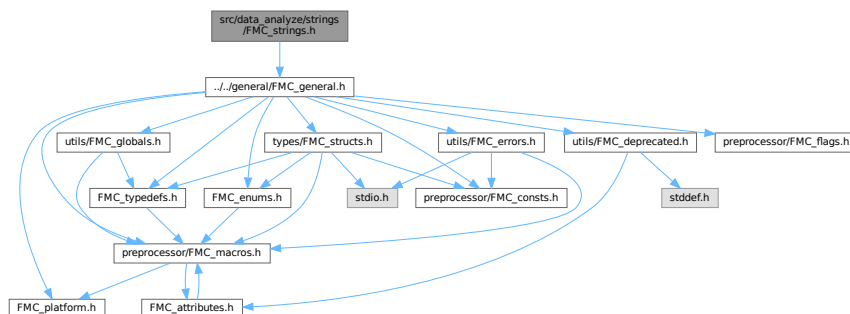
```

00001 /*
00002
00003 MIT License
00004
00005 Copyright (c) 2022 Axel PASCON
00006
00007 Permission is hereby granted, free of charge, to any person obtaining a copy
00008 of this software and associated documentation files (the "Software"), to deal
00009 in the Software without restriction, including without limitation the rights
00010 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011 copies of the Software, and to permit persons to whom the Software is
00012 furnished to do so, subject to the following conditions:
00013
00014 The above copyright notice and this permission notice shall be included in all
00015 copies or substantial portions of the Software.
00016
00017 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023 SOFTWARE.
00024
00025 */
00026

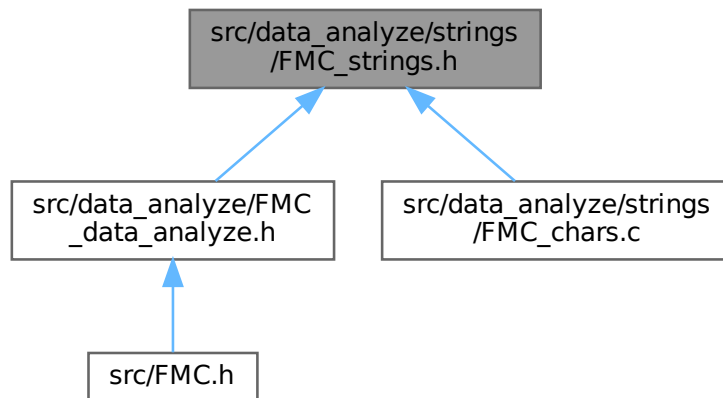
```

3.24 src/data_analyze/strings/FMC_strings.h File Reference

Include dependency graph for FMC_strings.h:



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



Macros

- `#define FMC_STRINGS_H`

3.24.1 Macro Definition Documentation

3.24.1.1 FMC_STRINGS_H

```
#define FMC_STRINGS_H
```

Definition at line 30 of file [FMC_strings.h](#).

3.25 FMC_strings.h

[Go to the documentation of this file.](#)

```

00001  /*
00002
00003  MIT License
00004
00005  Copyright (c) 2022 Axel PASCON
00006
00007  Permission is hereby granted, free of charge, to any person obtaining a copy
00008  of this software and associated documentation files (the "Software"), to deal
00009  in the Software without restriction, including without limitation the rights
00010  to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011  copies of the Software, and to permit persons to whom the Software is
00012  furnished to do so, subject to the following conditions:
00013
00014  The above copyright notice and this permission notice shall be included in all
00015  copies or substantial portions of the Software.
00016

```

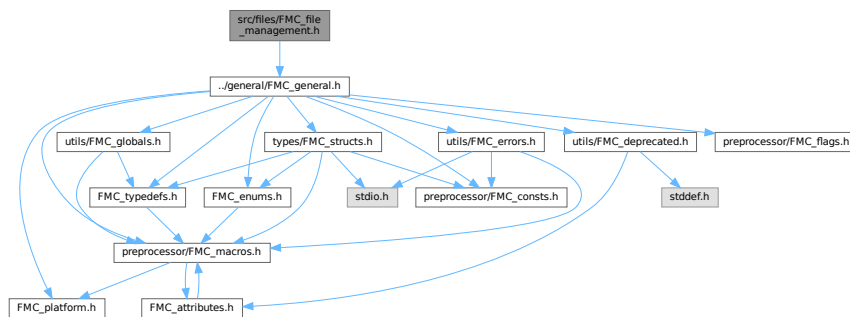
```

00017 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023 SOFTWARE.
00024
00025 */
00026
00027 #pragma once
00028
00029 #ifndef FMC_STRINGS_H
00030 #define FMC_STRINGS_H
00031
00032 #include "../general/FMC_general.h"
00033
00034
00035 #endif // FMC_STRINGS_H

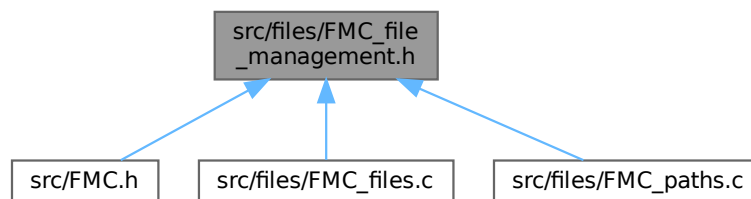
```

3.26 src/files/FMC_file_management.h File Reference

Include dependency graph for FMC_file_management.h:



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



Macros

- `#define FMC_FILE_MANAGEMENT_H`

Functions

- char * [FMC_cutFilename](#) (const char *restrict const path, char *restrict dirs, const size_t dirs_size)
- char * [FMC_extractFilename](#) (const char *restrict const path, char *restrict filename, const size_t filename_size)
Gets the filename from a complete path.
- char * [FMC_getExtension](#) (const char *restrict const path, char *restrict ext, const size_t ext_size)

3.26.1 Macro Definition Documentation

3.26.1.1 FMC_FILE_MANAGEMENT_H

```
#define FMC_FILE_MANAGEMENT_H
```

Definition at line 30 of file [FMC_file_management.h](#).

3.26.2 Function Documentation

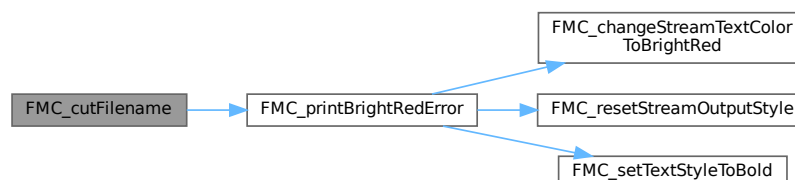
3.26.2.1 FMC_cutFilename()

```
char * FMC_cutFilename (
    const char *restrict const path,
    char *restrict dirs,
    const size_t dirs_size )
```

Definition at line 119 of file [FMC_paths.c](#).

References [FMC_makeMsg](#), [FMC_printBrightRedError\(\)](#), [MAX_FEXT_SIZE](#), [MAX_FNAME_SIZE](#), and [MAX_FPATH_SIZE](#).

Here is the call graph for this function:



3.26.2.2 FMC_extractFilename()

```
char * FMC_extractFilename (
    const char *restrict const path,
    char *restrict filename,
    const size_t filename_size )
```

Gets the filename from a complete path.

Author

Axel PASCON

Date

2023

Basically, this function only detects the last '/' or '\' character. For example, if the path is "C:\\Users\\someone\\Documents\\MyFile.txt", the function will return "MyFile.txt". If the path is "/home/someone/Desktop", then Desktop will be considered as the filename. This function is designed to only operate on strings, and do not check if the path is valid, is a file or a directory, etc.

Parameters

in	<i>path</i>	The path to extract the filename from.
out	<i>filename</i>	The buffer where the filename will be stored.
in	<i>filename_size</i>	The size of the filename buffer.

Returns

A pointer to the filename buffer.

Return values

<i>NULL</i>	If the path is NULL, if the filename buffer is NULL or if an error occurred. The error can be viewed by setting FMC_ENABLE_DEBUG to True .
-------------	--

Warning

The filename buffer must be at least as big as the path.

Note

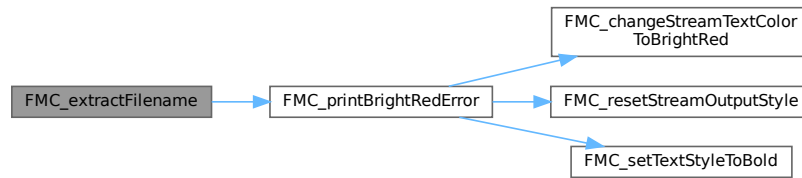
The maximum filename size is [MAX_FNAME_SIZE](#) . You can disable some warnings or error messages by defining FMC_ENABLE_DEBUG to [False](#) .

Definition at line 42 of file [FMC_paths.c](#).

References [FMC_makeMsg](#), [FMC_printBrightRedError\(\)](#), [MAX_FEXT_SIZE](#), [MAX_FNAME_SIZE](#), and [MAX_FPATH_SIZE](#).

Referenced by [FMC_getExtension\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



3.26.2.3 FMC_getExtension()

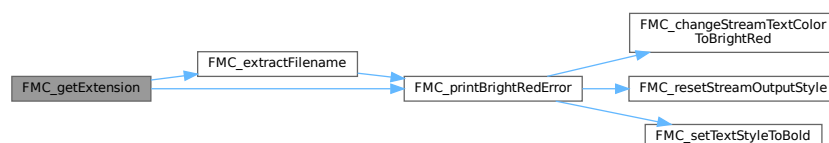
```

char * FMC_getExtension (
    const char *restrict const path,
    char *restrict ext,
    const size_t ext_size )
  
```

Definition at line 197 of file [FMC_paths.c](#).

References [FMC_extractFilename\(\)](#), [FMC_makeMsg](#), [FMC_printBrightRedError\(\)](#), [MAX_FEXT_SIZE](#), and [MAX_FNAME_SIZE](#).

Here is the call graph for this function:



3.27 FMC_file_management.h

[Go to the documentation of this file.](#)

```

00001  /*
00002
00003  MIT License
00004
00005  Copyright (c) 2022 Axel PASCON
00006
00007  Permission is hereby granted, free of charge, to any person obtaining a copy
00008  of this software and associated documentation files (the "Software"), to deal
00009  in the Software without restriction, including without limitation the rights
00010  to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011  copies of the Software, and to permit persons to whom the Software is
00012  furnished to do so, subject to the following conditions:
00013
00014  The above copyright notice and this permission notice shall be included in all
00015  copies or substantial portions of the Software.
00016
00017  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018  IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019  FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020  AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021  LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022  OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023  SOFTWARE.
00024
00025  */
00026
00027  #pragma once
00028
00029  #ifndef FMC_FILE_MANAGEMENT_H
00030  #define FMC_FILE_MANAGEMENT_H
00031
00032  #include "../general/FMC_general.h"
00033
00048  FMC_SHARED FMC_FUNC_HOT FMC_FUNC_WARN_UNUSED_RESULT FMC_FUNC_NONNULL(1, 2) char
    *FMC_extractFilename(const char * restrict const path, char * restrict filename, const size_t
    filename_size);
00049  FMC_SHARED FMC_FUNC_HOT FMC_FUNC_WARN_UNUSED_RESULT FMC_FUNC_NONNULL(1, 2) char *FMC_cutFilename(const
    char * restrict const path, char * restrict dirs, const size_t dirs_size);
00050  FMC_SHARED FMC_FUNC_HOT FMC_FUNC_WARN_UNUSED_RESULT FMC_FUNC_NONNULL(1, 2) char
    *FMC_getExtension(const char * restrict const path, char * restrict ext, const size_t ext_size);
00051
00052  #endif // FMC_FILE_MANAGEMENT_H

```

3.28 src/files/FMC_fileMan.c File Reference

3.29 FMC_fileMan.c

[Go to the documentation of this file.](#)

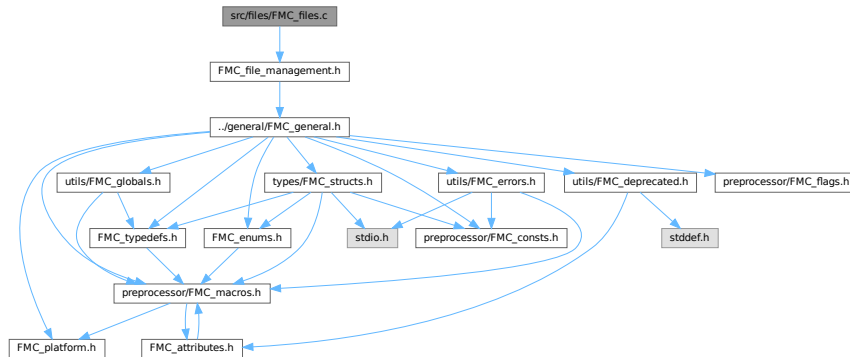
```

00001  /*
00002
00003  MIT License
00004
00005  Copyright (c) 2022-2023 Axel PASCON
00006
00007  Permission is hereby granted, free of charge, to any person obtaining a copy
00008  of this software and associated documentation files (the "Software"), to deal
00009  in the Software without restriction, including without limitation the rights
00010  to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011  copies of the Software, and to permit persons to whom the Software is
00012  furnished to do so, subject to the following conditions:
00013
00014  The above copyright notice and this permission notice shall be included in all
00015  copies or substantial portions of the Software.
00016
00017  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018  IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019  FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020  AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021  LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022  OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023  SOFTWARE.
00024
00025  */
00026

```


3.30 src/files/FMC_files.c File Reference

Include dependency graph for FMC_files.c:



3.31 FMC_files.c

[Go to the documentation of this file.](#)

```

00001  /*
00002
00003  MIT License
00004
00005  Copyright (c) 2022-2023 Axel PASCON
00006
00007  Permission is hereby granted, free of charge, to any person obtaining a copy
00008  of this software and associated documentation files (the "Software"), to deal
00009  in the Software without restriction, including without limitation the rights
00010  to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011  copies of the Software, and to permit persons to whom the Software is
00012  furnished to do so, subject to the following conditions:
00013
00014  The above copyright notice and this permission notice shall be included in all
00015  copies or substantial portions of the Software.
00016
00017  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018  IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019  FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020  AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021  LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022  OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023  SOFTWARE.
00024
00025  */
00026
00027  #include "FMC_file_management.h"
00028
00029  /*
00030  FMC_SHARED FMC_File *FMC_createFile(const char * const path)
00031  {
00032      FMC_File *file = NULL;
00033
00034  }*/

```


3.32.1.2 FMC_extractFilename()

```
char * FMC_extractFilename (
    const char *restrict const path,
    char *restrict filename,
    const size_t filename_size )
```

Gets the filename from a complete path.

Author

Axel PASCON

Date

2023

Basically, this function only detects the last '/' or '\' character. For example, if the path is "C:\\Users\\someone\\Documents\\MyFile.txt", the function will return "MyFile.txt". If the path is "/home/someone/Desktop", then Desktop will be considered as the filename. This function is designed to only operate on strings, and do not check if the path is valid, is a file or a directory, etc.

Parameters

in	<i>path</i>	The path to extract the filename from.
out	<i>filename</i>	The buffer where the filename will be stored.
in	<i>filename_size</i>	The size of the filename buffer.

Returns

A pointer to the filename buffer.

Return values

<i>NULL</i>	If the path is NULL, if the filename buffer is NULL or if an error occurred. The error can be viewed by setting FMC_ENABLE_DEBUG to True .
-------------	--

Warning

The filename buffer must be at least as big as the path.

Note

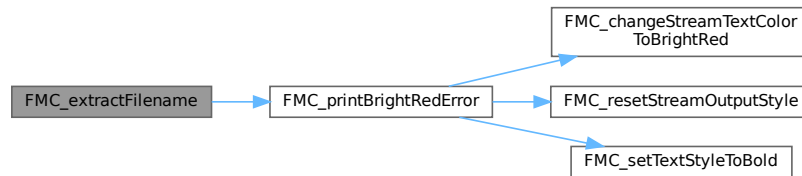
The maximum filename size is [MAX_FNAME_SIZE](#) . You can disable some warnings or error messages by defining FMC_ENABLE_DEBUG to [False](#) .

Definition at line 42 of file [FMC_paths.c](#).

References [FMC_makeMsg](#), [FMC_printBrightRedError\(\)](#), [MAX_FEXT_SIZE](#), [MAX_FNAME_SIZE](#), and [MAX_FPATH_SIZE](#).

Referenced by [FMC_getExtension\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



3.32.1.3 FMC_getExtension()

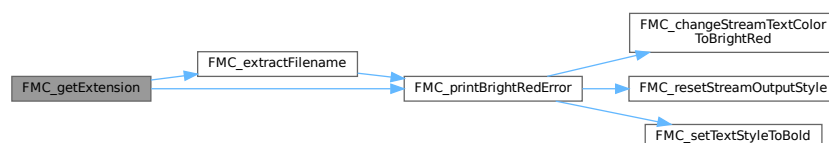
```

char * FMC_getExtension (
    const char *restrict const path,
    char *restrict ext,
    const size_t ext_size )
  
```

Definition at line 197 of file [FMC_paths.c](#).

References [FMC_extractFilename\(\)](#), [FMC_makeMsg](#), [FMC_printBrightRedError\(\)](#), [MAX_FEXT_SIZE](#), and [MAX_FNAME_SIZE](#).

Here is the call graph for this function:



3.33 FMC_paths.c

[Go to the documentation of this file.](#)

```

00001 /*
00002
00003 MIT License
00004
00005 Copyright (c) 2022-2023 Axel PASCON
00006
00007 Permission is hereby granted, free of charge, to any person obtaining a copy
00008 of this software and associated documentation files (the "Software"), to deal
00009 in the Software without restriction, including without limitation the rights
00010 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011 copies of the Software, and to permit persons to whom the Software is
00012 furnished to do so, subject to the following conditions:
00013
00014 The above copyright notice and this permission notice shall be included in all
00015 copies or substantial portions of the Software.
00016
00017 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023 SOFTWARE.
00024
00025 */
00026
00027 #include <string.h>
00028 #include <stdio.h>
00029
00030 #include "FMC_file_management.h"
00031
00032 #if defined(FMC_COMPILING_ON_WINDOWS) || defined(FMC_COMPILING_ON_MINGW)
00033 FMC_Bool FMC_ENABLE_DEBUG FMC_VAR_COMMON;
00034
00035 /*
00036  * For some reasons it doesn't compile on Windows without redeclaring the above variable. The funny
00037  * fact is that all the other files where
00038  * "FMC_ENABLE_DEBUG" appear compile well.
00039  * Moreover __attribute__((nonnull(...))) seems to interfere badly, optimizing away the first if of
00040  * the functions below, so it must be defined to
00041  * nothing.
00042  */
00043 #endif // FMC_COMPILING_ON_WINDOWS || FMC_COMPILING_ON_MINGW
00044
00045 FMC_SHARED FMC_FUNC_HOT FMC_FUNC_WARN_UNUSED_RESULT FMC_FUNC_NONNULL(1, 2) char
00046 *FMC_extractFilename(const char * restrict const path, char * restrict filename, const size_t
00047 filename_size)
00048 {
00049     #pragma GCC diagnostic ignored "-Wnonnull-compare" // get an error at compile time without this
00050     (because of attribute nonnull)
00051     if (!path || !filename)
00052     {
00053         if (FMC_ENABLE_DEBUG)
00054         {
00055             FMC_makeMsg(err_null, 4, "ERROR : ", "In function : ", __func__, ". At least one of the
00056             provided pointers is NULL.");
00057             FMC_printBrightRedError(stderr, err_null);
00058         }
00059         return NULL;
00060     }
00061     #pragma GCC diagnostic pop
00062     memset(filename, 0, filename_size);
00063     size_t path_len = 0;
00064     if ((path_len = strlen(path, MAX_FEXT_SIZE + MAX_FNAME_SIZE + MAX_FPATH_SIZE)) >= MAX_FEXT_SIZE +
00065     MAX_FNAME_SIZE + MAX_FPATH_SIZE)
00066     {
00067         FMC_makeMsg(err_path, 4, "ERROR : ", "In function : ", __func__, ". The provided path is too
00068         long (or doesn't contain any nul-character).");
00069         FMC_printBrightRedError(stderr, err_path);
00070         return NULL;
00071     }
00072     char path_cpy[MAX_FEXT_SIZE + MAX_FNAME_SIZE + MAX_FPATH_SIZE];
00073     strncpy(path_cpy, path, path_len+1);
00074     if (strcmp(path_cpy, path) != 0)
00075     {
00076         FMC_makeMsg(err_path2, 4, "FMC INTERNAL ERROR : ", "In function : ", __func__, ". strncpy
00077         failure.");
00078         FMC_printBrightRedError(stderr, err_path2);
00079         return NULL;
00080     }
00081     char *last_sep = NULL;
00082     last_sep = strrchr(path_cpy, (int)'/');

```

```

00074     if (!strchr(path_cpy, (int)'/') && !strchr(path_cpy, (int)'\\'))
00075     {
00076         filename = strncpy(filename, path_cpy, path_len+1);
00077         if (strcmp(filename, path_cpy) != 0)
00078         {
00079             FMC_makeMsg(err_path3, 4, "FMC INTERNAL ERROR : ", "In function : ", __func__, ". strncpy
failure.");
00080             FMC_printBrightRedError(stderr, err_path3);
00081             return NULL;
00082         }
00083         return filename;
00084     }
00085     else if (strchr(path_cpy, (int)'\\') && strchr(path_cpy, (int)'/'))
00086     {
00087         if (FMC_ENABLE_DEBUG)
00088         {
00089             FMC_makeMsg(err_path5, 4, "ERROR : ", "In function : ", __func__, ". The path contains
both '/' and '\\'.");
00090             FMC_printBrightRedError(stderr, err_path5);
00091         }
00092         return NULL;
00093     }
00094     else if (last_sep)
00095     {
00096         filename = strncpy(filename, last_sep+1, path_len+1);
00097         if (strcmp(filename, last_sep+1) != 0)
00098         {
00099             FMC_makeMsg(err_path4, 4, "FMC INTERNAL ERROR : ", "In function : ", __func__, ". strncpy
failure.");
00100             FMC_printBrightRedError(stderr, err_path4);
00101             return NULL;
00102         }
00103         return filename;
00104     }
00105     else
00106     {
00107         last_sep = strchr(path_cpy, (int)'\\');
00108         filename = strncpy(filename, last_sep+1, path_len+1);
00109         if (strcmp(filename, last_sep+1) != 0)
00110         {
00111             FMC_makeMsg(err_path4, 4, "FMC INTERNAL ERROR : ", "In function : ", __func__, ". strncpy
failure.");
00112             FMC_printBrightRedError(stderr, err_path4);
00113             return NULL;
00114         }
00115         return filename;
00116     }
00117 }
00118
00119 FMC_SHARED FMC_FUNC_HOT FMC_FUNC_WARN_UNUSED_RESULT FMC_FUNC_NONNULL(1, 2) char *FMC_cutFilename(const
char * restrict const path, char * restrict dirs, const size_t dirs_size)
00120 {
00121     #pragma GCC diagnostic ignored "-Wnonnull-compare" // get an error at compile time without this
(because of attribute nonnull defined on linux)
00122     if (!path || !dirs)
00123     {
00124         if (FMC_ENABLE_DEBUG)
00125         {
00126             FMC_makeMsg(err_null, 4, "ERROR : ", "In function : ", __func__, ". At least one of the
provided pointers is NULL.");
00127             FMC_printBrightRedError(stderr, err_null);
00128         }
00129         return NULL;
00130     }
00131     #pragma GCC diagnostic pop
00132     memset(dirs, 0, dirs_size);
00133     size_t path_len = 0;
00134     if ((path_len = strlen(path, MAX_FEXT_SIZE + MAX_FNAME_SIZE + MAX_FPATH_SIZE)) >= MAX_FEXT_SIZE +
MAX_FNAME_SIZE + MAX_FPATH_SIZE)
00135     {
00136         FMC_makeMsg(err_path, 4, "ERROR : ", "In function : ", __func__, ". The provided path is too
long (or doesn't contain any nul-character).");
00137         FMC_printBrightRedError(stderr, err_path);
00138         return NULL;
00139     }
00140     char path_cpy[MAX_FEXT_SIZE + MAX_FNAME_SIZE + MAX_FPATH_SIZE];
00141     strncpy(path_cpy, path, path_len+1);
00142     if (strcmp(path_cpy, path) != 0)
00143     {
00144         FMC_makeMsg(err_path2, 4, "FMC INTERNAL ERROR : ", "In function : ", __func__, ". strncpy
failure.");
00145         FMC_printBrightRedError(stderr, err_path2);
00146         return NULL;
00147     }
00148     char *last_sep = NULL;
00149     if ((last_sep = strchr(path_cpy, (int)'/')) && (last_sep = strchr(path_cpy, (int)'\\')))
00150     {

```

```

00151         if (FMC_ENABLE_DEBUG)
00152         {
00153             FMC_makeMsg(err_path5, 4, "ERROR : ", "In function : ", __func__, ". The path contains
both '/' and '\\'.");
00154             FMC_printBrightRedError(stderr, err_path5);
00155         }
00156         return NULL;
00157     }
00158
00159     else if ((last_sep = strrchr(path_cpy, (int)'/'))
00160     {
00161         strncpy(dirs, path_cpy, strlen(path_cpy, path_len) - strlen(last_sep, path_len) + 1);
00162         dirs[strlen(path_cpy, path_len) - strlen(last_sep, path_len) + 1] = '\0';
00163         return dirs;
00164     }
00165
00166     else if ((last_sep = strrchr(path_cpy, (int) '\\'))
00167     {
00168         strncpy(dirs, path_cpy, strlen(path_cpy, path_len) - strlen(last_sep, path_len) + 1);
00169         dirs[strlen(path_cpy, path_len) - strlen(last_sep, path_len) + 1] = '\0';
00170         return dirs;
00171     }
00172
00173     else if ((last_sep = strrchr(path_cpy, (int) '~'))
00174     {
00175         strncpy(dirs, "~/", 4);
00176         if (strcmp(dirs, "~/") != 0)
00177         {
00178             FMC_makeMsg(err_path4, 4, "FMC INTERNAL ERROR : ", "In function : ", __func__, ". strncpy
failure.");
00179             FMC_printBrightRedError(stderr, err_path4);
00180             return NULL;
00181         }
00182         return dirs;
00183     }
00184     else
00185     {
00186         dirs = strncpy(dirs, "./", 4);
00187         if (strcmp(dirs, "./") != 0)
00188         {
00189             FMC_makeMsg(err_path3, 4, "FMC INTERNAL ERROR : ", "In function : ", __func__, ". strncpy
failure.");
00190             FMC_printBrightRedError(stderr, err_path3);
00191             return NULL;
00192         }
00193         return dirs;
00194     }
00195 }
00196
00197 FMC_SHARED FMC_FUNC_HOT FMC_FUNC_WARN_UNUSED_RESULT FMC_FUNC_NONNULL(1, 2) char
*FMC_getExtension(const char * restrict const path, char * restrict ext, const size_t ext_size)
00198 {
00199     #pragma GCC diagnostic ignored "-Wnonnull-compare" // get an error at compile time without this
(because of attribute nonnull)
00200     if (!path || !ext)
00201     {
00202         if (FMC_ENABLE_DEBUG)
00203         {
00204             FMC_makeMsg(err_null, 4, "ERROR : ", "In function : ", __func__, ". At least one of the
provided pointers is NULL.");
00205             FMC_printBrightRedError(stderr, err_null);
00206         }
00207         return NULL;
00208     }
00209     #pragma GCC diagnostic pop
00210     memset(ext, 0, ext_size);
00211     char name[MAX_FNAME_SIZE];
00212     if (!FMC_extractFilename(path, name, MAX_FNAME_SIZE))
00213     {
00214         FMC_makeMsg(err_path6, 4, "FMC INTERNAL ERROR : ", "In function : ", __func__, ".
FMC_extractFilename call failed.");
00215         FMC_printBrightRedError(stderr, err_path6);
00216         return NULL;
00217     }
00218
00219     if (!strchr(name, (int) '.')) {strncpy(ext, "", 2); return ext;} // Could be modified (?)
00220     else
00221     {
00222         char *last_dot = NULL;
00223         if ((last_dot = strrchr(name, (int) '.'))
00224         {
00225             strncpy(ext, last_dot, strlen(last_dot+1, MAX_FEXT_SIZE)+1);
00226             return ext;
00227         }
00228         else
00229         {
00230             FMC_makeMsg(err_path7, 4, "FMC INTERNAL ERROR : ", "In function : ", __func__, ". strrchr

```


3.35 FMC.h

[Go to the documentation of this file.](#)

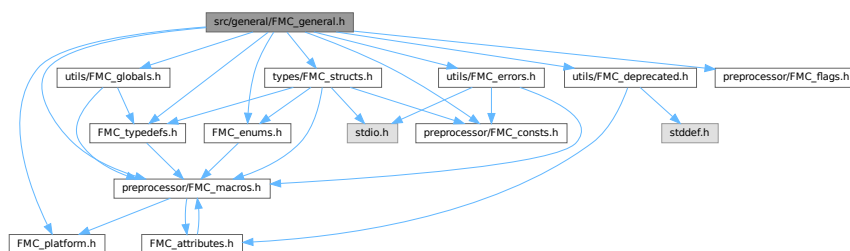
```

00001  /*
00002
00003  MIT License
00004
00005  Copyright (c) 2022 Axel PASCON
00006
00007  Permission is hereby granted, free of charge, to any person obtaining a copy
00008  of this software and associated documentation files (the "Software"), to deal
00009  in the Software without restriction, including without limitation the rights
00010  to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011  copies of the Software, and to permit persons to whom the Software is
00012  furnished to do so, subject to the following conditions:
00013
00014  The above copyright notice and this permission notice shall be included in all
00015  copies or substantial portions of the Software.
00016
00017  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018  IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019  FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020  AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021  LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022  OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023  SOFTWARE.
00024
00025  */
00026
00027  #pragma once
00028
00029  #ifndef FMC_H
00030  #define FMC_H
00031
00032  // includes
00033  #include "general/FMC_general.h"
00034  #include "code_utils/FMC_code_utils.h"
00035  #include "files/FMC_file_management.h"
00036  #include "data_analyze/FMC_data_analyze.h"
00037  #include "cpp/FMC_wrapper.h"
00038
00039
00040
00041
00042  #endif // FMC_H

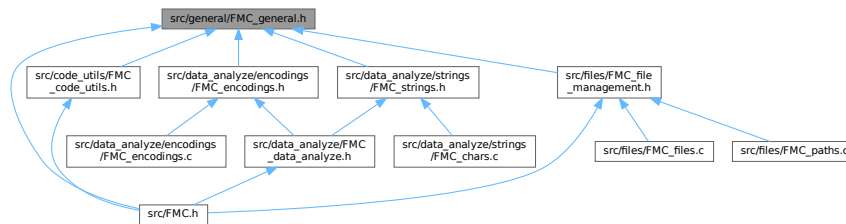
```

3.36 src/general/FMC_general.h File Reference

Include dependency graph for FMC_general.h:



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



Macros

- `#define FMC_DATA_H`

3.36.1 Macro Definition Documentation

3.36.1.1 FMC_DATA_H

```
#define FMC_DATA_H
```

Definition at line 30 of file `FMC_general.h`.

3.37 FMC_general.h

[Go to the documentation of this file.](#)

```
00001 /*
00002
00003 MIT License
00004
00005 Copyright (c) 2022 Axel PASCON
00006
00007 Permission is hereby granted, free of charge, to any person obtaining a copy
00008 of this software and associated documentation files (the "Software"), to deal
00009 in the Software without restriction, including without limitation the rights
00010 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011 copies of the Software, and to permit persons to whom the Software is
00012 furnished to do so, subject to the following conditions:
00013
00014 The above copyright notice and this permission notice shall be included in all
00015 copies or substantial portions of the Software.
00016
00017 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023 SOFTWARE.
00024
00025 */
00026
00027 #pragma once
00028
00029 #ifndef FMC_DATA_H
00030 #define FMC_DATA_H
```

3.38 src/general/preprocessor/FMC_attributes.h File Reference

```
graph TD
    A["src/general/preprocessor  
/FMC_attributes.h"] <--> B["FMC_macros.h"]
    B --> C["FMC_platform.h"]
```

3.39 FMC_attributes.h

[Go to the documentation of this file.](#)

```

00001  /*
00002
00003  MIT License
00004
00005  Copyright (c) 2023 Axel PASCON
00006
00007  Permission is hereby granted, free of charge, to any person obtaining a copy
00008  of this software and associated documentation files (the "Software"), to deal
00009  in the Software without restriction, including without limitation the rights
00010  to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011  copies of the Software, and to permit persons to whom the Software is
00012  furnished to do so, subject to the following conditions:
00013
00014  The above copyright notice and this permission notice shall be included in all
00015  copies or substantial portions of the Software.
00016
00017  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018  IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019  FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020  AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021  LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022  OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023  SOFTWARE.
00024
00025  */
00026
00027  #ifndef FMC_ATTRIBUTES_H
00028  #define FMC_ATTRIBUTES_H
00029
00030  #include "FMC_macros.h"
00031
00032
00033  #ifndef FMC_FUNC_ATTRIBUTES
00034      #define FMC_FUNC_ATTRIBUTES 1
00035
00036      #ifndef FMC_FUNC_ACCESS
00037          #define FMC_FUNC_ACCESS(access_type, ...) __attribute__((access(access_type, __VA_ARGS__)))
00038      #endif // FMC_FUNC_ACCESS
00039
00040      #ifndef FMC_FUNC_ALIAS
00041          #define FMC_FUNC_ALIAS(alias) __attribute__((alias(FMC_STRINGIZE(alias))))
00042      #endif // FMC_FUNC_ALIAS
00043
00044      #ifndef FMC_FUNC_ALWAYS_INLINE
00045          #define FMC_FUNC_ALWAYS_INLINE __attribute__((always_inline))
00046      #endif // FMC_FUNC_ALWAYS_INLINE
00047
00048      #ifndef FMC_FUNC_COLD
00049          #define FMC_FUNC_COLD __attribute__((cold))
00050      #endif // FMC_FUNC_COLD
00051
00052      #ifndef FMC_FUNC_CONST
00053          #define FMC_FUNC_CONST __attribute__((const))
00054      #endif // FMC_FUNC_CONST
00055
00056      #ifndef FMC_FUNC_CONSTRUCTOR
00057          #define FMC_FUNC_CONSTRUCTOR(priority) __attribute__((constructor(priority)))
00058      #endif // FMC_FUNC_CONSTRUCTOR
00059
00060      #ifndef FMC_FUNC_DESTRUCTOR
00061          #define FMC_FUNC_DESTRUCTOR(priority) __attribute__((destructor(priority)))
00062      #endif // FMC_FUNC_DESTRUCTOR
00063
00064      #ifndef FMC_FUNC_COPY
00065          #define FMC_FUNC_COPY(func) __attribute__((copy(func)))
00066      #endif // FMC_FUNC_COPY
00067
00068      #ifndef FMC_FUNC_DEPRECATED
00069          #define FMC_FUNC_DEPRECATED(msg) __attribute__((deprecated(FMC_STRINGIZE(msg))))
00070      #endif // FMC_FUNC_DEPRECATED
00071
00072      #ifndef FMC_FUNC_UNAVAILABLE
00073          #define FMC_FUNC_UNAVAILABLE(msg) __attribute__((unavailable(FMC_STRINGIZE(msg))))
00074      #endif // FMC_FUNC_UNAVAILABLE
00075
00076      #ifndef FMC_FUNC_ERROR
00077          #define FMC_FUNC_ERROR(msg) __attribute__((error(FMC_STRINGIZE(msg))))
00078      #endif // FMC_FUNC_ERROR
00079
00080      #ifndef FMC_FUNC_WARNING
00081          #define FMC_FUNC_WARNING(msg) __attribute__((warning(FMC_STRINGIZE(msg))))
00082      #endif // FMC_FUNC_WARNING

```

```

00083
00084 #ifndef FMC_FUNC_EXTERNALLY_VISIBLE
00085     #define FMC_FUNC_EXTERNALLY_VISIBLE __attribute__((externally_visible))
00086 #endif // FMC_FUNC_EXTERNALLY_VISIBLE
00087
00088 #ifndef FMC_FUNC_FLATTEN
00089     #define FMC_FUNC_FLATTEN __attribute__((flatten))
00090 #endif // FMC_FUNC_FLATTEN
00091
00092 #ifndef FMC_FUNC_FORMAT
00093     #define FMC_FUNC_FORMAT(func_fmt, fmt_pos, args_pos) __attribute__((format(func_fmt, fmt_pos,
args_pos)))
00094 #endif // FMC_FUNC_FORMAT
00095
00096 #ifndef FMC_FUNC_HOT
00097     #define FMC_FUNC_HOT __attribute__((hot))
00098 #endif // FMC_FUNC_HOT
00099
00100 #ifndef FMC_FUNC_MALLOC
00101     #define FMC_FUNC_MALLOC(...) __attribute__((malloc(__VA_ARGS__)))
00102 #endif // FMC_FUNC_MALLOC
00103
00104 #ifndef FMC_FUNC_NONNULL
00105     #if !(defined(FMC_COMPILING_ON_WINDOWS) || defined(FMC_COMPILING_ON_MINGW))
00106         #define FMC_FUNC_NONNULL(...) __attribute__((nonnull(__VA_ARGS__)))
00107     #else
00108         #define FMC_FUNC_NONNULL(...)
00109     #endif
00110 #endif // FMC_FUNC_NONNULL
00111
00112 #ifndef FMC_FUNC_NORETURN
00113     #define FMC_FUNC_NORETURN __attribute__((noreturn))
00114 #endif // FMC_FUNC_NORETURN
00115
00116 #ifndef FMC_FUNC_OPTIMIZE
00117     #define FMC_FUNC_OPTIMIZE(level) __attribute__((optimize(FMC_STRINGIZE(level))))
00118 #endif // FMC_FUNC_OPTIMIZE
00119
00120 #ifndef FMC_FUNC_PURE
00121     #define FMC_FUNC_PURE __attribute__((pure))
00122 #endif // FMC_FUNC_PURE
00123
00124 #ifndef FMC_FUNC_RETURNS_NONNULL
00125     #define FMC_FUNC_RETURNS_NONNULL __attribute__((returns_nonnull))
00126 #endif // FMC_FUNC_RETURNS_NONNULL
00127
00128 #ifndef FMC_FUNC_SECTION
00129     #define FMC_FUNC_SECTION(section_name) __attribute__((section(FMC_STRINGIZE(section_name))))
00130 #endif // FMC_FUNC_SECTION
00131
00132 #ifndef FMC_FUNC_SENTINEL
00133     #define FMC_FUNC_SENTINEL(pos) __attribute__((sentinel(pos)))
00134 #endif // FMC_FUNC_SENTINEL
00135
00136 #ifndef FMC_FUNC_STACK_PROTECT
00137     #define FMC_FUNC_STACK_PROTECT __attribute__((stack_protect))
00138 #endif // FMC_FUNC_STACK_PROTECT
00139
00140 #ifndef FMC_FUNC_SYMVER
00141     #define FMC_FUNC_SYMVER(name, major, minor, patch)
__attribute__((symver(FMC_STRINGIZE(FMC_CONCAT_6(name, @, v, major, minor, patch)))))
00142 #endif // FMC_FUNC_SYMVER
00143
00144 #ifndef FMC_FUNC_UNUSED
00145     #define FMC_FUNC_UNUSED __attribute__((unused))
00146 #endif // FMC_FUNC_UNUSED
00147
00148 #ifndef FMC_FUNC_USED
00149     #define FMC_FUNC_USED __attribute__((used))
00150 #endif // FMC_FUNC_USED
00151
00152 #ifndef FMC_FUNC_VISIBILITY
00153     #define FMC_FUNC_VISIBILITY(visibility_type)
__attribute__((visibility(FMC_STRINGIZE(visibility_type))))
00154 #endif // FMC_FUNC_VISIBILITY
00155
00156 #ifndef FMC_FUNC_WARN_UNUSED_RESULT
00157     #define FMC_FUNC_WARN_UNUSED_RESULT __attribute__((warn_unused_result))
00158 #endif // FMC_FUNC_WARN_UNUSED_RESULT
00159
00160 #ifndef FMC_FUNC_WEAK
00161     #define FMC_FUNC_WEAK __attribute__((weak))
00162 #endif // FMC_FUNC_WEAK
00163
00164 #ifndef FMC_FUNC_WEAK_REF
00165     #define FMC_FUNC_WEAK_REF(...) __attribute__((weakref(FMC_STRINGIZE(__VA_ARGS__))))
00166 #endif // FMC_FUNC_WEAK_REF

```

```

00167
00168     #ifndef FMC_FUNC_ZERO_REGISTERS
00169         #define FMC_FUNC_ZERO_REGISTERS(to_zero)
00170         __attribute__((zero_call_used_regs(FMC_STRINGIZE(to_zero))))
00171     #endif // FMC_FUNC_ZERO_REGISTERS
00172
00173
00174     #ifndef FMC_FUNC_STRONG_ALIAS
00175         #define FMC_FUNC_STRONG_ALIAS(func_name, aliased) FMC_FUNC_ALIAS(aliased)
00176         FMC_FUNC_COPY(aliased) __typeof__(aliased) func_name
00177     #endif // FMC_FUNC_STRONG_ALIAS
00178
00179     #ifndef FMC_FUNC_INLINE
00180         #define FMC_FUNC_INLINE inline FMC_FUNC_ALWAYS_INLINE
00181     #endif // FMC_FUNC_INLINE
00182
00183     #ifndef FMC_FUNC_PRINTF_FMT
00184         #define FMC_FUNC_PRINTF_FMT(fmt_pos, args_pos) FMC_FUNC_FORMAT(printf, fmt_pos, args_pos)
00185     #endif // FMC_FUNC_PRINTF_FMT
00186 #endif //FMC_FUNC_ATTRIBUTES
00187
00188 #ifndef FMC_VAR_ATTRIBUTES
00189     #define FMC_VAR_ATTRIBUTES
00190
00191     #ifndef FMC_VAR_ALIAS
00192         #define FMC_VAR_ALIAS(aliased) __attribute__((alias(FMC_STRINGIZE(aliased))))
00193     #endif // FMC_VAR_ALIAS
00194
00195     #ifndef FMC_VAR_CLEANUP
00196         #define FMC_VAR_CLEANUP(func_name) __attribute__((cleanup(func_name)))
00197     #endif // FMC_VAR_CLEANUP
00198
00199     #ifndef FMC_VAR_COMMON
00200         #define FMC_VAR_COMMON __attribute__((common))
00201     #endif // FMC_VAR_COMMON
00202
00203     #ifndef FMC_VAR_NO_COMMON
00204         #define FMC_VAR_NO_COMMON __attribute__((nocommon))
00205     #endif // FMC_VAR_NO_COMMON
00206
00207     #ifndef FMC_VAR_COPY
00208         #define FMC_VAR_COPY(var) __attribute__((copy(var)))
00209     #endif // FMC_VAR_COPY
00210
00211     #ifndef FMC_VAR_DEPRECATED
00212         #define FMC_VAR_DEPRECATED(msg) __attribute__((deprecated(FMC_STRINGIZE(msg))))
00213     #endif // FMC_VAR_DEPRECATED
00214
00215     #ifndef FMC_VAR_UNAVAILABLE
00216         #define FMC_VAR_UNAVAILABLE(msg) __attribute__((unavailable(FMC_STRINGIZE(msg))))
00217     #endif // FMC_VAR_UNAVAILABLE
00218
00219     #ifndef FMC_VAR_MACH_MODE
00220         #define FMC_VAR_MACH_MODE(mode) __attribute__((mode(mode)))
00221     #endif // FMC_VAR_MACH_MODE
00222
00223     #ifndef FMC_VAR_NON_STRING
00224         #define FMC_VAR_NON_STRING __attribute__((nonstring))
00225     #endif // FMC_VAR_NON_STRING
00226
00227     #ifndef FMC_VAR_SECTION
00228         #define FMC_VAR_SECTION(section_name) __attribute__((section(FMC_STRINGIZE(section_name))))
00229     #endif // FMC_VAR_SECTION
00230
00231     #ifndef FMC_VAR_UNUSED
00232         #define FMC_VAR_UNUSED __attribute__((unused))
00233     #endif // FMC_VAR_UNUSED
00234
00235     #ifndef FMC_VAR_USED
00236         #define FMC_VAR_USED __attribute__((used))
00237     #endif // FMC_VAR_USED
00238
00239     #ifndef FMC_VAR_UNINITIALIZED
00240         #define FMC_VAR_UNINITIALIZED __attribute__((uninitialized))
00241     #endif // FMC_VAR_UNINITIALIZED
00242
00243     #ifndef FMC_VAR_VISIBILITY
00244         #define FMC_VAR_VISIBILITY(visibility_type)
00245         __attribute__((visibility(FMC_STRINGIZE(visibility_type))))
00246     #endif // FMC_VAR_VISIBILITY
00247
00248     #ifndef FMC_VAR_WEAK
00249         #define FMC_VAR_WEAK __attribute__((weak))
00250     #endif // FMC_VAR_WEAK

```

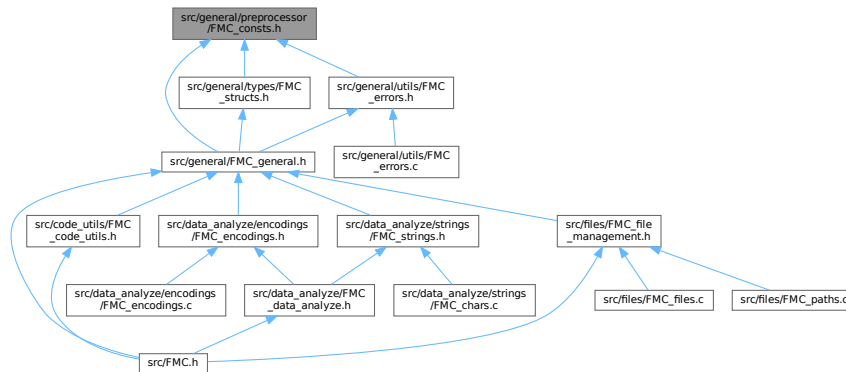
```

00251 #endif // FMC_VAR_ATTRIBUTES
00252
00253 #ifndef FMC_TYPE_ATTRIBUTES
00254     #define FMC_TYPE_ATTRIBUTES
00255
00256     #ifndef FMC_TYPE_DEPRECATED
00257         #define FMC_TYPE_DEPRECATED(msg) __attribute__((deprecated(FMC_STRINGIZE(msg))))
00258     #endif // FMC_TYPE_DEPRECATED
00259
00260     #ifndef FMC_TYPE_UNAVAILABLE
00261         #define FMC_TYPE_UNAVAILABLE(msg) __attribute__((unavailable(FMC_STRINGIZE(msg))))
00262     #endif // FMC_TYPE_UNAVAILABLE
00263
00264     #ifndef FMC_TYPE_MACH_MODE
00265         #define FMC_TYPE_MACH_MODE(mode) __attribute__((mode(mode)))
00266     #endif // FMC_TYPE_MACH_MODE
00267
00268     #ifndef FMC_TYPE_UNUSED
00269         #define FMC_TYPE_UNUSED __attribute__((unused))
00270     #endif // FMC_TYPE_UNUSED
00271
00272     #ifndef FMC_TYPE_VISIBILITY
00273         #define FMC_TYPE_VISIBILITY(visibility_type)
00274         __attribute__((visibility(FMC_STRINGIZE(visibility_type))))
00275     #endif // FMC_TYPE_VISIBILITY
00276 #endif // FMC_TYPE_ATTRIBUTES
00277
00278 #ifndef FMC_LABEL_ATTRIBUTES
00279     #define FMC_LABEL_ATTRIBUTES
00280
00281     #ifndef FMC_LABEL_UNUSED
00282         #define FMC_LABEL_UNUSED __attribute__((unused))
00283     #endif // FMC_LABEL_UNUSED
00284
00285     #ifndef FMC_LABEL_HOT
00286         #define FMC_LABEL_HOT __attribute__((hot))
00287     #endif // FMC_LABEL_HOT
00288
00289     #ifndef FMC_LABEL_COLD
00290         #define FMC_LABEL_COLD __attribute__((cold))
00291     #endif // FMC_LABEL_COLD
00292
00293 #endif // FMC_LABEL_ATTRIBUTES
00294
00295 #ifndef FMC_ENUM_ATTRIBUTES
00296     #define FMC_ENUM_ATTRIBUTES
00297
00298     #ifndef FMC_ENUM_DEPRECATED
00299         #define FMC_ENUM_DEPRECATED(msg) __attribute__((deprecated(FMC_STRINGIZE(msg))))
00300     #endif // FMC_ENUM_DEPRECATED
00301
00302     #ifndef FMC_ENUM_UNAVAILABLE
00303         #define FMC_ENUM_UNAVAILABLE(msg) __attribute__((unavailable(FMC_STRINGIZE(msg))))
00304     #endif // FMC_ENUM_UNAVAILABLE
00305
00306 #endif // FMC_ENUM_ATTRIBUTES
00307
00308 #ifndef FMC_STMT_ATTRIBUTES
00309     #define FMC_STMT_ATTRIBUTES
00310
00311     #ifndef FMC_STMT_FALLTHROUGH
00312         #define FMC_STMT_FALLTHROUGH __attribute__((fallthrough))
00313     #endif // FMC_STMT_FALLTHROUGH
00314
00315 #endif // FMC_STMT_ATTRIBUTES
00316
00317 #endif // FMC_ATTRIBUTES_H

```

3.40 src/general/preprocessor/FMC_consts.h File Reference

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



Macros

- #define BG_BLACK "\x1b[40m"
- #define BG_BLUE "\x1b[44m"
- #define BG_BRIGHT_BLACK "\x1b[100m"
- #define BG_BRIGHT_BLUE "\x1b[104m"
- #define BG_BRIGHT_CYAN "\x1b[106m"
- #define BG_BRIGHT_GREEN "\x1b[102m"
- #define BG_BRIGHT_MAGENTA "\x1b[105m"
- #define BG_BRIGHT_RED "\x1b[101m"
- #define BG_BRIGHT_WHITE "\x1b[107m"
- #define BG_BRIGHT_YELLOW "\x1b[103m"
- #define BG_CYAN "\x1b[46m"
- #define BG_GREEN "\x1b[42m"
- #define BG_MAGENTA "\x1b[45m"
- #define BG_RED "\x1b[41m"
- #define BG_WHITE "\x1b[47m"
- #define BG_YELLOW "\x1b[43m"
- #define False 0
- #define FG_BLACK "\x1b[30m"
- #define FG_BLUE "\x1b[34m"
- #define FG_BRIGHT_BLACK "\x1b[90m"
- #define FG_BRIGHT_BLUE "\x1b[94m"
- #define FG_BRIGHT_CYAN "\x1b[96m"
- #define FG_BRIGHT_GREEN "\x1b[92m"
- #define FG_BRIGHT_MAGENTA "\x1b[95m"
- #define FG_BRIGHT_RED "\x1b[91m"
- #define FG_BRIGHT_WHITE "\x1b[97m"
- #define FG_BRIGHT_YELLOW "\x1b[93m"
- #define FG_CYAN "\x1b[36m"
- #define FG_GREEN "\x1b[32m"
- #define FG_MAGENTA "\x1b[35m"
- #define FG_RED "\x1b[31m"
- #define FG_WHITE "\x1b[37m"

- `#define FG_YELLOW "\x1b[33m"`
- `#define FMC_BOOLEANS`
- `#define FMC_CONSTS_H`
- `#define FMC_MAX_PATH_COMPONENTS_SIZE`
- `#define FMC_STYLES`
- `#define MAX_FEXT_SIZE 50`
- `#define MAX_FNAME_SIZE 256`
- `#define MAX_FPATH_SIZE 512`
- `#define RESET "\x1b[0m"`
- `#define True 1`
- `#define TXT_BLINK "\x1b[5m"`
- `#define TXT_BOLD "\x1b[1m"`
- `#define TXT_DIM "\x1b[2m"`
- `#define TXT_HIDDEN "\x1b[8m"`
- `#define TXT_REVERSE "\x1b[7m"`
- `#define TXT_UNDERLINED "\x1b[4m"`

3.40.1 Macro Definition Documentation

3.40.1.1 BG_BLACK

```
#define BG_BLACK "\x1b[40m"
```

Definition at line 61 of file [FMC_consts.h](#).

3.40.1.2 BG_BLUE

```
#define BG_BLUE "\x1b[44m"
```

Definition at line 65 of file [FMC_consts.h](#).

3.40.1.3 BG_BRIGHT_BLACK

```
#define BG_BRIGHT_BLACK "\x1b[100m"
```

Definition at line 69 of file [FMC_consts.h](#).

3.40.1.4 BG_BRIGHT_BLUE

```
#define BG_BRIGHT_BLUE "\x1b[104m"
```

Definition at line 73 of file [FMC_consts.h](#).

3.40.1.5 BG_BRIGHT_CYAN

```
#define BG_BRIGHT_CYAN "\x1b[106m"
```

Definition at line 75 of file [FMC_consts.h](#).

3.40.1.6 BG_BRIGHT_GREEN

```
#define BG_BRIGHT_GREEN "\x1b[102m"
```

Definition at line 71 of file [FMC_consts.h](#).

3.40.1.7 BG_BRIGHT_MAGENTA

```
#define BG_BRIGHT_MAGENTA "\x1b[105m"
```

Definition at line 74 of file [FMC_consts.h](#).

3.40.1.8 BG_BRIGHT_RED

```
#define BG_BRIGHT_RED "\x1b[101m"
```

Definition at line 70 of file [FMC_consts.h](#).

3.40.1.9 BG_BRIGHT_WHITE

```
#define BG_BRIGHT_WHITE "\x1b[107m"
```

Definition at line 76 of file [FMC_consts.h](#).

3.40.1.10 BG_BRIGHT_YELLOW

```
#define BG_BRIGHT_YELLOW "\x1b[103m"
```

Definition at line 72 of file [FMC_consts.h](#).

3.40.1.11 BG_CYAN

```
#define BG_CYAN "\x1b[46m"
```

Definition at line 67 of file [FMC_consts.h](#).

3.40.1.12 BG_GREEN

```
#define BG_GREEN "\x1b[42m"
```

Definition at line 63 of file [FMC_consts.h](#).

3.40.1.13 BG_MAGENTA

```
#define BG_MAGENTA "\x1b[45m"
```

Definition at line 66 of file [FMC_consts.h](#).

3.40.1.14 BG_RED

```
#define BG_RED "\x1b[41m"
```

Definition at line 62 of file [FMC_consts.h](#).

3.40.1.15 BG_WHITE

```
#define BG_WHITE "\x1b[47m"
```

Definition at line 68 of file [FMC_consts.h](#).

3.40.1.16 BG_YELLOW

```
#define BG_YELLOW "\x1b[43m"
```

Definition at line 64 of file [FMC_consts.h](#).

3.40.1.17 False

```
#define False 0
```

Definition at line 94 of file [FMC_consts.h](#).

3.40.1.18 FG_BLACK

```
#define FG_BLACK "\x1b[30m"
```

Definition at line 44 of file [FMC_consts.h](#).

3.40.1.19 FG_BLUE

```
#define FG_BLUE "\x1b[34m"
```

Definition at line 48 of file [FMC_consts.h](#).

3.40.1.20 FG_BRIGHT_BLACK

```
#define FG_BRIGHT_BLACK "\x1b[90m"
```

Definition at line 52 of file [FMC_consts.h](#).

3.40.1.21 FG_BRIGHT_BLUE

```
#define FG_BRIGHT_BLUE "\x1b[94m"
```

Definition at line 56 of file [FMC_consts.h](#).

3.40.1.22 FG_BRIGHT_CYAN

```
#define FG_BRIGHT_CYAN "\x1b[96m"
```

Definition at line 58 of file [FMC_consts.h](#).

3.40.1.23 FG_BRIGHT_GREEN

```
#define FG_BRIGHT_GREEN "\x1b[92m"
```

Definition at line 54 of file [FMC_consts.h](#).

3.40.1.24 FG_BRIGHT_MAGENTA

```
#define FG_BRIGHT_MAGENTA "\x1b[95m"
```

Definition at line 57 of file [FMC_consts.h](#).

3.40.1.25 FG_BRIGHT_RED

```
#define FG_BRIGHT_RED "\x1b[91m"
```

Definition at line 53 of file [FMC_consts.h](#).

3.40.1.26 FG_BRIGHT_WHITE

```
#define FG_BRIGHT_WHITE "\x1b[97m"
```

Definition at line 59 of file [FMC_consts.h](#).

3.40.1.27 FG_BRIGHT_YELLOW

```
#define FG_BRIGHT_YELLOW "\x1b[93m"
```

Definition at line 55 of file [FMC_consts.h](#).

3.40.1.28 FG_CYAN

```
#define FG_CYAN "\x1b[36m"
```

Definition at line 50 of file [FMC_consts.h](#).

3.40.1.29 FG_GREEN

```
#define FG_GREEN "\x1b[32m"
```

Definition at line 46 of file [FMC_consts.h](#).

3.40.1.30 FG_MAGENTA

```
#define FG_MAGENTA "\x1b[35m"
```

Definition at line 49 of file [FMC_consts.h](#).

3.40.1.31 FG_RED

```
#define FG_RED "\x1b[31m"
```

Definition at line 45 of file [FMC_consts.h](#).

3.40.1.32 FG_WHITE

```
#define FG_WHITE "\x1b[37m"
```

Definition at line 51 of file [FMC_consts.h](#).

3.40.1.33 FG_YELLOW

```
#define FG_YELLOW "\x1b[33m"
```

Definition at line 47 of file [FMC_consts.h](#).

3.40.1.34 FMC_BOOLEANS

```
#define FMC_BOOLEANS
```

Definition at line 92 of file [FMC_consts.h](#).

3.40.1.35 FMC_CONSTS_H

```
#define FMC_CONSTS_H
```

Definition at line 30 of file [FMC_consts.h](#).

3.40.1.36 FMC_MAX_PATH_COMPONENTS_SIZE

```
#define FMC_MAX_PATH_COMPONENTS_SIZE
```

Definition at line 33 of file [FMC_consts.h](#).

3.40.1.37 FMC_STYLES

```
#define FMC_STYLES
```

Definition at line 40 of file [FMC_consts.h](#).

3.40.1.38 MAX_FEXT_SIZE

```
#define MAX_FEXT_SIZE 50
```

Definition at line 34 of file [FMC_consts.h](#).

3.40.1.39 MAX_FNAME_SIZE

```
#define MAX_FNAME_SIZE 256
```

Definition at line 35 of file [FMC_consts.h](#).

3.40.1.40 MAX_FPATH_SIZE

```
#define MAX_FPATH_SIZE 512
```

Definition at line 36 of file [FMC_consts.h](#).

3.40.1.41 RESET

```
#define RESET "\x1b[0m"
```

Definition at line 42 of file [FMC_consts.h](#).

3.40.1.42 True

```
#define True 1
```

Definition at line 93 of file [FMC_consts.h](#).

3.40.1.43 TXT_BLINK

```
#define TXT_BLINK "\x1b[5m"
```

Definition at line 81 of file [FMC_consts.h](#).

3.40.1.44 TXT_BOLD

```
#define TXT_BOLD "\x1b[1m"
```

Definition at line 78 of file [FMC_consts.h](#).

3.40.1.45 TXT_DIM

```
#define TXT_DIM "\x1b[2m"
```

Definition at line 79 of file [FMC_consts.h](#).

3.40.1.46 TXT_HIDDEN

```
#define TXT_HIDDEN "\x1b[8m"
```

Definition at line 83 of file [FMC_consts.h](#).

3.40.1.47 TXT_REVERSE

```
#define TXT_REVERSE "\x1b[7m"
```

Definition at line 82 of file [FMC_consts.h](#).

3.40.1.48 TXT_UNDERLINED

```
#define TXT_UNDERLINED "\x1b[4m"
```

Definition at line 80 of file [FMC_consts.h](#).

3.41 FMC_consts.h

[Go to the documentation of this file.](#)

```
00001 /*
00002
00003 MIT License
00004
00005 Copyright (c) 2022 Axel PASCON
00006
00007 Permission is hereby granted, free of charge, to any person obtaining a copy
00008 of this software and associated documentation files (the "Software"), to deal
00009 in the Software without restriction, including without limitation the rights
00010 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011 copies of the Software, and to permit persons to whom the Software is
00012 furnished to do so, subject to the following conditions:
00013
00014 The above copyright notice and this permission notice shall be included in all
00015 copies or substantial portions of the Software.
00016
00017 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023 SOFTWARE.
00024
00025 */
00026
00027 #pragma once
00028
00029 #ifndef FMC_CONSTS_H
00030 #define FMC_CONSTS_H
00031
00032 #ifndef FMC_MAX_PATH_COMPONENTS_SIZE
00033     #define FMC_MAX_PATH_COMPONENTS_SIZE
00034     #define MAX_FEXT_SIZE 50
00035     #define MAX_FNAME_SIZE 256
00036     #define MAX_FPATH_SIZE 512
00037 #endif
00038
00039 #ifndef FMC_STYLES
00040     #define FMC_STYLES
```

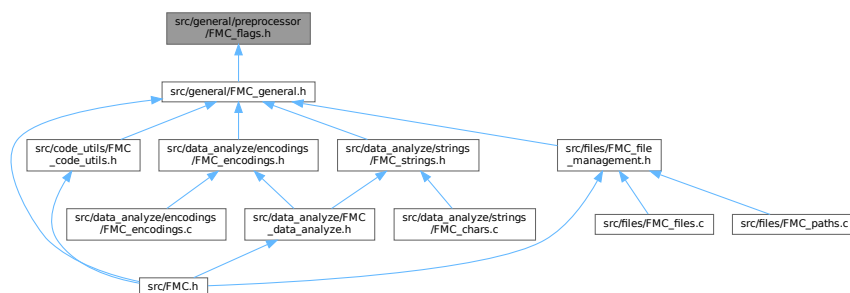
```

00041
00042 #define RESET "\x1b[0m"
00043
00044 #define FG_BLACK "\x1b[30m"
00045 #define FG_RED "\x1b[31m"
00046 #define FG_GREEN "\x1b[32m"
00047 #define FG_YELLOW "\x1b[33m"
00048 #define FG_BLUE "\x1b[34m"
00049 #define FG_MAGENTA "\x1b[35m"
00050 #define FG_CYAN "\x1b[36m"
00051 #define FG_WHITE "\x1b[37m"
00052 #define FG_BRIGHT_BLACK "\x1b[90m"
00053 #define FG_BRIGHT_RED "\x1b[91m"
00054 #define FG_BRIGHT_GREEN "\x1b[92m"
00055 #define FG_BRIGHT_YELLOW "\x1b[93m"
00056 #define FG_BRIGHT_BLUE "\x1b[94m"
00057 #define FG_BRIGHT_MAGENTA "\x1b[95m"
00058 #define FG_BRIGHT_CYAN "\x1b[96m"
00059 #define FG_BRIGHT_WHITE "\x1b[97m"
00060
00061 #define BG_BLACK "\x1b[40m"
00062 #define BG_RED "\x1b[41m"
00063 #define BG_GREEN "\x1b[42m"
00064 #define BG_YELLOW "\x1b[43m"
00065 #define BG_BLUE "\x1b[44m"
00066 #define BG_MAGENTA "\x1b[45m"
00067 #define BG_CYAN "\x1b[46m"
00068 #define BG_WHITE "\x1b[47m"
00069 #define BG_BRIGHT_BLACK "\x1b[100m"
00070 #define BG_BRIGHT_RED "\x1b[101m"
00071 #define BG_BRIGHT_GREEN "\x1b[102m"
00072 #define BG_BRIGHT_YELLOW "\x1b[103m"
00073 #define BG_BRIGHT_BLUE "\x1b[104m"
00074 #define BG_BRIGHT_MAGENTA "\x1b[105m"
00075 #define BG_BRIGHT_CYAN "\x1b[106m"
00076 #define BG_BRIGHT_WHITE "\x1b[107m"
00077
00078 #define TXT_BOLD "\x1b[1m"
00079 #define TXT_DIM "\x1b[2m"
00080 #define TXT_UNDERLINED "\x1b[4m"
00081 #define TXT_BLINK "\x1b[5m"
00082 #define TXT_REVERSE "\x1b[7m"
00083 #define TXT_HIDDEN "\x1b[8m"
00084
00085 #endif // FMC_STYLES
00086
00087 #if defined(FMC_BOOLEANS) || defined(True) || defined(False)
00088 #undef FMC_BOOLEANS
00089 #undef True
00090 #undef False
00091 #endif // FMC_BOOLEANS
00092 #define FMC_BOOLEANS
00093 #define True 1
00094 #define False 0
00095
00096 #endif // FMC_CONSTS_H

```

3.42 src/general/preprocessor/FMC_flags.h File Reference

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



Macros

- `#define ASCII` 64
- `#define FMC_ENCODING_FLAGS`
- `#define FMC_FLAGS`
- `#define UNKNOWN` 128
- `#define UTF16_BE` 8
- `#define UTF16_LE` 4
- `#define UTF32_BE` 32
- `#define UTF32_LE` 16
- `#define UTF8` 1
- `#define UTF8_BOM` 2

3.42.1 Macro Definition Documentation

3.42.1.1 ASCII

```
#define ASCII 64
```

Definition at line 40 of file [FMC_flags.h](#).

3.42.1.2 FMC_ENCODING_FLAGS

```
#define FMC_ENCODING_FLAGS
```

Definition at line 33 of file [FMC_flags.h](#).

3.42.1.3 FMC_FLAGS

```
#define FMC_FLAGS
```

Definition at line 30 of file [FMC_flags.h](#).

3.42.1.4 UNKNOWN

```
#define UNKNOWN 128
```

Definition at line 41 of file [FMC_flags.h](#).

3.42.1.5 UTF16_BE

```
#define UTF16_BE 8
```

Definition at line 37 of file [FMC_flags.h](#).

3.42.1.6 UTF16_LE

```
#define UTF16_LE 4
```

Definition at line 36 of file [FMC_flags.h](#).

3.42.1.7 UTF32_BE

```
#define UTF32_BE 32
```

Definition at line 39 of file [FMC_flags.h](#).

3.42.1.8 UTF32_LE

```
#define UTF32_LE 16
```

Definition at line 38 of file [FMC_flags.h](#).

3.42.1.9 UTF8

```
#define UTF8 1
```

Definition at line 34 of file [FMC_flags.h](#).

3.42.1.10 UTF8_BOM

```
#define UTF8_BOM 2
```

Definition at line 35 of file [FMC_flags.h](#).

3.43 FMC_flags.h

[Go to the documentation of this file.](#)

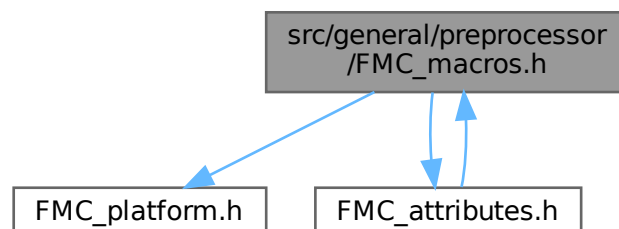
```

00001  /*
00002
00003  MIT License
00004
00005  Copyright (c) 2022 Axel PASCON
00006
00007  Permission is hereby granted, free of charge, to any person obtaining a copy
00008  of this software and associated documentation files (the "Software"), to deal
00009  in the Software without restriction, including without limitation the rights
00010  to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011  copies of the Software, and to permit persons to whom the Software is
00012  furnished to do so, subject to the following conditions:
00013
00014  The above copyright notice and this permission notice shall be included in all
00015  copies or substantial portions of the Software.
00016
00017  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018  IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019  FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020  AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021  LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022  OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023  SOFTWARE.
00024
00025  */
00026
00027  #pragma once
00028
00029  #ifndef FMC_FLAGS
00030  #define FMC_FLAGS
00031
00032  #ifndef FMC_ENCODING_FLAGS
00033      #define FMC_ENCODING_FLAGS
00034      #define UTF8 1
00035      #define UTF8_BOM 2
00036      #define UTF16_LE 4
00037      #define UTF16_BE 8
00038      #define UTF32_LE 16
00039      #define UTF32_BE 32
00040      #define ASCII 64
00041      #define UNKNOWN 128
00042  #endif
00043
00044  #endif // FMC_FLAGS

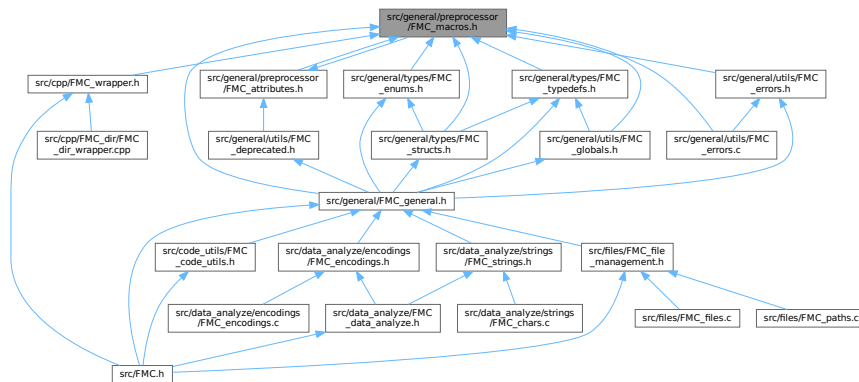
```

3.44 src/general/preprocessor/FMC_macros.h File Reference

Include dependency graph for FMC_macros.h:



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



Macros

- `#define FMC_COMPILE_TIME_ERROR(msg) _Pragma(Stringize(GCC error Stringize(msg)))`
- `#define FMC_DEFER(stmt, body) do body while (0); stmt`
- `#define FMC_ERROR_CHECK(cond, todo_stmt, enable_debug, todo_before)`
- `#define FMC_MACROS_H`
- `#define FMC_MAJOR_VERSION 1`
- `#define FMC_MINOR_VERSION 0`
- `#define FMC_PATCH_VERSION 0`
- `#define FMC_VERSION FMC_CONCAT_5(FMC_MAJOR_VERSION, FMC_PP_POINT(), FMC_MINOR_VERSION, FMC_PP_POINT(), FMC_PATCH_VERSION)`
- `#define FMC_VERSION_NUMBER FMC_CONCAT_2(FMC_MAJOR_VERSION*10000 + FMC_MINOR_VERSION*100 + FMC_PATCH_VERSION, L)`
- `#define FMC_VERSION_STRING FMC_STRINGIZE_5(FMC_MAJOR_VERSION, FMC_PP_POINT(), FMC_MINOR_VERSION, FMC_PP_POINT(), FMC_PATCH_VERSION)`

3.44.1 Macro Definition Documentation

3.44.1.1 FMC_COMPILE_TIME_ERROR

```
#define FMC_COMPILE_TIME_ERROR(  
    msg ) _Pragma(Stringize(GCC error Stringize(msg)))
```

Definition at line 170 of file [FMC_macros.h](#).

3.44.1.2 FMC_DEFER

```
#define FMC_DEFER(  
    stmt,  
    body ) do body while (0); stmt
```

Definition at line 99 of file [FMC_macros.h](#).

3.44.1.3 FMC_ERROR_CHECK

```
#define FMC_ERROR_CHECK(  
    cond,  
    todo_stmt,  
    enable_debug,  
    todo_before )
```

Value:

```
if (cond)  
{ if(enable_debug) todo_before  
  todo_stmt;  
}
```

\\
\\
\\

Definition at line 177 of file [FMC_macros.h](#).

3.44.1.4 FMC_MACROS_H

```
#define FMC_MACROS_H
```

Definition at line 31 of file [FMC_macros.h](#).

3.44.1.5 FMC_MAJOR_VERSION

```
#define FMC_MAJOR_VERSION 1
```

Definition at line 126 of file [FMC_macros.h](#).

3.44.1.6 FMC_MINOR_VERSION

```
#define FMC_MINOR_VERSION 0
```

Definition at line 127 of file [FMC_macros.h](#).

3.44.1.7 FMC_PATCH_VERSION

```
#define FMC_PATCH_VERSION 0
```

Definition at line 128 of file [FMC_macros.h](#).

3.44.1.8 FMC_VERSION

```
#define FMC_VERSION FMC_CONCAT_5(FMC_MAJOR_VERSION, FMC_PP_POINT(), FMC_MINOR_VERSION, FMC_PP_POINT(), FMC_PATCH_VERSION)
```

Definition at line 129 of file [FMC_macros.h](#).

3.44.1.9 FMC_VERSION_NUMBER

```
#define FMC_VERSION_NUMBER FMC_CONCAT_2(FMC_MAJOR_VERSION*10000 + FMC_MINOR_VERSION*100 + FMC_PATCH_VERSION, L)
```

Definition at line 131 of file [FMC_macros.h](#).

3.44.1.10 FMC_VERSION_STRING

```
#define FMC_VERSION_STRING FMC_STRINGIZE_5(FMC_MAJOR_VERSION, FMC_PP_POINT(), FMC_MINOR_VERSION, FMC_PP_POINT(), FMC_PATCH_VERSION)
```

Definition at line 130 of file [FMC_macros.h](#).

3.45 FMC_macros.h

[Go to the documentation of this file.](#)

```
00001 /*
00002
00003 MIT License
00004
00005 Copyright (c) 2022-2023 Axel PASCON
00006
00007 Permission is hereby granted, free of charge, to any person obtaining a copy
00008 of this software and associated documentation files (the "Software"), to deal
00009 in the Software without restriction, including without limitation the rights
00010 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011 copies of the Software, and to permit persons to whom the Software is
00012 furnished to do so, subject to the following conditions:
00013
00014 The above copyright notice and this permission notice shall be included in all
00015 copies or substantial portions of the Software.
00016
00017 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023 SOFTWARE.
00024
00025 */
00026
00027 #pragma once
00028
00029 #ifndef FMC_MACROS_H
00030 #define FMC_MACROS_H
00031 #include "FMC_platform.h"
00032 #include "FMC_attributes.h"
00033 #endif
```



```

00037
00038 /* Used to avoid false warnings (for example "attribute destructor/constructor does not take
00039 argument", when it actually can) */
00039 #if defined(__INTELLISENSE__ )
00040     #pragma diag_suppress 1094
00041 #endif
00042
00043 #ifndef FMC_PP_POINT
00044     #define FMC_PP_POINT() .
00045 #endif
00046
00047 #ifndef FMC_CONCAT_MACROS
00048     #define FMC_CONCAT_MACROS
00049     #define FMC_CONCAT10(x, y) x##y
00050     #define FMC_CONCAT9(x, y) FMC_CONCAT10(x, y)
00051     #define FMC_CONCAT8(x, y) FMC_CONCAT9(x, y)
00052     #define FMC_CONCAT7(x, y) FMC_CONCAT8(x, y)
00053     #define FMC_CONCAT6(x, y) FMC_CONCAT7(x, y)
00054     #define FMC_CONCAT5(x, y) FMC_CONCAT6(x, y)
00055     #define FMC_CONCAT4(x, y) FMC_CONCAT5(x, y)
00056     #define FMC_CONCAT3(x, y) FMC_CONCAT4(x, y)
00057     #define FMC_CONCAT2(x, y) FMC_CONCAT3(x, y)
00058     #define FMC_CONCAT(x, y) FMC_CONCAT2(x, y)
00059
00060     #define FMC_CONCAT_2(x, y) FMC_CONCAT(x, y)
00061     #define FMC_CONCAT_3(x, y, z) FMC_CONCAT(FMC_CONCAT(x, y), z)
00062     #define FMC_CONCAT_4(x, y, z, w) FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(x, y), z), w)
00063     #define FMC_CONCAT_5(x, y, z, w, v) FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(x, y), z), w), v)
00064     #define FMC_CONCAT_6(x, y, z, w, v, u) FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(x,
00065 y), z), w), v), u)
00066     #define FMC_CONCAT_7(x, y, z, w, v, u, t)
00067 FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(x, y), z), w), v), u), t)
00068     #define FMC_CONCAT_8(x, y, z, w, v, u, t, s)
00069 FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(x, y), z), w), v), u),
00070 t), s)
00071     #define FMC_CONCAT_9(x, y, z, w, v, u, t, s, r)
00072 FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(x, y), z), w), v), u), t), s), r)
00073 #endif
00074
00075 #ifndef FMC_STRINGIZE_MACROS
00076     #define FMC_STRINGIZE_MACROS
00077     #define FMC_STRINGIZE10(x) #x
00078     #define FMC_STRINGIZE9(x) FMC_STRINGIZE10(x)
00079     #define FMC_STRINGIZE8(x) FMC_STRINGIZE9(x)
00080     #define FMC_STRINGIZE7(x) FMC_STRINGIZE8(x)
00081     #define FMC_STRINGIZE6(x) FMC_STRINGIZE7(x)
00082     #define FMC_STRINGIZE5(x) FMC_STRINGIZE6(x)
00083     #define FMC_STRINGIZE4(x) FMC_STRINGIZE5(x)
00084     #define FMC_STRINGIZE3(x) FMC_STRINGIZE4(x)
00085     #define FMC_STRINGIZE2(x) FMC_STRINGIZE3(x)
00086     #define FMC_STRINGIZE(x) FMC_STRINGIZE2(x)
00087 #endif
00088
00089 #ifndef FMC_STRINGIZE_X
00090     #define FMC_STRINGIZE_X
00091     #define FMC_STRINGIZE_2(x, y) FMC_STRINGIZE(FMC_CONCAT(x, y))
00092     #define FMC_STRINGIZE_3(x, y, z) FMC_STRINGIZE(FMC_CONCAT(FMC_CONCAT(x, y), z))
00093     #define FMC_STRINGIZE_4(x, y, z, w) FMC_STRINGIZE(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(x, y), z), w))
00094     #define FMC_STRINGIZE_5(x, y, z, w, v)
00095 FMC_STRINGIZE(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(x, y), z), w), v))
00096     #define FMC_STRINGIZE_6(x, y, z, w, v, u)
00097 FMC_STRINGIZE(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(x, y), z), w), v), u))
00098     #define FMC_STRINGIZE_7(x, y, z, w, v, u, t)
00099 FMC_STRINGIZE(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(x, y), z), w), v), u), t))
00100     #define FMC_STRINGIZE_8(x, y, z, w, v, u, t, s)
00101 FMC_STRINGIZE(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(x, y), z), w), v), u), t), s))
00102     #define FMC_STRINGIZE_9(x, y, z, w, v, u, t, s, r)
00103 FMC_STRINGIZE(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(FMC_CONCAT(x, y), z), w), v), u), t), s), r))
00104 #endif
00105
00106 #ifndef FMC_DEFER
00107     #undef FMC_DEFER
00108 #endif
00109 #define FMC_DEFER(stmt, body) do body while (0); stmt
00110
00111
00112 #ifndef FMC_METHODS
00113     #define FMC_METHODS
00114     #define DECL_METHOD(name, ret, ...) \
00115         ret (*name)(__VA_ARGS__)
00116     #define INIT_STRUCT_METHOD(method, associated_function) \

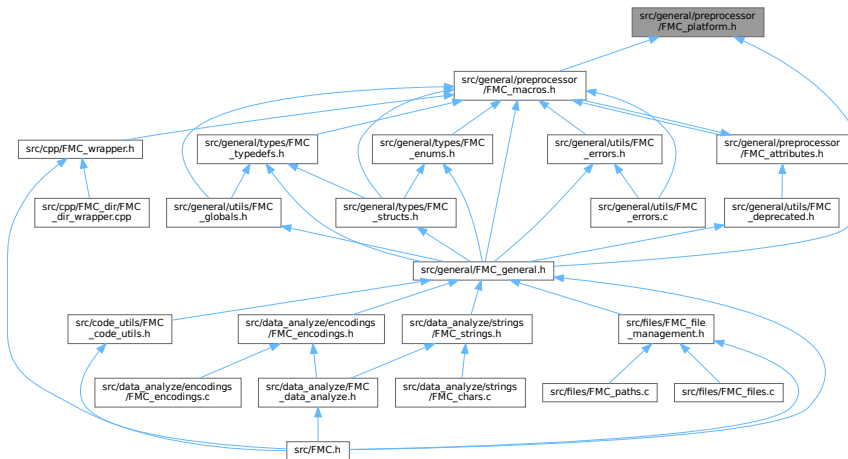
```

```

00109         .method = associated_function
00110
00111 #endif // FMC_METHODS
00112
00113 /*#ifndef FMC_OVERLOAD
00114     #define FMC_OVERLOAD(func)
00115 */
00116
00117 #ifdef FMC_VERSION
00118     #undef FMC_VERSION
00119     #undef FMC_VERSION_STRING
00120     #undef FMC_VERSION_NUMBER
00121     #undef FMC_MAJOR_VERSION
00122     #undef FMC_MINOR_VERSION
00123     #undef FMC_PATCH_VERSION
00124 #endif // FMC_VERSION
00125
00126 #define FMC_MAJOR_VERSION 1
00127 #define FMC_MINOR_VERSION 0
00128 #define FMC_PATCH_VERSION 0
00129 #define FMC_VERSION FMC_CONCAT_5(FMC_MAJOR_VERSION, FMC_PP_POINT(), FMC_MINOR_VERSION, FMC_PP_POINT(),
FMC_PATCH_VERSION)
00130 #define FMC_VERSION_STRING FMC_STRINGIZE_5(FMC_MAJOR_VERSION, FMC_PP_POINT(), FMC_MINOR_VERSION,
FMC_PP_POINT(), FMC_PATCH_VERSION)
00131 #define FMC_VERSION_NUMBER FMC_CONCAT_2(FMC_MAJOR_VERSION*10000 + FMC_MINOR_VERSION*100 +
FMC_PATCH_VERSION, L)
00132
00133 #ifndef FMC_alloca
00134     #define FMC_alloca(size) __builtin_alloca(size)
00135 #endif
00136
00137 #ifndef FMC_PROB
00138     #define FMC_PROB(true_expr, prob) __builtin_expect_with_probability(true_expr, 1, prob)
00139 #endif
00140
00141 #ifndef FMC_UNREACHABLE
00142     #define FMC_UNREACHABLE __builtin_unreachable()
00143 #endif
00144
00145 /* Maybe I'll have to modify this, even though it sounds fine to me now. */
00146 #ifndef FMC_SHARED
00147     #if FMC_COMPILING_ON_WINDOWS && !defined(FMC_STATIC)
00148         #if defined(FMC_BUILD_DLL)
00149             #define FMC_SHARED __declspec(dllexport)
00150         #elif defined(USE_FMC_DLL)
00151             #define FMC_SHARED __declspec(dllimport)
00152         #else
00153             #error "You must define FMC_BUILD_DLL to build the DLL or USE_FMC_DLL to use the built
DLL. To use or build the static library, please define FMC_STATIC."
00154         #endif
00155     #elif FMC_COMPILING_ON_WINDOWS && defined(FMC_STATIC)
00156         #define FMC_SHARED
00157     #elif FMC_COMPILING_ON_LINUX || FMC_COMPILING_ON_MACOS
00158         #if defined(FMC_STATIC) || defined(USE_FMC_DLL) || defined(FMC_BUILD_DLL)
00159             #warning "You don't have to specify FMC_STATIC, USE_FMC_DLL or FMC_BUILD_DLL on Linux,
Unix or Mac OS X. These are ignored on your system."
00160         #endif
00161         #define FMC_SHARED
00162     #else
00163         #error "Unsupported OS"
00164     #endif // PLATFORMS
00165 #endif // FMC_SHARED
00166
00167 #ifdef FMC_COMPILE_TIME_ERROR
00168     #undef FMC_COMPILE_TIME_ERROR
00169 #endif // FMC_COMPILE_TIME_ERROR
00170 #define FMC_COMPILE_TIME_ERROR(msg) _Pragma(Stringize(GCC error Stringize(msg)))
00171
00172
00173 #ifdef FMC_ERROR_CHECK
00174     #undef FMC_ERROR_CHECK
00175 #endif // FMC_ERROR_CHECK
00176 // thought about this for lisibility, not sure if I'll use it though
00177 #define FMC_ERROR_CHECK(cond, todo_stmt, enable_debug, todo_before) \
00178     if (cond) \
00179     {   if(enable_debug) todo_before \
00180         todo_stmt; \
00181     }
00182
00183 #endif // FMC_MACROS_H

```

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



[Go to the documentation of this file.](#)

```

00001 /*
00002
00003 MIT License
00004
00005 Copyright (c) 2022-2023 Axel PASCON
00006
00007 Permission is hereby granted, free of charge, to any person obtaining a copy
00008 of this software and associated documentation files (the "Software"), to deal
00009 in the Software without restriction, including without limitation the rights
00010 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011 copies of the Software, and to permit persons to whom the Software is
00012 furnished to do so, subject to the following conditions:
00013
00014 The above copyright notice and this permission notice shall be included in all
00015 copies or substantial portions of the Software.
00016
00017 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023 SOFTWARE.
00024
00025 */
00026
00027 #ifndef FMC_PLATFORM_H
00028 #define FMC_PLATFORM_H
00029
00030
00031 #if defined(FMC_COMPILING_ON_WINDOWS)
00032     #undef FMC_COMPILING_ON_WINDOWS
00033 #elif defined(FMC_COMPILING_ON_LINUX)
00034     #undef FMC_COMPILING_ON_LINUX
00035 #endif // OS detection
00036 #if defined(__WIN32) || defined(__WIN64) || defined(__WIN32__) || defined(__TOS_WIN__) ||
defined(__WINDOWS__)
00037     #define FMC_COMPILING_ON_WINDOWS 1
00038 #elif defined(__linux__) || defined(__linux) || defined(linux) || defined(__gnu_linux__)
00039     #define FMC_COMPILING_ON_LINUX 1
00040 #else
00041     #warning "This library hasn't been tested on this OS."
00042 #endif // OS management
00043

```

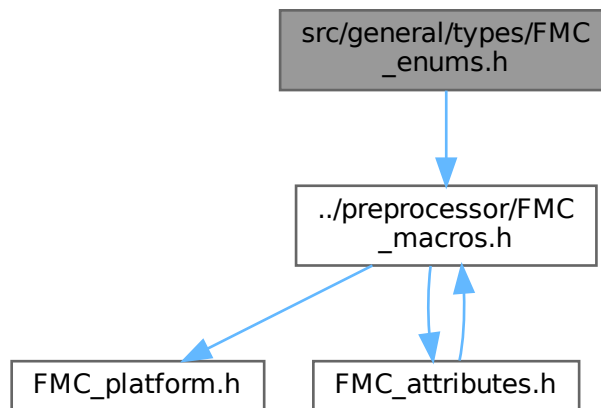
```

00044 #if defined(FMC_COMPILING_ON_MINGW)
00045     #undef FMC_COMPILING_ON_MINGW
00046 #elif defined(FMC_COMPILING_WITH_GCC)
00047     #undef FMC_COMPILING_WITH_GCC
00048 #endif // Compiler and environment detection
00049 #if defined(__MINGW32__) || defined(__MINGW64__) || defined(__MINGW32) || defined(__MINGW64) ||
    defined(__MINGW__)
00050     #define FMC_COMPILING_ON_MINGW 1
00051 #elif defined(__GNUC__) || defined(__GNUG__)
00052     #define FMC_COMPILING_WITH_GCC 1
00053 #else
00054     #warning "This library hasn't been tested on your compiler."
00055 #endif // Compiler and environment management
00056
00057 // check C17 standard
00058 #ifndef __cplusplus
00059     #if __STDC_VERSION__ < 201710L
00060         #error "FManC requires C17 standard or higher."
00061     #endif
00062 #else
00063     #if __cplusplus < 201703L
00064         #error "FManC requires C++17 standard or higher."
00065     #endif
00066 #endif
00067
00068 #endif /* FMC_PLATFORM_H */

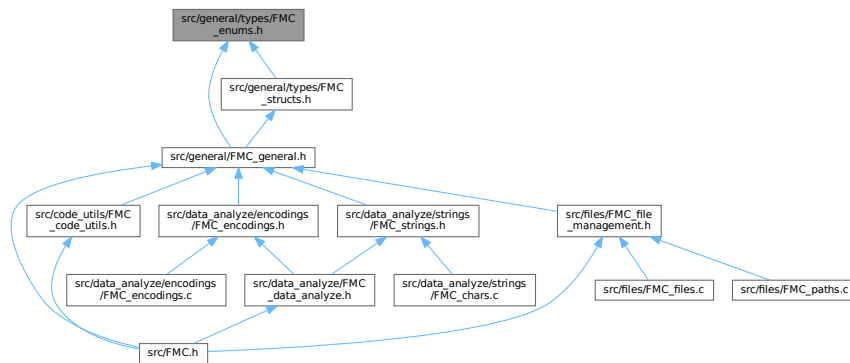
```

3.48 src/general/types/FMC_enums.h File Reference

Include dependency graph for FMC_enums.h:



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



Macros

- `#define FMC_ENUMS_H`

Typedefs

- `typedef enum FManC_Encodings FMC_Encodings`

Enumerations

- `enum FManC_Encodings {
utf8 = 1 , utf8_bom = 2 , utf16_le = 4 , utf16_be = 8 ,
utf32_le = 16 , utf32_be = 32 , ascii = 64 , unknown = 128 ,
error = 256 }`

3.48.1 Macro Definition Documentation

3.48.1.1 FMC_ENUMS_H

```
#define FMC_ENUMS_H
```

Definition at line 30 of file [FMC_enums.h](#).

3.48.2 Typedef Documentation

3.48.2.1 FMC_Encodings

```
typedef enum FManC_Encodings FMC_Encodings
```

Definition at line 47 of file [FMC_enums.h](#).

3.48.3 Enumeration Type Documentation

3.48.3.1 FManC_Encodings

```
enum FManC_Encodings
```

Enumerator

utf8	
utf8_bom	
utf16_le	
utf16_be	
utf32_le	
utf32_be	
ascii	
unknown	
error	

Definition at line 34 of file [FMC_enums.h](#).

3.49 FMC_enums.h

[Go to the documentation of this file.](#)

```
00001 /*
00002
00003 MIT License
00004
00005 Copyright (c) 2022-2023 Axel PASCON
00006
00007 Permission is hereby granted, free of charge, to any person obtaining a copy
00008 of this software and associated documentation files (the "Software"), to deal
00009 in the Software without restriction, including without limitation the rights
00010 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011 copies of the Software, and to permit persons to whom the Software is
00012 furnished to do so, subject to the following conditions:
00013
00014 The above copyright notice and this permission notice shall be included in all
00015 copies or substantial portions of the Software.
00016
00017 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023 SOFTWARE.
00024
00025 */
00026
00027 #pragma once
```

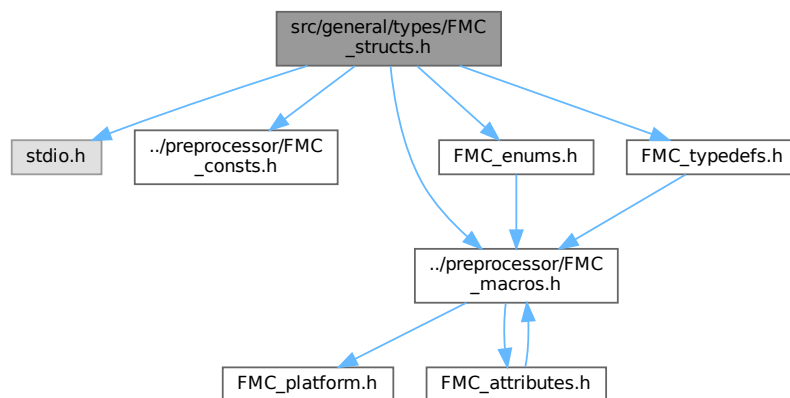
```

00028
00029 #ifndef FMC_ENUMS_H
00030 #define FMC_ENUMS_H
00031
00032 #include "../preprocessor/FMC_macros.h"
00033
00034 FMC_SHARED enum FManC_Encodings
00035 {
00036     utf8 = 1,
00037     utf8_bom = 2,
00038     utf16_le = 4,
00039     utf16_be = 8,
00040     utf32_le = 16,
00041     utf32_be = 32,
00042     ascii = 64,
00043     unknown = 128,
00044     error = 256
00045 };
00046
00047 typedef enum FManC_Encodings FMC_Encodings;
00048
00049 #endif // FMC_ENUMS_H

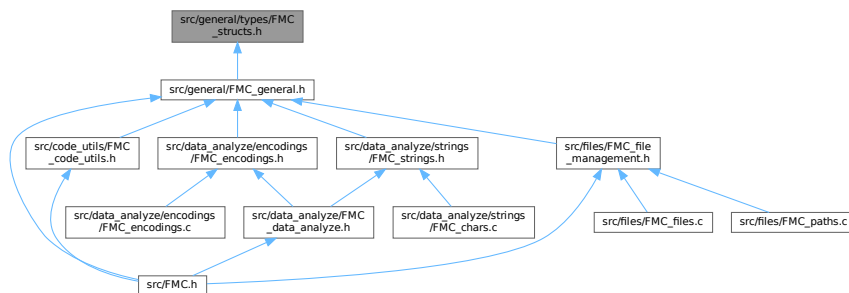
```

3.50 src/general/types/FMC_structs.h File Reference

Include dependency graph for FMC_structs.h:



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



Data Structures

- struct [FManC_Char](#)
- struct [FManC_CharComp](#)
- struct [FManC_CStrView](#)
- struct [FManC_File](#)
- struct [FManC_String](#)
- struct [FManC_StrOcc](#)

Macros

- `#define` [FMC_STRUCTS_H](#)

Typedefs

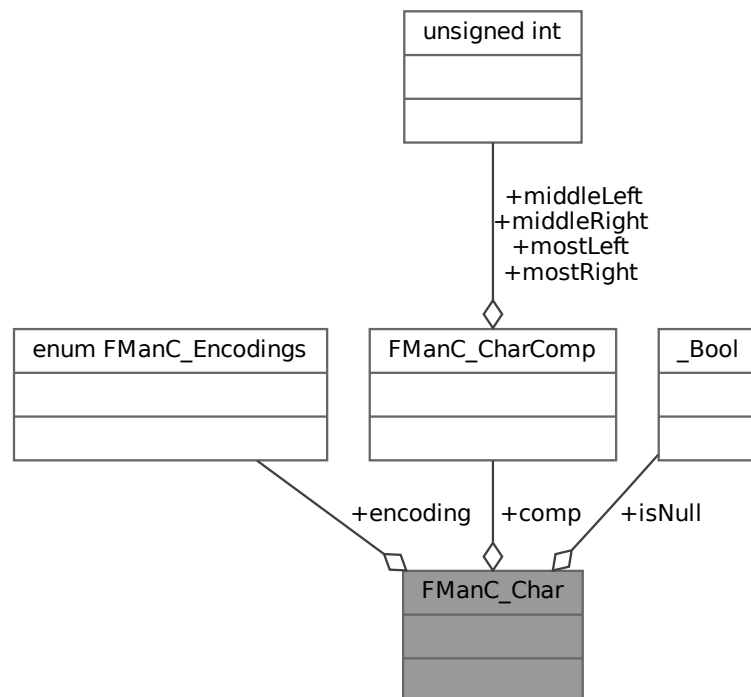
- typedef struct [FManC_Char](#) [FMC_Char](#)
- typedef struct [FManC_CharComp](#) [FMC_CharComp](#)
- typedef struct [FManC_CStrView](#) [FMC_CStrView](#)
- typedef struct [FManC_File](#) [FMC_File](#)
- typedef struct [FManC_String](#) [FMC_String](#)
- typedef struct [FManC_StrOcc](#) [FMC_StrOcc](#)

3.50.1 Data Structure Documentation

3.50.1.1 struct FManC_Char

Definition at line 69 of file [FMC_structs.h](#).

Collaboration diagram for FManC_Char:



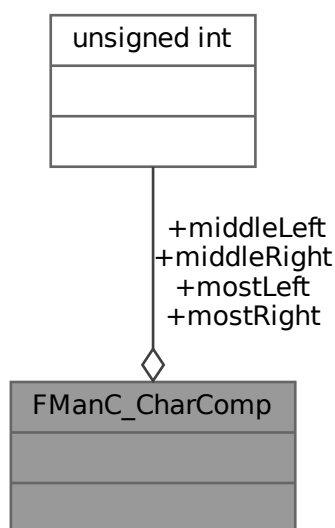
Data Fields

FMC_CharComp	comp	
FMC_Encodings	encoding	
FMC_CharControl	isNull	

3.50.1.2 struct FManC_CharComp

Definition at line 59 of file [FMC_structs.h](#).

Collaboration diagram for FManC_CharComp:



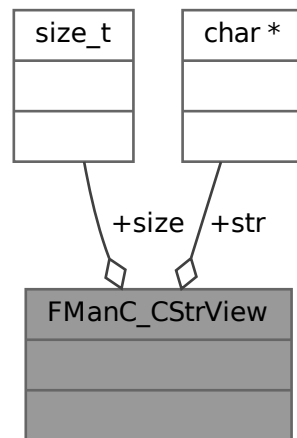
Data Fields

unsigned int	middleLeft: 8	
unsigned int	middleRight: 8	
unsigned int	mostLeft: 8	
unsigned int	mostRight: 8	

3.50.1.3 struct FManC_CStrView

Definition at line 87 of file [FMC_structs.h](#).

Collaboration diagram for FManC_CStrView:



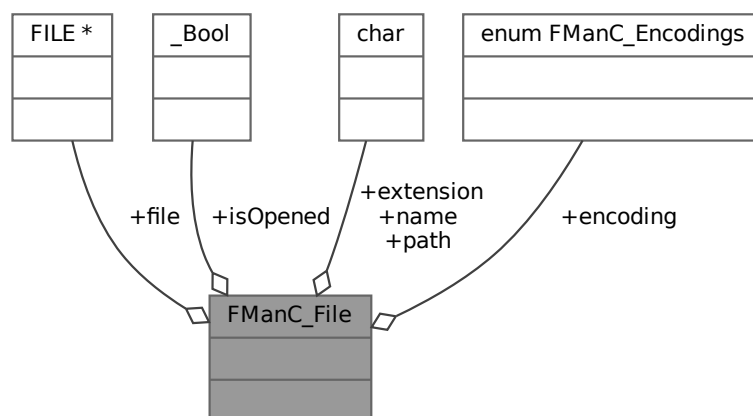
Data Fields

size_t	size	
char *	str	

3.50.1.4 struct FManC_File

Definition at line 39 of file [FMC_structs.h](#).

Collaboration diagram for FManC_File:



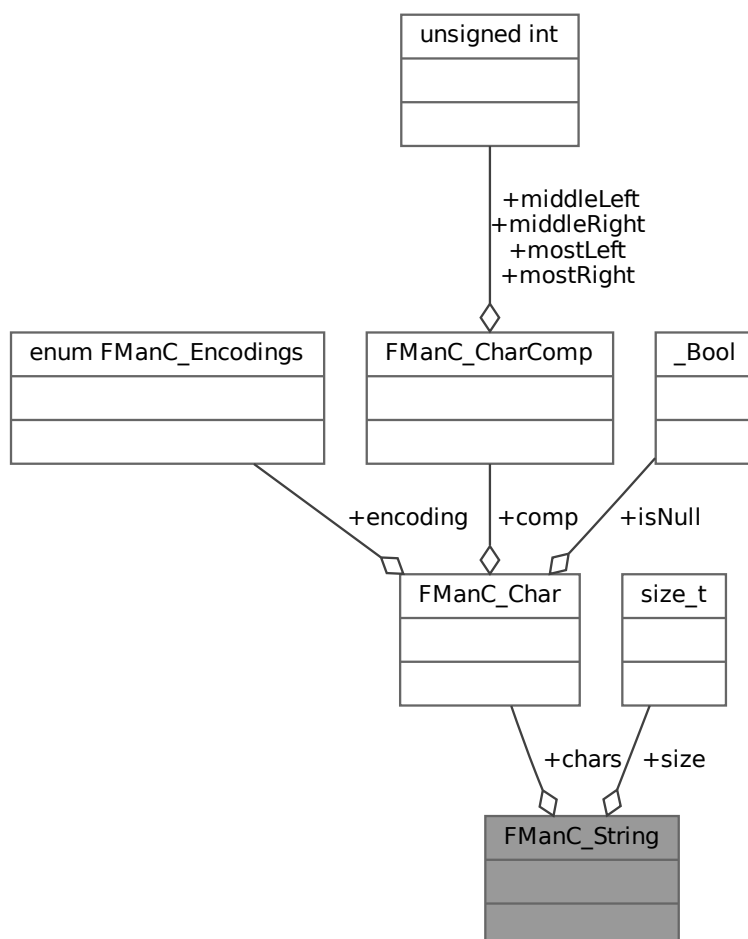
Data Fields

FMC_Encodings	encoding	
char	extension[MAX_FEXT_SIZE]	
FILE *	file	
FMC_FileState	isOpened	
char	name[MAX_FNAME_SIZE]	
char	path[MAX_FPATH_SIZE]	

3.50.1.5 struct FManC_String

Definition at line 79 of file [FMC_structs.h](#).

Collaboration diagram for FManC_String:



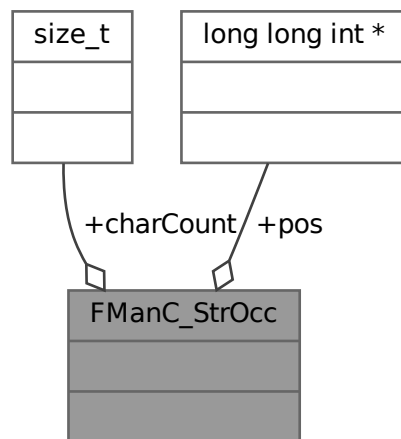
Data Fields

FMC_Char *	chars	
size_t	size	

3.50.1.6 struct FManC_StrOcc

Definition at line 51 of file [FMC_structs.h](#).

Collaboration diagram for FManC_StrOcc:



Data Fields

size_t	charCount	
long long int *	pos	

3.50.2 Macro Definition Documentation**3.50.2.1 FMC_STRUCTS_H**

```
#define FMC_STRUCTS_H
```

Definition at line 30 of file [FMC_structs.h](#).

3.50.3 Typedef Documentation

3.50.3.1 FMC_Char

```
typedef struct FManC_Char FMC_Char
```

Definition at line 76 of file [FMC_structs.h](#).

3.50.3.2 FMC_CharComp

```
typedef struct FManC_CharComp FMC_CharComp
```

Definition at line 67 of file [FMC_structs.h](#).

3.50.3.3 FMC_CStrView

```
typedef struct FManC_CStrView FMC_CStrView
```

Definition at line 93 of file [FMC_structs.h](#).

3.50.3.4 FMC_File

```
typedef struct FManC_File FMC_File
```

Definition at line 49 of file [FMC_structs.h](#).

3.50.3.5 FMC_String

```
typedef struct FManC_String FMC_String
```

Definition at line 85 of file [FMC_structs.h](#).

3.50.3.6 FMC_StrOcc

```
typedef struct FManC_StrOcc FMC_StrOcc
```

Definition at line 57 of file [FMC_structs.h](#).

3.51 FMC_structs.h

[Go to the documentation of this file.](#)

```

00001  /*
00002
00003  MIT License
00004
00005  Copyright (c) 2022-2023 Axel PASCON
00006
00007  Permission is hereby granted, free of charge, to any person obtaining a copy
00008  of this software and associated documentation files (the "Software"), to deal
00009  in the Software without restriction, including without limitation the rights
00010  to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011  copies of the Software, and to permit persons to whom the Software is
00012  furnished to do so, subject to the following conditions:
00013
00014  The above copyright notice and this permission notice shall be included in all
00015  copies or substantial portions of the Software.
00016
00017  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018  IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019  FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020  AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021  LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022  OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023  SOFTWARE.
00024
00025  */
00026
00027  #pragma once
00028
00029  #ifndef FMC_STRUCTS_H
00030  #define FMC_STRUCTS_H
00031
00032
00033  #include <stdio.h>
00034  #include "../preprocessor/FMC_consts.h"
00035  #include "../preprocessor/FMC_macros.h"
00036  #include "FMC_enums.h"
00037  #include "FMC_typedefs.h"
00038
00039  FMC_SHARED struct FManC_File
00040  {
00041      FILE *file;
00042      FMC_FileState isOpened;
00043      char path[MAX_FPATH_SIZE];
00044      char name[MAX_FNAME_SIZE];
00045      char extension[MAX_FEXT_SIZE];
00046      FMC_Encodings encoding;
00047  };
00048
00049  typedef struct FManC_File FMC_File;
00050
00051  FMC_SHARED struct FManC_StrOcc
00052  {
00053      size_t charCount;
00054      long long int *pos;
00055  };
00056
00057  typedef struct FManC_StrOcc FMC_StrOcc;
00058
00059  FMC_SHARED struct FManC_CharComp
00060  {
00061      unsigned int mostLeft : 8;
00062      unsigned int middleLeft : 8;
00063      unsigned int middleRight : 8;
00064      unsigned int mostRight : 8;
00065  };
00066
00067  typedef struct FManC_CharComp FMC_CharComp;
00068
00069  FMC_SHARED struct FManC_Char
00070  {
00071      FMC_Encodings encoding;
00072      FMC_CharComp comp;
00073      FMC_CharControl isNull;
00074  };
00075
00076  typedef struct FManC_Char FMC_Char;
00077
00078
00079  FMC_SHARED struct FManC_String
00080  {
00081      FMC_Char *chars;
00082      size_t size;

```

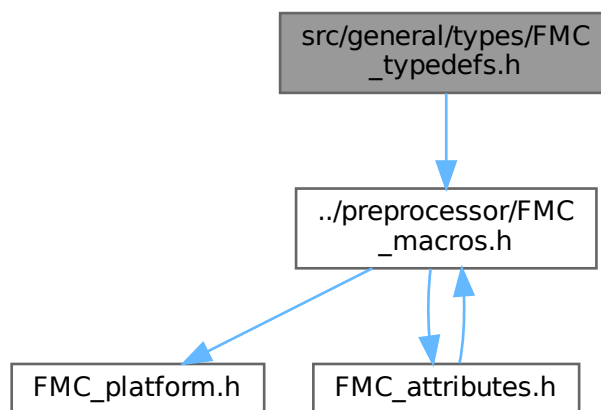
```

00083 };
00084
00085 typedef struct FManC_String FMC_String;
00086
00087 FMC_SHARED struct FManC_CStrView
00088 {
00089     size_t size;
00090     char *str;
00091 };
00092
00093 typedef struct FManC_CStrView FMC_CStrView;
00094
00095 /*#include <threads.h>
00096
00097 FMC_SHARED struct FManC_ArenaElement
00098 {
00099     void* current;
00100     size_t alignment;
00101 };
00102
00103 FMC_SHARED struct FManC_Arena
00104 {
00105     void* start;
00106     void* end;
00107 };*/
00108
00109 #endif // FMC_STRUCTS_H
00110
00111 #endif // FMC_STRUCTS_H

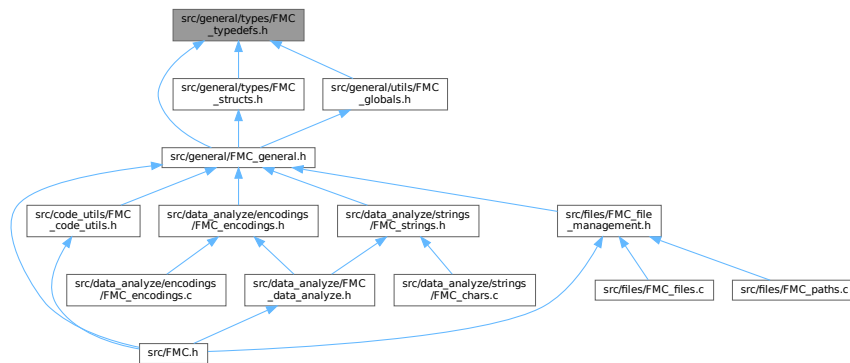
```

3.52 src/general/types/FMC_typedefs.h File Reference

Include dependency graph for FMC_typedefs.h:



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



Macros

- `#define FMC_TYPEDEFS_H`

Typedefs

- `typedef _Bool FMC_Bool`
- `typedef _Bool FMC_CharControl`
- `typedef _Bool FMC_FileState`
- `typedef int found_bs_n`
- `typedef int found_bs_r_bs_n`
- `typedef int found_bs_t`

3.52.1 Macro Definition Documentation

3.52.1.1 FMC_TYPEDEFS_H

```
#define FMC_TYPEDEFS_H
```

Definition at line 30 of file [FMC_typedefs.h](#).

3.52.2 Typedef Documentation

3.52.2.1 FMC_Bool

```
typedef _Bool FMC_Bool
```

Definition at line 39 of file [FMC_typedefs.h](#).

3.52.2.2 FMC_CharControl

```
typedef _Bool FMC_CharControl
```

Definition at line 37 of file [FMC_typedefs.h](#).

3.52.2.3 FMC_FileState

```
typedef _Bool FMC_FileState
```

Definition at line 38 of file [FMC_typedefs.h](#).

3.52.2.4 found_bs_n

```
typedef int found_bs_n
```

Definition at line 34 of file [FMC_typedefs.h](#).

3.52.2.5 found_bs_r_bs_n

```
typedef int found_bs_r_bs_n
```

Definition at line 36 of file [FMC_typedefs.h](#).

3.52.2.6 found_bs_t

```
typedef int found_bs_t
```

Definition at line 35 of file [FMC_typedefs.h](#).

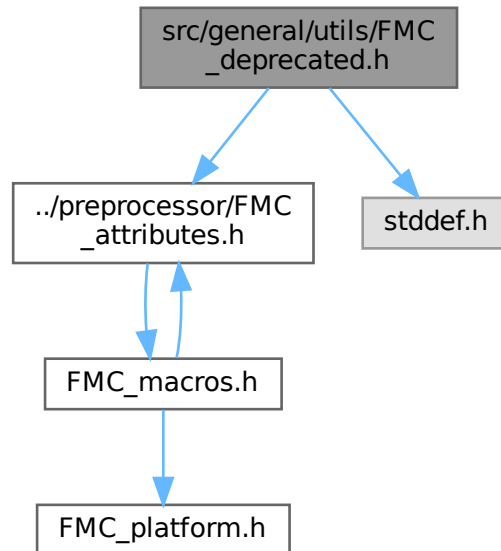
3.53 FMC_typedefs.h

[Go to the documentation of this file.](#)

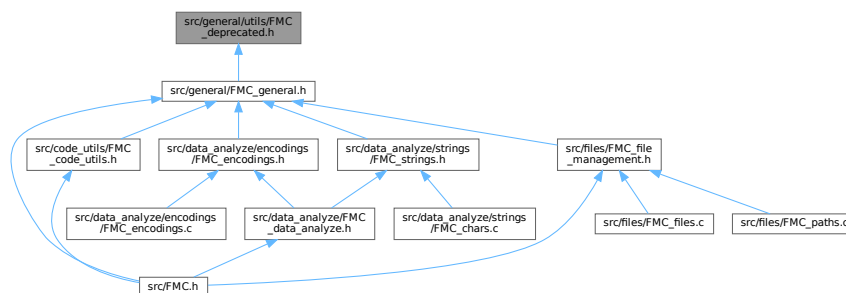
```
00001 /*
00002
00003 MIT License
00004
00005 Copyright (c) 2022-2023 Axel PASCON
00006
00007 Permission is hereby granted, free of charge, to any person obtaining a copy
00008 of this software and associated documentation files (the "Software"), to deal
00009 in the Software without restriction, including without limitation the rights
00010 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011 copies of the Software, and to permit persons to whom the Software is
00012 furnished to do so, subject to the following conditions:
00013
00014 The above copyright notice and this permission notice shall be included in all
00015 copies or substantial portions of the Software.
00016
00017 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023 SOFTWARE.
00024
00025 */
00026
00027 #pragma once
00028
00029 #ifndef FMC_TYPEDEFS_H
00030 #define FMC_TYPEDEFS_H
00031
00032 #include "../preprocessor/FMC_macros.h"
00033
00034 typedef int found_bs_n;
00035 typedef int found_bs_t;
00036 typedef int found_bs_r_bs_n;
00037 typedef _Bool FMC_CharControl;
00038 typedef _Bool FMC_FileState;
00039 typedef _Bool FMC_Bool;
00040
00041
00042 #endif // FMC_TYPEDEFS_H
```

3.54 src/general/utis/FMC_deprecated.h File Reference

Include dependency graph for FMC_deprecated.h:



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



Functions

- [FMC_FUNC_UNAVAILABLE](#) (This function is not anymore available in the library since the version 1.0.0. Use [FMC_cutFilename](#) instead) void fgetFilePath(char *sourceFilePath
- [FMC_FUNC_UNAVAILABLE](#) (This function is not anymore available in the library since the version 1.0.0. Use [FMC_extractFilename](#) instead) void fgetFileName(char *sourceFilePath
- [FMC_FUNC_UNAVAILABLE](#) (This function is not anymore available in the library since the version 1.0.0. Use [FMC_getExtension](#) instead) void fgetFileExtension(char *sourceFilePath
- [FMC_FUNC_UNAVAILABLE](#) (This function is not anymore available in the library since the version 1.0.0.) char *copyFileWithoutTabAndLineBreak(char *sourceFilePath
- [FMC_TYPE_UNAVAILABLE](#) (This type is not anymore available in the library since the version 1.0.0.) struct FMANC_SO

Variables

- char * [extension](#)
- char * [fileName](#)
- char * [filePath](#)
- char ** [pathToCopy](#)
- char * [toSearch](#)

3.54.1 Function Documentation

3.54.1.1 FMC_FUNC_UNAVAILABLE() [1/4]

```
FMC_FUNC_UNAVAILABLE (
    This function is not anymore available in the library since the version 1.0.0.
    Use FMC\_cutFilename instead )
```

3.54.1.2 FMC_FUNC_UNAVAILABLE() [2/4]

```
FMC_FUNC_UNAVAILABLE (
    This function is not anymore available in the library since the version 1.0.0.
    Use FMC\_extractFilename instead )
```

3.54.1.3 FMC_FUNC_UNAVAILABLE() [3/4]

```
FMC_FUNC_UNAVAILABLE (
    This function is not anymore available in the library since the version 1.0.0.
    Use FMC\_getExtension instead )
```

3.54.1.4 FMC_FUNC_UNAVAILABLE() [4/4]

```
FMC_FUNC_UNAVAILABLE (
    This function is not anymore available in the library since the version 1.0. 0.
    )
```

3.54.1.5 FMC_TYPE_UNAVAILABLE()

```
FMC_TYPE_UNAVAILABLE (
    This type is not anymore available in the library since the version 1.0. 0. )
```

Definition at line 8 of file [FMC_deprecated.h](#).

3.54.2 Variable Documentation

3.54.2.1 extension

`char* extension`

Definition at line 28 of file [FMC_deprecated.h](#).

3.54.2.2 fileName

`char* fileName`

Definition at line 22 of file [FMC_deprecated.h](#).

3.54.2.3 filePath

`char* filePath`

Definition at line 25 of file [FMC_deprecated.h](#).

3.54.2.4 pathToCopy

`char** pathToCopy`

Definition at line 19 of file [FMC_deprecated.h](#).

3.54.2.5 toSearch

`char* toSearch`

Definition at line 40 of file [FMC_deprecated.h](#).

3.55 FMC_deprecated.h

[Go to the documentation of this file.](#)

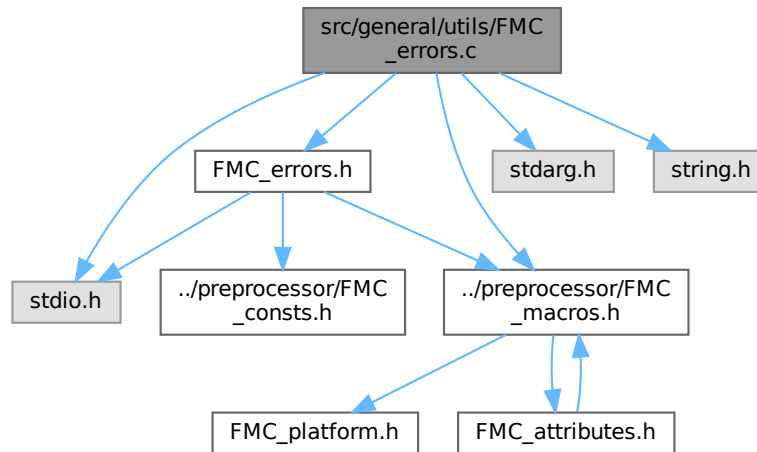
```

00001 #ifndef FMC_DEPRECATED_H
00002 #define FMC_DEPRECATED_H
00003
00004 #include "../preprocessor/FMC_attributes.h"
00005 #include <stddef.h>
00006
00007 #if !defined(BUILDING_FMANC)
00008 FMC_TYPE_UNAVAILABLE(This type is not anymore available in the library since the version 1.0.0.)
00009 struct FMANC_SO
00010 {
00011     size_t charCount;
00012     long long int *pos;
00013 };
00014
00015 FMC_TYPE_UNAVAILABLE(This type is not anymore available in the library since the version 1.0.0.)
00016 typedef struct FMANC_SO stringOccurrences;
00017
00018 FMC_FUNC_UNAVAILABLE(This function is not anymore available in the library since the version 1.0.0.)
00019 char *copyFileWithoutTabAndLineBreak(char *sourceFilePath, char **pathToCopy);
00020
00021 FMC_FUNC_UNAVAILABLE(This function is not anymore available in the library since the version 1.0.0.
00022 Use FMC_extractFilename instead)
00023 void fgetFileName(char *sourceFilePath, char *fileName);
00024
00025 FMC_FUNC_UNAVAILABLE(This function is not anymore available in the library since the version 1.0.0.
00026 Use FMC_cutFilename instead)
00027 void fgetFilePath(char *sourceFilePath, char *filePath);
00028
00029 FMC_FUNC_UNAVAILABLE(This function is not anymore available in the library since the version 1.0.0.
00030 Use FMC_getExtension instead)
00031 void fgetFileExtension(char *sourceFilePath, char *extension);
00032
00033 FMC_FUNC_UNAVAILABLE(This function is not anymore available in the library since the version 1.0.0.)
00034 size_t countCharInFile(char *filePath);
00035
00036 FMC_FUNC_UNAVAILABLE(This function is not anymore available in the library since the version 1.0.0.)
00037 stringOccurrences *init_StringOccurrences(size_t sizeOfString);
00038
00039 FMC_FUNC_UNAVAILABLE(This function is not anymore available in the library since the version 1.0.0.)
00040 void free_stringOccurrences();
00041
00042 FMC_FUNC_UNAVAILABLE(This function is not anymore available in the library since the version 1.0.0.)
00043 stringOccurrences *searchStringInFile(char *filePath, char *toSearch);
00044
00045 FMC_FUNC_UNAVAILABLE(This function is not anymore available in the library since the version 1.0.0.)
00046 int deleteCStyleComments(char *filePath);
00047
00048 #endif // BUILDING_FMANC
00049 #endif // FMC_DEPRECATED_H

```

3.56 src/general/utls/FMC_errors.c File Reference

Include dependency graph for FMC_errors.c:



Functions

- void [FMC_changeStreamTextColorToBlue](#) (FILE *stream)
- void [FMC_changeStreamTextColorToBrightBlue](#) (FILE *stream)
- void [FMC_changeStreamTextColorToBrightCyan](#) (FILE *stream)
- void [FMC_changeStreamTextColorToBrightGreen](#) (FILE *stream)
- void [FMC_changeStreamTextColorToBrightMagenta](#) (FILE *stream)
- void [FMC_changeStreamTextColorToBrightRed](#) (FILE *stream)
- void [FMC_changeStreamTextColorToBrightWhite](#) (FILE *stream)
- void [FMC_changeStreamTextColorToBrightYellow](#) (FILE *stream)
- void [FMC_changeStreamTextColorToCyan](#) (FILE *stream)
- void [FMC_changeStreamTextColorToGreen](#) (FILE *stream)
- void [FMC_changeStreamTextColorToMagenta](#) (FILE *stream)
- void [FMC_changeStreamTextColorToRed](#) (FILE *stream)
- void [FMC_changeStreamTextColorToWhite](#) (FILE *stream)
- void [FMC_changeStreamTextColorToYellow](#) (FILE *stream)
- void [FMC_makeMsg_f](#) (char *buff, unsigned int argc,...)
- void [FMC_printBlueError](#) (FILE *stream, const char *text)
- void [FMC_printBlueText](#) (FILE *stream, const char *text)
- void [FMC_printBrightBlueError](#) (FILE *stream, const char *text)
- void [FMC_printBrightBlueText](#) (FILE *stream, const char *text)
- void [FMC_printBrightCyanError](#) (FILE *stream, const char *text)
- void [FMC_printBrightCyanText](#) (FILE *stream, const char *text)
- void [FMC_printBrightGreenError](#) (FILE *stream, const char *text)
- void [FMC_printBrightGreenText](#) (FILE *stream, const char *text)
- void [FMC_printBrightMagentaError](#) (FILE *stream, const char *text)
- void [FMC_printBrightMagentaText](#) (FILE *stream, const char *text)
- void [FMC_printBrightRedError](#) (FILE *stream, const char *text)
- void [FMC_printBrightRedText](#) (FILE *stream, const char *text)

- void [FMC_printBrightWhiteError](#) (FILE *stream, const char *text)
- void [FMC_printBrightWhiteText](#) (FILE *stream, const char *text)
- void [FMC_printBrightYellowError](#) (FILE *stream, const char *text)
- void [FMC_printBrightYellowText](#) (FILE *stream, const char *text)
- void [FMC_printCyanError](#) (FILE *stream, const char *text)
- void [FMC_printCyanText](#) (FILE *stream, const char *text)
- void [FMC_printGreenError](#) (FILE *stream, const char *text)
- void [FMC_printGreenText](#) (FILE *stream, const char *text)
- void [FMC_printMagentaError](#) (FILE *stream, const char *text)
- void [FMC_printMagentaText](#) (FILE *stream, const char *text)
- void [FMC_printRedError](#) (FILE *stream, const char *text)
- void [FMC_printRedText](#) (FILE *stream, const char *text)
- void [FMC_printWhiteError](#) (FILE *stream, const char *text)
- void [FMC_printWhiteText](#) (FILE *stream, const char *text)
- void [FMC_printYellowError](#) (FILE *stream, const char *text)
- void [FMC_printYellowText](#) (FILE *stream, const char *text)
- void [FMC_resetStreamOutputStyle](#) (FILE *stream)

3.56.1 Function Documentation

3.56.1.1 FMC_changeStreamTextColorToBlue()

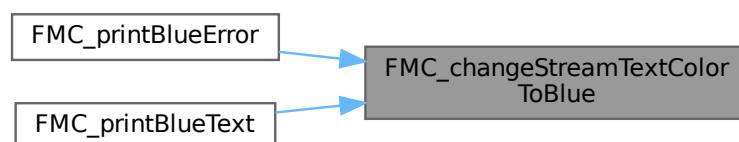
```
void FMC_changeStreamTextColorToBlue (
    FILE * stream )
```

Definition at line 63 of file [FMC_errors.h](#).

References [FG_BLUE](#).

Referenced by [FMC_printBlueError\(\)](#), and [FMC_printBlueText\(\)](#).

Here is the caller graph for this function:



3.56.1.2 FMC_changeStreamTextColorToBrightBlue()

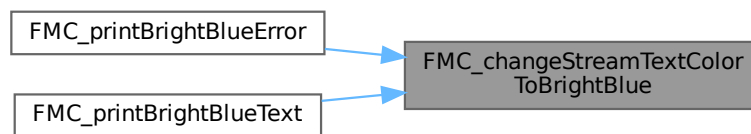
```
void FMC_changeStreamTextColorToBrightBlue (
    FILE * stream )
```

Definition at line 98 of file [FMC_errors.h](#).

References [FG_BRIGHT_BLUE](#).

Referenced by [FMC_printBrightBlueError\(\)](#), and [FMC_printBrightBlueText\(\)](#).

Here is the caller graph for this function:



3.56.1.3 FMC_changeStreamTextColorToBrightCyan()

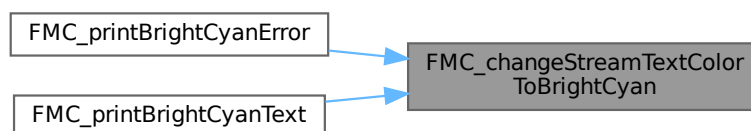
```
void FMC_changeStreamTextColorToBrightCyan (
    FILE * stream )
```

Definition at line 108 of file [FMC_errors.h](#).

References [FG_BRIGHT_CYAN](#).

Referenced by [FMC_printBrightCyanError\(\)](#), and [FMC_printBrightCyanText\(\)](#).

Here is the caller graph for this function:



3.56.1.4 FMC_changeStreamTextColorToBrightGreen()

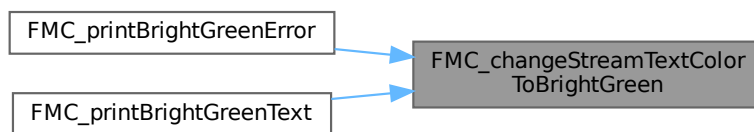
```
void FMC_changeStreamTextColorToBrightGreen (
    FILE * stream )
```

Definition at line 88 of file [FMC_errors.h](#).

References [FG_BRIGHT_GREEN](#).

Referenced by [FMC_printBrightGreenError\(\)](#), and [FMC_printBrightGreenText\(\)](#).

Here is the caller graph for this function:



3.56.1.5 FMC_changeStreamTextColorToBrightMagenta()

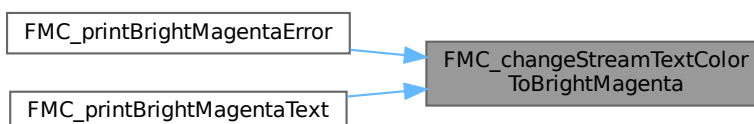
```
void FMC_changeStreamTextColorToBrightMagenta (
    FILE * stream )
```

Definition at line 103 of file [FMC_errors.h](#).

References [FG_BRIGHT_MAGENTA](#).

Referenced by [FMC_printBrightMagentaError\(\)](#), and [FMC_printBrightMagentaText\(\)](#).

Here is the caller graph for this function:



3.56.1.6 FMC_changeStreamTextColorToBrightRed()

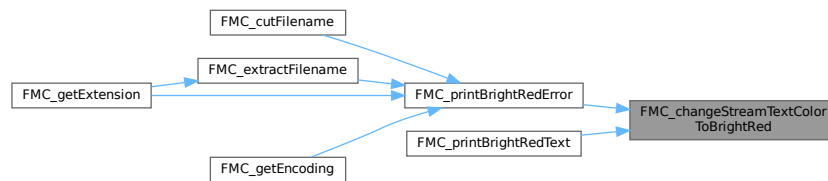
```
void FMC_changeStreamTextColorToBrightRed (
    FILE * stream )
```

Definition at line 83 of file [FMC_errors.h](#).

References [FG_BRIGHT_RED](#).

Referenced by [FMC_printBrightRedError\(\)](#), and [FMC_printBrightRedText\(\)](#).

Here is the caller graph for this function:



3.56.1.7 FMC_changeStreamTextColorToBrightWhite()

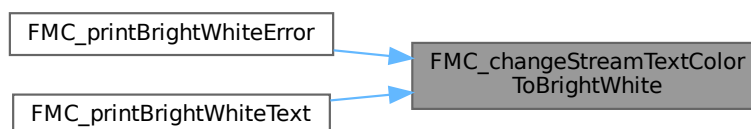
```
void FMC_changeStreamTextColorToBrightWhite (
    FILE * stream )
```

Definition at line 113 of file [FMC_errors.h](#).

References [FG_BRIGHT_WHITE](#).

Referenced by [FMC_printBrightWhiteError\(\)](#), and [FMC_printBrightWhiteText\(\)](#).

Here is the caller graph for this function:



3.56.1.8 FMC_changeStreamTextColorToBrightYellow()

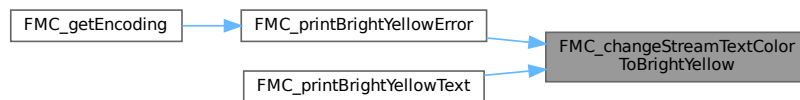
```
void FMC_changeStreamTextColorToBrightYellow (
    FILE * stream )
```

Definition at line 93 of file [FMC_errors.h](#).

References [FG_BRIGHT_YELLOW](#).

Referenced by [FMC_printBrightYellowError\(\)](#), and [FMC_printBrightYellowText\(\)](#).

Here is the caller graph for this function:



3.56.1.9 FMC_changeStreamTextColorToCyan()

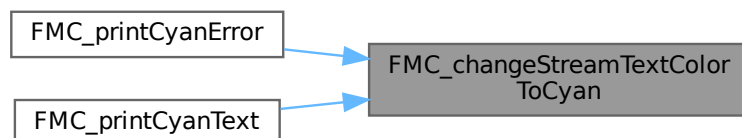
```
void FMC_changeStreamTextColorToCyan (
    FILE * stream )
```

Definition at line 73 of file [FMC_errors.h](#).

References [FG_CYAN](#).

Referenced by [FMC_printCyanError\(\)](#), and [FMC_printCyanText\(\)](#).

Here is the caller graph for this function:



3.56.1.10 FMC_changeStreamTextColorToGreen()

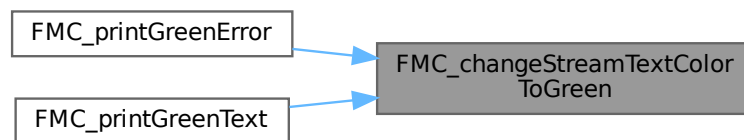
```
void FMC_changeStreamTextColorToGreen (
    FILE * stream )
```

Definition at line 53 of file [FMC_errors.h](#).

References [FG_GREEN](#).

Referenced by [FMC_printGreenError\(\)](#), and [FMC_printGreenText\(\)](#).

Here is the caller graph for this function:



3.56.1.11 FMC_changeStreamTextColorToMagenta()

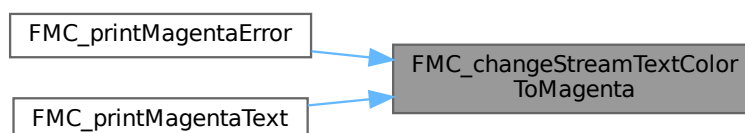
```
void FMC_changeStreamTextColorToMagenta (
    FILE * stream )
```

Definition at line 68 of file [FMC_errors.h](#).

References [FG_MAGENTA](#).

Referenced by [FMC_printMagentaError\(\)](#), and [FMC_printMagentaText\(\)](#).

Here is the caller graph for this function:



3.56.1.12 FMC_changeStreamTextColorToRed()

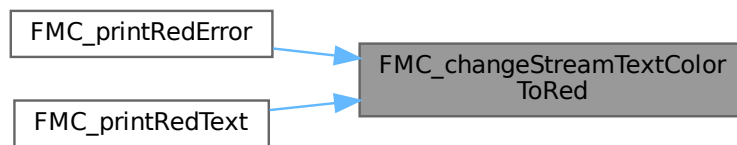
```
void FMC_changeStreamTextColorToRed (  
    FILE * stream )
```

Definition at line 48 of file [FMC_errors.h](#).

References [FG_RED](#).

Referenced by [FMC_printRedError\(\)](#), and [FMC_printRedText\(\)](#).

Here is the caller graph for this function:



3.56.1.13 FMC_changeStreamTextColorToWhite()

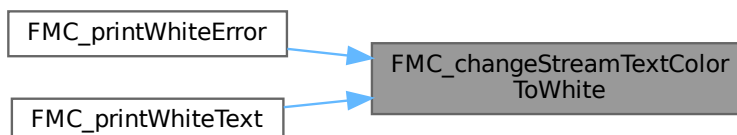
```
void FMC_changeStreamTextColorToWhite (  
    FILE * stream )
```

Definition at line 78 of file [FMC_errors.h](#).

References [FG_WHITE](#).

Referenced by [FMC_printWhiteError\(\)](#), and [FMC_printWhiteText\(\)](#).

Here is the caller graph for this function:



3.56.1.14 FMC_changeStreamTextColorToYellow()

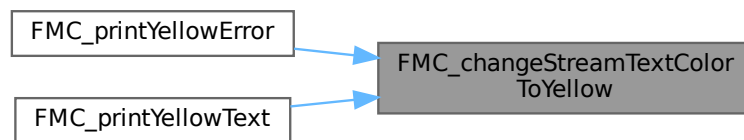
```
void FMC_changeStreamTextColorToYellow (
    FILE * stream )
```

Definition at line 58 of file [FMC_errors.h](#).

References [FG_YELLOW](#).

Referenced by [FMC_printYellowError\(\)](#), and [FMC_printYellowText\(\)](#).

Here is the caller graph for this function:



3.56.1.15 FMC_makeMsg_f()

```
void FMC_makeMsg_f (
    char * buff,
    unsigned int argc,
    ... )
```

Definition at line 33 of file [FMC_errors.c](#).

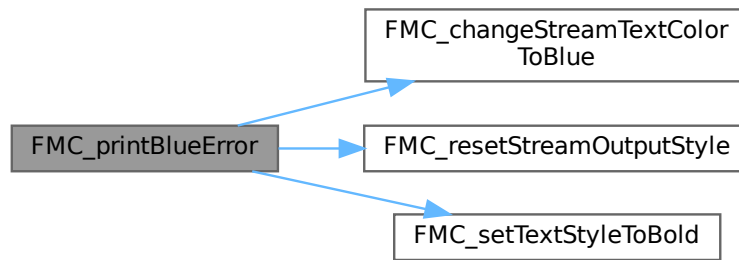
3.56.1.16 FMC_printBlueError()

```
void FMC_printBlueError (
    FILE * stream,
    const char * text )
```

Definition at line 341 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBlue\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



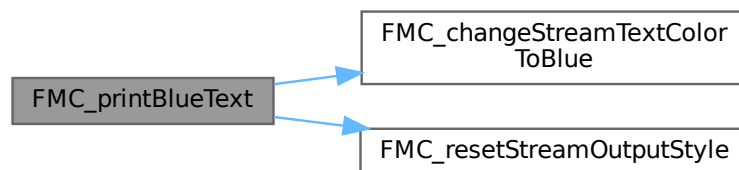
3.56.1.17 FMC_printBlueText()

```
void FMC_printBlueText (
    FILE * stream,
    const char * text )
```

Definition at line 240 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBlue\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



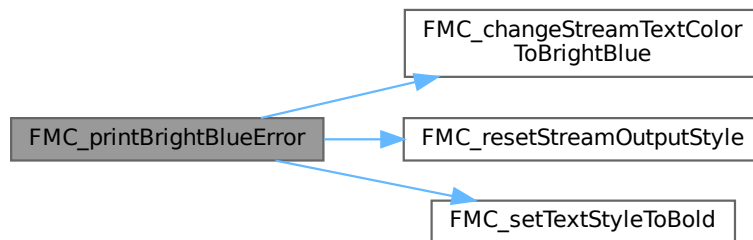
3.56.1.18 FMC_printBrightBlueError()

```
void FMC_printBrightBlueError (
    FILE * stream,
    const char * text )
```

Definition at line 397 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightBlue\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



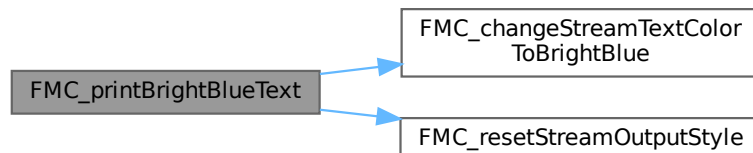
3.56.1.19 FMC_printBrightBlueText()

```
void FMC_printBrightBlueText (
    FILE * stream,
    const char * text )
```

Definition at line 289 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightBlue\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



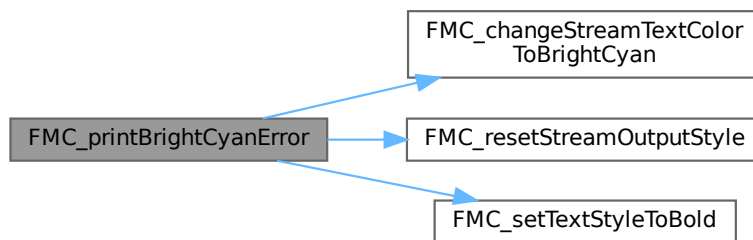
3.56.1.20 FMC_printBrightCyanError()

```
void FMC_printBrightCyanError (
    FILE * stream,
    const char * text )
```

Definition at line 413 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightCyan\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



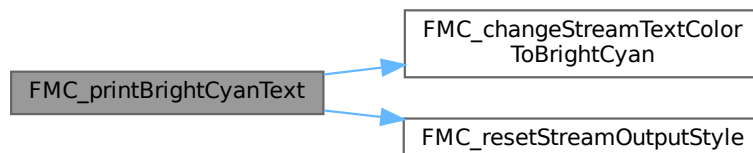
3.56.1.21 FMC_printBrightCyanText()

```
void FMC_printBrightCyanText (
    FILE * stream,
    const char * text )
```

Definition at line 303 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightCyan\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



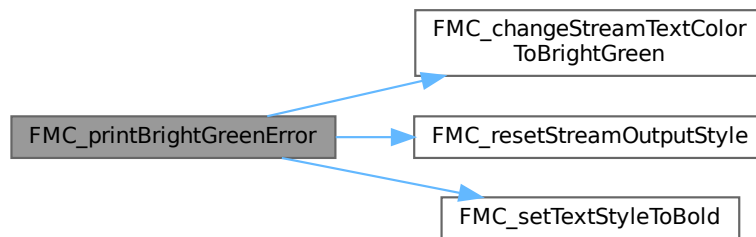
3.56.1.22 FMC_printBrightGreenError()

```
void FMC_printBrightGreenError (
    FILE * stream,
    const char * text )
```

Definition at line 381 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightGreen\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



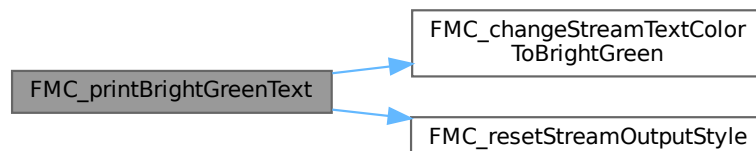
3.56.1.23 FMC_printBrightGreenText()

```
void FMC_printBrightGreenText (
    FILE * stream,
    const char * text )
```

Definition at line 275 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightGreen\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



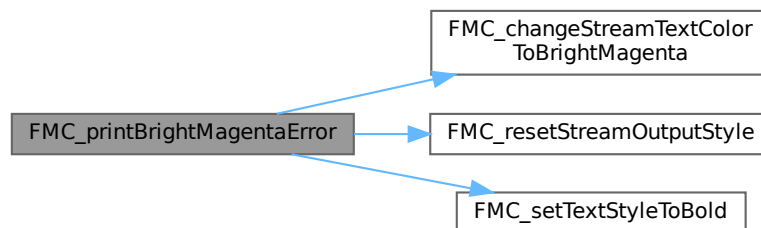
3.56.1.24 FMC_printBrightMagentaError()

```
void FMC_printBrightMagentaError (
    FILE * stream,
    const char * text )
```

Definition at line 405 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightMagenta\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



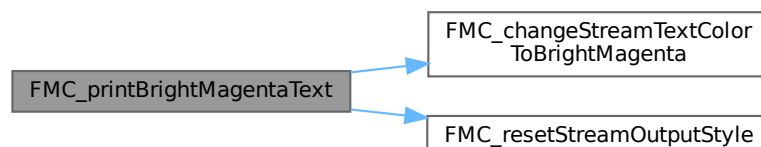
3.56.1.25 FMC_printBrightMagentaText()

```
void FMC_printBrightMagentaText (
    FILE * stream,
    const char * text )
```

Definition at line 296 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightMagenta\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



3.56.1.26 FMC_printBrightRedError()

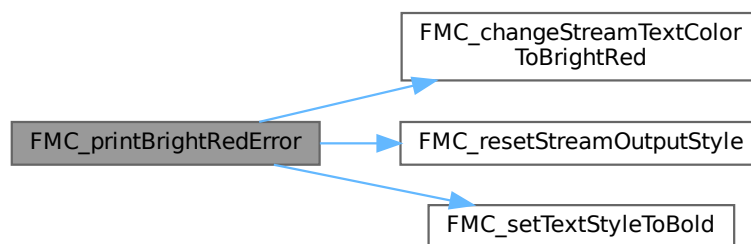
```
void FMC_printBrightRedError (
    FILE * stream,
    const char * text )
```

Definition at line 373 of file [FMC_errors.h](#).

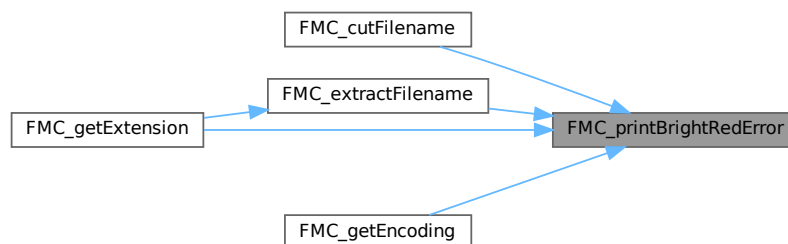
References [FMC_changeStreamTextColorToBrightRed\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Referenced by [FMC_cutFilename\(\)](#), [FMC_extractFilename\(\)](#), [FMC_getEncoding\(\)](#), and [FMC_getExtension\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:

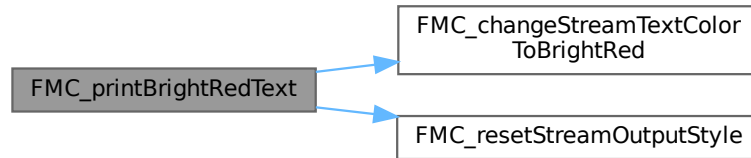
**3.56.1.27 FMC_printBrightRedText()**

```
void FMC_printBrightRedText (
    FILE * stream,
    const char * text )
```

Definition at line 268 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightRed\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



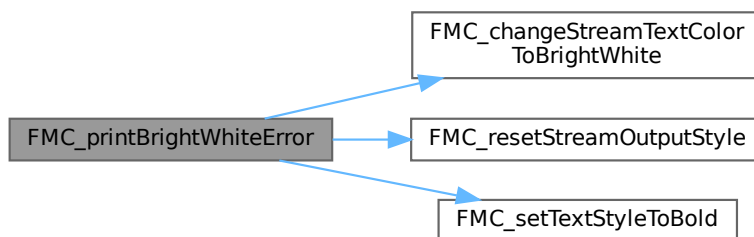
3.56.1.28 FMC_printBrightWhiteError()

```
void FMC_printBrightWhiteError (
    FILE * stream,
    const char * text )
```

Definition at line 421 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightWhite\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



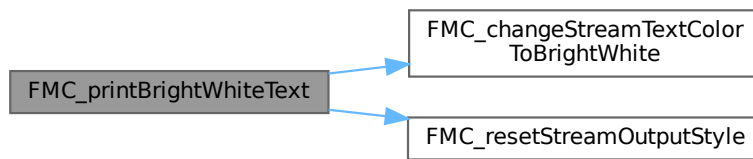
3.56.1.29 FMC_printBrightWhiteText()

```
void FMC_printBrightWhiteText (
    FILE * stream,
    const char * text )
```

Definition at line 310 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightWhite\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



3.56.1.30 FMC_printBrightYellowError()

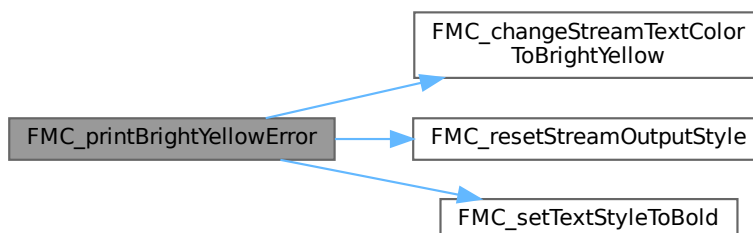
```
void FMC_printBrightYellowError (
    FILE * stream,
    const char * text )
```

Definition at line 389 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightYellow\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Referenced by [FMC_getEncoding\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



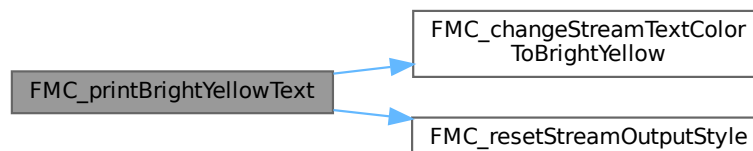
3.56.1.31 FMC_printBrightYellowText()

```
void FMC_printBrightYellowText (
    FILE * stream,
    const char * text )
```

Definition at line 282 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightYellow\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



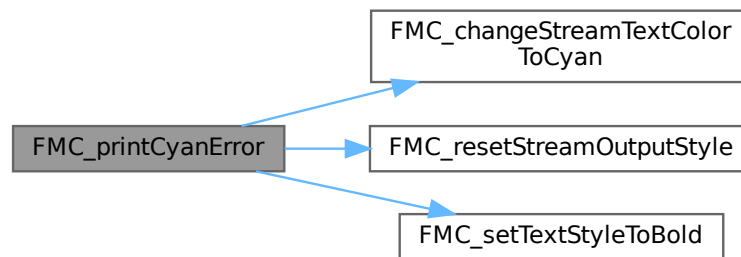
3.56.1.32 FMC_printCyanError()

```
void FMC_printCyanError (
    FILE * stream,
    const char * text )
```

Definition at line 357 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToCyan\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



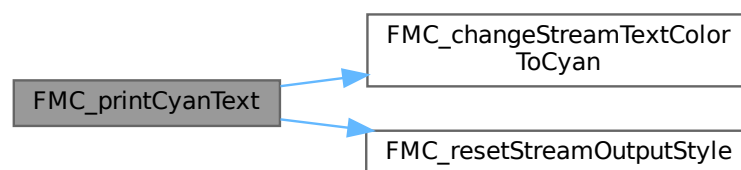
3.56.1.33 FMC_printCyanText()

```
void FMC_printCyanText (
    FILE * stream,
    const char * text )
```

Definition at line 254 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToCyan\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



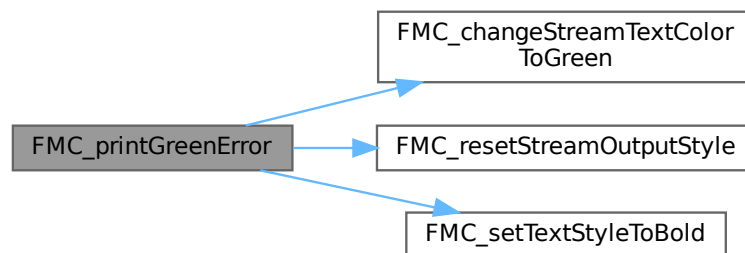
3.56.1.34 FMC_printGreenError()

```
void FMC_printGreenError (
    FILE * stream,
    const char * text )
```

Definition at line 325 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToGreen\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



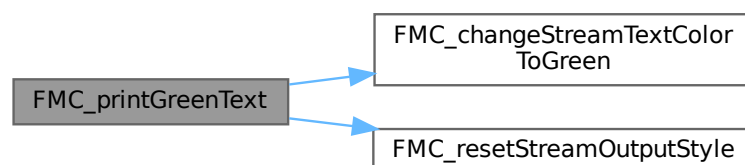
3.56.1.35 FMC_printGreenText()

```
void FMC_printGreenText (
    FILE * stream,
    const char * text )
```

Definition at line 226 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToGreen\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



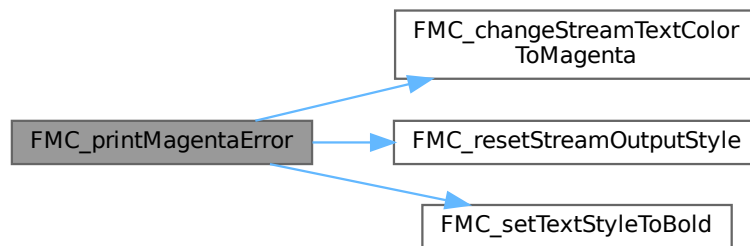
3.56.1.36 FMC_printMagentaError()

```
void FMC_printMagentaError (
    FILE * stream,
    const char * text )
```

Definition at line 349 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToMagenta\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



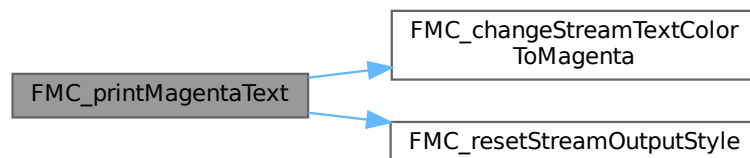
3.56.1.37 FMC_printMagentaText()

```
void FMC_printMagentaText (
    FILE * stream,
    const char * text )
```

Definition at line 247 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToMagenta\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



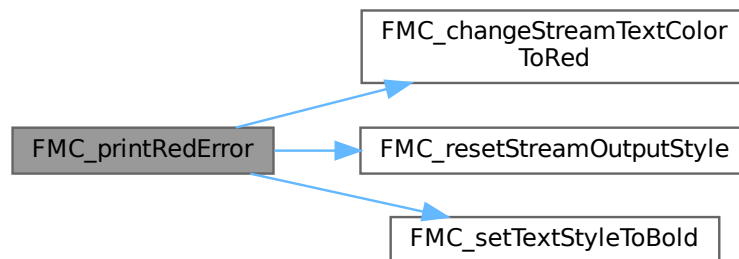
3.56.1.38 FMC_printRedError()

```
void FMC_printRedError (
    FILE * stream,
    const char * text )
```

Definition at line 317 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToRed\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



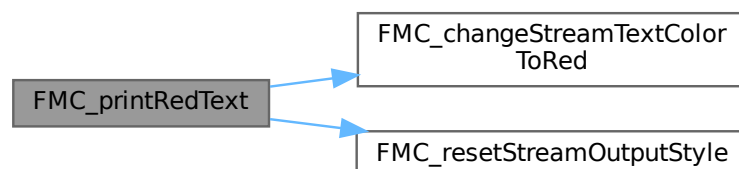
3.56.1.39 FMC_printRedText()

```
void FMC_printRedText (
    FILE * stream,
    const char * text )
```

Definition at line 219 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToRed\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



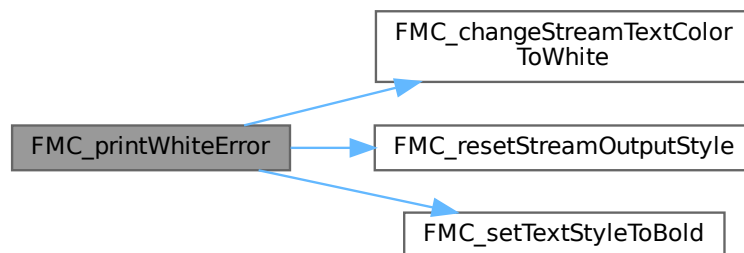
3.56.1.40 FMC_printWhiteError()

```
void FMC_printWhiteError (
    FILE * stream,
    const char * text )
```

Definition at line 365 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToWhite\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



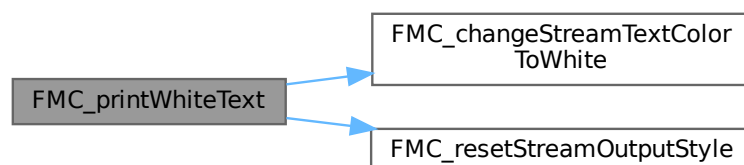
3.56.1.41 FMC_printWhiteText()

```
void FMC_printWhiteText (
    FILE * stream,
    const char * text )
```

Definition at line 261 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToWhite\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



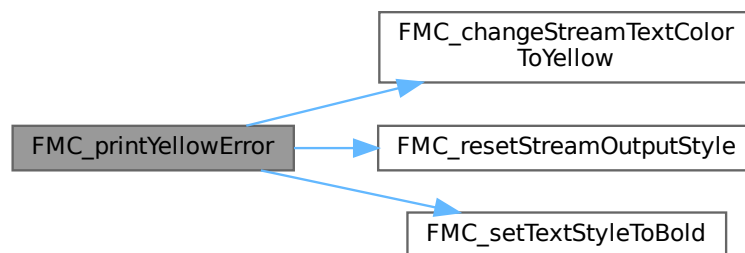
3.56.1.42 FMC_printYellowError()

```
void FMC_printYellowError (
    FILE * stream,
    const char * text )
```

Definition at line 333 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToYellow\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



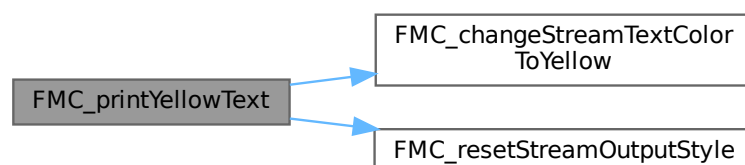
3.56.1.43 FMC_printYellowText()

```
void FMC_printYellowText (
    FILE * stream,
    const char * text )
```

Definition at line 233 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToYellow\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



3.56.1.44 FMC_resetStreamOutputStyle()

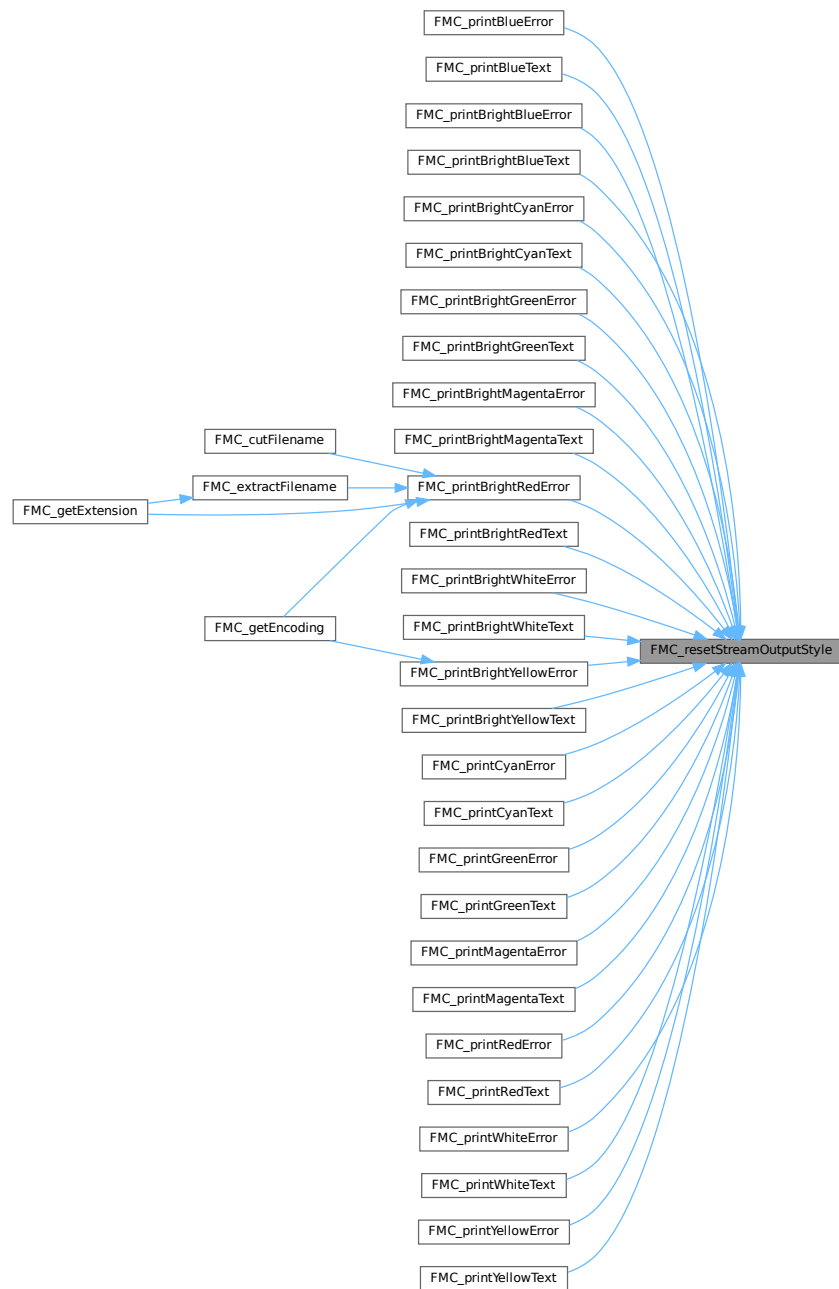
```
void FMC_resetStreamOutputStyle (
    FILE * stream )
```

Definition at line 42 of file [FMC_errors.h](#).

References [RESET](#).

Referenced by [FMC_printBlueError\(\)](#), [FMC_printBlueText\(\)](#), [FMC_printBrightBlueError\(\)](#), [FMC_printBrightBlueText\(\)](#), [FMC_printBrightCyanError\(\)](#), [FMC_printBrightCyanText\(\)](#), [FMC_printBrightGreenError\(\)](#), [FMC_printBrightGreenText\(\)](#), [FMC_printBrightMagentaError\(\)](#), [FMC_printBrightMagentaText\(\)](#), [FMC_printBrightRedError\(\)](#), [FMC_printBrightRedText\(\)](#), [FMC_printBrightWhiteError\(\)](#), [FMC_printBrightWhiteText\(\)](#), [FMC_printBrightYellowError\(\)](#), [FMC_printBrightYellowText\(\)](#), [FMC_printCyanError\(\)](#), [FMC_printCyanText\(\)](#), [FMC_printGreenError\(\)](#), [FMC_printGreenText\(\)](#), [FMC_printMagentaError\(\)](#), [FMC_printMagentaText\(\)](#), [FMC_printRedError\(\)](#), [FMC_printRedText\(\)](#), [FMC_printWhiteError\(\)](#), [FMC_printWhiteText\(\)](#), [FMC_printYellowError\(\)](#), and [FMC_printYellowText\(\)](#).

Here is the caller graph for this function:



3.57 FMC_errors.c

[Go to the documentation of this file.](#)

```

00001  /*
00002
00003  MIT License
00004
00005  Copyright (c) 2022 Axel PASCON
00006
00007  Permission is hereby granted, free of charge, to any person obtaining a copy
00008  of this software and associated documentation files (the "Software"), to deal

```



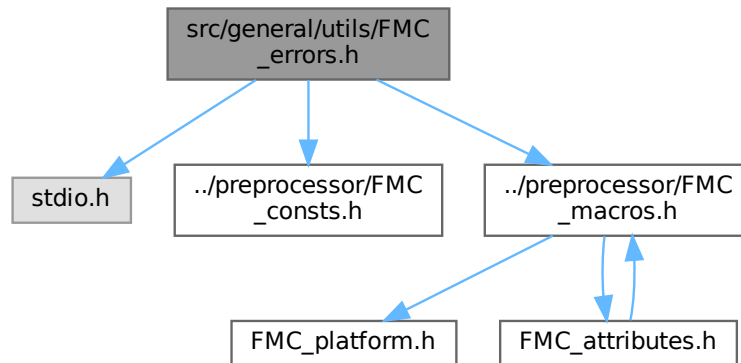
```

00009 in the Software without restriction, including without limitation the rights
00010 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011 copies of the Software, and to permit persons to whom the Software is
00012 furnished to do so, subject to the following conditions:
00013
00014 The above copyright notice and this permission notice shall be included in all
00015 copies or substantial portions of the Software.
00016
00017 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023 SOFTWARE.
00024
00025 */
00026
00027 #include "FMC_errors.h"
00028 #include "../preprocessor/FMC_macros.h"
00029 #include <stdio.h>
00030 #include <stdarg.h>
00031 #include <string.h>
00032
00033 FMC_SHARED void FMC_makeMsg_f(char *buff, unsigned int argc, ...)
00034 {
00035     va_list args;
00036     va_start(args, argc);
00037     for (unsigned int i = 0; i < argc; i++)
00038     {
00039         char *arg = va_arg(args, char *);
00040         buff = strcat(buff, arg);
00041     }
00042     va_end(args);
00043 }
00044
00045 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_resetStreamOutputStyle(FILE *stream);
00046 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToRed(FILE *stream);
00047 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToGreen(FILE *stream);
00048 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToYellow(FILE *stream);
00049 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToBlue(FILE *stream);
00050 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToMagenta(FILE *stream);
00051 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToCyan(FILE *stream);
00052 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToWhite(FILE *stream);
00053 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToBrightRed(FILE *stream);
00054 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToBrightGreen(FILE *stream);
00055 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToBrightYellow(FILE *stream);
00056 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToBrightBlue(FILE *stream);
00057 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToBrightMagenta(FILE *stream);
00058 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToBrightCyan(FILE *stream);
00059 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToBrightWhite(FILE *stream);
00060
00061 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printRedText(FILE *stream, const char *text);
00062 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printGreenText(FILE *stream, const char *text);
00063 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printYellowText(FILE *stream, const char *text);
00064 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBlueText(FILE *stream, const char *text);
00065 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printMagentaText(FILE *stream, const char *text);
00066 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printCyanText(FILE *stream, const char *text);
00067 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printWhiteText(FILE *stream, const char *text);
00068 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightRedText(FILE *stream, const char *text);
00069 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightGreenText(FILE *stream, const char *text);
00070 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightYellowText(FILE *stream, const char
    *text);
00071 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightBlueText(FILE *stream, const char *text);
00072 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightMagentaText(FILE *stream, const char
    *text);
00073 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightCyanText(FILE *stream, const char *text);
00074 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightWhiteText(FILE *stream, const char *text);
00075
00076 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printRedError(FILE *stream, const char *text);
00077 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printGreenError(FILE *stream, const char *text);
00078 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printYellowError(FILE *stream, const char *text);
00079 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBlueError(FILE *stream, const char *text);
00080 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printMagentaError(FILE *stream, const char *text);
00081 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printCyanError(FILE *stream, const char *text);
00082 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printWhiteError(FILE *stream, const char *text);
00083 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightRedError(FILE *stream, const char *text);
00084 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightGreenError(FILE *stream, const char
    *text);
00085 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightYellowError(FILE *stream, const char
    *text);
00086 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightBlueError(FILE *stream, const char *text);
00087 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightMagentaError(FILE *stream, const char
    *text);
00088 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightCyanError(FILE *stream, const char *text);
00089 extern FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightWhiteError(FILE *stream, const char
    *text);

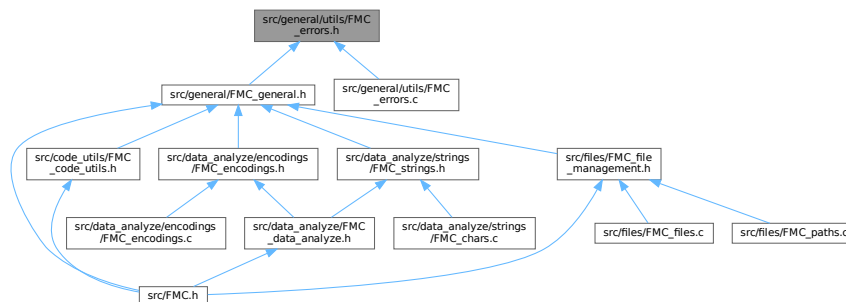
```

3.58 src/general/utis/FMC_errors.h File Reference

Include dependency graph for FMC_errors.h:



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



Macros

- `#define FMC_ERRORS`
- `#define FMC_makeMsg(err_var_name, argc, ...)`

Functions

- `void FMC_changeStreamTextColorToBlue (FILE *stream)`
- `void FMC_changeStreamTextColorToBrightBlue (FILE *stream)`
- `void FMC_changeStreamTextColorToBrightCyan (FILE *stream)`
- `void FMC_changeStreamTextColorToBrightGreen (FILE *stream)`
- `void FMC_changeStreamTextColorToBrightMagenta (FILE *stream)`
- `void FMC_changeStreamTextColorToBrightRed (FILE *stream)`
- `void FMC_changeStreamTextColorToBrightWhite (FILE *stream)`

- void [FMC_changeStreamTextColorToBrightYellow](#) (FILE *stream)
- void [FMC_changeStreamTextColorToCyan](#) (FILE *stream)
- void [FMC_changeStreamTextColorToGreen](#) (FILE *stream)
- void [FMC_changeStreamTextColorToMagenta](#) (FILE *stream)
- void [FMC_changeStreamTextColorToRed](#) (FILE *stream)
- void [FMC_changeStreamTextColorToWhite](#) (FILE *stream)
- void [FMC_changeStreamTextColorToYellow](#) (FILE *stream)
- void [FMC_makeMsg_f](#) (char *buff, unsigned int argc,...)
- void [FMC_printBlueError](#) (FILE *stream, const char *text)
- void [FMC_printBlueText](#) (FILE *stream, const char *text)
- void [FMC_printBrightBlueError](#) (FILE *stream, const char *text)
- void [FMC_printBrightBlueText](#) (FILE *stream, const char *text)
- void [FMC_printBrightCyanError](#) (FILE *stream, const char *text)
- void [FMC_printBrightCyanText](#) (FILE *stream, const char *text)
- void [FMC_printBrightGreenError](#) (FILE *stream, const char *text)
- void [FMC_printBrightGreenText](#) (FILE *stream, const char *text)
- void [FMC_printBrightMagentaError](#) (FILE *stream, const char *text)
- void [FMC_printBrightMagentaText](#) (FILE *stream, const char *text)
- void [FMC_printBrightRedError](#) (FILE *stream, const char *text)
- void [FMC_printBrightRedText](#) (FILE *stream, const char *text)
- void [FMC_printBrightWhiteError](#) (FILE *stream, const char *text)
- void [FMC_printBrightWhiteText](#) (FILE *stream, const char *text)
- void [FMC_printBrightYellowError](#) (FILE *stream, const char *text)
- void [FMC_printBrightYellowText](#) (FILE *stream, const char *text)
- void [FMC_printCyanError](#) (FILE *stream, const char *text)
- void [FMC_printCyanText](#) (FILE *stream, const char *text)
- void [FMC_printGreenError](#) (FILE *stream, const char *text)
- void [FMC_printGreenText](#) (FILE *stream, const char *text)
- void [FMC_printMagentaError](#) (FILE *stream, const char *text)
- void [FMC_printMagentaText](#) (FILE *stream, const char *text)
- void [FMC_printRedError](#) (FILE *stream, const char *text)
- void [FMC_printRedText](#) (FILE *stream, const char *text)
- void [FMC_printWhiteError](#) (FILE *stream, const char *text)
- void [FMC_printWhiteText](#) (FILE *stream, const char *text)
- void [FMC_printYellowError](#) (FILE *stream, const char *text)
- void [FMC_printYellowText](#) (FILE *stream, const char *text)
- void [FMC_resetStreamOutputStyle](#) (FILE *stream)
- void [FMC_setBGStreamColorToBlue](#) (FILE *stream)
- void [FMC_setBGStreamColorToBrightBlue](#) (FILE *stream)
- void [FMC_setBGStreamColorToBrightCyan](#) (FILE *stream)
- void [FMC_setBGStreamColorToBrightGreen](#) (FILE *stream)
- void [FMC_setBGStreamColorToBrightMagenta](#) (FILE *stream)
- void [FMC_setBGStreamColorToBrightRed](#) (FILE *stream)
- void [FMC_setBGStreamColorToBrightWhite](#) (FILE *stream)
- void [FMC_setBGStreamColorToBrightYellow](#) (FILE *stream)
- void [FMC_setBGStreamColorToCyan](#) (FILE *stream)
- void [FMC_setBGStreamColorToGreen](#) (FILE *stream)
- void [FMC_setBGStreamColorToMagenta](#) (FILE *stream)
- void [FMC_setBGStreamColorToRed](#) (FILE *stream)
- void [FMC_setBGStreamColorToWhite](#) (FILE *stream)
- void [FMC_setBGStreamColorToYellow](#) (FILE *stream)
- void [FMC_setTextStyleToBlink](#) (FILE *stream)
- void [FMC_setTextStyleToBold](#) (FILE *stream)
- void [FMC_setTextStyleToDim](#) (FILE *stream)
- void [FMC_setTextStyleToHidden](#) (FILE *stream)
- void [FMC_setTextStyleToReverse](#) (FILE *stream)
- void [FMC_setTextStyleToUnderlined](#) (FILE *stream)

3.58.1 Macro Definition Documentation

3.58.1.1 FMC_ERRORS

```
#define FMC_ERRORS
```

Definition at line 30 of file [FMC_errors.h](#).

3.58.1.2 FMC_makeMsg

```
#define FMC_makeMsg(  
    err_var_name,  
    argc,  
    ... )
```

Value:

```
char err_var_name[256] = {"\0"};  
FMC_makeMsg_f(err_var_name, argc, __VA_ARGS__) \
```

Definition at line 38 of file [FMC_errors.h](#).

3.58.2 Function Documentation

3.58.2.1 FMC_changeStreamTextColorToBlue()

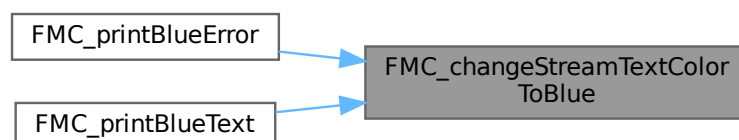
```
void FMC_changeStreamTextColorToBlue (  
    FILE * stream )
```

Definition at line 63 of file [FMC_errors.h](#).

References [FG_BLUE](#).

Referenced by [FMC_printBlueError\(\)](#), and [FMC_printBlueText\(\)](#).

Here is the caller graph for this function:



3.58.2.2 FMC_changeStreamTextColorToBrightBlue()

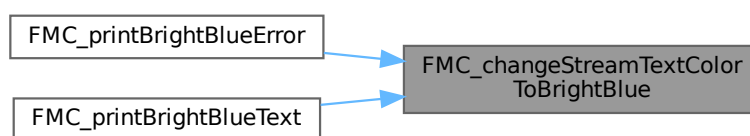
```
void FMC_changeStreamTextColorToBrightBlue (
    FILE * stream )
```

Definition at line 98 of file [FMC_errors.h](#).

References [FG_BRIGHT_BLUE](#).

Referenced by [FMC_printBrightBlueError\(\)](#), and [FMC_printBrightBlueText\(\)](#).

Here is the caller graph for this function:



3.58.2.3 FMC_changeStreamTextColorToBrightCyan()

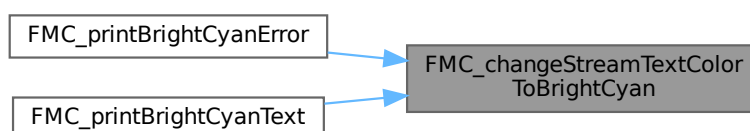
```
void FMC_changeStreamTextColorToBrightCyan (
    FILE * stream )
```

Definition at line 108 of file [FMC_errors.h](#).

References [FG_BRIGHT_CYAN](#).

Referenced by [FMC_printBrightCyanError\(\)](#), and [FMC_printBrightCyanText\(\)](#).

Here is the caller graph for this function:



3.58.2.4 FMC_changeStreamTextColorToBrightGreen()

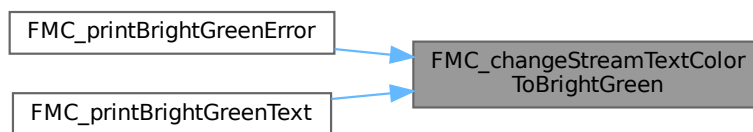
```
void FMC_changeStreamTextColorToBrightGreen (
    FILE * stream )
```

Definition at line 88 of file [FMC_errors.h](#).

References [FG_BRIGHT_GREEN](#).

Referenced by [FMC_printBrightGreenError\(\)](#), and [FMC_printBrightGreenText\(\)](#).

Here is the caller graph for this function:



3.58.2.5 FMC_changeStreamTextColorToBrightMagenta()

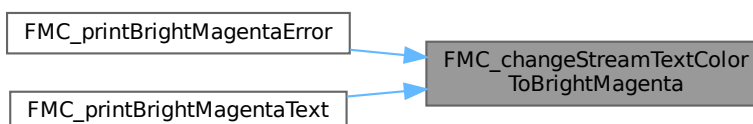
```
void FMC_changeStreamTextColorToBrightMagenta (
    FILE * stream )
```

Definition at line 103 of file [FMC_errors.h](#).

References [FG_BRIGHT_MAGENTA](#).

Referenced by [FMC_printBrightMagentaError\(\)](#), and [FMC_printBrightMagentaText\(\)](#).

Here is the caller graph for this function:



3.58.2.6 FMC_changeStreamTextColorToBrightRed()

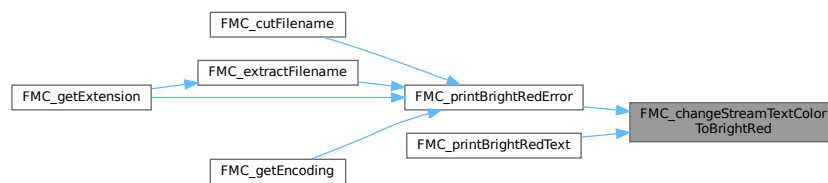
```
void FMC_changeStreamTextColorToBrightRed (
    FILE * stream )
```

Definition at line 83 of file [FMC_errors.h](#).

References [FG_BRIGHT_RED](#).

Referenced by [FMC_printBrightRedError\(\)](#), and [FMC_printBrightRedText\(\)](#).

Here is the caller graph for this function:



3.58.2.7 FMC_changeStreamTextColorToBrightWhite()

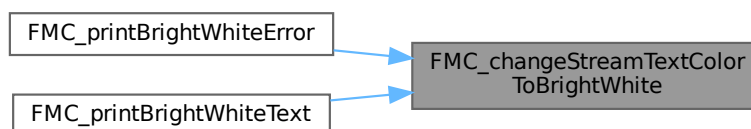
```
void FMC_changeStreamTextColorToBrightWhite (
    FILE * stream )
```

Definition at line 113 of file [FMC_errors.h](#).

References [FG_BRIGHT_WHITE](#).

Referenced by [FMC_printBrightWhiteError\(\)](#), and [FMC_printBrightWhiteText\(\)](#).

Here is the caller graph for this function:



3.58.2.8 FMC_changeStreamTextColorToBrightYellow()

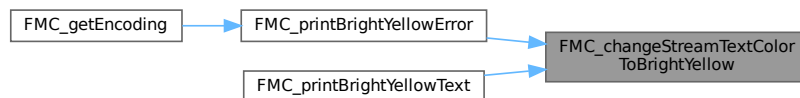
```
void FMC_changeStreamTextColorToBrightYellow (  
    FILE * stream )
```

Definition at line 93 of file [FMC_errors.h](#).

References [FG_BRIGHT_YELLOW](#).

Referenced by [FMC_printBrightYellowError\(\)](#), and [FMC_printBrightYellowText\(\)](#).

Here is the caller graph for this function:



3.58.2.9 FMC_changeStreamTextColorToCyan()

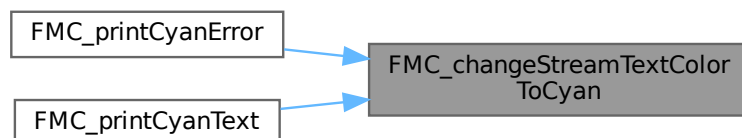
```
void FMC_changeStreamTextColorToCyan (  
    FILE * stream )
```

Definition at line 73 of file [FMC_errors.h](#).

References [FG_CYAN](#).

Referenced by [FMC_printCyanError\(\)](#), and [FMC_printCyanText\(\)](#).

Here is the caller graph for this function:



3.58.2.10 FMC_changeStreamTextColorToGreen()

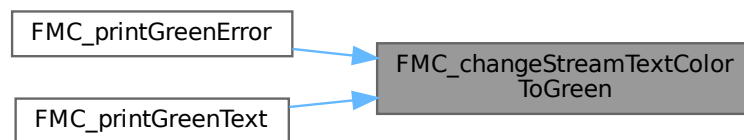
```
void FMC_changeStreamTextColorToGreen (
    FILE * stream )
```

Definition at line 53 of file [FMC_errors.h](#).

References [FG_GREEN](#).

Referenced by [FMC_printGreenError\(\)](#), and [FMC_printGreenText\(\)](#).

Here is the caller graph for this function:



3.58.2.11 FMC_changeStreamTextColorToMagenta()

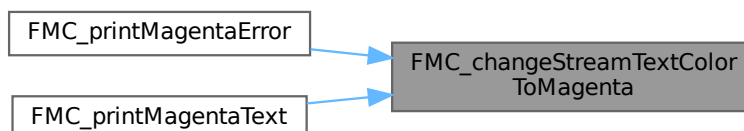
```
void FMC_changeStreamTextColorToMagenta (
    FILE * stream )
```

Definition at line 68 of file [FMC_errors.h](#).

References [FG_MAGENTA](#).

Referenced by [FMC_printMagentaError\(\)](#), and [FMC_printMagentaText\(\)](#).

Here is the caller graph for this function:



3.58.2.12 FMC_changeStreamTextColorToRed()

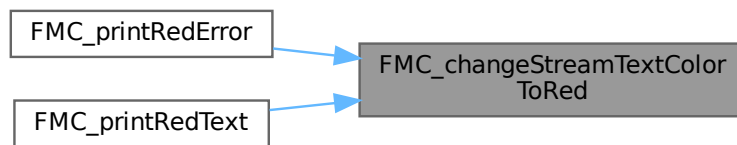
```
void FMC_changeStreamTextColorToRed (  
    FILE * stream )
```

Definition at line 48 of file [FMC_errors.h](#).

References [FG_RED](#).

Referenced by [FMC_printRedError\(\)](#), and [FMC_printRedText\(\)](#).

Here is the caller graph for this function:



3.58.2.13 FMC_changeStreamTextColorToWhite()

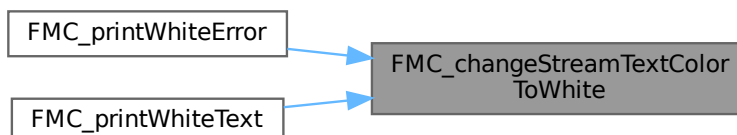
```
void FMC_changeStreamTextColorToWhite (  
    FILE * stream )
```

Definition at line 78 of file [FMC_errors.h](#).

References [FG_WHITE](#).

Referenced by [FMC_printWhiteError\(\)](#), and [FMC_printWhiteText\(\)](#).

Here is the caller graph for this function:



3.58.2.14 FMC_changeStreamTextColorToYellow()

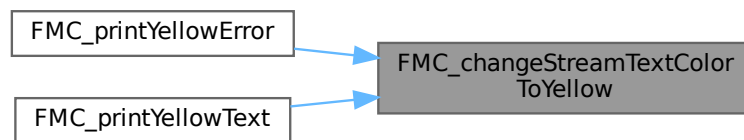
```
void FMC_changeStreamTextColorToYellow (
    FILE * stream )
```

Definition at line 58 of file [FMC_errors.h](#).

References [FG_YELLOW](#).

Referenced by [FMC_printYellowError\(\)](#), and [FMC_printYellowText\(\)](#).

Here is the caller graph for this function:



3.58.2.15 FMC_makeMsg_f()

```
void FMC_makeMsg_f (
    char * buff,
    unsigned int argc,
    ... )
```

Definition at line 33 of file [FMC_errors.c](#).

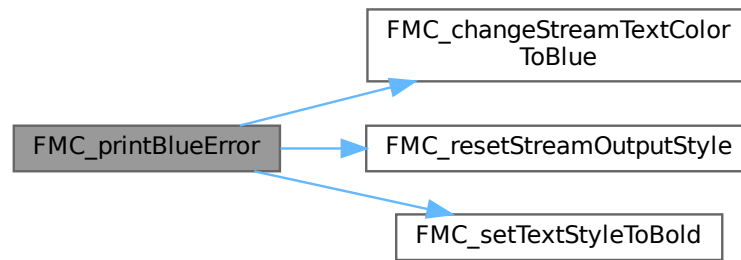
3.58.2.16 FMC_printBlueError()

```
void FMC_printBlueError (
    FILE * stream,
    const char * text )
```

Definition at line 341 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBlue\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



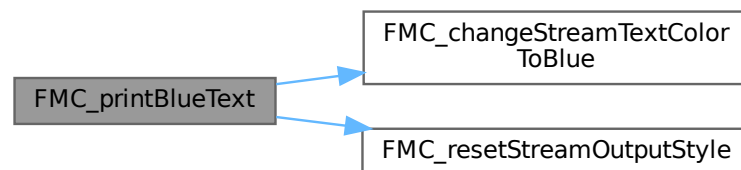
3.58.2.17 FMC_printBlueText()

```
void FMC_printBlueText (
    FILE * stream,
    const char * text )
```

Definition at line 240 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBlue\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



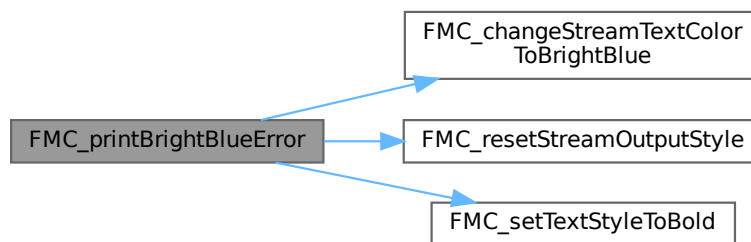
3.58.2.18 FMC_printBrightBlueError()

```
void FMC_printBrightBlueError (
    FILE * stream,
    const char * text )
```

Definition at line 397 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightBlue\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



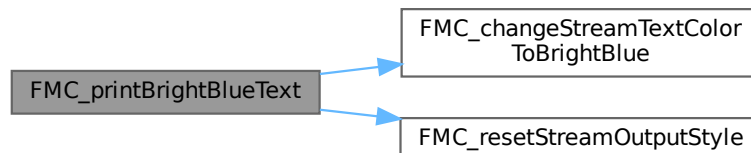
3.58.2.19 FMC_printBrightBlueText()

```
void FMC_printBrightBlueText (
    FILE * stream,
    const char * text )
```

Definition at line 289 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightBlue\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



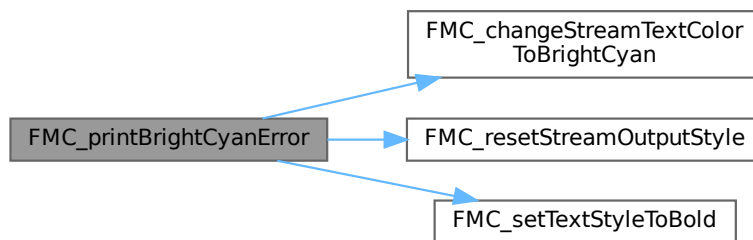
3.58.2.20 FMC_printBrightCyanError()

```
void FMC_printBrightCyanError (
    FILE * stream,
    const char * text )
```

Definition at line 413 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightCyan\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



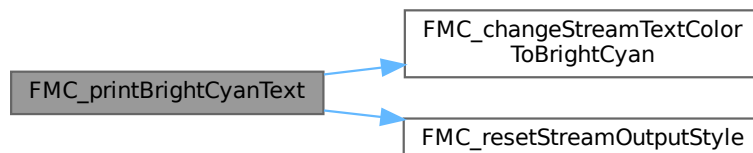
3.58.2.21 FMC_printBrightCyanText()

```
void FMC_printBrightCyanText (
    FILE * stream,
    const char * text )
```

Definition at line 303 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightCyan\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



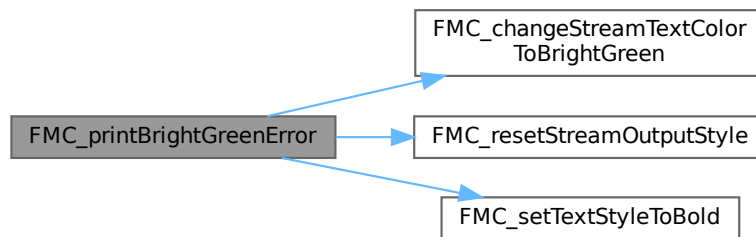
3.58.2.22 FMC_printBrightGreenError()

```
void FMC_printBrightGreenError (
    FILE * stream,
    const char * text )
```

Definition at line 381 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightGreen\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



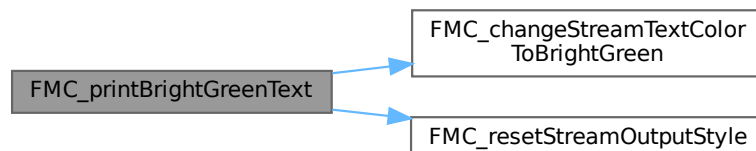
3.58.2.23 FMC_printBrightGreenText()

```
void FMC_printBrightGreenText (
    FILE * stream,
    const char * text )
```

Definition at line 275 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightGreen\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



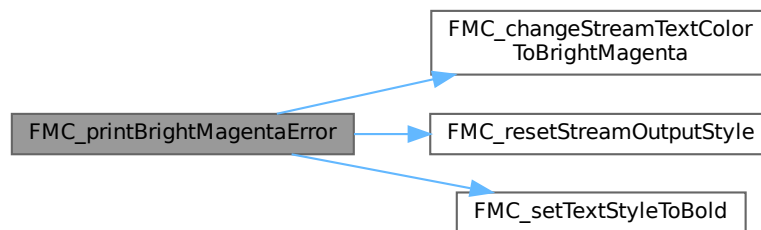
3.58.2.24 FMC_printBrightMagentaError()

```
void FMC_printBrightMagentaError (
    FILE * stream,
    const char * text )
```

Definition at line 405 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightMagenta\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



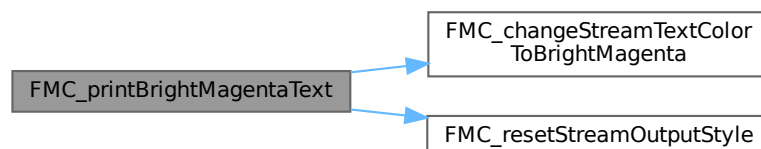
3.58.2.25 FMC_printBrightMagentaText()

```
void FMC_printBrightMagentaText (
    FILE * stream,
    const char * text )
```

Definition at line 296 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightMagenta\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



3.58.2.26 FMC_printBrightRedError()

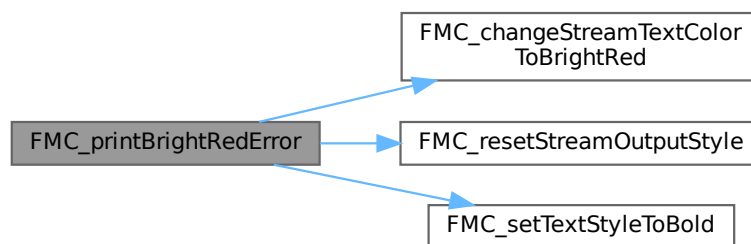
```
void FMC_printBrightRedError (
    FILE * stream,
    const char * text )
```

Definition at line 373 of file [FMC_errors.h](#).

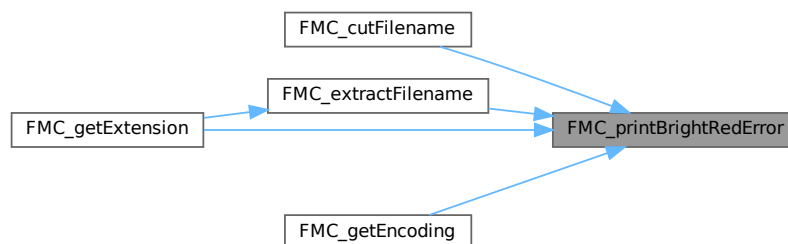
References [FMC_changeStreamTextColorToBrightRed\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Referenced by [FMC_cutFilename\(\)](#), [FMC_extractFilename\(\)](#), [FMC_getEncoding\(\)](#), and [FMC_getExtension\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



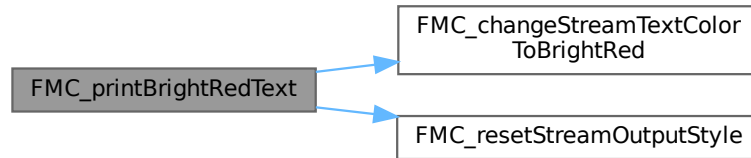
3.58.2.27 FMC_printBrightRedText()

```
void FMC_printBrightRedText (
    FILE * stream,
    const char * text )
```

Definition at line 268 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightRed\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



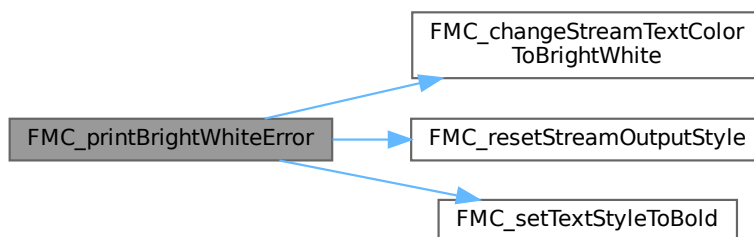
3.58.2.28 FMC_printBrightWhiteError()

```
void FMC_printBrightWhiteError (
    FILE * stream,
    const char * text )
```

Definition at line 421 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightWhite\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



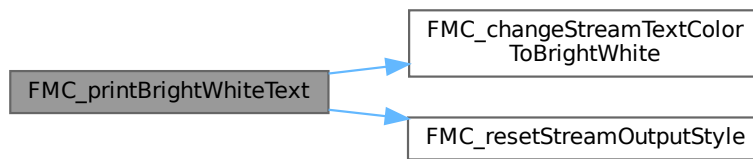
3.58.2.29 FMC_printBrightWhiteText()

```
void FMC_printBrightWhiteText (
    FILE * stream,
    const char * text )
```

Definition at line 310 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightWhite\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



3.58.2.30 FMC_printBrightYellowError()

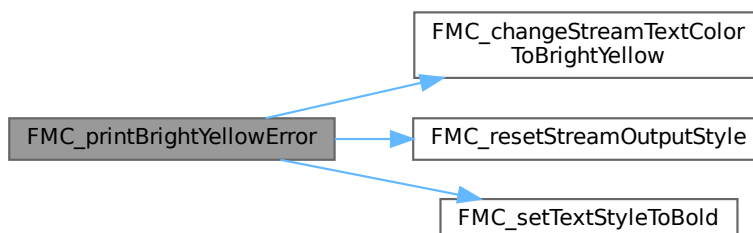
```
void FMC_printBrightYellowError (
    FILE * stream,
    const char * text )
```

Definition at line 389 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightYellow\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Referenced by [FMC_getEncoding\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



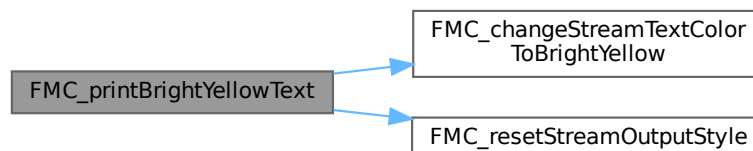
3.58.2.31 FMC_printBrightYellowText()

```
void FMC_printBrightYellowText (
    FILE * stream,
    const char * text )
```

Definition at line 282 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToBrightYellow\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



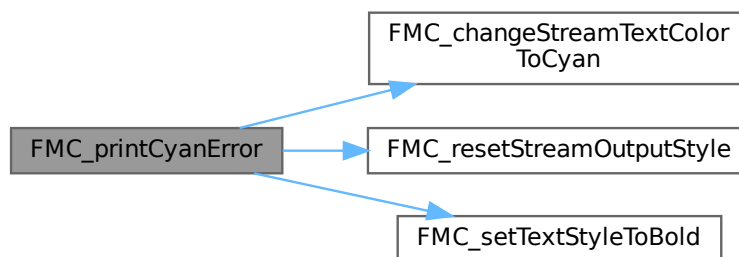
3.58.2.32 FMC_printCyanError()

```
void FMC_printCyanError (
    FILE * stream,
    const char * text )
```

Definition at line 357 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToCyan\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



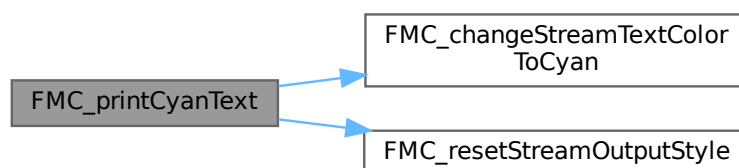
3.58.2.33 FMC_printCyanText()

```
void FMC_printCyanText (
    FILE * stream,
    const char * text )
```

Definition at line 254 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToCyan\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



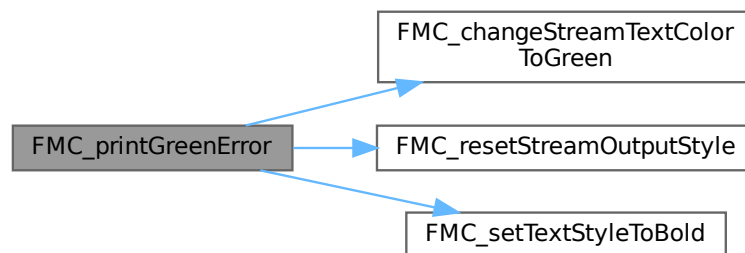
3.58.2.34 FMC_printGreenError()

```
void FMC_printGreenError (
    FILE * stream,
    const char * text )
```

Definition at line 325 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToGreen\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



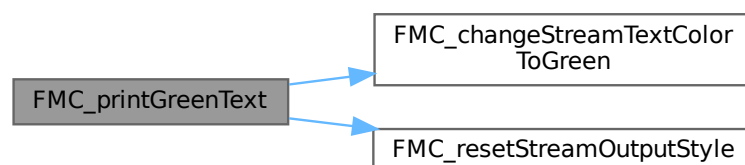
3.58.2.35 FMC_printGreenText()

```
void FMC_printGreenText (
    FILE * stream,
    const char * text )
```

Definition at line 226 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToGreen\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



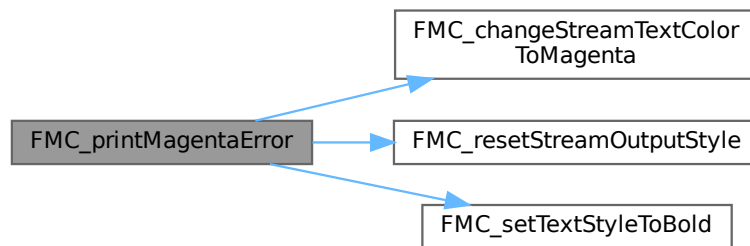
3.58.2.36 FMC_printMagentaError()

```
void FMC_printMagentaError (
    FILE * stream,
    const char * text )
```

Definition at line 349 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToMagenta\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



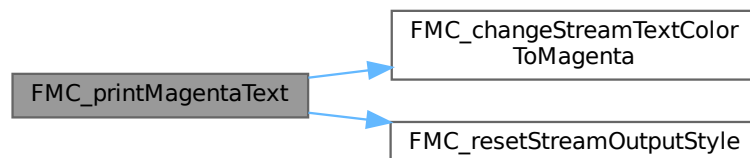
3.58.2.37 FMC_printMagentaText()

```
void FMC_printMagentaText (
    FILE * stream,
    const char * text )
```

Definition at line 247 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToMagenta\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



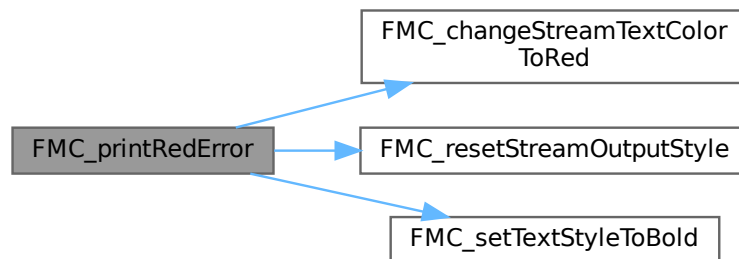
3.58.2.38 FMC_printRedError()

```
void FMC_printRedError (
    FILE * stream,
    const char * text )
```

Definition at line 317 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToRed\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



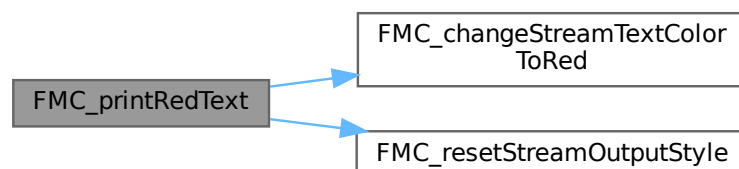
3.58.2.39 FMC_printRedText()

```
void FMC_printRedText (
    FILE * stream,
    const char * text )
```

Definition at line 219 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToRed\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



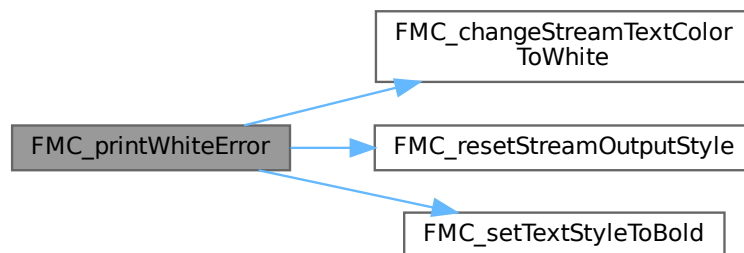
3.58.2.40 FMC_printWhiteError()

```
void FMC_printWhiteError (
    FILE * stream,
    const char * text )
```

Definition at line 365 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToWhite\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



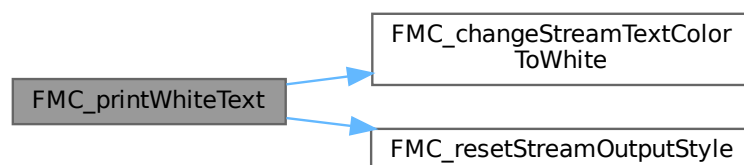
3.58.2.41 FMC_printWhiteText()

```
void FMC_printWhiteText (
    FILE * stream,
    const char * text )
```

Definition at line 261 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToWhite\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



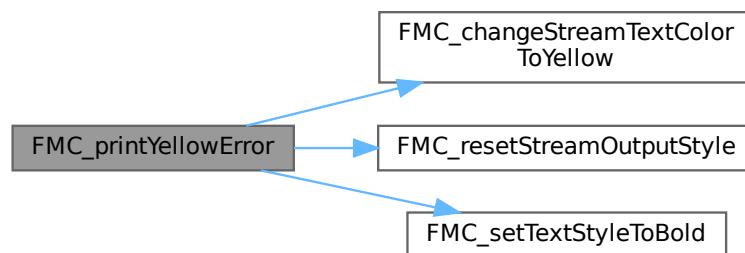
3.58.2.42 FMC_printYellowError()

```
void FMC_printYellowError (
    FILE * stream,
    const char * text )
```

Definition at line 333 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToYellow\(\)](#), [FMC_resetStreamOutputStyle\(\)](#), and [FMC_setTextStyleToBold\(\)](#).

Here is the call graph for this function:



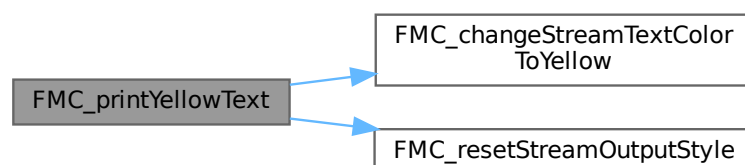
3.58.2.43 FMC_printYellowText()

```
void FMC_printYellowText (
    FILE * stream,
    const char * text )
```

Definition at line 233 of file [FMC_errors.h](#).

References [FMC_changeStreamTextColorToYellow\(\)](#), and [FMC_resetStreamOutputStyle\(\)](#).

Here is the call graph for this function:



3.58.2.44 FMC_resetStreamOutputStyle()

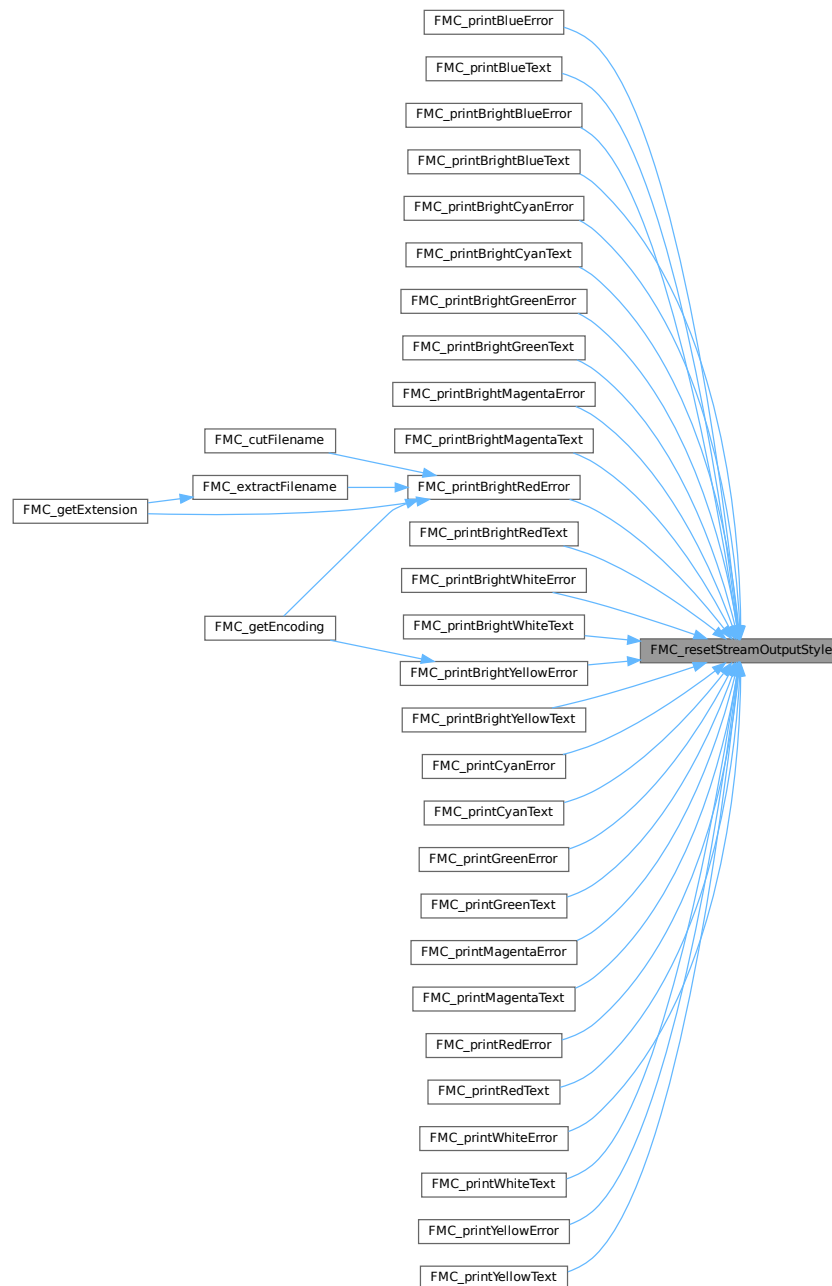
```
void FMC_resetStreamOutputStyle (
    FILE * stream )
```

Definition at line 42 of file [FMC_errors.h](#).

References [RESET](#).

Referenced by [FMC_printBlueError\(\)](#), [FMC_printBlueText\(\)](#), [FMC_printBrightBlueError\(\)](#), [FMC_printBrightBlueText\(\)](#), [FMC_printBrightCyanError\(\)](#), [FMC_printBrightCyanText\(\)](#), [FMC_printBrightGreenError\(\)](#), [FMC_printBrightGreenText\(\)](#), [FMC_printBrightMagentaError\(\)](#), [FMC_printBrightMagentaText\(\)](#), [FMC_printBrightRedError\(\)](#), [FMC_printBrightRedText\(\)](#), [FMC_printBrightWhiteError\(\)](#), [FMC_printBrightWhiteText\(\)](#), [FMC_printBrightYellowError\(\)](#), [FMC_printBrightYellowText\(\)](#), [FMC_printCyanError\(\)](#), [FMC_printCyanText\(\)](#), [FMC_printGreenError\(\)](#), [FMC_printGreenText\(\)](#), [FMC_printMagentaError\(\)](#), [FMC_printMagentaText\(\)](#), [FMC_printRedError\(\)](#), [FMC_printRedText\(\)](#), [FMC_printWhiteError\(\)](#), [FMC_printWhiteText\(\)](#), [FMC_printYellowError\(\)](#), and [FMC_printYellowText\(\)](#).

Here is the caller graph for this function:



3.58.2.45 FMC_setBGStreamColorToBlue()

```
void FMC_setBGStreamColorToBlue (
    FILE * stream )
```

Definition at line 133 of file [FMC_errors.h](#).

References [BG_BLUE](#).

3.58.2.46 FMC_setBGStreamColorToBrightBlue()

```
void FMC_setBGStreamColorToBrightBlue (
    FILE * stream )
```

Definition at line 168 of file [FMC_errors.h](#).

References [BG_BRIGHT_BLUE](#).

3.58.2.47 FMC_setBGStreamColorToBrightCyan()

```
void FMC_setBGStreamColorToBrightCyan (
    FILE * stream )
```

Definition at line 178 of file [FMC_errors.h](#).

References [BG_BRIGHT_CYAN](#).

3.58.2.48 FMC_setBGStreamColorToBrightGreen()

```
void FMC_setBGStreamColorToBrightGreen (
    FILE * stream )
```

Definition at line 158 of file [FMC_errors.h](#).

References [BG_BRIGHT_GREEN](#).

3.58.2.49 FMC_setBGStreamColorToBrightMagenta()

```
void FMC_setBGStreamColorToBrightMagenta (
    FILE * stream )
```

Definition at line 173 of file [FMC_errors.h](#).

References [BG_BRIGHT_MAGENTA](#).

3.58.2.50 FMC_setBGStreamColorToBrightRed()

```
void FMC_setBGStreamColorToBrightRed (
    FILE * stream )
```

Definition at line 153 of file [FMC_errors.h](#).

References [BG_BRIGHT_RED](#).

3.58.2.51 FMC_setBGStreamColorToBrightWhite()

```
void FMC_setBGStreamColorToBrightWhite (
    FILE * stream )
```

Definition at line 183 of file [FMC_errors.h](#).

References [BG_BRIGHT_WHITE](#).

3.58.2.52 FMC_setBGStreamColorToBrightYellow()

```
void FMC_setBGStreamColorToBrightYellow (
    FILE * stream )
```

Definition at line 163 of file [FMC_errors.h](#).

References [BG_BRIGHT_YELLOW](#).

3.58.2.53 FMC_setBGStreamColorToCyan()

```
void FMC_setBGStreamColorToCyan (
    FILE * stream )
```

Definition at line 143 of file [FMC_errors.h](#).

References [BG_CYAN](#).

3.58.2.54 FMC_setBGStreamColorToGreen()

```
void FMC_setBGStreamColorToGreen (
    FILE * stream )
```

Definition at line 123 of file [FMC_errors.h](#).

References [BG_GREEN](#).

3.58.2.55 FMC_setBGStreamColorToMagenta()

```
void FMC_setBGStreamColorToMagenta (
    FILE * stream )
```

Definition at line 138 of file [FMC_errors.h](#).

References [BG_MAGENTA](#).

3.58.2.56 FMC_setBGStreamColorToRed()

```
void FMC_setBGStreamColorToRed (
    FILE * stream )
```

Definition at line 118 of file [FMC_errors.h](#).

References [BG_RED](#).

3.58.2.57 FMC_setBGStreamColorToWhite()

```
void FMC_setBGStreamColorToWhite (
    FILE * stream )
```

Definition at line 148 of file [FMC_errors.h](#).

References [BG_WHITE](#).

3.58.2.58 FMC_setBGStreamColorToYellow()

```
void FMC_setBGStreamColorToYellow (
    FILE * stream )
```

Definition at line 128 of file [FMC_errors.h](#).

References [BG_YELLOW](#).

3.58.2.59 FMC_setTextStyleToBlink()

```
void FMC_setTextStyleToBlink (
    FILE * stream )
```

Definition at line 203 of file [FMC_errors.h](#).

References [TXT_BLINK](#).

3.58.2.60 FMC_setTextStyleToBold()

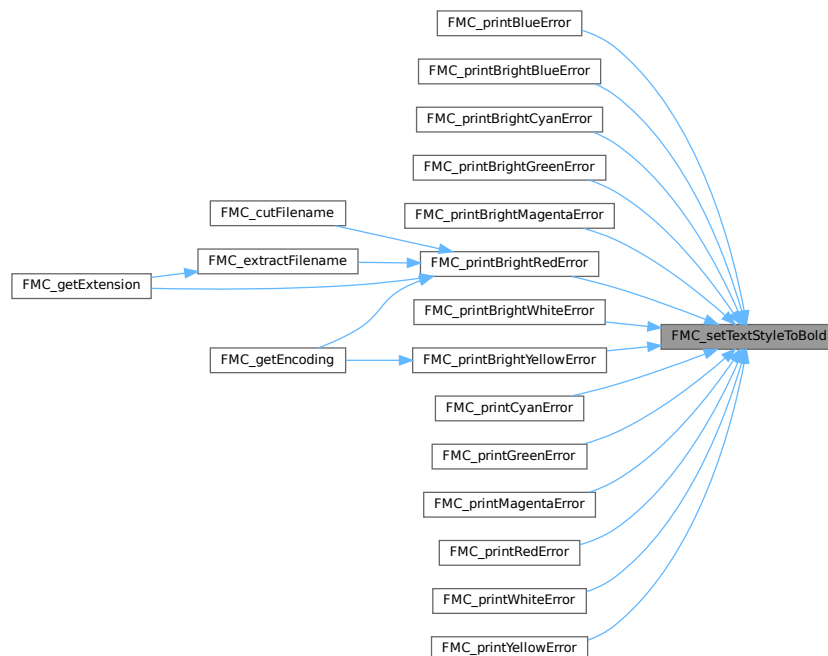
```
void FMC_setTextStyleToBold (
    FILE * stream )
```

Definition at line 188 of file [FMC_errors.h](#).

References [TXT_BOLD](#).

Referenced by [FMC_printBlueError\(\)](#), [FMC_printBrightBlueError\(\)](#), [FMC_printBrightCyanError\(\)](#), [FMC_printBrightGreenError\(\)](#), [FMC_printBrightMagentaError\(\)](#), [FMC_printBrightRedError\(\)](#), [FMC_printBrightWhiteError\(\)](#), [FMC_printBrightYellowError\(\)](#), [FMC_printCyanError\(\)](#), [FMC_printGreenError\(\)](#), [FMC_printMagentaError\(\)](#), [FMC_printRedError\(\)](#), [FMC_printWhiteError\(\)](#), and [FMC_printYellowError\(\)](#).

Here is the caller graph for this function:



3.58.2.61 FMC_setTextStyleToDim()

```
void FMC_setTextStyleToDim (
    FILE * stream )
```

Definition at line 193 of file [FMC_errors.h](#).

References [TXT_DIM](#).

3.58.2.62 FMC_setTextStyleToHidden()

```
void FMC_setTextStyleToHidden (
    FILE * stream )
```

Definition at line 213 of file [FMC_errors.h](#).

References [TXT_HIDDEN](#).

3.58.2.63 FMC_setTextStyleToReverse()

```
void FMC_setTextStyleToReverse (
    FILE * stream )
```

Definition at line 208 of file [FMC_errors.h](#).

References [TXT_REVERSE](#).

3.58.2.64 FMC_setTextStyleToUnderlined()

```
void FMC_setTextStyleToUnderlined (
    FILE * stream )
```

Definition at line 198 of file [FMC_errors.h](#).

References [TXT_UNDERLINED](#).

3.59 FMC_errors.h

[Go to the documentation of this file.](#)

```
00001 /*
00002
00003 MIT License
00004
00005 Copyright (c) 2022 Axel PASCON
00006
00007 Permission is hereby granted, free of charge, to any person obtaining a copy
00008 of this software and associated documentation files (the "Software"), to deal
00009 in the Software without restriction, including without limitation the rights
00010 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011 copies of the Software, and to permit persons to whom the Software is
00012 furnished to do so, subject to the following conditions:
00013
00014 The above copyright notice and this permission notice shall be included in all
00015 copies or substantial portions of the Software.
00016
00017 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023 SOFTWARE.
00024
00025 */
00026
00027 #pragma once
```

```

00028
00029 #ifndef FMC_ERRORS
00030 #define FMC_ERRORS
00031
00032 #include <stdio.h>
00033 #include "../preprocessor/FMC_consts.h"
00034 #include "../preprocessor/FMC_macros.h"
00035
00036 FMC_SHARED FMC_FUNC_NONNULL(1) void FMC_makeMsg_f(char *buff, unsigned int argc, ...);
00037
00038 #define FMC_makeMsg(err_var_name, argc, ...) \
00039     char err_var_name[256] = {"\0"}; \
00040     FMC_makeMsg_f(err_var_name, argc, __VA_ARGS__)
00041
00042 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_resetStreamOutputStyle(FILE *stream)
00043 {
00044     fprintf(stream, RESET);
00045 }
00046
00047
00048 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToRed(FILE *stream)
00049 {
00050     fprintf(stream, FG_RED);
00051 }
00052
00053 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToGreen(FILE *stream)
00054 {
00055     fprintf(stream, FG_GREEN);
00056 }
00057
00058 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToYellow(FILE *stream)
00059 {
00060     fprintf(stream, FG_YELLOW);
00061 }
00062
00063 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToBlue(FILE *stream)
00064 {
00065     fprintf(stream, FG_BLUE);
00066 }
00067
00068 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToMagenta(FILE *stream)
00069 {
00070     fprintf(stream, FG_MAGENTA);
00071 }
00072
00073 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToCyan(FILE *stream)
00074 {
00075     fprintf(stream, FG_CYAN);
00076 }
00077
00078 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToWhite(FILE *stream)
00079 {
00080     fprintf(stream, FG_WHITE);
00081 }
00082
00083 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToBrightRed(FILE *stream)
00084 {
00085     fprintf(stream, FG_BRIGHT_RED);
00086 }
00087
00088 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToBrightGreen(FILE *stream)
00089 {
00090     fprintf(stream, FG_BRIGHT_GREEN);
00091 }
00092
00093 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToBrightYellow(FILE *stream)
00094 {
00095     fprintf(stream, FG_BRIGHT_YELLOW);
00096 }
00097
00098 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToBrightBlue(FILE *stream)
00099 {
00100     fprintf(stream, FG_BRIGHT_BLUE);
00101 }
00102
00103 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToBrightMagenta(FILE *stream)
00104 {
00105     fprintf(stream, FG_BRIGHT_MAGENTA);
00106 }
00107
00108 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToBrightCyan(FILE *stream)
00109 {
00110     fprintf(stream, FG_BRIGHT_CYAN);
00111 }
00112
00113 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_changeStreamTextColorToBrightWhite(FILE *stream)
00114 {

```

```
00115     fprintf(stream, FG_BRIGHT_WHITE);
00116 }
00117
00118 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setBGStreamColorToRed(FILE *stream)
00119 {
00120     fprintf(stream, BG_RED);
00121 }
00122
00123 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setBGStreamColorToGreen(FILE *stream)
00124 {
00125     fprintf(stream, BG_GREEN);
00126 }
00127
00128 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setBGStreamColorToYellow(FILE *stream)
00129 {
00130     fprintf(stream, BG_YELLOW);
00131 }
00132
00133 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setBGStreamColorToBlue(FILE *stream)
00134 {
00135     fprintf(stream, BG_BLUE);
00136 }
00137
00138 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setBGStreamColorToMagenta(FILE *stream)
00139 {
00140     fprintf(stream, BG_MAGENTA);
00141 }
00142
00143 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setBGStreamColorToCyan(FILE *stream)
00144 {
00145     fprintf(stream, BG_CYAN);
00146 }
00147
00148 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setBGStreamColorToWhite(FILE *stream)
00149 {
00150     fprintf(stream, BG_WHITE);
00151 }
00152
00153 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setBGStreamColorToBrightRed(FILE *stream)
00154 {
00155     fprintf(stream, BG_BRIGHT_RED);
00156 }
00157
00158 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setBGStreamColorToBrightGreen(FILE *stream)
00159 {
00160     fprintf(stream, BG_BRIGHT_GREEN);
00161 }
00162
00163 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setBGStreamColorToBrightYellow(FILE *stream)
00164 {
00165     fprintf(stream, BG_BRIGHT_YELLOW);
00166 }
00167
00168 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setBGStreamColorToBrightBlue(FILE *stream)
00169 {
00170     fprintf(stream, BG_BRIGHT_BLUE);
00171 }
00172
00173 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setBGStreamColorToBrightMagenta(FILE *stream)
00174 {
00175     fprintf(stream, BG_BRIGHT_MAGENTA);
00176 }
00177
00178 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setBGStreamColorToBrightCyan(FILE *stream)
00179 {
00180     fprintf(stream, BG_BRIGHT_CYAN);
00181 }
00182
00183 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setBGStreamColorToBrightWhite(FILE *stream)
00184 {
00185     fprintf(stream, BG_BRIGHT_WHITE);
00186 }
00187
00188 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setTextStyleToBold(FILE *stream)
00189 {
00190     fprintf(stream, TXT_BOLD);
00191 }
00192
00193 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setTextStyleToDim(FILE *stream)
00194 {
00195     fprintf(stream, TXT_DIM);
00196 }
00197
00198 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setTextStyleToUnderlined(FILE *stream)
00199 {
00200     fprintf(stream, TXT_UNDERLINED);
00201 }
```

```
00202
00203 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setTextStyleToBlink(FILE *stream)
00204 {
00205     fprintf(stream, TXT_BLINK);
00206 }
00207
00208 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setTextStyleToReverse(FILE *stream)
00209 {
00210     fprintf(stream, TXT_REVERSE);
00211 }
00212
00213 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_setTextStyleToHidden(FILE *stream)
00214 {
00215     fprintf(stream, TXT_HIDDEN);
00216 }
00217
00218
00219 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printRedText(FILE *stream, const char *text)
00220 {
00221     FMC_changeStreamTextColorToRed(stream);
00222     fprintf(stream, "%s\n", text);
00223     FMC_resetStreamOutputStyle(stream);
00224 }
00225
00226 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printGreenText(FILE *stream, const char *text)
00227 {
00228     FMC_changeStreamTextColorToGreen(stream);
00229     fprintf(stream, "%s\n", text);
00230     FMC_resetStreamOutputStyle(stream);
00231 }
00232
00233 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printYellowText(FILE *stream, const char *text)
00234 {
00235     FMC_changeStreamTextColorToYellow(stream);
00236     fprintf(stream, "%s\n", text);
00237     FMC_resetStreamOutputStyle(stream);
00238 }
00239
00240 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBlueText(FILE *stream, const char *text)
00241 {
00242     FMC_changeStreamTextColorToBlue(stream);
00243     fprintf(stream, "%s\n", text);
00244     FMC_resetStreamOutputStyle(stream);
00245 }
00246
00247 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printMagentaText(FILE *stream, const char *text)
00248 {
00249     FMC_changeStreamTextColorToMagenta(stream);
00250     fprintf(stream, "%s\n", text);
00251     FMC_resetStreamOutputStyle(stream);
00252 }
00253
00254 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printCyanText(FILE *stream, const char *text)
00255 {
00256     FMC_changeStreamTextColorToCyan(stream);
00257     fprintf(stream, "%s\n", text);
00258     FMC_resetStreamOutputStyle(stream);
00259 }
00260
00261 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printWhiteText(FILE *stream, const char *text)
00262 {
00263     FMC_changeStreamTextColorToWhite(stream);
00264     fprintf(stream, "%s\n", text);
00265     FMC_resetStreamOutputStyle(stream);
00266 }
00267
00268 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightRedText(FILE *stream, const char *text)
00269 {
00270     FMC_changeStreamTextColorToBrightRed(stream);
00271     fprintf(stream, "%s\n", text);
00272     FMC_resetStreamOutputStyle(stream);
00273 }
00274
00275 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightGreenText(FILE *stream, const char *text)
00276 {
00277     FMC_changeStreamTextColorToBrightGreen(stream);
00278     fprintf(stream, "%s\n", text);
00279     FMC_resetStreamOutputStyle(stream);
00280 }
00281
00282 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightYellowText(FILE *stream, const char *text)
00283 {
00284     FMC_changeStreamTextColorToBrightYellow(stream);
00285     fprintf(stream, "%s\n", text);
00286     FMC_resetStreamOutputStyle(stream);
00287 }
00288
```

```

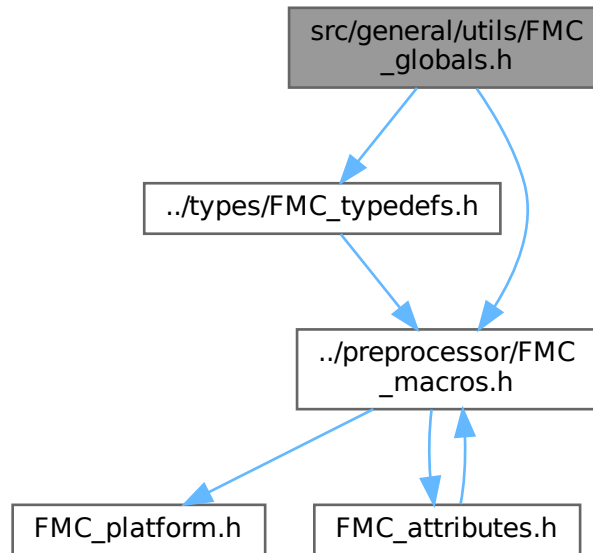
00289 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightBlueText(FILE *stream, const char *text)
00290 {
00291     FMC_changeStreamTextColorToBrightBlue(stream);
00292     fprintf(stream, "%s\n", text);
00293     FMC_resetStreamOutputStyle(stream);
00294 }
00295
00296 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightMagentaText(FILE *stream, const char *text)
00297 {
00298     FMC_changeStreamTextColorToBrightMagenta(stream);
00299     fprintf(stream, "%s\n", text);
00300     FMC_resetStreamOutputStyle(stream);
00301 }
00302
00303 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightCyanText(FILE *stream, const char *text)
00304 {
00305     FMC_changeStreamTextColorToBrightCyan(stream);
00306     fprintf(stream, "%s\n", text);
00307     FMC_resetStreamOutputStyle(stream);
00308 }
00309
00310 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightWhiteText(FILE *stream, const char *text)
00311 {
00312     FMC_changeStreamTextColorToBrightWhite(stream);
00313     fprintf(stream, "%s\n", text);
00314     FMC_resetStreamOutputStyle(stream);
00315 }
00316
00317 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printRedError(FILE *stream, const char *text)
00318 {
00319     FMC_changeStreamTextColorToRed(stream);
00320     FMC_setTextStyleToBold(stream);
00321     fprintf(stream, "%s\n", text);
00322     FMC_resetStreamOutputStyle(stream);
00323 }
00324
00325 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printGreenError(FILE *stream, const char *text)
00326 {
00327     FMC_changeStreamTextColorToGreen(stream);
00328     FMC_setTextStyleToBold(stream);
00329     fprintf(stream, "%s\n", text);
00330     FMC_resetStreamOutputStyle(stream);
00331 }
00332
00333 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printYellowError(FILE *stream, const char *text)
00334 {
00335     FMC_changeStreamTextColorToYellow(stream);
00336     FMC_setTextStyleToBold(stream);
00337     fprintf(stream, "%s\n", text);
00338     FMC_resetStreamOutputStyle(stream);
00339 }
00340
00341 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBlueError(FILE *stream, const char *text)
00342 {
00343     FMC_changeStreamTextColorToBlue(stream);
00344     FMC_setTextStyleToBold(stream);
00345     fprintf(stream, "%s\n", text);
00346     FMC_resetStreamOutputStyle(stream);
00347 }
00348
00349 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printMagentaError(FILE *stream, const char *text)
00350 {
00351     FMC_changeStreamTextColorToMagenta(stream);
00352     FMC_setTextStyleToBold(stream);
00353     fprintf(stream, "%s\n", text);
00354     FMC_resetStreamOutputStyle(stream);
00355 }
00356
00357 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printCyanError(FILE *stream, const char *text)
00358 {
00359     FMC_changeStreamTextColorToCyan(stream);
00360     FMC_setTextStyleToBold(stream);
00361     fprintf(stream, "%s\n", text);
00362     FMC_resetStreamOutputStyle(stream);
00363 }
00364
00365 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printWhiteError(FILE *stream, const char *text)
00366 {
00367     FMC_changeStreamTextColorToWhite(stream);
00368     FMC_setTextStyleToBold(stream);
00369     fprintf(stream, "%s\n", text);
00370     FMC_resetStreamOutputStyle(stream);
00371 }
00372
00373 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightRedError(FILE *stream, const char *text)
00374 {
00375     FMC_changeStreamTextColorToBrightRed(stream);

```

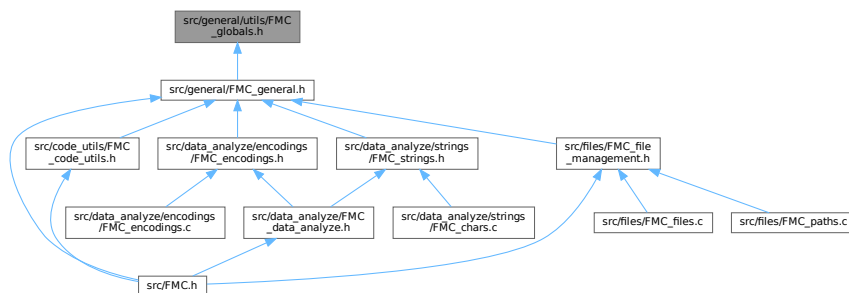
```
00376     FMC_setTextStyleToBold(stream);
00377     fprintf(stream, "%s\n", text);
00378     FMC_resetStreamOutputStyle(stream);
00379 }
00380
00381 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightGreenError(FILE *stream, const char *text)
00382 {
00383     FMC_changeStreamTextColorToBrightGreen(stream);
00384     FMC_setTextStyleToBold(stream);
00385     fprintf(stream, "%s\n", text);
00386     FMC_resetStreamOutputStyle(stream);
00387 }
00388
00389 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightYellowError(FILE *stream, const char *text)
00390 {
00391     FMC_changeStreamTextColorToBrightYellow(stream);
00392     FMC_setTextStyleToBold(stream);
00393     fprintf(stream, "%s\n", text);
00394     FMC_resetStreamOutputStyle(stream);
00395 }
00396
00397 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightBlueError(FILE *stream, const char *text)
00398 {
00399     FMC_changeStreamTextColorToBrightBlue(stream);
00400     FMC_setTextStyleToBold(stream);
00401     fprintf(stream, "%s\n", text);
00402     FMC_resetStreamOutputStyle(stream);
00403 }
00404
00405 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightMagentaError(FILE *stream, const char *text)
00406 {
00407     FMC_changeStreamTextColorToBrightMagenta(stream);
00408     FMC_setTextStyleToBold(stream);
00409     fprintf(stream, "%s\n", text);
00410     FMC_resetStreamOutputStyle(stream);
00411 }
00412
00413 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightCyanError(FILE *stream, const char *text)
00414 {
00415     FMC_changeStreamTextColorToBrightCyan(stream);
00416     FMC_setTextStyleToBold(stream);
00417     fprintf(stream, "%s\n", text);
00418     FMC_resetStreamOutputStyle(stream);
00419 }
00420
00421 FMC_FUNC_FLATTEN FMC_FUNC_INLINE void FMC_printBrightWhiteError(FILE *stream, const char *text)
00422 {
00423     FMC_changeStreamTextColorToBrightWhite(stream);
00424     FMC_setTextStyleToBold(stream);
00425     fprintf(stream, "%s\n", text);
00426     FMC_resetStreamOutputStyle(stream);
00427 }
00428
00429 #endif // FMC_ERRORS
```

3.60 src/general/utls/FMC_globals.h File Reference

Include dependency graph for FMC_globals.h:



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



Variables

- `FMC_Bool` `FMC_ENABLE_DEBUG` `FMC_VAR_COMMON`

3.60.1 Variable Documentation

3.60.1.1 FMC_VAR_COMMON

```
FMC_Bool FMC_ENABLE_DEBUG FMC_VAR_COMMON [extern]
```

3.61 FMC_globals.h

[Go to the documentation of this file.](#)

```
00001 /*
00002
00003 MIT License
00004
00005 Copyright (c) 2022-2023 Axel PASCON
00006
00007 Permission is hereby granted, free of charge, to any person obtaining a copy
00008 of this software and associated documentation files (the "Software"), to deal
00009 in the Software without restriction, including without limitation the rights
00010 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
00011 copies of the Software, and to permit persons to whom the Software is
00012 furnished to do so, subject to the following conditions:
00013
00014 The above copyright notice and this permission notice shall be included in all
00015 copies or substantial portions of the Software.
00016
00017 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
00018 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00019 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
00020 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00021 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00022 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
00023 SOFTWARE.
00024
00025 */
00026
00027 #ifndef FMC_GLOBALS_H
00028 #define FMC_GLOBALS_H
00029
00030 #include "../types/FMC_typedefs.h"
00031 #include "../preprocessor/FMC_macros.h"
00032
00033 extern FMC_Bool FMC_ENABLE_DEBUG FMC_VAR_COMMON;
00034
00035 #endif // FMC_GLOBALS_H
```


Index

`__STDC_WANT_LIB_EXT1__`
 FMC_encodings.c, [52](#)

ASCII
 FMC_flags.h, [95](#)

ascii
 FMC_enums.h, [106](#)

BG_BLACK
 FMC_consts.h, [85](#)

BG_BLUE
 FMC_consts.h, [85](#)

BG_BRIGHT_BLACK
 FMC_consts.h, [85](#)

BG_BRIGHT_BLUE
 FMC_consts.h, [85](#)

BG_BRIGHT_CYAN
 FMC_consts.h, [86](#)

BG_BRIGHT_GREEN
 FMC_consts.h, [86](#)

BG_BRIGHT_MAGENTA
 FMC_consts.h, [86](#)

BG_BRIGHT_RED
 FMC_consts.h, [86](#)

BG_BRIGHT_WHITE
 FMC_consts.h, [86](#)

BG_BRIGHT_YELLOW
 FMC_consts.h, [86](#)

BG_CYAN
 FMC_consts.h, [87](#)

BG_GREEN
 FMC_consts.h, [87](#)

BG_MAGENTA
 FMC_consts.h, [87](#)

BG_RED
 FMC_consts.h, [87](#)

BG_WHITE
 FMC_consts.h, [87](#)

BG_YELLOW
 FMC_consts.h, [87](#)

docs/documentation_pages/main_page.dox, [5](#)

error
 FMC_enums.h, [106](#)

extension
 FMC_deprecated.h, [121](#)

False
 FMC_consts.h, [88](#)

FG_BLACK
 FMC_consts.h, [88](#)

FG_BLUE
 FMC_consts.h, [88](#)

FG_BRIGHT_BLACK
 FMC_consts.h, [88](#)

FG_BRIGHT_BLUE
 FMC_consts.h, [88](#)

FG_BRIGHT_CYAN
 FMC_consts.h, [88](#)

FG_BRIGHT_GREEN
 FMC_consts.h, [89](#)

FG_BRIGHT_MAGENTA
 FMC_consts.h, [89](#)

FG_BRIGHT_RED
 FMC_consts.h, [89](#)

FG_BRIGHT_WHITE
 FMC_consts.h, [89](#)

FG_BRIGHT_YELLOW
 FMC_consts.h, [89](#)

FG_CYAN
 FMC_consts.h, [89](#)

FG_GREEN
 FMC_consts.h, [90](#)

FG_MAGENTA
 FMC_consts.h, [90](#)

FG_RED
 FMC_consts.h, [90](#)

FG_WHITE
 FMC_consts.h, [90](#)

FG_YELLOW
 FMC_consts.h, [90](#)

fileName
 FMC_deprecated.h, [121](#)

filePath
 FMC_deprecated.h, [121](#)

FManC_Char, [108](#)

FManC_CharComp, [109](#)

FManC_CStrView, [109](#)

FManC_Encodings
 FMC_enums.h, [106](#)

FManC_File, [110](#)

FManC_String, [111](#)

FManC_StrOcc, [112](#)

FMC.h
 FMC_H, [76](#)

FMC_Bool
 FMC_typedefs.h, [116](#)

FMC_BOOLEANS
 FMC_consts.h, [90](#)

- FMC_changeStreamTextColorToBlue
 - FMC_errors.c, [124](#)
 - FMC_errors.h, [152](#)
- FMC_changeStreamTextColorToBrightBlue
 - FMC_errors.c, [124](#)
 - FMC_errors.h, [152](#)
- FMC_changeStreamTextColorToBrightCyan
 - FMC_errors.c, [125](#)
 - FMC_errors.h, [153](#)
- FMC_changeStreamTextColorToBrightGreen
 - FMC_errors.c, [125](#)
 - FMC_errors.h, [153](#)
- FMC_changeStreamTextColorToBrightMagenta
 - FMC_errors.c, [126](#)
 - FMC_errors.h, [154](#)
- FMC_changeStreamTextColorToBrightRed
 - FMC_errors.c, [126](#)
 - FMC_errors.h, [154](#)
- FMC_changeStreamTextColorToBrightWhite
 - FMC_errors.c, [127](#)
 - FMC_errors.h, [155](#)
- FMC_changeStreamTextColorToBrightYellow
 - FMC_errors.c, [127](#)
 - FMC_errors.h, [155](#)
- FMC_changeStreamTextColorToCyan
 - FMC_errors.c, [128](#)
 - FMC_errors.h, [156](#)
- FMC_changeStreamTextColorToGreen
 - FMC_errors.c, [128](#)
 - FMC_errors.h, [156](#)
- FMC_changeStreamTextColorToMagenta
 - FMC_errors.c, [129](#)
 - FMC_errors.h, [157](#)
- FMC_changeStreamTextColorToRed
 - FMC_errors.c, [129](#)
 - FMC_errors.h, [157](#)
- FMC_changeStreamTextColorToWhite
 - FMC_errors.c, [130](#)
 - FMC_errors.h, [158](#)
- FMC_changeStreamTextColorToYellow
 - FMC_errors.c, [130](#)
 - FMC_errors.h, [158](#)
- FMC_Char
 - FMC_structs.h, [113](#)
- FMC_CharComp
 - FMC_structs.h, [113](#)
- FMC_CharControl
 - FMC_typedefs.h, [117](#)
- FMC_checkEncodingFlag
 - FMC_encodings.c, [53](#)
 - FMC_encodings.h, [58](#)
- FMC_code_utils.h
 - FMC_CODE_UTILS_H, [6](#)
- FMC_CODE_UTILS_H
 - FMC_code_utils.h, [6](#)
- FMC_COMPILE_TIME_ERROR
 - FMC_macros.h, [98](#)
- FMC_consts.h
- BG_BLACK, [85](#)
- BG_BLUE, [85](#)
- BG_BRIGHT_BLACK, [85](#)
- BG_BRIGHT_BLUE, [85](#)
- BG_BRIGHT_CYAN, [86](#)
- BG_BRIGHT_GREEN, [86](#)
- BG_BRIGHT_MAGENTA, [86](#)
- BG_BRIGHT_RED, [86](#)
- BG_BRIGHT_WHITE, [86](#)
- BG_BRIGHT_YELLOW, [86](#)
- BG_CYAN, [87](#)
- BG_GREEN, [87](#)
- BG_MAGENTA, [87](#)
- BG_RED, [87](#)
- BG_WHITE, [87](#)
- BG_YELLOW, [87](#)
- False, [88](#)
- FG_BLACK, [88](#)
- FG_BLUE, [88](#)
- FG_BRIGHT_BLACK, [88](#)
- FG_BRIGHT_BLUE, [88](#)
- FG_BRIGHT_CYAN, [88](#)
- FG_BRIGHT_GREEN, [89](#)
- FG_BRIGHT_MAGENTA, [89](#)
- FG_BRIGHT_RED, [89](#)
- FG_BRIGHT_WHITE, [89](#)
- FG_BRIGHT_YELLOW, [89](#)
- FG_CYAN, [89](#)
- FG_GREEN, [90](#)
- FG_MAGENTA, [90](#)
- FG_RED, [90](#)
- FG_WHITE, [90](#)
- FG_YELLOW, [90](#)
- FMC_BOOLEANS, [90](#)
- FMC_CONSTS_H, [91](#)
- FMC_MAX_PATH_COMPONENTS_SIZE, [91](#)
- FMC_STYLES, [91](#)
- MAX_FEXT_SIZE, [91](#)
- MAX_FNAME_SIZE, [91](#)
- MAX_FPATH_SIZE, [91](#)
- RESET, [92](#)
- True, [92](#)
- TXT_BLINK, [92](#)
- TXT_BOLD, [92](#)
- TXT_DIM, [92](#)
- TXT_HIDDEN, [92](#)
- TXT_REVERSE, [93](#)
- TXT_UNDERLINED, [93](#)
- FMC_CONSTS_H
 - FMC_consts.h, [91](#)
- FMC_CStrView
 - FMC_structs.h, [113](#)
- FMC_cutFilename
 - FMC_file_management.h, [65](#)
 - FMC_paths.c, [70](#)
- FMC_data_analyze.h
 - FMC_DATA_ANALYZE_H, [60](#)
- FMC_DATA_ANALYZE_H

- FMC_data_analyze.h, 60
- FMC_DATA_H
 - FMC_general.h, 78
- FMC_DEFER
 - FMC_macros.h, 98
- FMC_deprecated.h
 - extension, 121
 - fileName, 121
 - filePath, 121
 - FMC_FUNC_UNAVAILABLE, 120
 - FMC_TYPE_UNAVAILABLE, 120
 - pathToCopy, 121
 - toSearch, 121
- FMC_dir.cpp
 - FMC_dirExists_, 8
 - FMC_getAbsolutePath_, 8
 - FMC_getCurrentPath_, 8
 - FMC_isBlock_, 9
 - FMC_isCharFile_, 9
 - FMC_isDir_, 10
 - FMC_isEmpty_, 10
 - FMC_isFIFO_, 11
 - FMC_isOther_, 11
 - FMC_isRegFile_, 12
 - FMC_isSocket_, 12
 - FMC_isSymLink_, 13
- FMC_dir.hpp
 - FMC_dirExists_, 17
 - FMC_getAbsolutePath_, 17
 - FMC_getCurrentPath_, 17
 - FMC_isBlock_, 18
 - FMC_isCharFile_, 18
 - FMC_isDir_, 19
 - FMC_isEmpty_, 19
 - FMC_isFIFO_, 20
 - FMC_isOther_, 20
 - FMC_isRegFile_, 21
 - FMC_isSocket_, 21
 - FMC_isSymLink_, 22
- FMC_dir_wrapper.cpp
 - FMC_dirExists, 25
 - FMC_getAbsolutePath, 25
 - FMC_getCurrentPath, 26
 - FMC_isBlock, 27
 - FMC_isCharFile, 28
 - FMC_isDir, 29
 - FMC_isEmpty, 30
 - FMC_isFIFO, 31
 - FMC_isOther, 32
 - FMC_isRegFile, 33
 - FMC_isSocket, 34
 - FMC_isSymLink, 35
- FMC_dirExists
 - FMC_dir_wrapper.cpp, 25
 - FMC_wrapper.h, 39
- FMC_dirExists_
 - FMC_dir.cpp, 8
 - FMC_dir.hpp, 17
- FMC_ENCODING_FLAGS
 - FMC_flags.h, 95
- FMC_ENCODINGS
 - FMC_encodings.h, 58
- FMC_Encodings
 - FMC_enums.h, 105
- FMC_encodings.c
 - __STDC_WANT_LIB_EXT1__, 52
 - FMC_checkEncodingFlag, 53
 - FMC_getEncoding, 53
- FMC_encodings.h
 - FMC_checkEncodingFlag, 58
 - FMC_ENCODINGS, 58
 - FMC_getEncoding, 58
- FMC_enums.h
 - ascii, 106
 - error, 106
 - FManC_Encodings, 106
 - FMC_Encodings, 105
 - FMC_ENUMS_H, 105
 - unknown, 106
 - utf16_be, 106
 - utf16_le, 106
 - utf32_be, 106
 - utf32_le, 106
 - utf8, 106
 - utf8_bom, 106
- FMC_ENUMS_H
 - FMC_enums.h, 105
- FMC_ERROR_CHECK
 - FMC_macros.h, 98
- FMC_ERRORS
 - FMC_errors.h, 152
- FMC_errors.c
 - FMC_changeStreamTextColorToBlue, 124
 - FMC_changeStreamTextColorToBrightBlue, 124
 - FMC_changeStreamTextColorToBrightCyan, 125
 - FMC_changeStreamTextColorToBrightGreen, 125
 - FMC_changeStreamTextColorToBrightMagenta, 126
 - FMC_changeStreamTextColorToBrightRed, 126
 - FMC_changeStreamTextColorToBrightWhite, 127
 - FMC_changeStreamTextColorToBrightYellow, 127
 - FMC_changeStreamTextColorToCyan, 128
 - FMC_changeStreamTextColorToGreen, 128
 - FMC_changeStreamTextColorToMagenta, 129
 - FMC_changeStreamTextColorToRed, 129
 - FMC_changeStreamTextColorToWhite, 130
 - FMC_changeStreamTextColorToYellow, 130
 - FMC_makeMsg_f, 131
 - FMC_printBlueError, 131
 - FMC_printBlueText, 132
 - FMC_printBrightBlueError, 132
 - FMC_printBrightBlueText, 133
 - FMC_printBrightCyanError, 133
 - FMC_printBrightCyanText, 134
 - FMC_printBrightGreenError, 134
 - FMC_printBrightGreenText, 135

- FMC_printBrightMagentaError, 135
- FMC_printBrightMagentaText, 136
- FMC_printBrightRedError, 136
- FMC_printBrightRedText, 137
- FMC_printBrightWhiteError, 138
- FMC_printBrightWhiteText, 138
- FMC_printBrightYellowError, 139
- FMC_printBrightYellowText, 140
- FMC_printCyanError, 140
- FMC_printCyanText, 141
- FMC_printGreenError, 141
- FMC_printGreenText, 142
- FMC_printMagentaError, 142
- FMC_printMagentaText, 143
- FMC_printRedError, 143
- FMC_printRedText, 144
- FMC_printWhiteError, 144
- FMC_printWhiteText, 145
- FMC_printYellowError, 145
- FMC_printYellowText, 146
- FMC_resetStreamOutputStyle, 146
- FMC_errors.h
 - FMC_changeStreamTextColorToBlue, 152
 - FMC_changeStreamTextColorToBrightBlue, 152
 - FMC_changeStreamTextColorToBrightCyan, 153
 - FMC_changeStreamTextColorToBrightGreen, 153
 - FMC_changeStreamTextColorToBrightMagenta, 154
 - FMC_changeStreamTextColorToBrightRed, 154
 - FMC_changeStreamTextColorToBrightWhite, 155
 - FMC_changeStreamTextColorToBrightYellow, 155
 - FMC_changeStreamTextColorToCyan, 156
 - FMC_changeStreamTextColorToGreen, 156
 - FMC_changeStreamTextColorToMagenta, 157
 - FMC_changeStreamTextColorToRed, 157
 - FMC_changeStreamTextColorToWhite, 158
 - FMC_changeStreamTextColorToYellow, 158
 - FMC_ERRORS, 152
 - FMC_makeMsg, 152
 - FMC_makeMsg_f, 159
 - FMC_printBlueError, 159
 - FMC_printBlueText, 160
 - FMC_printBrightBlueError, 160
 - FMC_printBrightBlueText, 161
 - FMC_printBrightCyanError, 161
 - FMC_printBrightCyanText, 162
 - FMC_printBrightGreenError, 162
 - FMC_printBrightGreenText, 163
 - FMC_printBrightMagentaError, 163
 - FMC_printBrightMagentaText, 164
 - FMC_printBrightRedError, 164
 - FMC_printBrightRedText, 165
 - FMC_printBrightWhiteError, 166
 - FMC_printBrightWhiteText, 166
 - FMC_printBrightYellowError, 167
 - FMC_printBrightYellowText, 168
 - FMC_printCyanError, 168
 - FMC_printCyanText, 169
 - FMC_printGreenError, 169
 - FMC_printGreenText, 170
 - FMC_printMagentaError, 170
 - FMC_printMagentaText, 171
 - FMC_printRedError, 171
 - FMC_printRedText, 172
 - FMC_printWhiteError, 172
 - FMC_printWhiteText, 173
 - FMC_printYellowError, 173
 - FMC_printYellowText, 174
 - FMC_resetStreamOutputStyle, 174
 - FMC_setBGStreamColorToBlue, 176
 - FMC_setBGStreamColorToBrightBlue, 176
 - FMC_setBGStreamColorToBrightCyan, 177
 - FMC_setBGStreamColorToBrightGreen, 177
 - FMC_setBGStreamColorToBrightMagenta, 177
 - FMC_setBGStreamColorToBrightRed, 177
 - FMC_setBGStreamColorToBrightWhite, 177
 - FMC_setBGStreamColorToBrightYellow, 178
 - FMC_setBGStreamColorToCyan, 178
 - FMC_setBGStreamColorToGreen, 178
 - FMC_setBGStreamColorToMagenta, 178
 - FMC_setBGStreamColorToRed, 178
 - FMC_setBGStreamColorToWhite, 179
 - FMC_setBGStreamColorToYellow, 179
 - FMC_setTextStyleToBlink, 179
 - FMC_setTextStyleToBold, 179
 - FMC_setTextStyleToDim, 180
 - FMC_setTextStyleToHidden, 180
 - FMC_setTextStyleToReverse, 181
 - FMC_setTextStyleToUnderlined, 181
- FMC_extractFilename
 - FMC_file_management.h, 65
 - FMC_paths.c, 70
- FMC_File
 - FMC_structs.h, 113
- FMC_file_management.h
 - FMC_cutFilename, 65
 - FMC_extractFilename, 65
 - FMC_FILE_MANAGEMENT_H, 65
 - FMC_getExtension, 67
- FMC_FILE_MANAGEMENT_H
 - FMC_file_management.h, 65
- FMC_FileState
 - FMC_typedefs.h, 117
- FMC_FLAGS
 - FMC_flags.h, 95
- FMC_flags.h
 - ASCII, 95
 - FMC_ENCODING_FLAGS, 95
 - FMC_FLAGS, 95
 - UNKNOWN, 95
 - UTF16_BE, 95
 - UTF16_LE, 96
 - UTF32_BE, 96
 - UTF32_LE, 96
 - UTF8, 96
 - UTF8_BOM, 96

FMC_FUNC_UNAVAILABLE
 FMC_deprecated.h, 120
FMC_general.h
 FMC_DATA_H, 78
FMC_getAbsolutePath
 FMC_dir_wrapper.cpp, 25
 FMC_wrapper.h, 40
FMC_getAbsolutePath_
 FMC_dir.cpp, 8
 FMC_dir.hpp, 17
FMC_getCurrentPath
 FMC_dir_wrapper.cpp, 26
 FMC_wrapper.h, 41
FMC_getCurrentPath_
 FMC_dir.cpp, 8
 FMC_dir.hpp, 17
FMC_getEncoding
 FMC_encodings.c, 53
 FMC_encodings.h, 58
FMC_getExtension
 FMC_file_management.h, 67
 FMC_paths.c, 72
FMC_globals.h
 FMC_VAR_COMMON, 187
FMC_H
 FMC.h, 76
FMC_isBlock
 FMC_dir_wrapper.cpp, 27
 FMC_wrapper.h, 42
FMC_isBlock_
 FMC_dir.cpp, 9
 FMC_dir.hpp, 18
FMC_isCharFile
 FMC_dir_wrapper.cpp, 28
 FMC_wrapper.h, 43
FMC_isCharFile_
 FMC_dir.cpp, 9
 FMC_dir.hpp, 18
FMC_isDir
 FMC_dir_wrapper.cpp, 29
 FMC_wrapper.h, 44
FMC_isDir_
 FMC_dir.cpp, 10
 FMC_dir.hpp, 19
FMC_isEmpty
 FMC_dir_wrapper.cpp, 30
 FMC_wrapper.h, 45
FMC_isEmpty_
 FMC_dir.cpp, 10
 FMC_dir.hpp, 19
FMC_isFIFO
 FMC_dir_wrapper.cpp, 31
 FMC_wrapper.h, 46
FMC_isFIFO_
 FMC_dir.cpp, 11
 FMC_dir.hpp, 20
FMC_isOther
 FMC_dir_wrapper.cpp, 32
 FMC_wrapper.h, 47
FMC_isOther_
 FMC_dir.cpp, 11
 FMC_dir.hpp, 20
FMC_isRegFile
 FMC_dir_wrapper.cpp, 33
 FMC_wrapper.h, 48
FMC_isRegFile_
 FMC_dir.cpp, 12
 FMC_dir.hpp, 21
FMC_isSocket
 FMC_dir_wrapper.cpp, 34
 FMC_wrapper.h, 49
FMC_isSocket_
 FMC_dir.cpp, 12
 FMC_dir.hpp, 21
FMC_isSymLink
 FMC_dir_wrapper.cpp, 35
 FMC_wrapper.h, 50
FMC_isSymLink_
 FMC_dir.cpp, 13
 FMC_dir.hpp, 22
FMC_macros.h
 FMC_COMPILE_TIME_ERROR, 98
 FMC_DEFER, 98
 FMC_ERROR_CHECK, 98
 FMC_MACROS_H, 99
 FMC_MAJOR_VERSION, 99
 FMC_MINOR_VERSION, 99
 FMC_PATCH_VERSION, 99
 FMC_VERSION, 99
 FMC_VERSION_NUMBER, 100
 FMC_VERSION_STRING, 100
FMC_MACROS_H
 FMC_macros.h, 99
FMC_MAJOR_VERSION
 FMC_macros.h, 99
FMC_makeMsg
 FMC_errors.h, 152
FMC_makeMsg_f
 FMC_errors.c, 131
 FMC_errors.h, 159
FMC_MAX_PATH_COMPONENTS_SIZE
 FMC_consts.h, 91
FMC_MINOR_VERSION
 FMC_macros.h, 99
FMC_PATCH_VERSION
 FMC_macros.h, 99
FMC_paths.c
 FMC_cutFilename, 70
 FMC_extractFilename, 70
 FMC_getExtension, 72
FMC_printBlueError
 FMC_errors.c, 131
 FMC_errors.h, 159
FMC_printBlueText
 FMC_errors.c, 132
 FMC_errors.h, 160

FMC_printBrightBlueError
FMC_errors.c, 132
FMC_errors.h, 160

FMC_printBrightBlueText
FMC_errors.c, 133
FMC_errors.h, 161

FMC_printBrightCyanError
FMC_errors.c, 133
FMC_errors.h, 161

FMC_printBrightCyanText
FMC_errors.c, 134
FMC_errors.h, 162

FMC_printBrightGreenError
FMC_errors.c, 134
FMC_errors.h, 162

FMC_printBrightGreenText
FMC_errors.c, 135
FMC_errors.h, 163

FMC_printBrightMagentaError
FMC_errors.c, 135
FMC_errors.h, 163

FMC_printBrightMagentaText
FMC_errors.c, 136
FMC_errors.h, 164

FMC_printBrightRedError
FMC_errors.c, 136
FMC_errors.h, 164

FMC_printBrightRedText
FMC_errors.c, 137
FMC_errors.h, 165

FMC_printBrightWhiteError
FMC_errors.c, 138
FMC_errors.h, 166

FMC_printBrightWhiteText
FMC_errors.c, 138
FMC_errors.h, 166

FMC_printBrightYellowError
FMC_errors.c, 139
FMC_errors.h, 167

FMC_printBrightYellowText
FMC_errors.c, 140
FMC_errors.h, 168

FMC_printCyanError
FMC_errors.c, 140
FMC_errors.h, 168

FMC_printCyanText
FMC_errors.c, 141
FMC_errors.h, 169

FMC_printGreenError
FMC_errors.c, 141
FMC_errors.h, 169

FMC_printGreenText
FMC_errors.c, 142
FMC_errors.h, 170

FMC_printMagentaError
FMC_errors.c, 142
FMC_errors.h, 170

FMC_printMagentaText
FMC_errors.c, 143
FMC_errors.h, 171

FMC_printRedError
FMC_errors.c, 143
FMC_errors.h, 171

FMC_printRedText
FMC_errors.c, 144
FMC_errors.h, 172

FMC_printWhiteError
FMC_errors.c, 144
FMC_errors.h, 172

FMC_printWhiteText
FMC_errors.c, 145
FMC_errors.h, 173

FMC_printYellowError
FMC_errors.c, 145
FMC_errors.h, 173

FMC_printYellowText
FMC_errors.c, 146
FMC_errors.h, 174

FMC_resetStreamOutputStyle
FMC_errors.c, 146
FMC_errors.h, 174

FMC_setBGStreamColorToBlue
FMC_errors.h, 176

FMC_setBGStreamColorToBrightBlue
FMC_errors.h, 176

FMC_setBGStreamColorToBrightCyan
FMC_errors.h, 177

FMC_setBGStreamColorToBrightGreen
FMC_errors.h, 177

FMC_setBGStreamColorToBrightMagenta
FMC_errors.h, 177

FMC_setBGStreamColorToBrightRed
FMC_errors.h, 177

FMC_setBGStreamColorToBrightWhite
FMC_errors.h, 177

FMC_setBGStreamColorToBrightYellow
FMC_errors.h, 178

FMC_setBGStreamColorToCyan
FMC_errors.h, 178

FMC_setBGStreamColorToGreen
FMC_errors.h, 178

FMC_setBGStreamColorToMagenta
FMC_errors.h, 178

FMC_setBGStreamColorToRed
FMC_errors.h, 178

FMC_setBGStreamColorToWhite
FMC_errors.h, 179

FMC_setBGStreamColorToYellow
FMC_errors.h, 179

FMC_setTextStyleToBlink
FMC_errors.h, 179

FMC_setTextStyleToBold
FMC_errors.h, 179

FMC_setTextStyleToDim
FMC_errors.h, 180

FMC_setTextStyleToHidden

- FMC_errors.h, 180
- FMC_setTextStyleToReverse
 - FMC_errors.h, 181
- FMC_setTextStyleToUnderlined
 - FMC_errors.h, 181
- FMC_String
 - FMC_structs.h, 113
- FMC_strings.h
 - FMC_STRINGS_H, 63
- FMC_STRINGS_H
 - FMC_strings.h, 63
- FMC_StrOcc
 - FMC_structs.h, 113
- FMC_structs.h
 - FMC_Char, 113
 - FMC_CharComp, 113
 - FMC_CStrView, 113
 - FMC_File, 113
 - FMC_String, 113
 - FMC_StrOcc, 113
 - FMC_STRUCTS_H, 112
- FMC_STRUCTS_H
 - FMC_structs.h, 112
- FMC_STYLES
 - FMC_consts.h, 91
- FMC_TYPE_UNAVAILABLE
 - FMC_deprecated.h, 120
- FMC_typedefs.h
 - FMC_Bool, 116
 - FMC_CharControl, 117
 - FMC_FileState, 117
 - FMC_TYPEDEFS_H, 116
 - found_bs_n, 117
 - found_bs_r_bs_n, 117
 - found_bs_t, 117
- FMC_TYPEDEFS_H
 - FMC_typedefs.h, 116
- FMC_VAR_COMMON
 - FMC_globals.h, 187
- FMC_VERSION
 - FMC_macros.h, 99
- FMC_VERSION_NUMBER
 - FMC_macros.h, 100
- FMC_VERSION_STRING
 - FMC_macros.h, 100
- FMC_wrapper.h
 - FMC_dirExists, 39
 - FMC_getAbsolutePath, 40
 - FMC_getCurrentPath, 41
 - FMC_isBlock, 42
 - FMC_isCharFile, 43
 - FMC_isDir, 44
 - FMC_isEmpty, 45
 - FMC_isFIFO, 46
 - FMC_isOther, 47
 - FMC_isRegFile, 48
 - FMC_isSocket, 49
 - FMC_isSymLink, 50
- found_bs_n
 - FMC_typedefs.h, 117
- found_bs_r_bs_n
 - FMC_typedefs.h, 117
- found_bs_t
 - FMC_typedefs.h, 117
- MAX_FEXT_SIZE
 - FMC_consts.h, 91
- MAX_FNAME_SIZE
 - FMC_consts.h, 91
- MAX_FPATH_SIZE
 - FMC_consts.h, 91
- pathToCopy
 - FMC_deprecated.h, 121
- RESET
 - FMC_consts.h, 92
- src/code_utils/FMC_code_utils.h, 5, 6
- src/code_utils/FMC_codeUtils.c, 6
- src/cpp/FMC_dir/FMC_dir.cpp, 7, 14
- src/cpp/FMC_dir/FMC_dir.hpp, 16, 23
- src/cpp/FMC_dir/FMC_dir_wrapper.cpp, 24, 36
- src/cpp/FMC_wrapper.h, 38, 51
- src/data_analyze/encodings/FMC_encodings.c, 52, 54
- src/data_analyze/encodings/FMC_encodings.h, 57, 59
- src/data_analyze/FMC_data_analyze.h, 59, 60
- src/data_analyze/strings/FMC_chars.c, 61
- src/data_analyze/strings/FMC_strings.c, 62
- src/data_analyze/strings/FMC_strings.h, 62, 63
- src/files/FMC_file_management.h, 64, 68
- src/files/FMC_fileMan.c, 68
- src/files/FMC_files.c, 69
- src/files/FMC_paths.c, 70, 73
- src/FMC.h, 76, 77
- src/general/FMC_general.h, 77, 78
- src/general/preprocessor/FMC_attributes.h, 79, 80
- src/general/preprocessor/FMC_consts.h, 84, 93
- src/general/preprocessor/FMC_flags.h, 94, 97
- src/general/preprocessor/FMC_macros.h, 97, 100
- src/general/preprocessor/FMC_platform.h, 103
- src/general/types/FMC_enums.h, 104, 106
- src/general/types/FMC_structs.h, 107, 114
- src/general/types/FMC_typedefs.h, 115, 118
- src/general/utills/FMC_deprecated.h, 119, 122
- src/general/utills/FMC_errors.c, 123, 148
- src/general/utills/FMC_errors.h, 150, 181
- src/general/utills/FMC_globals.h, 187, 188
- toSearch
 - FMC_deprecated.h, 121
- True
 - FMC_consts.h, 92
- TXT_BLINK
 - FMC_consts.h, 92
- TXT_BOLD
 - FMC_consts.h, 92

TXT_DIM
 FMC_consts.h, [92](#)
TXT_HIDDEN
 FMC_consts.h, [92](#)
TXT_REVERSE
 FMC_consts.h, [93](#)
TXT_UNDERLINED
 FMC_consts.h, [93](#)

UNKNOWN
 FMC_flags.h, [95](#)
unknown
 FMC_enums.h, [106](#)
UTF16_BE
 FMC_flags.h, [95](#)
utf16_be
 FMC_enums.h, [106](#)
UTF16_LE
 FMC_flags.h, [96](#)
utf16_le
 FMC_enums.h, [106](#)
UTF32_BE
 FMC_flags.h, [96](#)
utf32_be
 FMC_enums.h, [106](#)
UTF32_LE
 FMC_flags.h, [96](#)
utf32_le
 FMC_enums.h, [106](#)
UTF8
 FMC_flags.h, [96](#)
utf8
 FMC_enums.h, [106](#)
UTF8_BOM
 FMC_flags.h, [96](#)
utf8_bom
 FMC_enums.h, [106](#)