

Embroidermodder

Generated by Doxygen 1.9.4

1 Overview	1
1.0.1 License	1
2 About	2
2.1 The Embroidermodder Project and Team	2
2.1.1 "Core Development Team"	2
2.2 for Embroidermodder 2, libembroidery and all other related code	3
2.2.1 Embroidermodder 1	3
2.2.2 Features	3
2.2.3 "Build and Install"	4
2.2.4 History	5
2.3 Contact us	5
3 Downloads	5
3.1 Alpha Build	5
4 Changelog	6
5 Ideas	6
6 Formats	6
6.1 Overview	6
6.1.1 Read/Write Support Levels	6
6.1.2 Table of Format Support Levels	7
6.1.3 Format Support	9
7 Geometry and Algorithms	9
7.1 To Do	9
7.1.1 Development	10
7.1.2 Testing	10
7.1.3 Contributing	10
7.1.4 Embroidermodder Project Coding Standards	11
7.1.5 Version Control	11
7.1.6 Donations	11
7.1.7 Embroidermodder Project Coding Standards	12
7.1.8 Ideas	13
7.1.9 Electronics development	15
7.1.10 Development	15
7.2 Embroiderbot and Libembroidery on Embedded Systems	16
7.2.1 Compatible Boards	16
7.2.2 Arduino Considerations	16
7.2.3 Space	16
7.2.4 Tables	17
7.2.5 Current Pattern Memory Management	17

7.2.6 Special Notes	17
7.2.7 The Assembly Split	17
7.3 The Embroider Command Line Program	17
7.3.1 Embroider pipeline	18
7.3.2 embroider CLI	18
8 GNU Free Documentation License	18
9 Contributor Covenant Code of Conduct	24
9.1 Our Pledge	24
9.2 Our Standards	24
9.3 Enforcement Responsibilities	24
9.4 Scope	24
9.5 Enforcement	25
9.6 Enforcement Guidelines	25
9.6.1 1. Correction	25
9.6.2 2. Warning	25
9.6.3 3. Temporary Ban	25
9.6.4 4. Permanent Ban	25
9.7 Attribution	26
10 Privacy Policy for Embroidery Viewer	26
10.0.1 CONTACT US	26
11 Todo List	26
12 Namespace Index	30
12.1 Namespace List	30
13 Hierarchical Index	30
13.1 Class Hierarchy	30
14 Class Index	33
14.1 Class List	33
15 File Index	35
15.1 File List	35
16 Namespace Documentation	39
16.1 em2_dev_script Namespace Reference	39
16.1.1 Detailed Description	39
16.1.2 Variable Documentation	39
17 Class Documentation	40
17.1 _bcf_directory Struct Reference	40
17.1.1 Detailed Description	40

17.1.2 Member Data Documentation	40
17.2 _bcf_directory_entry Struct Reference	40
17.2.1 Member Data Documentation	41
17.3 _bcf_file Struct Reference	42
17.3.1 Member Data Documentation	42
17.4 _bcf_file_difat Struct Reference	43
17.4.1 Member Data Documentation	43
17.5 _bcf_file_fat Struct Reference	43
17.5.1 Member Data Documentation	44
17.6 _bcf_file_header Struct Reference	44
17.6.1 Detailed Description	45
17.6.2 Member Data Documentation	45
17.7 _vp3Hoop Struct Reference	46
17.7.1 Member Data Documentation	47
17.8 Application Class Reference	49
17.8.1 Detailed Description	49
17.8.2 Constructor & Destructor Documentation	49
17.8.3 Member Function Documentation	50
17.8.4 Member Data Documentation	50
17.9 CmdPrompt Class Reference	50
17.9.1 Detailed Description	52
17.9.2 Constructor & Destructor Documentation	52
17.9.3 Member Function Documentation	53
17.9.4 Member Data Documentation	59
17.10 CmdPromptHandle Class Reference	59
17.10.1 Detailed Description	60
17.10.2 Constructor & Destructor Documentation	60
17.10.3 Member Function Documentation	61
17.10.4 Member Data Documentation	62
17.11 CmdPromptHistory Class Reference	62
17.11.1 Detailed Description	63
17.11.2 Constructor & Destructor Documentation	63
17.11.3 Member Function Documentation	63
17.11.4 Member Data Documentation	65
17.12 CmdPromptInput Class Reference	65
17.12.1 Constructor & Destructor Documentation	67
17.12.2 Member Function Documentation	67
17.12.3 Member Data Documentation	72
17.13 CmdPromptSplitter Class Reference	73
17.13.1 Detailed Description	74
17.13.2 Constructor & Destructor Documentation	74
17.13.3 Member Function Documentation	74

17.14 Compress Struct Reference	75
17.14.1 Member Data Documentation	75
17.15 EmbAlignedDim_ Struct Reference	76
17.15.1 Member Data Documentation	76
17.16 EmbAngularDim_ Struct Reference	76
17.16.1 Member Data Documentation	77
17.17 EmbArc_ Struct Reference	77
17.17.1 Detailed Description	77
17.17.2 Member Data Documentation	77
17.18 EmbArcLengthDim_ Struct Reference	78
17.18.1 Member Data Documentation	78
17.19 EmbArray_ Struct Reference	78
17.19.1 Member Data Documentation	78
17.20 EmbBezier_ Struct Reference	79
17.20.1 Member Data Documentation	79
17.21 EmbBlock_ Struct Reference	80
17.21.1 Member Data Documentation	80
17.22 EmbCircle_ Struct Reference	80
17.22.1 Member Data Documentation	80
17.23 EmbColor_ Struct Reference	81
17.23.1 Detailed Description	81
17.23.2 Member Data Documentation	81
17.24 EmbDetailsDialog Class Reference	81
17.24.1 Detailed Description	82
17.24.2 Constructor & Destructor Documentation	82
17.24.3 Member Function Documentation	83
17.24.4 Member Data Documentation	83
17.25 EmbDiameterDim_ Struct Reference	84
17.25.1 Member Data Documentation	84
17.26 EmbEllipse_ Struct Reference	84
17.26.1 Member Data Documentation	85
17.27 EmbFormatList_ Struct Reference	85
17.27.1 Member Data Documentation	85
17.28 EmbGeometry_ Struct Reference	86
17.28.1 Member Data Documentation	87
17.29 EmbImage_ Struct Reference	88
17.29.1 Member Data Documentation	89
17.30 EmbInfiniteLine_ Struct Reference	89
17.30.1 Member Data Documentation	90
17.31 EmbLayer_ Struct Reference	90
17.31.1 Member Data Documentation	90
17.32 EmbLeaderDim_ Struct Reference	90

17.32.1 Member Data Documentation	91
17.33 EmbLine_ Struct Reference	91
17.33.1 Member Data Documentation	91
17.34 EmbLinearDim_ Struct Reference	92
17.34.1 Member Data Documentation	92
17.35 EmbOrdinateDim_ Struct Reference	92
17.35.1 Member Data Documentation	92
17.36 EmbPath_ Struct Reference	92
17.36.1 Member Data Documentation	93
17.37 EmbPattern_ Struct Reference	93
17.37.1 Member Data Documentation	94
17.38 EmbPoint_ Struct Reference	94
17.38.1 Member Data Documentation	95
17.39 EmbRadiusDim_ Struct Reference	95
17.39.1 Member Data Documentation	95
17.40 EmbRay_ Struct Reference	96
17.40.1 Member Data Documentation	96
17.41 EmbRect_ Struct Reference	96
17.41.1 Member Data Documentation	96
17.42 EmbSatinOutline_ Struct Reference	97
17.42.1 Member Data Documentation	97
17.43 EmbSpline_ Struct Reference	98
17.43.1 Member Data Documentation	98
17.44 EmbStitch_ Struct Reference	98
17.44.1 Member Data Documentation	98
17.45 EmbTextMulti_ Struct Reference	99
17.45.1 Member Data Documentation	99
17.46 EmbTextSingle_ Struct Reference	99
17.46.1 Member Data Documentation	100
17.47 EmbThread_ Struct Reference	100
17.47.1 Member Data Documentation	100
17.48 EmbTime_ Struct Reference	101
17.48.1 Member Data Documentation	101
17.49 EmbVector_ Struct Reference	102
17.49.1 Detailed Description	102
17.49.2 Member Data Documentation	102
17.50 Geometry Class Reference	102
17.50.1 Detailed Description	107
17.50.2 Member Enumeration Documentation	107
17.50.3 Constructor & Destructor Documentation	107
17.50.4 Member Function Documentation	109
17.50.5 Member Data Documentation	128

17.51 hoop_padding Struct Reference	131
17.51.1 Member Data Documentation	131
17.52 Huffman Struct Reference	131
17.52.1 Member Data Documentation	132
17.53 ImageWidget Class Reference	132
17.53.1 Detailed Description	133
17.53.2 Constructor & Destructor Documentation	133
17.53.3 Member Function Documentation	133
17.53.4 Member Data Documentation	134
17.54 LayerManager Class Reference	134
17.54.1 Detailed Description	135
17.54.2 Constructor & Destructor Documentation	135
17.54.3 Member Function Documentation	135
17.54.4 Member Data Documentation	136
17.55 LSYSTEM Struct Reference	136
17.55.1 Member Data Documentation	136
17.56 MainWindow Class Reference	137
17.56.1 Detailed Description	140
17.56.2 Constructor & Destructor Documentation	141
17.56.3 Member Function Documentation	141
17.56.4 Member Data Documentation	154
17.57 MdiArea Class Reference	155
17.57.1 Constructor & Destructor Documentation	156
17.57.2 Member Function Documentation	157
17.57.3 Member Data Documentation	159
17.58 MdiWindow Class Reference	160
17.58.1 Constructor & Destructor Documentation	161
17.58.2 Member Function Documentation	162
17.58.3 Member Data Documentation	166
17.59 Node_ Struct Reference	168
17.59.1 Member Data Documentation	168
17.60 PreviewDialog Class Reference	169
17.60.1 Constructor & Destructor Documentation	169
17.60.2 Member Data Documentation	169
17.61 PropertyEditor Class Reference	170
17.61.1 Constructor & Destructor Documentation	171
17.61.2 Member Function Documentation	171
17.61.3 Member Data Documentation	174
17.62 SaveObject Class Reference	176
17.62.1 Constructor & Destructor Documentation	177
17.62.2 Member Function Documentation	178
17.62.3 Member Data Documentation	186

17.63 SelectBox Class Reference	186
17.63.1 Constructor & Destructor Documentation	187
17.63.2 Member Function Documentation	187
17.63.3 Member Data Documentation	188
17.64 Settings_Dialog Class Reference	189
17.64.1 Constructor & Destructor Documentation	190
17.64.2 Member Function Documentation	191
17.64.3 Member Data Documentation	196
17.65 StatusBar Class Reference	197
17.65.1 Detailed Description	197
17.65.2 Constructor & Destructor Documentation	197
17.65.3 Member Function Documentation	197
17.65.4 Member Data Documentation	198
17.66 StxThread_ Struct Reference	198
17.66.1 Member Data Documentation	198
17.67 SubDescriptor_ Struct Reference	198
17.67.1 Member Data Documentation	199
17.68 SvgAttribute_ Struct Reference	199
17.68.1 Member Data Documentation	199
17.69 thread_color_ Struct Reference	199
17.69.1 Member Data Documentation	200
17.70 ThredExtension_ Struct Reference	200
17.70.1 Member Data Documentation	200
17.71 ThredHeader_ Struct Reference	201
17.71.1 Member Data Documentation	201
17.72 UndoableCommand Class Reference	201
17.72.1 Constructor & Destructor Documentation	202
17.72.2 Member Function Documentation	203
17.72.3 Member Data Documentation	203
17.73 UndoEditor Class Reference	204
17.73.1 Constructor & Destructor Documentation	205
17.73.2 Member Function Documentation	205
17.73.3 Member Data Documentation	206
17.74 View Class Reference	206
17.74.1 Constructor & Destructor Documentation	209
17.74.2 Member Function Documentation	209
17.74.3 Member Data Documentation	216
17.75 VipHeader_ Struct Reference	219
17.75.1 Member Data Documentation	220
18 File Documentation	221
18.1 CODE_OF_CONDUCT.md File Reference	221

18.2 embroidermodder2/cmdprompt.cpp File Reference	221
18.2.1 Detailed Description	221
18.3 embroidermodder2/em2_dev_script.py File Reference	221
18.4 embroidermodder2/embdetails-dialog.cpp File Reference	221
18.5 embroidermodder2/embroidermodder.cpp File Reference	221
18.5.1 Function Documentation	221
18.5.2 Variable Documentation	222
18.6 embroidermodder2/embroidermodder.h File Reference	222
18.6.1 Detailed Description	226
18.6.2 Macro Definition Documentation	226
18.6.3 Typedef Documentation	226
18.6.4 Enumeration Type Documentation	227
18.6.5 Function Documentation	228
18.6.6 Variable Documentation	236
18.7 embroidermodder.h	238
18.8 embroidermodder2/imagewidget.cpp File Reference	256
18.9 embroidermodder2/interface.cpp File Reference	256
18.9.1 Detailed Description	257
18.9.2 Function Documentation	257
18.10 embroidermodder2/layer-manager.cpp File Reference	263
18.10.1 Detailed Description	263
18.11 embroidermodder2/mainwindow-menus.cpp File Reference	263
18.11.1 Function Documentation	263
18.12 embroidermodder2/mainwindow-toolbars.cpp File Reference	264
18.13 embroidermodder2/mainwindow.cpp File Reference	264
18.13.1 Enumeration Type Documentation	269
18.13.2 Function Documentation	271
18.13.3 Variable Documentation	292
18.14 embroidermodder2/mdiarea.cpp File Reference	294
18.15 embroidermodder2/mdiwindow.cpp File Reference	294
18.15.1 Function Documentation	294
18.16 embroidermodder2/objects.cpp File Reference	294
18.16.1 Function Documentation	295
18.17 embroidermodder2/preview-dialog.cpp File Reference	295
18.18 embroidermodder2/property-editor.cpp File Reference	295
18.18.1 Function Documentation	296
18.18.2 Variable Documentation	296
18.19 embroidermodder2/README.md File Reference	297
18.20 embroidermodder2/selectbox.cpp File Reference	297
18.21 embroidermodder2/settings-dialog.cpp File Reference	297
18.21.1 Function Documentation	297
18.21.2 Variable Documentation	298

18.22 embroidermodder2/statusbar.cpp File Reference	299
18.23 embroidermodder2/undo-commands.cpp File Reference	299
18.24 embroidermodder2/undo-editor.cpp File Reference	299
18.24.1 Detailed Description	300
18.25 embroidermodder2/view.cpp File Reference	300
18.25.1 Detailed Description	300
18.25.2 Function Documentation	300
18.26 extern/libembroidery/src/array.c File Reference	300
18.26.1 Function Documentation	301
18.27 extern/libembroidery/src/compress.c File Reference	302
18.27.1 Detailed Description	303
18.27.2 Function Documentation	303
18.27.3 Variable Documentation	304
18.28 extern/libembroidery/src/embroidery.h File Reference	304
18.28.1 Macro Definition Documentation	311
18.28.2 Typedef Documentation	319
18.28.3 Function Documentation	321
18.28.4 Variable Documentation	336
18.29 embroidery.h	337
18.30 extern/libembroidery/src/embroidery_internal.h File Reference	344
18.30.1 Macro Definition Documentation	352
18.30.2 Typedef Documentation	360
18.30.3 Enumeration Type Documentation	361
18.30.4 Function Documentation	362
18.30.5 Variable Documentation	384
18.31 embroidery_internal.h	384
18.32 extern/libembroidery/src/encoding.c File Reference	391
18.32.1 Detailed Description	392
18.32.2 Function Documentation	392
18.33 extern/libembroidery/src/fill.c File Reference	394
18.33.1 Function Documentation	395
18.33.2 Variable Documentation	398
18.34 extern/libembroidery/src/formats.c File Reference	399
18.34.1 Function Documentation	400
18.34.2 Variable Documentation	402
18.35 extern/libembroidery/src/formats/format_100.c File Reference	403
18.35.1 Detailed Description	403
18.35.2 Function Documentation	403
18.36 extern/libembroidery/src/formats/format_10o.c File Reference	403
18.36.1 Detailed Description	403
18.36.2 Function Documentation	404
18.37 extern/libembroidery/src/formats/format_art.c File Reference	404

18.37.1 Detailed Description	404
18.37.2 Function Documentation	404
18.38 extern/libembroidery/src/formats/format_bmc.c File Reference	404
18.38.1 Detailed Description	405
18.38.2 Function Documentation	405
18.39 extern/libembroidery/src/formats/format_bro.c File Reference	405
18.39.1 Detailed Description	405
18.39.2 Function Documentation	405
18.40 extern/libembroidery/src/formats/format_cnd.c File Reference	405
18.40.1 Detailed Description	406
18.40.2 Function Documentation	406
18.41 extern/libembroidery/src/formats/format_col.c File Reference	406
18.41.1 Detailed Description	406
18.41.2 Function Documentation	407
18.42 extern/libembroidery/src/formats/format_csd.c File Reference	407
18.42.1 Detailed Description	407
18.42.2 Macro Definition Documentation	407
18.42.3 Function Documentation	407
18.42.4 Variable Documentation	408
18.43 extern/libembroidery/src/formats/format_csv.c File Reference	408
18.43.1 Detailed Description	409
18.43.2 Function Documentation	409
18.44 extern/libembroidery/src/formats/format_dat.c File Reference	409
18.44.1 Function Documentation	409
18.45 extern/libembroidery/src/formats/format_dem.c File Reference	410
18.45.1 Detailed Description	410
18.45.2 Function Documentation	410
18.46 extern/libembroidery/src/formats/format_dsb.c File Reference	410
18.46.1 Detailed Description	410
18.46.2 Function Documentation	411
18.47 extern/libembroidery/src/formats/format_dst.c File Reference	411
18.47.1 Detailed Description	411
18.47.2 Macro Definition Documentation	412
18.47.3 Function Documentation	412
18.48 extern/libembroidery/src/formats/format_dsz.c File Reference	413
18.48.1 Function Documentation	413
18.49 extern/libembroidery/src/formats/format_dxf.c File Reference	413
18.49.1 Function Documentation	414
18.50 extern/libembroidery/src/formats/format_edr.c File Reference	414
18.50.1 Function Documentation	414
18.51 extern/libembroidery/src/formats/format_emd.c File Reference	415
18.51.1 Detailed Description	415

18.51.2 Function Documentation	415
18.52 extern/libembroidery/src/formats/format_exp.c File Reference	415
18.52.1 Function Documentation	415
18.53 extern/libembroidery/src/formats/format_exy.c File Reference	416
18.53.1 Function Documentation	416
18.54 extern/libembroidery/src/formats/format_ey.c File Reference	416
18.54.1 Function Documentation	416
18.55 extern/libembroidery/src/formats/format_fxy.c File Reference	417
18.55.1 Function Documentation	417
18.56 extern/libembroidery/src/formats/format_gc.c File Reference	417
18.56.1 Function Documentation	417
18.57 extern/libembroidery/src/formats/format_gnc.c File Reference	418
18.57.1 Function Documentation	418
18.58 extern/libembroidery/src/formats/format_gt.c File Reference	418
18.58.1 Function Documentation	418
18.59 extern/libembroidery/src/formats/format_hus.c File Reference	419
18.59.1 Function Documentation	419
18.60 extern/libembroidery/src/formats/format_inb.c File Reference	420
18.60.1 Function Documentation	420
18.61 extern/libembroidery/src/formats/format_inf.c File Reference	420
18.61.1 Function Documentation	420
18.62 extern/libembroidery/src/formats/format_jef.c File Reference	421
18.62.1 Function Documentation	421
18.63 extern/libembroidery/src/formats/format_ksm.c File Reference	422
18.63.1 Function Documentation	422
18.64 extern/libembroidery/src/formats/format_max.c File Reference	422
18.64.1 Function Documentation	423
18.64.2 Variable Documentation	423
18.65 extern/libembroidery/src/formats/format_mit.c File Reference	423
18.65.1 Function Documentation	423
18.66 extern/libembroidery/src/formats/format_new.c File Reference	424
18.66.1 Function Documentation	424
18.67 extern/libembroidery/src/formats/format_ofm.c File Reference	424
18.67.1 Function Documentation	425
18.68 extern/libembroidery/src/formats/format_pcd.c File Reference	425
18.68.1 Function Documentation	426
18.69 extern/libembroidery/src/formats/format_pcm.c File Reference	426
18.69.1 Function Documentation	426
18.70 extern/libembroidery/src/formats/format_pcq.c File Reference	426
18.70.1 Function Documentation	427
18.71 extern/libembroidery/src/formats/format_pcs.c File Reference	427
18.71.1 Function Documentation	427

18.72	extern/libembroidery/src/formats/format_pec.c File Reference	427
18.72.1	Function Documentation	428
18.73	extern/libembroidery/src/formats/format_pel.c File Reference	429
18.73.1	Function Documentation	429
18.74	extern/libembroidery/src/formats/format_pem.c File Reference	429
18.74.1	Function Documentation	429
18.75	extern/libembroidery/src/formats/format_pes.c File Reference	430
18.75.1	Function Documentation	430
18.75.2	Variable Documentation	432
18.76	extern/libembroidery/src/formats/format_phb.c File Reference	432
18.76.1	Function Documentation	432
18.77	extern/libembroidery/src/formats/format_phc.c File Reference	433
18.77.1	Function Documentation	433
18.78	extern/libembroidery/src/formats/format_plt.c File Reference	433
18.78.1	Function Documentation	433
18.79	extern/libembroidery/src/formats/format_rgb.c File Reference	434
18.79.1	Function Documentation	434
18.80	extern/libembroidery/src/formats/format_sew.c File Reference	434
18.80.1	Function Documentation	434
18.81	extern/libembroidery/src/formats/format_shv.c File Reference	435
18.81.1	Function Documentation	435
18.82	extern/libembroidery/src/formats/format_sst.c File Reference	435
18.82.1	Function Documentation	435
18.83	extern/libembroidery/src/formats/format_stx.c File Reference	436
18.83.1	Function Documentation	436
18.84	extern/libembroidery/src/formats/format_svg.c File Reference	436
18.84.1	Function Documentation	437
18.84.2	Variable Documentation	437
18.85	extern/libembroidery/src/formats/format_t01.c File Reference	438
18.85.1	Function Documentation	438
18.86	extern/libembroidery/src/formats/format_t09.c File Reference	438
18.86.1	Function Documentation	438
18.87	extern/libembroidery/src/formats/format_tap.c File Reference	439
18.87.1	Function Documentation	439
18.88	extern/libembroidery/src/formats/format_thr.c File Reference	439
18.88.1	Function Documentation	439
18.89	extern/libembroidery/src/formats/format_txt.c File Reference	440
18.89.1	Function Documentation	440
18.90	extern/libembroidery/src/formats/format_u00.c File Reference	440
18.90.1	Function Documentation	440
18.91	extern/libembroidery/src/formats/format_u01.c File Reference	441
18.91.1	Function Documentation	441

18.92	extern/libembroidery/src/formats/format_vip.c File Reference	441
18.92.1	Function Documentation	442
18.92.2	Variable Documentation	442
18.93	extern/libembroidery/src/formats/format_vp3.c File Reference	443
18.93.1	Function Documentation	443
18.94	extern/libembroidery/src/formats/format_xxx.c File Reference	444
18.94.1	Function Documentation	444
18.95	extern/libembroidery/src/formats/format_zsk.c File Reference	445
18.95.1	Detailed Description	445
18.95.2	Function Documentation	445
18.96	extern/libembroidery/src/geometry.c File Reference	445
18.96.1	Function Documentation	446
18.97	extern/libembroidery/src/geometry/arc.c File Reference	447
18.97.1	Function Documentation	447
18.98	extern/libembroidery/src/geometry/circle.c File Reference	450
18.98.1	Function Documentation	451
18.99	extern/libembroidery/src/geometry/ellipse.c File Reference	451
18.99.1	Function Documentation	452
18.100	extern/libembroidery/src/geometry/functions.c File Reference	453
18.100.1	Function Documentation	453
18.101	extern/libembroidery/src/geometry/line.c File Reference	454
18.101.1	Function Documentation	454
18.102	extern/libembroidery/src/geometry/path.c File Reference	454
18.103	extern/libembroidery/src/geometry/polygon.c File Reference	454
18.104	extern/libembroidery/src/geometry/polyline.c File Reference	454
18.105	extern/libembroidery/src/geometry/rect.c File Reference	455
18.105.1	Function Documentation	455
18.106	extern/libembroidery/src/geometry/text.c File Reference	455
18.106.1	Function Documentation	455
18.107	extern/libembroidery/src/geometry/vector.c File Reference	457
18.107.1	Function Documentation	457
18.108	extern/libembroidery/src/image.c File Reference	459
18.108.1	Detailed Description	459
18.108.2	Function Documentation	459
18.109	extern/libembroidery/src/main.c File Reference	460
18.109.1	Macro Definition Documentation	462
18.109.2	Function Documentation	465
18.109.3	Variable Documentation	471
18.110	extern/libembroidery/src/pattern.c File Reference	471
18.110.1	Detailed Description	472
18.110.2	Function Documentation	472
18.111	extern/libembroidery/src/thread-color.c File Reference	476

18.111.1 Function Documentation	477
18.111.2 Variable Documentation	477
18.112 privacy_policy.md File Reference	478
Bibliography	479
Index	481

1 Overview

Version

2.0.0-alpha

Author

The Embroidermodder Team

(UNDER MAJOR RESTRUCTURING, PLEASE WAIT FOR VERSION 2)

<http://www.libembroidery.org>

Embroidermodder is a free machine embroidery application. The newest version, Embroidermodder 2 can:

- edit and create embroidery designs
- estimate the amount of thread and machine time needed to stitch a design
- convert embroidery files to a variety of formats
- upscale or downscale designs
- run on Windows, Mac and Linux

Embroidermodder 2 is very much a work in progress since we're doing a ground up rewrite to an interface in C using the GUI toolkit SDL2. The reasoning for this is detailed in the issues tab.

For a more in-depth look at what we are developing read our [website](#) which includes these docs as well as the up-to date printer-friendly versions. These discuss recent changes, plans and has user and developer guides for all the Embroidermodder projects.

To see what we're focussing on right now, see the [Open Collective News](#).

1.0.1 License

The source code is under the terms of the zlib license: see `LICENSE.md` in the source code directory.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

2 About

2.1 The Embroidermodder Project and Team

The *Embroidermodder 2* project is a collection of small software utilities for manipulating, converting and creating embroidery files in all major embroidery machine formats. The program *Embroidermodder 2* itself is a larger graphical user interface (GUI) which is at the heart of the project.

The tools and associated documents are:

- This website (www.libembroidery.org), which is maintained [here](#).
- [The manual](#) covering all these projects.
- The GUI (*embroidermodder*), maintained [here](#).
- The core library of low-level functions: [libembroidery](#).
- The CLI *embroider* which is part of [libembroidery](#).
- Mobile embroidery format viewers and tools ([EmbroideryMobile](#))).
- Specs for an open hardware embroidery machine called Embroiderbot (not started yet) which is also part of [libembroidery](#).

They all tools to make the standard user experience of working with an embroidery machine better without expensive software which is locked to specific manufacturers and formats. But ultimately we hope that the core *Embroidermodder 2* is a practical, ever-present tool in larger workshops, small cottage industry workshops and personal hobbyist's bedrooms.

Embroidermodder 2 is licensed under the zlib license and we aim to keep all of our tools open source and free of charge. If you would like to support the project check out our [Open Collective](#) group. If you would like to help, please join us on GitHub. This document is written as developer training as well helping new users (see the last sections) so this is the place to learn how to start changing the code.

The Embroidermodder Team is the collection of people who've submitted patches, artwork and documentation to our three projects. The team was established by Jonathan Greig and Josh Varga. The full list is actively maintained below.

2.1.1 "Core Development Team"

Embroidermodder 2:

- [Jonathan Greig](#)
- [Josh Varga](#)
- [Robin Swift](#)

Embroidermodder 1:

- [Josh Varga](#)
- [Mark Pontius](#)

2.2 for Embroidermodder 2, libembroidery and all other related code

If you have contributed and wish to be added to this list, alter the [README on Embroidermodder github page](#) and we'll copy it to the libembroidery source code since that is credited to "The Embroidermodder Team".

2.2.1 Embroidermodder 1

The Embroidermodder Team is also inspired by the original Embroidermodder that was built by Mark Pontius and the same Josh Varga on SourceForge which unfortunately appears to have died from linkrot. We may create a distribution on here to be the official "legacy" Embroidermodder code but likely in a separate repository because it's GNU GPL v3 and this code is written to be zlib (that is, permissive licensed) all the way down.

One reason why this is useful is that the rewrite by Jonathan Greig, John Varga and Robin Swift for Embroidermodder 2 should have no regressions: no features present in v1 should be missing in v2.

2.2.2 Features

Embroidermodder 2 has many advanced features that enable you to create awesome designs quicker, tweak existing designs to perfection, and can be fully customized to fit your workflow.

A summary of these features:

- Cross Platform
- Realistic rendering
- Various grid types and auto-adjusting rulers
- Many measurement tools
- Add text to any design
- Supports many formats
- Batch Conversion
- Scripting API

2.2.2.1 Cross Platform If you use multiple operating systems, it's important to choose software that works on all of them.

Embroidermodder 2 runs on Windows, Linux and Mac OS X. Let's not forget the [Raspberry Pi](#).

2.2.2.2 Realistic Rendering It is important to be able to visualize what a design will look like when stitched and our pseudo "3D" realistic rendering helps achieve this.

Realistic rendering sample #1:

Realistic rendering sample #2:

Realistic rendering sample #3:

Various grid types and auto-adjusting rulers

Making use of the automatically adjusting ruler in conjunction with the grid will ensure your design is properly sized and fits within your embroidery hoop area.

Use rectangular, circular or isometric grids to construct your masterpiece!

Multiple grids and rulers in action:

2.2.2.3 Realistic Rendering Taking measurements is a critical part of creating great designs. Whether you are designing mission critical embroidered space suits for NASA or some other far out design for your next meet-up, you will have precise measurement tools at your command to make it happen. You can locate individual points or find distances between any 2 points anywhere in the design!

Take quick and accurate measurements:

2.2.2.4 Add text to any design Need to make company apparel for all of your employees with individual names on them? No sweat. Just simply add text to your existing design or create one from scratch, quickly and easily. Didn't get it the right size or made a typo? No problem. Just select the text and update it with the property editor.

Add text and adjust its properties quickly:

2.2.2.5 Supports many formats Embroidery machines all accept different formats. There are so many formats available that it can sometimes be confusing whether a design will work with your machine.

Embroidermodder 2 supports a wide variety of embroidery formats as well as several vector formats, such as SVG and DXF. This allows you to worry less about which designs you can use.

2.2.2.6 Batch Conversion Need to send a client several different formats? Just use libembroidery-convert, our command line utility which supports batch file conversion.

There are a multitude of formats to choose from:

2.2.2.7 Scripting API If you've got programming skills and there is a feature that isn't currently available that you absolutely cannot live without, you have the capability to create your own custom commands for Embroidermodder 2. We provide an QtScript API which exposes various application functionality so that it is possible to extend the application without requiring a new release. If you have created a command that you think is worth including in the next release, just [contact us](#) and we will review it for functionality, bugs, and finally inclusion.

An Embroidermodder 2 command excerpt:

2.2.3 "Build and Install"

Assuming you already have the SDL2 libraries you can proceed to using the fast build, which assumes you want to build and test locally.

The fast build should be:

```
bash build.sh
```

or, on Windows:

```
.\build.bat
```

Then run using the `run.bat` or `run.sh` scripts in the `build/` directory.

Otherwise, follow the instructions below.

If you plan to install the dev version to your system (we recommend you wait for the official installers and beta release first) then use the CMake build instead.

2.2.3.1 Install on Desktop We recommend that if you want to install the development version you use the CMake build. Like this:

```
git submodule init
git submodule update

mkdir build
cd build
cmake ..
cmake --build .
sudo cmake --install .
```

These lines are written into the file:

```
./build_install.sh
```

On Windows use the next section.

2.2.4 History

Embroidermodder 1 was started by Mark Pontius in 2004 while staying up all night with his son in his first couple months. When Mark returned to his day job, he lacked the time to continue the project. Mark made the decision to focus on his family and work, and in 2005, Mark gave full control of the project to Josh Varga so that Embroidermodder could continue its growth.

Embroidermodder 2 was conceived in mid 2011 when Jonathan Greig and Josh Varga discussed the possibility of making a cross-platform version. It is currently in active development and will run on GNU/Linux, Mac OS X, Microsoft Windows and Raspberry Pi.

All [Embroidermodder downloads](#) are hosted on SourceForge.

The [source code for Embroidermodder 1](#) has always been hosted on Sourceforge.

The [source code for Embroidermodder 2](#) was moved to GitHub on July 18, 2013.

The [website for Embroidermodder](#) was moved to GitHub on September 9, 2013.

2.3 Contact us

For general questions email: [embroidermodder at gmail.com](mailto:embroidermodder@gmail.com)

To request a new feature [open an issue on the main Embroidermodder GitHub repository](#). We'll move it to the correct repository.

3 Downloads

3.1 Alpha Build

This is a highly experimental build: we recommend users wait for the beta release when the basic features are functional.

Visit our [GitHub Releases page](#) for the current build. Unfortunately, earlier builds went down with the Sourceforge page we hosted them on.

4 Changelog

5 Ideas

Stuff that is now supposed to be generated by Doxygen:

Todo Bibliography style to plainnat.

Todo Serif font for printed docs.

Todo US letter paper version of printed docs.

6 Formats

6.1 Overview

6.1.1 Read/Write Support Levels

The table of read/write format support levels uses the status levels described here:

Status Label	Description
<code>rw-none</code>	Either the format produces no output, reporting an error. Or it produces a Tajima dst file as an alternative.
<code>rw-poor</code>	A file somewhat similar to our examples is produced. We don't know how well it runs on machines in practice as we don't have any user reports or personal tests.
<code>rw-basic</code>	Simple files in this format run well on machines that use this format.
<code>rw-standard</code>	Files with non-standard features work on machines and we have good documentation on the format.
<code>rw-reliable</code>	All known features don't cause crashes. Almost all work as expected.
<code>rw-complete</code>	All known features of the format work on machines that use this format. Translations from and to this format preserve all features present in both.

These can be split into `r-basic w-none`, for example, if they don't match.

So all formats can, in principle, have good read and good write support, because it's defined in relation to files that we have described the formats for.

Status Label	Description
<code>test-none</code>	No tests have been written to test the specifics of the format.
<code>test-basic</code>	Stitch Lists and/or colors have read/write tests.
<code>test-thorough</code>	All features of that format has at least one test.
<code>test-fuzz</code>	Can test the format for uses of features that we haven't thought of by feeding in nonsense that is designed to push possibly dangerous weaknesses to reveal themselves.
<code>test-complete</code>	Both thorough and fuzz testing is covered.

6.1.1.1 Test Support Levels So all formats can, in principle, have complete testing support, because it's defined in relation to files that we have described the formats for.

Status Label	Description
doc-none	We haven't researched this beyond finding example files.
doc-basic	We have a rough sketch of the size and contents of the header if there is one. We know the basic stitch encoding (if there is one), but not necessarily all stitch features.
doc-standard	We know some good sources and/or have tested all the features that appear to exist. They mostly work the way we have described.
doc-good	All features that were described somewhere have been covered here or we have thoroughly tested our ideas against other softwares and hardwares and they work as expected.
doc-complete	There is a known official description and our description covers all the same features.

6.1.1.2 Documentation Support Levels Not all formats can have complete documentation because it's based on what information is publically available. So the total score is reported in the table below based on what level we think is available.

6.1.1.3 Overall Support Since the overall support level is the combination of these 4 factors, but rather than summing up their values it's an issue of the minimum support of the 4.

Status Label	Description
read-only	If write support is none and read support is not none.
write-only	If read support is none and write support is not none.
unstable	If both read and write support are not none but testing or documentation is none.
basic	If all ratings are better than none.
reliable	If all ratings are better than basic.
complete	If all ratings could not reasonably be better (for example any improvements rely on information that we may never have access to). This is the only status that can be revoked, since if the format changes or new documentation is released it is no longer complete.
experimental	For all other scenarios.

6.1.2 Table of Format Support Levels

Overview of documentation support by format.

Format	Ratings	Score
Toyota Embroidery Format (.100)	rw-basic doc-none test-none	unstable
Toyota Embroidery Format (.10o)	rw-basic doc-none test-none	unstable
Bernina Embroidery Format (.art)	rw-none doc-none test-none	experimental
Bitmap Cache Embroidery Format (.bmc)	r-basic w-none doc-none test-none	unstable
Bits and Volts Embroidery Format (.bro)	rw-none doc-none test-none	experimental
Melco Embroidery Format (.cnd)	rw-none doc-none test-none	experimental
Embroidery Thread Color Format (.col)	rw-basic doc-none test-none	experimental
Singer Embroidery Format (.csd)	rw-none doc-none test-none	experimental
Comma Separated Values (.csv)	rw-none doc-none test-none	experimental
Barudan Embroidery Format (.dat)	rw-none doc-none test-none	experimental
Melco Embroidery Format (.dem)	rw-none doc-none test-none	experimental
Barudan Embroidery Format (.dsb)	rw-none doc-none test-none	experimental

Format	Ratings	Score
Tajima Embroidery Format (.dst)	rw-none doc-none test-none	experimental
ZSK USA Embroidery Format (.dsz)	rw-none doc-none test-none	experimental
Drawing Exchange Format (.dxf)	rw-none doc-none test-none	experimental
Embird Embroidery Format (.edr)	rw-none doc-none test-none	experimental
Elna Embroidery Format (.emd)	rw-none doc-none test-none	experimental
Melco Embroidery Format (.exp)	rw-none doc-none test-none	experimental
Eltac Embroidery Format (.exy)	rw-none doc-none test-none	experimental
Sierra Expanded Embroidery Format (.eys)	rw-none doc-none test-none	experimental
Fortron Embroidery Format (.fxy)	rw-none doc-none test-none	experimental
Smoothie G-Code Embroidery Format (.gc)	rw-none doc-none test-none	experimental
Great Notions Embroidery Format (.gnc)	rw-none doc-none test-none	experimental
Gold Thread Embroidery Format (.gt)	rw-none doc-none test-none	experimental
Husqvarna Viking Embroidery Format (.hus)	rw-none doc-none test-none	experimental
Inbro Embroidery Format (.inb)	rw-none doc-none test-none	experimental
Embroidery Color Format (.inf)	rw-none doc-none test-none	experimental
Janome Embroidery Format (.jef)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.ksm)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.max)	rw-none doc-none test-none	experimental
Mitsubishi Embroidery Format (.mit)	rw-none doc-none test-none	experimental
Ameco Embroidery Format (.new)	rw-none doc-none test-none	experimental
Melco Embroidery Format (.ofm)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.pcd)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.pcm)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.pcq)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.pcs)	rw-none doc-none test-none	experimental
Brother Embroidery Format (.pec)	rw-none doc-none test-none	experimental
Brother Embroidery Format (.pel)	rw-none doc-none test-none	experimental
Brother Embroidery Format (.pem)	rw-none doc-none test-none	experimental
Brother Embroidery Format (.pes)	rw-none doc-none test-none	experimental
Brother Embroidery Format (.phb)	rw-none doc-none test-none	experimental
Brother Embroidery Format (.phc)	rw-none doc-none test-none	experimental
AutoCAD Embroidery Format (.plt)	rw-none doc-none test-none	experimental
RGB Embroidery Format (.rgb)	rw-none doc-none test-none	experimental
Janome Embroidery Format (.sew)	rw-none doc-none test-none	experimental
Husqvarna Viking Embroidery Format (.shv)	rw-none doc-none test-none	experimental
Sunstar Embroidery Format (.sst)	rw-none doc-none test-none	experimental
Data Stitch Embroidery Format (.stx)	rw-none doc-none test-none	experimental
Scalable Vector Graphics (.svg)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.t01)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.t09)	rw-none doc-none test-none	experimental
Happy Embroidery Format (.tap)	rw-none doc-none test-none	experimental
ThredWorks Embroidery Format (.thr)	rw-none doc-none test-none	experimental
Text File (.txt)	rw-none doc-none test-none	experimental
Barudan Embroidery Format (.u00)	rw-none doc-none test-none	experimental
Barudan Embroidery Format (.u01)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.vip)	rw-none doc-none test-none	experimental
Pfaff Embroidery Format (.vp3)	rw-none doc-none test-none	experimental
Singer Embroidery Format (.xxx)	rw-none doc-none test-none	experimental

Format	Ratings	Score
ZSK USA Embroidery Format (.zsk)	rw-none doc-none test-none	experimental

6.1.3 Format Support

| FORMAT | READ | WRITE | NOTES | |-----|-----|-----|-----| | 10o | YES | | read (need to fix external color loading) (maybe find out what ctrl | code flags of 0x10, 0x08, 0x04, and 0x02 mean) | | 100 | | | none (4 byte codes) 61 00 10 09 (type, type2, x, y ?) x | y (signed char) | | 100 | | | none (4 byte codes) 61 00 10 09 (type, type2, x, y ?) x & y (signed char) | | art | | | none | | bro | YES | | read (complete)(maybe figure out detail of header) | | cnd | | | none | | col | | | (color file no design) read(final) write(final) | | csd | YES | | read (complete) | | dat | | | read () | | dem | | | none (looks like just encrypted cnd) | | dsb | YES | | read (unknown how well) (stitch data looks same as 10o) | | dst | YES | | read (complete) / write(unknown) | | dsz | YES | | read (unknown) | | dxf | | | read (Port to C. needs refactored) | | edr | | | read (C version is broken) / write (complete) | | emd | | | read (unknown) | | exp | YES | | read (unknown) / write(unknown) | | exy | YES | | read (need to fix external color loading) | | fxy | YES | | read (need to fix external color loading) | | gnc | | | none | | gt | | | read (need to fix external color loading) | | hus | YES | | read (unknown) / write (C version is broken) | | inb | YES | | read (buggy?) | | jef | YES | | write (need to fix the offsets when it is moving to another spot) | | ksm | YES | | read (unknown) / write (unknown) | | pcd | | | pcm | | | pcq | | | read (Port to C) | | pcs | BUGGY | | read (buggy / colors are not correct / after reading, writing any other format is messed up) | | pec | | | read / write (without embedded images, sometimes overlooks some stitches leaving a gap) | | pel | | | none | | pem | | | none | | pes | YES | | phb | | | phc | | | rgb | | | sew | YES | | shv | | | read (C version is broken) | | sst | | | none | | svg | YES | | tap | YES | | read (unknown) | | u01 | | | vip | YES | | vp3 | YES | | xxx | YES | | zsk | | | read (complete) |

Todo Josh, Review this section and move any info still valid or needing work into TODO comments in the actual libembroidery code. Many items in this list are out of date and do not reflect the current status of libembroidery. When finished, delete this file.

- Test that all formats read data in correct scale (format details should match other programs)
- Add which formats to work with to preferences.
- Check for memory leaks
- Update all formats without color to check for edr or rgb files
- Fix issues with DST (VERY important that DST work well)

Todo Support for Singer FHE, CHE (Compucon) formats?

7 Geometry and Algorithms

7.1 To Do

Todo (Arduino) Fix emb-outline files

Todo (Arduino) Fix thread-color files

Todo (Arduino) Logging of Last Stitch Location to External USB Storage(commonly available and easily replaced) ...wait until TRE is available to avoid rework

Todo (Arduino) inotool.org - seems like the logical solution for Nightly/CI builds

Todo (Arduino) Smoothieboard experiments

Todo (testing) looping test that reads 10 times while running valgrind. See `\texttt{embPattern_loadExternalColorFile()}` Arduino leak note for more info.

7.1.1 Development

If you wish to develop with us you can chat via the contact email on the [website]\url{ <https://libembroidery.org>} or in the issues tab on the [github page]\url{ <https://github.com/Embroidermodder/Embroidermodder/issues>}. People have been polite and friendly in these conversations and I (Robin) have really enjoyed them. If we do have any arguments please note we have a [Code of Conduct] [CODE_OF_CONDUCT.md](#) so there is a consistent policy to enforce when dealing with these arguments.

The first thing you should try is building from source using the [build advice](build) above. Then read some of the [manual] \url{ https://libembroidery.org/embroidermodder_2.0_manual.pdf} to get the general layout of the source code and what we are currently planning.

7.1.2 Testing

To find unfixed errors run the tests by launching from the command line with:

```
$ embroidermodder --test
```

then dig through the output. It's currently not worth reporting the errors, since there are so many but if you can fix anything reported here you can submit a PR.

7.1.3 Contributing

7.1.3.1 Funding The easiest way to help is to fund development (see the Donate button above), since we can't afford to spend a lot of time developing and only have limited kit to test out libembroidery on.

7.1.3.2 Programming and Engineering Should you want to get into the code itself:

- Low level C developers are needed for the base library libembroidery.
- Low level assembly programmers are needed for translating some of libembroidery to Embroider↔Bot.
- Hardware Engineers to help design our own kitbashed embroidery machine EmbroiderBot, one of the original project aims in 2013.
- Scheme developers and C/SDL developers to help build the GUI.
- Scheme developers to help add designs for generating of custom stitch-filled emblems like the heart or dolfi. Note that this happens in Embroidermodder not libembroidery (which assumes that you already have a function available).

7.1.3.3 Writing We also need people familiar with the software and the general machine embroidery ecosystem to contribute to the [documentation](#).

We need researchers to find references for the documentation: colour tables, machine specifications etc. The history is murky and often very poorly maintained so if you know anything from working in the industry that you can share: it'd be appreciated!

7.1.4 Embroidermodder Project Coding Standards

A basic set of guidelines to use when submitting code.

Code structure is mre important than style, so first we advise you read "Design" and experimenting before getting into the specifics of code style.

7.1.4.1 Where Code Goes Anything that deals with the specifics of embroidery file formats, threads, rendering to images, embroidery machinery or command line interfaces should go in `libembroidery` not here.

7.1.4.2 Non-compiled Files Go

Todo Like most user interfaces Embroidermodder is mostly data, so here we will have a list describing where each CSV goes.

7.1.4.3 in which we break style on purpose Most style guides advise you to keep functions short. We make a few pointed exceptions to this where the overall health and functionality of the source code should benefit.

The `actuator` function will always be a mess and it should be: we're keeping the total source lines of code down by encoding all user action into a descrete sequence of strings that are all below `TEXTT{ _STRING_LENGTH }` in length. See the section on the actuator (TODO) describing why any other solution we could think here would mean more more code without a payoff in speed of execution or clarity.

7.1.5 Version Control

Being an open source project, developers can grab the latest code at any time and attempt to build it themselves. We try our best to ensure that it will build smoothly at any time, although occasionally we do break the build. In these instances, please provide a patch, pull request which fixes the issue or open an issue and notify us of the problem, as we may not be aware of it and we can build fine.

Try to group commits based on what they are related to: features/bugs/comments/graphics/commands/etc...

7.1.6 Donations

Creating software that interfaces with hardware is costly. A summary of some of the costs involved:

- Developer time for 2 core developers
- Computer equipment and parts
- Embroidery machinery
- Various electronics for kitbashing Embroiderbot
- Consumable materials (thread, fabric, stabilizer, etc...)

If you have found our software useful, please consider funding further development by donating to the project on Open Collective (`url{ https://opencollective.com/embroidermodder }`).

7.1.7 Embroidermodder Project Coding Standards

Rather than maintain our own standard for style, please defer to the Python's PEP 7 [\[3\]](#) for C style and emulating that in C++.

A basic set of guidelines to use when submitting code. Defer to the PEP7 standard with the following additions:

- All files and directories shall be lowercase and contain no spaces.
- Structs and class names should use LeadingCapitals.
- Enums and constants should be BLOCK_CAPITALS.
- Class members and functions without a parent class should be snake_case. With the exception of when one of the words is a "class" name from libembroidery in which case it has the middle capitals like this: `embArray_add`.
- Don't use exceptions.
- Don't use ternary operator (?:) in place of if/else.
- Don't repeat a variable name that already occurs in an outer scope.

7.1.7.1 Version Control Being an open source project, developers can grab the latest code at any time and attempt to build it themselves. We try our best to ensure that it will build smoothly at any time, although occasionally we do break the build. In these instances, please provide a patch, pull request which fixes the issue or open an issue and notify us of the problem, as we may not be aware of it and we can build fine.

Try to group commits based on what they are related to: features/bugs/comments/graphics/commands/etc...

7.1.7.2 Comments When writing code, sometimes there are items that we know can be improved, incomplete or need special clarification. In these cases, use the types of comments shown below. They are pretty standard and are highlighted by many editors to make reviewing code easier. We also use shell scripts to parse the code to find all of these occurrences so someone wanting to go on a bug hunt will be able to easily see which areas of the code need more love.

libembroidery and Embroidermodder are written in C and adheres to C89 standards. This means that any C99 or C++ comments will show up as errors when compiling with gcc. In any C code, you must use:

```
/* Use C Style Comments within code blocks.
 *
 * Use Doxygen style code blocks to place todo, bug, hack, warning,
 * and note items like this:
 *
 * \todo EXAMPLE: This code clearly needs more work or further review.
 *
 * \bug This code is definitely wrong. It needs fixed.
 *
 * \hack This code shouldn't be written this way or I don't
 * feel right about it. There may a better solution
 *
 * \warning Think twice (or more times) before changing this code.
 * I put this here for a good reason.
 *
 * \note This comment is much more important than lesser comments.
 */
```

7.1.8 Ideas

7.1.8.1 Why this document I've been trying to make this document indirectly through the Github issues page and the website we're building but I think a straightforward, plain-text file needs to be the ultimate backup for this. Then I can have a printout while I'm working on the project.

7.1.8.2 Qt and dependencies I'm switching to SDL2 (which is a whole other conversation) which means we can ship it with the source code package meaning only a basic build environment is necessary to build it.

7.1.8.3 Documentation Can we treat the website being a duplicate of the docs a non-starter? I'd be happier with tex/pdf only and (I know this is counter-intuitive) one per project.

7.1.8.4 Social Platform So... all the issues and project boards etc. being on Github is all well and good assuming that we have our own copies. But we don't if Github goes down or some other major player takes over the space and we have to move (again, since this started on SourceForge).

This file is a backup for that which is why I'm repeating myself between them.

7.1.8.5 Identify the meaning of these TODO items

- Saving CSV/SVG (rt) + CSV read/write UNKNOWN interpreted as COLOR bug #179
- Lego Mindstorms NXT/EV3 ports and/or commands

7.1.8.6 Progress Chart The chart of successful from-to conversions (previously a separate issue) is something that should appear in the README.

7.1.8.7 Standard The criteria for a good Pull Request from an outside developer has these properties, from most to least important:

- No regressions on testing.
- Add a feature, bug fix or documentation that is already agreed on through GitHub issues or some other way with a core developer.
- No GUI specific code should be in libembroidery, that's for Embroidermodder.
- Pedantic/ansi C unless there's a good reason to use another language.
- Meet the style above (i.e. [PEP 7](#), [Code Lay-out](#)). We'll just fix the style if the code's good and it's not a lot of work.
- `embroider` should be in POSIX style as a command line program.
- No dependencies that aren't "standard", i.e. use only the C Standard Library.

7.1.8.8 Image Fitting A currently unsolved problem in development that warrants further research is the scenario where a user wants to feed embroider an image that can then be .

7.1.8.9 To Place A *right-handed coordinate system* is one where up is positive and right is positive. Left-handed is up is positive, left is positive. Screens often use down is positive, right is positive, including the OpenGL standard so when switching between graphics formats and stitch formats we need to use a vertical flip (`embPattern_
flip`).

`0x20` is the space symbol, so when padding either 0 or space is preferred and in the case of space use the literal '`'`'.

7.1.8.10 To Do We currently need help with:

- Thorough descriptions of each embroidery format.
- Finding resources for each of the branded thread libraries (along with a full citation for documentation).
- Finding resources for each geometric algorithm used (along with a full citation for documentation).
- Completing the full `--full-test-suite` with no segfaults and at least a clear error message (for example `not implemented yet`).
- Identifying `best guesses` for filling in missing information when going from, say `.csv` to a late `.pes` version. What should the default be when the data doesn't clarify?
- Improving the written documentation.
- Funding, see the Sponsor button above. We can treat this as `work` and put far more hours in with broad support in small donations from people who want specific features.

Beyond this the development targets are categories sorted into:

- Basic Features
- Code quality and user friendliness
- embroider CLI
- Documentation
- GUI
- electronics development

7.1.8.11 Basic features

- Incorporate `#if 0`d parts of `libembroidery.c`.
- Interpret how to write formats that have a read mode from the source code and vice versa.
- Document the specifics of the file formats here for embroidery machine specific formats. Find websites and other sources that break down the binary formats we currently don't understand.
- Find more and better documentation of the structure of the headers for the formats we do understand.

7.1.8.12 Code quality and user friendliness

- Document all structs, macros and functions (will contribute directly on the web version).
- Incorporate experimental code, improve support for language bindings.
- Make stitch `x, y` into an `EmbVector`.

7.1.8.13 Documentation

- Create csv data files for thread tables.
- Convert tex to markdown, make tex an output of `build.bash`.
- Run `sloccount` on `extern/` and `.` (and `)` so we know the current scale of the project, aim to get this number low. Report the total as part of the documentation.
- Try to get as much of the source code that we maintain into C as possible so new developers don't need to learn multiple languages to have an effect. This bars the embedded parts of the code.

7.1.8.14 GUI

- Make EmbroideryMobile (Android) also backend to `libembroidery` with a Java wrapper.
- Make EmbroideryMobile (iOS) also backend to `libembroidery` with a Swift wrapper.
- Share some of the MobileViewer and iMobileViewer layout with the main EM2. Perhaps combine those 3 into the Embroidermodder repository so there are 4 repositories total.
- Convert layout data to JSON format and use cJSON for parsing.

7.1.9 Electronics development

- Currently experimenting with Fritzing8, upload netlists to embroiderbot when they can run simulations using the asm in `libembroidery`.
- Create a common assembly for data that is the same across chipsets `libembroidery_data_internal.s`.
- Make the defines part of `embroidery.h` all systems and the function list `c code only`. That way we can share some development between assembly and C versions.

7.1.10 Development

7.1.10.1 Contributing If you're interested in getting involved, here's some guidance for new developers. Currently The Embroidermodder Team is all hobbyists with an interest in making embroidery machines more open and user friendly. If you'd like to support us in some other way you can donate to our Open Collective page (click the Donate button) so we can spend more time working on the project.

All code written for `libembroidery` should be ANSI C89 compliant if it is C. Using other languages should only be used where necessary to support bindings.

7.1.10.2 Debug If you wish to help with development, run this debug script and send us the error log.

```
#!/bin/bash

rm -fr libembroidery-debug

git clone http://github.com/embroidermodder/libembroidery libembroidery-debug
cd libembroidery-debug

cmake -DCMAKE_BUILD_TYPE=DEBUG .
cmake --build . --config=DEBUG

valgrind ./embroider --full-test-suite
```

While we will attempt to maintain good results from this script as part of normal development it should be the first point of failure on any system we haven't tested or format we understand less.

7.1.10.3 Binary download We need a current `embroider` command line program download, so people can update without building.

7.2 Embroiderbot and Libembroidery on Embedded Systems

The libembroidery library is designed to support embedded environments, so it can be used in CNC applications.

7.2.1 Compatible Boards

We recommend using an Arduino greater specs. That being said, we have had success using an Arduino Uno R3 but this will likely require further optimization and other improvements to ensure continued compatibility with the Uno. See below for more information.

7.2.2 Arduino Considerations

There are two main concerns here: Flash Storage and SRAM.

libembroidery continually outgrows the 32KB of Flash storage on the Arduino Uno and every time this occurs, a decision has to be made as to what capabilities should be included or omitted. While reading files is the main focus on arduino, writing files may also play a bigger role in the future. Long term, it would be most practical to handle the inclusion or omission of any feature via a single configuration header file that the user can modify to suit their needs.

SRAM is in extremely limited supply and it will deplete quickly so any dynamic allocation should occur early during the setup phase of the sketch and sparingly or not at all later in the sketch. To help minimize SRAM consumption on Arduino and ensure libembroidery can be used in any way the sketch creator desires, it is required that any sketch using libembroidery must implement event handlers. See the `ino-event` source and header files for more information.

There is also an excellent article by Bill Earl on the Adafruit Learning System which covers these topics in more depth: <http://learn.adafruit.com/memories-of-an-arduino?view=all>.

7.2.3 Space

Since a stitch takes 3 bytes of storage and many patterns use more than 10k stitches, we can't assume that the pattern will fit in memory. Therefore we will need to buffer the current pattern on and off storage in small chunks. By the same reasoning, we can't load all of one struct before looping so we will need functions similar to `binaryReadInt16` for each struct.

This means the `EmbArray` approach won't work since we need to load each element and dynamic memory management is unnecessary because the arrays lie in storage.

Todo Replace `EmbArray` functions with `embPattern` load functions.

7.2.4 Tables

All thread tables and large text blocks are too big to compile directly into the source code. Instead we can package the library with a data packet that is compiled from an assembly program in raw format so the specific padding can be controlled.

In the user section above we will make it clear that this file needs to be loaded on the pattern USB/SD card or the program won't function.

Todo Start file with a list of offsets to data with a corresponding table to load into with macro constants for each label needed.

7.2.5 Current Pattern Memory Management

It will be simpler to make one file per EmbArray so we keep an EmbFile* and a length, so no malloc call is necessary. So there needs to be a consistent tmpfile naming scheme.

Todo For each pattern generate a random string of hexadecimal and append it to the filenames like `stitch↵List_A16F.dat`. Need to check for a file which indicates that this string has been used already.

7.2.6 Special Notes

Due to historical reasons and to remain compatible with the Arduino 1.0 IDE, this folder must be called "utility". Refer to the arduino build process for more info: <https://arduino.github.io/arduino-cli/0.19/sketch-build-process/>.

libembroidery relies on the Arduino SD library for reading files. See the ino-file source and header files for more information.

7.2.7 The Assembly Split

One problem to the problem of supporting both systems with abundant memory (such as a 2010s or later desktop) and with scarce memory (such as embedded systems) is that they don't share the same assembly language. To deal with this: there will be two equivalent software which are hand engineered to be similar but one will be in C and the other in the assembly dialects we support.

All assembly will be intended for embedded systems only, since a slightly smaller set of features will be supported. However, we will write a `x86` version since that can be tested.

That way the work that has been done to simplify the C code can be applied to the assembly versions.

7.3 The Embroider Command Line Program

Todo Move back to libembroidery now we have the combined docs build.

7.3.1 Embroider pipeline

Adjectives apply to every following noun so

```
embroider --satin 0.3,0.6 --thickness 2 --circle 10,20,5 \  
  --border 3 --disc 30,40,10 --arc 30,50,10,60 output.pes
```

Creates:

- a circle with properties: thickness 2, satin 0.3,0.6
- a disc with properties:
- an arc with properties:

in that order then writes them to the output file `output.pes`.

7.3.2 embroider CLI

- Make `-circle` flag to add a circle to the current pattern.
- Make `-rect` flag to add a rectangle to the current pattern.
- Make `-fill` flag to set the current satin fill algorithm for the current geometry. (for example `-fill crosses -circle 11,13,10` fills a circle with center 11mm, 13mm with radius 10mm with crosses).
- Make `-ellipse` flag to add to ellipse to the current pattern.
- Make `-bezier` flag to add a bezier curve to the current pattern.

8 GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <https://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

8.0.0.0.1 0. PREAMBLE The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

8.0.0.0.2 1. APPLICABILITY AND DEFINITIONS This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

8.0.0.0.3 2. VERBATIM COPYING You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

8.0.0.0.4 3. COPYING IN QUANTITY If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

8.0.0.0.5 4. MODIFICATIONS You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.

- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

8.0.0.0.6 5. COMBINING DOCUMENTS You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

8.0.0.0.7 6. COLLECTIONS OF DOCUMENTS You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

8.0.0.0.8 7. AGGREGATION WITH INDEPENDENT WORKS A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8.0.0.0.9 8. TRANSLATION Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

8.0.0.0.10 9. TERMINATION You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

8.0.0.0.11 10. FUTURE REVISIONS OF THIS LICENSE The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <https://www.gnu.org/licenses/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

8.0.0.0.12 11. RELICENSING "Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

8.0.0.1 ADDENDUM: How to use this License for your documents To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

9 Contributor Covenant Code of Conduct

9.1 Our Pledge

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

We pledge to act and interact in ways that contribute to an open, welcoming, diverse, inclusive, and healthy community.

9.2 Our Standards

Examples of behavior that contributes to a positive environment for our community include:

- Demonstrating empathy and kindness toward other people
- Being respectful of differing opinions, viewpoints, and experiences
- Giving and gracefully accepting constructive feedback
- Accepting responsibility and apologizing to those affected by our mistakes, and learning from the experience
- Focusing on what is best not just for us as individuals, but for the overall community

Examples of unacceptable behavior include:

- The use of sexualized language or imagery, and sexual attention or advances of any kind
- Trolling, insulting or derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or email address, without their explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

9.3 Enforcement Responsibilities

Community leaders are responsible for clarifying and enforcing our standards of acceptable behavior and will take appropriate and fair corrective action in response to any behavior that they deem inappropriate, threatening, offensive, or harmful.

Community leaders have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, and will communicate reasons for moderation decisions when appropriate.

9.4 Scope

This Code of Conduct applies within all community spaces, and also applies when an individual is officially representing the community in public spaces. Examples of representing our community include using an official e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event.

9.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported to the community leaders responsible for enforcement at embroidermodder@gmail.com. All complaints will be reviewed and investigated promptly and fairly.

All community leaders are obligated to respect the privacy and security of the reporter of any incident.

9.6 Enforcement Guidelines

Community leaders will follow these Community Impact Guidelines in determining the consequences for any action they deem in violation of this Code of Conduct:

9.6.1 1. Correction

Community Impact: Use of inappropriate language or other behavior deemed unprofessional or unwelcome in the community.

Consequence: A private, written warning from community leaders, providing clarity around the nature of the violation and an explanation of why the behavior was inappropriate. A public apology may be requested.

9.6.2 2. Warning

Community Impact: A violation through a single incident or series of actions.

Consequence: A warning with consequences for continued behavior. No interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, for a specified period of time. This includes avoiding interactions in community spaces as well as external channels like social media. Violating these terms may lead to a temporary or permanent ban.

9.6.3 3. Temporary Ban

Community Impact: A serious violation of community standards, including sustained inappropriate behavior.

Consequence: A temporary ban from any sort of interaction or public communication with the community for a specified period of time. No public or private interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, is allowed during this period. Violating these terms may lead to a permanent ban.

9.6.4 4. Permanent Ban

Community Impact: Demonstrating a pattern of violation of community standards, including sustained inappropriate behavior, harassment of an individual, or aggression toward or disparagement of classes of individuals.

Consequence: A permanent ban from any sort of public interaction within the community.

9.7 Attribution

This Code of Conduct is adapted from the [Contributor Covenant](https://www.contributor-covenant.org/version/2/0/code_of_conduct.html), version 2.0, available at https://www.contributor-covenant.org/version/2/0/code_of_conduct.html.

Community Impact Guidelines were inspired by [Mozilla's code of conduct enforcement ladder](#).

For answers to common questions about this code of conduct, see the FAQ at <https://www.contributor-covenant.org/faq>. Translations are available at <https://www.contributor-covenant.org/translations>.

10 Privacy Policy for Embroidery Viewer

Last updated December 15, 2021

Embroidermodder (“we” or “us” or “our”) respects the privacy of our users (“user” or “you”). This Privacy Policy explains how we collect, use, disclose, and safeguard your information when you visit our mobile application (the “Application”). Please read this Privacy Policy carefully. IF YOU DO NOT AGREE WITH THE TERMS OF THIS PRIVACY POLICY, PLEASE DO NOT ACCESS THE APPLICATION.

We reserve the right to make changes to this Privacy Policy at any time and for any reason. We will alert you about any changes by updating the “Last updated” date of this Privacy Policy. You are encouraged to periodically review this Privacy Policy to stay informed of updates. You will be deemed to have been made aware of, will be subject to, and will be deemed to have accepted the changes in any revised Privacy Policy by your continued use of the [Application](#) after the date such revised Privacy Policy is posted.

This Privacy Policy does not apply to the third-party online/mobile store from which you install the [Application](#) or make payments. We are not responsible for any of the data collected by any such third party.

We do not knowingly collect information from anyone other than what is already provided by the app store. If you become aware of any data we have collected, please contact us using the contact information provided below.

10.0.1 CONTACT US

If you have questions or comments about this Privacy Policy, please contact us at:

Embroidermodder@gmail.com

11 Todo List

Member [about_action](#) (String args)

these should all be static, since other files use the actuator to call them.

Member [bcf_directory](#)

possibly add a directory tree in the future.

Member [bcf_file_header](#)

CLSID should be a separate type.

Member [binaryWritelnt](#) (FILE *f, int data)

replace with embInt_read

Member [binaryWriteIntBE](#) (FILE *f, int data)

replace with embInt_read

Member [binaryWriteShort](#) (FILE *f, short data)

replace with embInt_read

Member [binaryWriteUInt](#) (FILE *f, unsigned int data)

replace with embInt_read

Member [binaryWriteUIntBE](#) (FILE *f, unsigned int data)

replace with embInt_read

Member [binaryWriteUShort](#) (FILE *f, unsigned short data)

replace with embInt_read

Member [binaryWriteUShortBE](#) (FILE *f, unsigned short data)

replace with embInt_read

Member [copy_trim](#) (char const *s)

decription

Member [day_vision_action](#) (String args)

Make day vision color settings.

Member [decode_t01_record](#) (unsigned char b[3], int *x, int *y, int *flags)

remove the unused return argument.

Member [embArc_print](#) (EmbArc arc)

move to [arc.c](#)

Member [embGeometry_vulcanize](#) (EmbGeometry *obj)

Review. This could be controlled by a simple flag.

Member [embPattern_correctForMaxStitchLength](#) (EmbPattern *p, EmbReal maxStitchLength, EmbReal maxJumpLength)

The params determine the max XY movement rather than the length. They need renamed or clarified further.

Member [embPattern_stitchEllipse](#) (EmbPattern *p, EmbEllipse ellipse, int thread_index, int style)

finish stitchEllipse

Member [embPattern_stitchPath](#) (EmbPattern *p, EmbPath path, int thread_index, int style)

finish stitch path

Member [embPattern_stitchPolygon](#) (EmbPattern *p, EmbPolygon polygon, int thread_index, int style)

finish stitch polygon

Member [embPattern_stitchPolyline](#) (EmbPattern *p, EmbPolyline polyline, int thread_index, int style)

finish stitch polyline

Member [embVector_multiply](#) (EmbVector vector, EmbReal magnitude, EmbVector *result)

make result return argument.

Member [embVector_normalize](#) (EmbVector vector, EmbVector *result)

make result return argument.

File [format_art.c](#)

Find a source.

File [format_bmc.c](#)

Find a source.

File [format_cnd.c](#)

Find a source.

Page **Formats**

Josh, Review this section and move any info still valid or needing work into TODO comments in the actual libembroidery code. Many items in this list are out of date and do not reflect the current status of libembroidery. When finished, delete this file.

Support for Singer FHE, CHE (Compucon) formats?

Member **formatTable [numberOfFormats]**

This list needs reviewed in case some stitch formats also can contain object data (EMBFORMAT_↔ STCHANDOBJ). *

Member **fread_int32_be (FILE *f)**

replace with embInt_read

Member **fread_uint16 (FILE *f)**

replace with embInt_read

Member **generate_dragon_curve (char *state, int iterations)**

find citation for paper folding method

Page **Geometry and Algorithms**

(Arduino) inotool.org - seems like the logical solution for Nightly/CI builds

(Arduino) Logging of Last Stitch Location to External USB Storage(commonly available and easily replaced) ...wait until TRE is available to avoid rework

(Arduino) Fix emb-outline files

(Arduino)Fix thread-color files

Move back to libembroidery now we have the combined docs build.

For each pattern generate a random string of hexadecimal and append it to the filenames like `stitchList\↔_A16F.dat`. Need to check for a file which indicates that this string has been used already.

Start file with a list of offsets to data with a corresponding table to load into with macro constants for each label needed.

Replace EmbArray functions with embPattern load functions.

Like most user interfaces Embroidermodder is mostly data, so here we will have a list describing where each CSV goes.

(testing) looping test that reads 10 times while running valgrind. See `\texttt{embPattern_loadExternalColorFile()}` Arduino leak note for more info.

(Arduino) Smoothieboard experiments

Member **Geometry::calculateArcData (EmbArc arc)**

convert this to update and make it Type sensitive.

Page **Ideas**

Bibliography style to plainnat.

US letter paper version of printed docs.

Serif font for printed docs.

Member **MainWindow::createAllActions ()**

Set What's This Context Help to statusTip for now so there is some infos there. Make custom whats this context help popup with more descriptive help than just the status bar/tip one liner(short but not real long) with a hyper-link in the custom popup at the bottom to open full help file description. Ex: like wxPython AGW's SuperToolTip. ACTION->setWhatsThis(statusTip);

Finish All Commands ... <.< If an action calls a script then there will be an entry in config that is a StringList to be interpreted as a script.

Member **MdiWindow::saveBMC ()**

Save a Brother PEL image (An 8bpp, 130x113 pixel monochromatic? bitmap image) Why 8bpp when only 1bpp is needed?

Should BMC be limited to ~32KB or is this a mix up with Bitmap Cache?

Is there/should there be other embedded data in the bitmap besides the image itself?

Member `night_vision_action` (String args)

Make night vision color settings.

Member `OBJ_LTYPE`

Use color chart in formats/format-dxf.h for this

Member `PropertyEditor::clearAllFields` ()

DimAligned
DimAngular
DimArcLength
DimDiameter
DimLeader
DimLinear
DimOrdinate
DimRadius

Member `PropertyEditor::createComboBoxSelected` ()

document this

Member `PropertyEditor::createToolButtonQSelect` ()

document this

Member `PropertyEditor::eventFilter` (QObject *obj, QEvent *event)

document this

Member `PropertyEditor::~~PropertyEditor` ()

document this

Member `SaveObject::addPath` (EmbPattern *pattern, QGraphicsItem *item)

Reimplement `addPolyline()` using the libembroidery C API

Member `SaveObject::save` (QString fileName)

Before saving to a stitch only format, Embroidermodder needs to calculate the optimal path to minimize jump stitches. Also based upon which layer needs to be stitched first, the path to the next object needs to be hidden beneath fills that will come later. When finding the optimal path, we need to take into account the color of the thread, as we do not want to try to hide dark colored stitches beneath light colored fills.

Member `SaveObject::toPolyline` (EmbPattern *pattern, const QPointF &objPos, const QPainterPath &objPath, QString layer, const QColor &color, QString lineType, QString lineWeight)

FIX EmbPolyline* polyObject = embPolyline_init(pointList, color_out, 1); //TODO: proper lineType embPattern->_addPolylineAbs(pattern, polyObject);

Member `set_enabled` (QObject *parent, const char *key, bool enabled)

error reporting.

Member `set_visibility` (QObject *parent, const char *key, bool visibility)

error reporting.

Member `SubDescriptor_::colorCode`

better variable naming

Member `SubDescriptor_::someInt`

better variable naming

Member `SubDescriptor_::someOtherInt`

better variable naming

Member `validFileFormat` (String fileName)

check the file exists on the system, rename to validFile?

Member `View::mouseMoveEvent` (QMouseEvent *event)

turn move into an actuator call.

12 Namespace Index

12.1 Namespace List

Here is a list of all namespaces with brief descriptions:

em2_dev_script	39
--------------------------------	----

13 Hierarchical Index

13.1 Class Hierarchy

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

_bcf_directory	40
_bcf_directory_entry	40
_bcf_file	42
_bcf_file_difat	43
_bcf_file_fat	43
_bcf_file_header	44
_vp3Hoop	46
Compress	75
EmbAlignedDim_	76
EmbAngularDim_	76
EmbArc_	77
EmbArcLengthDim_	78
EmbArray_	78
EmbBezier_	79
EmbBlock_	80
EmbCircle_	80
EmbColor_	81
EmbDiameterDim_	84
EmbEllipse_	84
EmbFormatList_	85
EmbGeometry_	86
EmblImage_	88

EmblInfiniteLine_	89
EmbLayer_	90
EmbLeaderDim_	90
EmbLine_	91
EmbLinearDim_	92
EmbOrdinateDim_	92
EmbPath_	92
EmbPattern_	93
EmbPoint_	94
EmbRadiusDim_	95
EmbRay_	96
EmbRect_	96
EmbSatinOutline_	97
EmbSpline_	98
EmbStitch_	98
EmbTextMulti_	99
EmbTextSingle_	99
EmbThread_	100
EmbTime_	101
EmbVector_	102
hoop_padding	131
Huffman	131
LSYSTEM	136
Node_	168
QApplication	
Application	49
QDialog	
EmbDetailsDialog	81
LayerManager	134
Settings_Dialog	189
QDockWidget	
PropertyEditor	170
UndoEditor	204
QFileDialog	

PreviewDialog	169
QGraphicsPathItem	
Geometry	102
QGraphicsView	
View	206
QLineEdit	
CmdPromptInput	65
QMainWindow	
MainWindow	137
QMdiArea	
MdiArea	155
QMdiSubWindow	
MdiWindow	160
QObject	
SaveObject	176
QRubberBand	
SelectBox	186
QSplitter	
CmdPromptSplitter	73
QSplitterHandle	
CmdPromptHandle	59
QStatusBar	
StatusBar	197
QTextBrowser	
CmdPromptHistory	62
QUndoCommand	
UndoableCommand	201
QWidget	
CmdPrompt	50
ImageWidget	132
StxThread_	198
SubDescriptor_	198
SvgAttribute_	199
thread_color_	199
ThredExtension_	200
ThredHeader_	201
VipHeader_	219

14 Class Index

14.1 Class List

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

_bcf_directory	40
_bcf_directory_entry	40
_bcf_file	42
_bcf_file_difat	43
_bcf_file_fat	43
_bcf_file_header	44
_vp3Hoop	46
Application	49
CmdPrompt 50	
CmdPromptHandle 59	
CmdPromptHistory The Command Prompt History class	62
CmdPromptInput	65
CmdPromptSplitter 73	
Compress	75
EmbAlignedDim_	76
EmbAngularDim_	76
EmbArc_ Absolute position (not relative)	77
EmbArcLengthDim_	78
EmbArray_	78
EmbBezier_	79
EmbBlock_	80
EmbCircle_	80
EmbColor_	81
EmbDetailsDialog 81	
EmbDiameterDim_	84

EmbEllipse_	84
EmbFormatList_	85
EmbGeometry_	86
EmbImage_	88
EmbInfiniteLine_	89
EmbLayer_	90
EmbLeaderDim_	90
EmbLine_	91
EmbLinearDim_	92
EmbOrdinateDim_	92
EmbPath_	92
EmbPattern_	93
EmbPoint_	94
EmbRadiusDim_	95
EmbRay_	96
EmbRect_	96
EmbSatinOutline_	97
EmbSpline_	98
EmbStitch_	98
EmbTextMulti_	99
EmbTextSingle_	99
EmbThread_	100
EmbTime_	101
EmbVector_	102
Geometry	
The Geometry class	102
hoop_padding	131
Huffman	131
ImageWidget	
132	
LayerManager	
134	
LSYSTEM	136

MainWindow	
The MainWindow class	137
MdiArea	155
MdiWindow	160
Node_	168
PreviewDialog	169
PropertyEditor	170
SaveObject	176
SelectBox	186
Settings_Dialog	189
StatusBar	197
StxThread_	198
SubDescriptor_	198
SvgAttribute_	199
thread_color_	199
ThredExtension_	200
ThredHeader_	201
UndoableCommand	201
UndoEditor	204
View	206
VipHeader_	219

15 File Index

15.1 File List

Here is a list of all files with brief descriptions:

embroidermodder2/cmdprompt.cpp	221
embroidermodder2/em2_dev_script.py	221
embroidermodder2/embdetails-dialog.cpp	221
embroidermodder2/embroidermodder.cpp	221
embroidermodder2/embroidermodder.h	222
embroidermodder2/imagewidget.cpp	256

embroidermodder2/interface.cpp	256
embroidermodder2/layer-manager.cpp	263
embroidermodder2/mainwindow-menus.cpp	263
embroidermodder2/mainwindow-toolbars.cpp	264
embroidermodder2/mainwindow.cpp	264
embroidermodder2/mdiarea.cpp	294
embroidermodder2/mdiwindow.cpp	294
embroidermodder2/objects.cpp	294
embroidermodder2/preview-dialog.cpp	295
embroidermodder2/property-editor.cpp	295
embroidermodder2/selectbox.cpp	297
embroidermodder2/settings-dialog.cpp	297
embroidermodder2/statusbar.cpp	299
embroidermodder2/undo-commands.cpp	299
embroidermodder2/undo-editor.cpp	299
embroidermodder2/view.cpp	300
extern/libembroidery/src/array.c	300
extern/libembroidery/src/compress.c	302
extern/libembroidery/src/embroidery.h	304
extern/libembroidery/src/embroidery_internal.h	344
extern/libembroidery/src/encoding.c	391
extern/libembroidery/src/fill.c	394
extern/libembroidery/src/formats.c	399
extern/libembroidery/src/geometry.c	445
extern/libembroidery/src/image.c	459
extern/libembroidery/src/main.c	460
extern/libembroidery/src/pattern.c	471
extern/libembroidery/src/thread-color.c	476
extern/libembroidery/src/formats/format_100.c	403
extern/libembroidery/src/formats/format_10o.c	403
extern/libembroidery/src/formats/format_art.c	404
extern/libembroidery/src/formats/format_bmc.c	404

extern/libembroidery/src/formats/ format_bro.c	405
extern/libembroidery/src/formats/ format_cnd.c	405
extern/libembroidery/src/formats/ format_col.c	406
extern/libembroidery/src/formats/ format_csd.c	407
extern/libembroidery/src/formats/ format_csv.c	408
extern/libembroidery/src/formats/ format_dat.c	409
extern/libembroidery/src/formats/ format_dem.c	410
extern/libembroidery/src/formats/ format_dsb.c	410
extern/libembroidery/src/formats/ format_dst.c	411
extern/libembroidery/src/formats/ format_dsz.c	413
extern/libembroidery/src/formats/ format_dxf.c	413
extern/libembroidery/src/formats/ format_edr.c	414
extern/libembroidery/src/formats/ format_emd.c	415
extern/libembroidery/src/formats/ format_exp.c	415
extern/libembroidery/src/formats/ format_exy.c	416
extern/libembroidery/src/formats/ format_eyc.c	416
extern/libembroidery/src/formats/ format_fxy.c	417
extern/libembroidery/src/formats/ format_gc.c	417
extern/libembroidery/src/formats/ format_gnc.c	418
extern/libembroidery/src/formats/ format_gt.c	418
extern/libembroidery/src/formats/ format_hus.c	419
extern/libembroidery/src/formats/ format_inb.c	420
extern/libembroidery/src/formats/ format_inf.c	420
extern/libembroidery/src/formats/ format_jef.c	421
extern/libembroidery/src/formats/ format_ksm.c	422
extern/libembroidery/src/formats/ format_max.c	422
extern/libembroidery/src/formats/ format_mit.c	423
extern/libembroidery/src/formats/ format_new.c	424
extern/libembroidery/src/formats/ format_ofm.c	424
extern/libembroidery/src/formats/ format_pcd.c	425
extern/libembroidery/src/formats/ format_pcm.c	426
extern/libembroidery/src/formats/ format_pcq.c	426

extern/libembroidery/src/formats/ format_pcs.c	427
extern/libembroidery/src/formats/ format_pec.c	427
extern/libembroidery/src/formats/ format_pel.c	429
extern/libembroidery/src/formats/ format_pem.c	429
extern/libembroidery/src/formats/ format_pes.c	430
extern/libembroidery/src/formats/ format_phb.c	432
extern/libembroidery/src/formats/ format_phc.c	433
extern/libembroidery/src/formats/ format_plt.c	433
extern/libembroidery/src/formats/ format_rgb.c	434
extern/libembroidery/src/formats/ format_sew.c	434
extern/libembroidery/src/formats/ format_shv.c	435
extern/libembroidery/src/formats/ format_sst.c	435
extern/libembroidery/src/formats/ format_stx.c	436
extern/libembroidery/src/formats/ format_svg.c	436
extern/libembroidery/src/formats/ format_t01.c	438
extern/libembroidery/src/formats/ format_t09.c	438
extern/libembroidery/src/formats/ format_tap.c	439
extern/libembroidery/src/formats/ format_thr.c	439
extern/libembroidery/src/formats/ format_txt.c	440
extern/libembroidery/src/formats/ format_u00.c	440
extern/libembroidery/src/formats/ format_u01.c	441
extern/libembroidery/src/formats/ format_vip.c	441
extern/libembroidery/src/formats/ format_vp3.c	443
extern/libembroidery/src/formats/ format_xxx.c	444
extern/libembroidery/src/formats/ format_zsk.c	445
extern/libembroidery/src/geometry/ arc.c	447
extern/libembroidery/src/geometry/ circle.c	450
extern/libembroidery/src/geometry/ ellipse.c	451
extern/libembroidery/src/geometry/ functions.c	453
extern/libembroidery/src/geometry/ line.c	454
extern/libembroidery/src/geometry/ path.c	454
extern/libembroidery/src/geometry/ polygon.c	454

extern/libembroidery/src/geometry/polyline.c	454
extern/libembroidery/src/geometry/rect.c	455
extern/libembroidery/src/geometry/text.c	455
extern/libembroidery/src/geometry/vector.c	457

16 Namespace Documentation

16.1 em2_dev_script Namespace Reference

Variables

- string [header](#)
- dictionary [d](#) = {}
- [s](#) = f.read()

16.1.1 Detailed Description

Embroidermodder 2.

Copyright 2013-2023 The Embroidermodder Team
 Embroidermodder 2 is Open Source Software.
 See LICENSE for licensing terms.

In order to improve the config.toml configuration file, we can't rely on Embroidermodder2 itself since it will crash on attempting to load poorly formed data. Instead, we run checks with this script to see that config.toml is well-formed as toml and it can make recommendations to the developers on what to do about missing data like a compiler would.

16.1.2 Variable Documentation

16.1.2.1 [d](#) `d = {}`

16.1.2.2 [header](#) `string header`

Initial value:

```
1 = """
2 # Embroidermodder 2.
3 #
4 # -----
5 #
6 # Copyright 2013-2023 The Embroidermodder Team
7 # Embroidermodder 2 is Open Source Software.
8 # See LICENSE for licensing terms.
9 #
10 # -----
11
12 """
```

16.1.2.3 `s = f.read()`

17 Class Documentation

17.1 `_bcf_directory` Struct Reference

```
#include <embroidery_internal.h>
```

Public Attributes

- `bcf_directory_entry * dirEntries`
- unsigned int `maxNumberOfDirectoryEntries`

17.1.1 Detailed Description

Todo possibly add a directory tree in the future.

17.1.2 Member Data Documentation

17.1.2.1 `dirEntries` `bcf_directory_entry* dirEntries`

17.1.2.2 `maxNumberOfDirectoryEntries` `unsigned int maxNumberOfDirectoryEntries`

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

- `extern/libembroidery/src/embroidery_internal.h`

17.2 `_bcf_directory_entry` Struct Reference

```
#include <embroidery_internal.h>
```

Public Attributes

- char `directoryEntryName` [32]
- unsigned short `directoryEntryNameLength`
- unsigned char `objectType`
- unsigned char `colorFlag`
- unsigned int `leftSiblingId`
- unsigned int `rightSiblingId`
- unsigned int `childId`
- unsigned char `CLSID` [16]
- unsigned int `stateBits`
- `EmbTime` `creationTime`
- `EmbTime` `modifiedTime`
- unsigned int `startingSectorLocation`
- unsigned long `streamSize`
- unsigned int `streamSizeHigh`
- struct `_bcf_directory_entry * next`

17.2.1 Member Data Documentation

17.2.1.1 childId `unsigned int childId`

17.2.1.2 CLSID `unsigned char CLSID[16]`

17.2.1.3 colorFlag `unsigned char colorFlag`

17.2.1.4 creationTime `EmbTime creationTime`

17.2.1.5 directoryEntryName `char directoryEntryName[32]`

17.2.1.6 directoryEntryNameLength `unsigned short directoryEntryNameLength`

17.2.1.7 leftSiblingId `unsigned int leftSiblingId`

17.2.1.8 modifiedTime `EmbTime modifiedTime`

17.2.1.9 next `struct _bcf_directory_entry* next`

17.2.1.10 objectType `unsigned char objectType`

17.2.1.11 rightSiblingId unsigned int rightSiblingId

17.2.1.12 startingSectorLocation unsigned int startingSectorLocation

17.2.1.13 stateBits unsigned int stateBits

17.2.1.14 streamSize unsigned long streamSize

17.2.1.15 streamSizeHigh unsigned int streamSizeHigh

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

- [extern/libembroidery/src/embroidery_internal.h](#)

17.3 _bcf_file Struct Reference

```
#include <embroidery_internal.h>
```

Public Attributes

- [bcf_file_header](#) header
- [bcf_file_difat](#) * difat
- [bcf_file_fat](#) * fat
- [bcf_directory](#) * directory

17.3.1 Member Data Documentation

17.3.1.1 difat [bcf_file_difat](#)* difat

The header for the CompoundFile

17.3.1.2 directory [bcf_directory](#)* directory

The File Allocation Table for the Compound File

17.3.1.3 `fat` `bcf_file_fat*` `fat`

The "Double Indirect FAT" for the CompoundFile

17.3.1.4 `header` `bcf_file_header` `header`

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

- `extern/libembroidery/src/embroidery_internal.h`

17.4 `_bcf_file_difat` Struct Reference

```
#include <embroidery_internal.h>
```

Public Attributes

- unsigned int `fatSectorCount`
- unsigned int `fatSectorEntries` [109]
- unsigned int `sectorSize`

17.4.1 Member Data Documentation

17.4.1.1 `fatSectorCount` unsigned int `fatSectorCount`

17.4.1.2 `fatSectorEntries` unsigned int `fatSectorEntries`[109]

17.4.1.3 `sectorSize` unsigned int `sectorSize`

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

- `extern/libembroidery/src/embroidery_internal.h`

17.5 `_bcf_file_fat` Struct Reference

```
#include <embroidery_internal.h>
```

Public Attributes

- int [fatEntryCount](#)
- unsigned int [fatEntries](#) [255]
- unsigned int [numberOfEntriesInFatSector](#)

17.5.1 Member Data Documentation

17.5.1.1 fatEntries unsigned int fatEntries[255]

17.5.1.2 fatEntryCount int fatEntryCount

17.5.1.3 numberOfEntriesInFatSector unsigned int numberOfEntriesInFatSector

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

- extern/libembroidery/src/[embroidery_internal.h](#)

17.6 _bcf_file_header Struct Reference

```
#include <embroidery_internal.h>
```

Public Attributes

- unsigned char [signature](#) [8]
- unsigned char [CLSID](#) [16]
- unsigned short [minorVersion](#)
- unsigned short [majorVersion](#)
- unsigned short [byteOrder](#)
- unsigned short [sectorShift](#)
- unsigned short [miniSectorShift](#)
- unsigned short [reserved1](#)
- unsigned int [reserved2](#)
- unsigned int [numberOfDirectorySectors](#)
- unsigned int [numberOfFATSectors](#)
- unsigned int [firstDirectorySectorLocation](#)
- unsigned int [transactionSignatureNumber](#)
- unsigned int [miniStreamCutoffSize](#)
- unsigned int [firstMiniFATSectorLocation](#)
- unsigned int [numberOfMiniFatSectors](#)
- unsigned int [firstDifatSectorLocation](#)
- unsigned int [numberOfDifatSectors](#)

17.6.1 Detailed Description

Todo CLSID should be a separate type.

17.6.2 Member Data Documentation

17.6.2.1 `byteOrder` `unsigned short byteOrder`

17.6.2.2 `CLSID` `unsigned char CLSID[16]`

17.6.2.3 `firstDifatSectorLocation` `unsigned int firstDifatSectorLocation`

17.6.2.4 `firstDirectorySectorLocation` `unsigned int firstDirectorySectorLocation`

17.6.2.5 `firstMiniFATSectorLocation` `unsigned int firstMiniFATSectorLocation`

17.6.2.6 `majorVersion` `unsigned short majorVersion`

17.6.2.7 `miniSectorShift` `unsigned short miniSectorShift`

17.6.2.8 `miniStreamCutoffSize` `unsigned int miniStreamCutoffSize`

17.6.2.9 `minorVersion` `unsigned short minorVersion`

17.6.2.10 numberOfDifatSectors unsigned int numberOfDifatSectors

17.6.2.11 numberOfDirectorySectors unsigned int numberOfDirectorySectors

17.6.2.12 numberOfFATSectors unsigned int numberOfFATSectors

17.6.2.13 numberOfMiniFatSectors unsigned int numberOfMiniFatSectors

17.6.2.14 reserved1 unsigned short reserved1

17.6.2.15 reserved2 unsigned int reserved2

17.6.2.16 sectorShift unsigned short sectorShift

17.6.2.17 signature unsigned char signature[8]

17.6.2.18 transactionSignatureNumber unsigned int transactionSignatureNumber

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

- [extern/libembroidery/src/embroidery_internal.h](#)

17.7 _vp3Hoop Struct Reference

```
#include <embroidery_internal.h>
```

Public Attributes

- int [right](#)
- int [bottom](#)
- int [left](#)
- int [top](#)
- int [threadLength](#)
- char [unknown2](#)
- unsigned char [numberOfColors](#)
- unsigned short [unknown3](#)
- int [unknown4](#)
- int [numberOfBytesRemaining](#)
- int [xOffset](#)
- int [yOffset](#)
- unsigned char [byte1](#)
- unsigned char [byte2](#)
- unsigned char [byte3](#)
- int [right2](#)
- int [left2](#)
- int [bottom2](#)
- int [top2](#)
- int [width](#)
- int [height](#)

17.7.1 Member Data Documentation

17.7.1.1 bottom `int bottom`

17.7.1.2 bottom2 `int bottom2`

17.7.1.3 byte1 `unsigned char byte1`

17.7.1.4 byte2 `unsigned char byte2`

17.7.1.5 byte3 `unsigned char byte3`

17.7.1.6 height `int height`

17.7.1.7 left `int left`

17.7.1.8 left2 `int left2`

17.7.1.9 numberOfBytesRemaining `int numberOfBytesRemaining`

17.7.1.10 numberOfColors `unsigned char numberOfColors`

17.7.1.11 right `int right`

17.7.1.12 right2 `int right2`

17.7.1.13 threadLength `int threadLength`

17.7.1.14 top `int top`

17.7.1.15 top2 `int top2`

17.7.1.16 unknown2 `char unknown2`

17.7.1.17 unknown3 `unsigned short unknown3`

17.7.1.18 unknown4 `int unknown4`

17.7.1.19 width `int width`

17.7.1.20 xOffset `int xOffset`

17.7.1.21 yOffset `int yOffset`

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

- `extern/libembroidery/src/embroidery_internal.h`

17.8 Application Class Reference

```
#include <embroidermodder.h>
```

Public Member Functions

- [Application](#) (int argc, char **argv)
[Application::Application.](#)
- void [setMainWin](#) ([MainWindow](#) *mainWin)

Public Attributes

- [MainWindow](#) * [__mainWin](#)

Protected Member Functions

- virtual bool [event](#) (QEvent *e)
[Application::event.](#)

17.8.1 Detailed Description

Note

On Mac, if the user drops a file on the app's Dock icon, or uses Open As, then this is how the app actually opens the file.

17.8.2 Constructor & Destructor Documentation

17.8.2.1 Application() [Application](#) (
 int argc,
 char ** argv)

[Application::Application.](#)

Parameters

<i>argc</i>	
<i>argv</i>	

17.8.3 Member Function Documentation

17.8.3.1 event() `bool event (`
 `QEvent * event) [protected], [virtual]`

[Application::event.](#)

Parameters

<i>event</i>	
--------------	--

Returns

17.8.3.2 setMainWin() `void setMainWin (`
 `MainWindow * mainWin) [inline]`

17.8.4 Member Data Documentation

17.8.4.1 __mainWin `MainWindow* __mainWin`

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

- [embroidermodder2/embroidermodder.h](#)
- [embroidermodder2/embroidermodder.cpp](#)

17.9 CmdPrompt Class Reference

```
#include <embroidermodder.h>
```


Public Slots

- void [setCurrentText](#) (QString txt)
CmdPrompt::setCurrentText.
- void [setHistory](#) (QString txt)
CmdPrompt::setHistory.
- void [setPrefix](#) (QString txt)
CmdPrompt::setPrefix.
- void [appendHistory](#) (QString txt)
CmdPrompt::appendHistory.
- void [alert](#) (QString txt)
CmdPrompt::alert.
- void [startBlinking](#) ()
CmdPrompt::startBlinking.
- void [stopBlinking](#) ()
CmdPrompt::stopBlinking.
- void [blink](#) ()
CmdPrompt::blink.
- void [setPromptTextColor](#) (const QColor &)
CmdPrompt::setPromptTextColor.
- void [setPromptBackgroundColor](#) (const QColor &)
CmdPrompt::setPromptBackgroundColor.
- void [setPromptFontFamily](#) (QString)
CmdPrompt::setPromptFontFamily.
- void [setPromptFontStyle](#) (QString)
CmdPrompt::setPromptFontStyle.
- void [setPromptFontSize](#) (int)
CmdPrompt::setPromptFontSize.
- void [floatingChanged](#) (bool)
CmdPrompt::floatingChanged.
- void [saveHistory](#) (QString fileName, bool html)
CmdPrompt::saveHistory.

Signals

- void [appendTheHistory](#) (QString txt, int prefixLength)
- void [startCommand](#) (QString cmd)
- void [runCommand](#) (QString cmd, QString cmdtxt)
- void [deletePressed](#) ()
- void [tabPressed](#) ()
- void [escapePressed](#) ()
- void [upPressed](#) ()
- void [downPressed](#) ()
- void [F1Pressed](#) ()
- void [F2Pressed](#) ()
- void [F3Pressed](#) ()
- void [F4Pressed](#) ()
- void [F5Pressed](#) ()
- void [F6Pressed](#) ()
- void [F7Pressed](#) ()
- void [F8Pressed](#) ()
- void [F9Pressed](#) ()
- void [F10Pressed](#) ()
- void [F11Pressed](#) ()

- void [F12Pressed](#) ()
- void [cutPressed](#) ()
- void [copyPressed](#) ()
- void [pastePressed](#) ()
- void [selectAllPressed](#) ()
- void [undoPressed](#) ()
- void [redoPressed](#) ()
- void [shiftPressed](#) ()
- void [shiftReleased](#) ()
- void [showSettings](#) ()
- void [historyAppended](#) (QString txt)

Public Member Functions

- [CmdPrompt](#) (QWidget *parent=0)
[CmdPrompt::CmdPrompt.](#)
- [~CmdPrompt](#) ()
[CmdPrompt::~~CmdPrompt.](#)
- void [updateStyle](#) ()
[CmdPrompt::updateStyle.](#)

Public Attributes

- [CmdPromptInput](#) * [promptInput](#)
- [CmdPromptHistory](#) * [promptHistory](#)
- QVBoxLayout * [promptVBoxLayout](#)
- QFrame * [promptDivider](#)
- [CmdPromptSplitter](#) * [promptSplitter](#)
- QHash< QString, QString > * [styleHash](#)
- QTimer * [blinkTimer](#)
- bool [blinkState](#)

17.9.1 Detailed Description

17.9.2 Constructor & Destructor Documentation

17.9.2.1 [CmdPrompt\(\)](#) [CmdPrompt](#) (QWidget * *parent* = 0)

[CmdPrompt::CmdPrompt.](#)

Parameters

<i>parent</i>	
---------------	--

17.9.2.2 ~CmdPrompt() ~CmdPrompt ()

[CmdPrompt::~CmdPrompt.](#)

17.9.3 Member Function Documentation

17.9.3.1 alert void alert (QString txt) [slot]

[CmdPrompt::alert.](#)

Parameters

<i>txt</i>	
------------	--

17.9.3.2 appendHistory void appendHistory (QString txt) [slot]

[CmdPrompt::appendHistory.](#)

Parameters

<i>txt</i>	
------------	--

17.9.3.3 appendTheHistory void appendTheHistory (QString txt, int prefixLength) [signal]

17.9.3.4 blink void blink () [slot]

[CmdPrompt::blink.](#)

17.9.3.5 copyPressed void copyPressed () [signal]

17.9.3.6 cutPressed `void cutPressed () [signal]`

17.9.3.7 deletePressed `void deletePressed () [signal]`

17.9.3.8 downPressed `void downPressed () [signal]`

17.9.3.9 escapePressed `void escapePressed () [signal]`

17.9.3.10 F10Pressed `void F10Pressed () [signal]`

17.9.3.11 F11Pressed `void F11Pressed () [signal]`

17.9.3.12 F12Pressed `void F12Pressed () [signal]`

17.9.3.13 F1Pressed `void F1Pressed () [signal]`

17.9.3.14 F2Pressed `void F2Pressed () [signal]`

17.9.3.15 F3Pressed `void F3Pressed () [signal]`

17.9.3.16 F4Pressed `void F4Pressed () [signal]`

17.9.3.17 F5Pressed void F5Pressed () [signal]

17.9.3.18 F6Pressed void F6Pressed () [signal]

17.9.3.19 F7Pressed void F7Pressed () [signal]

17.9.3.20 F8Pressed void F8Pressed () [signal]

17.9.3.21 F9Pressed void F9Pressed () [signal]

17.9.3.22 floatingChanged void floatingChanged (
 bool *isFloating*) [slot]

[CmdPrompt::floatingChanged.](#)

Parameters

<i>isFloating</i>	
-------------------	--

17.9.3.23 historyAppended void historyAppended (
 QString *txt*) [signal]

17.9.3.24 pastePressed void pastePressed () [signal]

17.9.3.25 redoPressed void redoPressed () [signal]

17.9.3.26 runCommand void runCommand (
 QString *cmd*,
 QString *cmdtxt*) [signal]

17.9.3.27 saveHistory void saveHistory (
 QString *fileName*,
 bool *html*) [slot]

[CmdPrompt::saveHistory](#).

Parameters

<i>fileName</i>	
<i>html</i>	

17.9.3.28 selectAllPressed void selectAllPressed () [signal]

17.9.3.29 setCurrentText void setCurrentText (
 QString *txt*) [inline], [slot]

17.9.3.30 setHistory void setHistory (
 QString *txt*) [inline], [slot]

17.9.3.31 setPrefix void setPrefix (
 QString *txt*) [slot]

[CmdPrompt::setPrefix](#).

Parameters

<i>txt</i>	
------------	--

17.9.3.32 setPromptBackgroundColor void setPromptBackgroundColor (
 const QColor & *color*) [slot]

[CmdPrompt::setPromptBackgroundColor](#).

Parameters

<i>color</i>	
--------------	--

17.9.3.33 setPromptFontFamily void setPromptFontFamily (
 QString *family*) [slot]

[CmdPrompt::setPromptFontFamily.](#)

Parameters

<i>family</i>	
---------------	--

17.9.3.34 setPromptFontSize void setPromptFontSize (
 int *size*) [slot]

[CmdPrompt::setPromptFontSize.](#)

Parameters

<i>size</i>	
-------------	--

17.9.3.35 setPromptFontStyle void setPromptFontStyle (
 QString *style*) [slot]

[CmdPrompt::setPromptFontStyle.](#)

Parameters

<i>style</i>	
--------------	--

17.9.3.36 setPromptTextColor void setPromptTextColor (
 const QColor & *color*) [slot]

[CmdPrompt::setPromptTextColor.](#)

Parameters

<i>color</i>	
--------------	--

17.9.3.37 shiftPressed `void shiftPressed () [signal]`

17.9.3.38 shiftReleased `void shiftReleased () [signal]`

17.9.3.39 showSettings `void showSettings () [signal]`

17.9.3.40 startBlinking `void startBlinking () [slot]`

[CmdPrompt::startBlinking.](#)

17.9.3.41 startCommand `void startCommand (
 QString cmd) [signal]`

17.9.3.42 stopBlinking `void stopBlinking () [slot]`

[CmdPrompt::stopBlinking.](#)

17.9.3.43 tabPressed `void tabPressed () [signal]`

17.9.3.44 undoPressed `void undoPressed () [signal]`

17.9.3.45 updateStyle() `void updateStyle ()`

[CmdPrompt::updateStyle.](#)

17.9.3.46 upPressed `void upPressed () [signal]`

17.9.4 Member Data Documentation

17.9.4.1 blinkState `bool blinkState`

17.9.4.2 blinkTimer `QTimer* blinkTimer`

17.9.4.3 promptDivider `QFrame* promptDivider`

17.9.4.4 promptHistory `CmdPromptHistory* promptHistory`

17.9.4.5 promptInput `CmdPromptInput* promptInput`

17.9.4.6 promptSplitter `CmdPromptSplitter* promptSplitter`

17.9.4.7 promptVBoxLayout `QVBoxLayout* promptVBoxLayout`

17.9.4.8 styleHash `QHash<QString, QString>* styleHash`

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

- embroidermodder2/[embroidermodder.h](#)
- embroidermodder2/[cmdprompt.cpp](#)

17.10 CmdPromptHandle Class Reference

```
#include <embroidermodder.h>
```

Signals

- void [handlePressed](#) (int y)
- void [handleReleased](#) (int y)
- void [handleMoved](#) (int y)

Public Member Functions

- [CmdPromptHandle](#) (Qt::Orientation orientation, QSplitter *parent)
[CmdPromptHandle::CmdPromptHandle.](#)
- [~CmdPromptHandle](#) ()
[CmdPromptHandle::~~CmdPromptHandle.](#)

Public Attributes

- int [pressY](#)
- int [releaseY](#)
- int [moveY](#)

Protected Member Functions

- void [mousePressEvent](#) (QMouseEvent *e)
[CmdPromptHandle::mousePressEvent.](#)
- void [mouseReleaseEvent](#) (QMouseEvent *e)
[CmdPromptHandle::mouseReleaseEvent.](#)
- void [mouseMoveEvent](#) (QMouseEvent *e)
[CmdPromptHandle::mouseMoveEvent.](#)

17.10.1 Detailed Description

17.10.2 Constructor & Destructor Documentation

17.10.2.1 [CmdPromptHandle\(\)](#) [CmdPromptHandle](#) (Qt::Orientation orientation, QSplitter * parent)

[CmdPromptHandle::CmdPromptHandle.](#)

Parameters

<i>orientation</i>	
<i>parent</i>	

17.10.2.2 `~CmdPromptHandle()` `~CmdPromptHandle ()`

[CmdPromptHandle::~~CmdPromptHandle.](#)

17.10.3 Member Function Documentation

17.10.3.1 `handleMoved` `void handleMoved (`
`int y) [signal]`

17.10.3.2 `handlePressed` `void handlePressed (`
`int y) [signal]`

17.10.3.3 `handleReleased` `void handleReleased (`
`int y) [signal]`

17.10.3.4 `mouseMoveEvent()` `void mouseMoveEvent (`
`QMouseEvent * e) [protected]`

[CmdPromptHandle::mouseMoveEvent.](#)

Parameters

<i>e</i>	The mouse event.
----------	------------------

17.10.3.5 `mousePressEvent()` `void mousePressEvent (`
`QMouseEvent * e) [protected]`

[CmdPromptHandle::mousePressEvent.](#)

Parameters

<i>e</i>	
----------	--

17.10.3.6 `mouseReleaseEvent()` `void mouseReleaseEvent (`
`QMouseEvent * e) [protected]`

[CmdPromptHandle::mouseReleaseEvent](#).

Parameters

<i>e</i>	The mouse event.
----------	------------------

17.10.4 Member Data Documentation

17.10.4.1 moveY `int moveY`

17.10.4.2 pressY `int pressY`

17.10.4.3 releaseY `int releaseY`

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

- embroidermodder2/[embroidermodder.h](#)
- embroidermodder2/[cmdprompt.cpp](#)

17.11 CmdPromptHistory Class Reference

The Command Prompt History class.

```
#include <embroidermodder.h>
```

Public Slots

- void [appendHistory](#) (QString txt, int prefixLength)
[CmdPromptHistory::appendHistory](#).
- void [startResizeHistory](#) (int y)
[CmdPromptHistory::startResizeHistory](#).
- void [stopResizeHistory](#) (int y)
[CmdPromptHistory::stopResizeHistory](#).
- void [resizeHistory](#) (int y)
[CmdPromptHistory::resizeHistory](#).

Signals

- void [historyAppended](#) (QString txt)

Public Member Functions

- [CmdPromptHistory](#) (QWidget *parent=0)
CmdPromptHistory::CmdPromptHistory.
- [~CmdPromptHistory](#) ()
CmdPromptHistory::~~CmdPromptHistory.
- [QString applyFormatting](#) (QString txt, int prefixLength)
CmdPromptHistory::applyFormatting.

Public Attributes

- int [tmpHeight](#)

Protected Member Functions

- void [contextMenuEvent](#) (QContextMenuEvent *event)
CmdPromptHistory::contextMenuEvent.

17.11.1 Detailed Description

The Command Prompt History class.

17.11.2 Constructor & Destructor Documentation

17.11.2.1 CmdPromptHistory() [CmdPromptHistory](#) (
 QWidget * *parent* = 0)

[CmdPromptHistory::CmdPromptHistory.](#)

Parameters

<i>parent</i>	The QWidget that it sits in.
---------------	------------------------------

17.11.2.2 ~CmdPromptHistory() [~CmdPromptHistory](#) ()

[CmdPromptHistory::~~CmdPromptHistory.](#)

17.11.3 Member Function Documentation

17.11.3.1 appendHistory void appendHistory (
 QString *txt*,
 int *prefixLength*) [slot]

[CmdPromptHistory::appendHistory.](#)

Parameters

<i>txt</i>	
<i>prefixLength</i>	

17.11.3.2 applyFormatting() QString applyFormatting (
 QString *txt*,
 int *prefixLength*)

[CmdPromptHistory::applyFormatting.](#)

Parameters

<i>txt</i>	
<i>prefixLength</i>	

Returns

17.11.3.3 contextMenuEvent() void contextMenuEvent (
 QContextMenuEvent * *event*) [protected]

[CmdPromptHistory::contextMenuEvent.](#)

Parameters

<i>event</i>	
--------------	--

17.11.3.4 historyAppended void historyAppended (
 QString *txt*) [signal]

17.11.3.5 resizeHistory void resizeHistory (
 int *y*) [slot]

[CmdPromptHistory::resizeHistory.](#)

Parameters

<i>y</i>	
----------	--

17.11.3.6 startResizeHistory void startResizeHistory (
int *y*) [slot]

[CmdPromptHistory::startResizeHistory.](#)

17.11.3.7 stopResizeHistory void stopResizeHistory (
int *y*) [slot]

[CmdPromptHistory::stopResizeHistory.](#)

17.11.4 Member Data Documentation

17.11.4.1 tmpHeight int tmpHeight

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

- embroidermodder2/[embroidermodder.h](#)
- embroidermodder2/[cmdprompt.cpp](#)

17.12 CmdPromptInput Class Reference

```
#include <embroidermodder.h>
```

Public Slots

- void [endCommand](#) ()
[CmdPromptInput::endCommand.](#)
- void [processInput](#) (void)
[CmdPromptInput::processInput.](#)
- void [checkSelection](#) ()
[CmdPromptInput::checkSelection.](#)
- void [updateCurrentText](#) (QString txt)
[CmdPromptInput::updateCurrentText.](#)
- void [checkEditedText](#) (QString txt)
[CmdPromptInput::checkEditedText.](#)
- void [checkChangedText](#) (QString txt)
[CmdPromptInput::checkChangedText.](#)
- void [checkCursorPosition](#) (int oldpos, int newpos)
[CmdPromptInput::checkCursorPosition.](#)

Signals

- void [appendHistory](#) (QString txt, int prefixLength)
- void [startCommand](#) (QString cmd)
- void [runCommand](#) (QString cmd, QString cmdtxt)
- void [deletePressed](#) ()
- void [tabPressed](#) ()
- void [escapePressed](#) ()
- void [upPressed](#) ()
- void [downPressed](#) ()
- void [F1Pressed](#) ()
- void [F2Pressed](#) ()
- void [F3Pressed](#) ()
- void [F4Pressed](#) ()
- void [F5Pressed](#) ()
- void [F6Pressed](#) ()
- void [F7Pressed](#) ()
- void [F8Pressed](#) ()
- void [F9Pressed](#) ()
- void [F10Pressed](#) ()
- void [F11Pressed](#) ()
- void [F12Pressed](#) ()
- void [cutPressed](#) ()
- void [copyPressed](#) ()
- void [pastePressed](#) ()
- void [selectAllPressed](#) ()
- void [undoPressed](#) ()
- void [redoPressed](#) ()
- void [shiftPressed](#) ()
- void [shiftReleased](#) ()
- void [showSettings](#) ()
- void [stopBlinking](#) ()

Public Member Functions

- [CmdPromptInput](#) (QWidget *parent=0)
CmdPromptInput::CmdPromptInput.
- [~CmdPromptInput](#) ()
- void [changeFormatting](#) (std::vector< QTextLayout::FormatRange > formats)
CmdPromptInput::changeFormatting.
- void [clearFormatting](#) ()
CmdPromptInput::clearFormatting.
- void [applyFormatting](#) ()
CmdPromptInput::applyFormatting.

Public Attributes

- QString [curText](#)
- QString [defaultPrefix](#)
- QString [prefix](#)
- QString [lastCmd](#)
- QString [curCmd](#)
- bool [cmdActive](#)
- bool [rapidFireEnabled](#)
- bool [isBlinking](#)

Protected Member Functions

- void [contextMenuEvent](#) (QContextMenuEvent *event)
[CmdPromptInput::contextMenuEvent](#).
- bool [eventFilter](#) (QObject *obj, QEvent *event)
[CmdPromptInput::eventFilter](#).

Private Slots

- void [copyClip](#) ()
[CmdPromptInput::copyClip](#).
- void [pasteClip](#) ()
[CmdPromptInput::pasteClip](#).

17.12.1 Constructor & Destructor Documentation

17.12.1.1 CmdPromptInput() [CmdPromptInput](#) (
QWidget * parent = 0)

[CmdPromptInput::CmdPromptInput](#).

Parameters

<i>parent</i>	
---------------	--

17.12.1.2 ~CmdPromptInput() [~CmdPromptInput](#) () [inline]

17.12.2 Member Function Documentation

17.12.2.1 appendHistory void [appendHistory](#) (
QString txt,
int *prefixLength*) [signal]

17.12.2.2 applyFormatting() void [applyFormatting](#) ()
[CmdPromptInput::applyFormatting](#).

17.12.2.3 changeFormatting() void [changeFormatting](#) (
std::vector< QTextLayout::FormatRange > *formats*)
[CmdPromptInput::changeFormatting](#).

Parameters

<i>formats</i>	
----------------	--

17.12.2.4 checkChangedText void checkChangedText (
 QString *txt*) [slot]

[CmdPromptInput::checkChangedText.](#)

Parameters

<i>txt</i>	
------------	--

17.12.2.5 checkCursorPosition void checkCursorPosition (
 int *oldpos*,
 int *newpos*) [slot]

[CmdPromptInput::checkCursorPosition.](#)

Parameters

<i>oldpos</i>	
<i>newpos</i>	

17.12.2.6 checkEditedText void checkEditedText (
 QString *txt*) [slot]

[CmdPromptInput::checkEditedText.](#)

Parameters

<i>txt</i>	
------------	--

17.12.2.7 checkSelection void checkSelection () [slot]

[CmdPromptInput::checkSelection.](#)

17.12.2.8 clearFormatting() `void clearFormatting ()`

[CmdPromptInput::clearFormatting.](#)

17.12.2.9 contextMenuEvent() `void contextMenuEvent (
 QContextMenuEvent * event) [protected]`

[CmdPromptInput::contextMenuEvent.](#)

Parameters

<i>event</i>	
--------------	--

17.12.2.10 copyClip `void copyClip () [private], [slot]`

[CmdPromptInput::copyClip.](#)

17.12.2.11 copyPressed `void copyPressed () [signal]`

17.12.2.12 cutPressed `void cutPressed () [signal]`

17.12.2.13 deletePressed `void deletePressed () [signal]`

17.12.2.14 downPressed `void downPressed () [signal]`

17.12.2.15 endCommand `void endCommand () [slot]`

[CmdPromptInput::endCommand.](#)

17.12.2.16 escapePressed `void escapePressed () [signal]`

17.12.2.17 eventFilter() `bool eventFilter (
 QObject * obj,
 QEvent * event) [protected]`

[CmdPromptInput::eventFilter.](#)

Parameters

<i>obj</i>	
<i>event</i>	

Returns

17.12.2.18 F10Pressed `void F10Pressed () [signal]`

17.12.2.19 F11Pressed `void F11Pressed () [signal]`

17.12.2.20 F12Pressed `void F12Pressed () [signal]`

17.12.2.21 F1Pressed `void F1Pressed () [signal]`

17.12.2.22 F2Pressed `void F2Pressed () [signal]`

17.12.2.23 F3Pressed `void F3Pressed () [signal]`

17.12.2.24 F4Pressed `void F4Pressed () [signal]`

17.12.2.25 F5Pressed `void F5Pressed () [signal]`

17.12.2.26 F6Pressed `void F6Pressed () [signal]`

17.12.2.27 F7Pressed void F7Pressed () [signal]

17.12.2.28 F8Pressed void F8Pressed () [signal]

17.12.2.29 F9Pressed void F9Pressed () [signal]

17.12.2.30 pasteClip void pasteClip () [private], [slot]

[CmdPromptInput::pasteClip.](#)

17.12.2.31 pastePressed void pastePressed () [signal]

17.12.2.32 processInput void processInput (
 void) [slot]

[CmdPromptInput::processInput.](#)

17.12.2.33 redoPressed void redoPressed () [signal]

17.12.2.34 runCommand void runCommand (
 QString cmd,
 QString cmdtxt) [signal]

17.12.2.35 selectAllPressed void selectAllPressed () [signal]

17.12.2.36 shiftPressed void shiftPressed () [signal]

17.12.2.37 shiftReleased `void shiftReleased () [signal]`

17.12.2.38 showSettings `void showSettings () [signal]`

17.12.2.39 startCommand `void startCommand (
 QString cmd) [signal]`

17.12.2.40 stopBlinking `void stopBlinking () [signal]`

17.12.2.41 tabPressed `void tabPressed () [signal]`

17.12.2.42 undoPressed `void undoPressed () [signal]`

17.12.2.43 updateCurrentText `void updateCurrentText (
 QString txt) [slot]`

[CmdPromptInput::updateCurrentText.](#)

Parameters

<i>txt</i>	
------------	--

17.12.2.44 upPressed `void upPressed () [signal]`

17.12.3 Member Data Documentation

17.12.3.1 cmdActive `bool cmdActive`

17.12.3.2 curCmd `QString curCmd`

17.12.3.3 curText `QString curText`

17.12.3.4 defaultPrefix `QString defaultPrefix`

17.12.3.5 isBlinking `bool isBlinking`

17.12.3.6 lastCmd `QString lastCmd`

17.12.3.7 prefix `QString prefix`

17.12.3.8 rapidFireEnabled `bool rapidFireEnabled`

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

- [embroidermodder2/embroidermodder.h](#)
- [embroidermodder2/cmdprompt.cpp](#)

17.13 CmdPromptSplitter Class Reference

```
#include <embroidermodder.h>
```

Signals

- void [pressResizeHistory](#) (int y)
- void [releaseResizeHistory](#) (int y)
- void [moveResizeHistory](#) (int y)

Public Member Functions

- [CmdPromptSplitter](#) (QWidget *parent=0)
CmdPromptSplitter::CmdPromptSplitter.
- [~CmdPromptSplitter](#) ()
CmdPromptSplitter::~~CmdPromptSplitter.

Protected Member Functions

- `QSplitterHandle * createHandle ()`
[CmdPromptSplitter::createHandle.](#)

17.13.1 Detailed Description

17.13.2 Constructor & Destructor Documentation

17.13.2.1 CmdPromptSplitter() `CmdPromptSplitter (`
`QWidget * parent = 0)`

[CmdPromptSplitter::CmdPromptSplitter.](#)

Parameters

<i>parent</i>	
---------------	--

17.13.2.2 ~CmdPromptSplitter() `~CmdPromptSplitter ()`

[CmdPromptSplitter::~~CmdPromptSplitter.](#)

17.13.3 Member Function Documentation

17.13.3.1 createHandle() `QSplitterHandle * createHandle ()` [protected]

[CmdPromptSplitter::createHandle.](#)

Returns

17.13.3.2 moveResizeHistory `void moveResizeHistory (`
`int y)` [signal]

17.13.3.3 pressResizeHistory void pressResizeHistory (
int y) [signal]

17.13.3.4 releaseResizeHistory void releaseResizeHistory (
int y) [signal]

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

- embroidermodder2/[embroidermodder.h](#)
- embroidermodder2/[cmdprompt.cpp](#)

17.14 Compress Struct Reference

```
#include <embroidery_internal.h>
```

Public Attributes

- int [bit_position](#)
- char * [input_data](#)
- int [input_length](#)
- int [bits_total](#)
- int [block_elements](#)
- [huffman](#) [character_length_huffman](#)
- [huffman](#) [character_huffman](#)
- [huffman](#) [distance_huffman](#)

17.14.1 Member Data Documentation

17.14.1.1 bit_position int bit_position

17.14.1.2 bits_total int bits_total

17.14.1.3 block_elements int block_elements

17.14.1.4 character_huffman [huffman](#) character_huffman

17.14.1.5 character_length_huffman [huffman](#) character_length_huffman

17.14.1.6 distance_huffman [huffman](#) distance_huffman

17.14.1.7 input_data char* input_data

17.14.1.8 input_length int input_length

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

- extern/libembroidery/src/[embroidery_internal.h](#)

17.15 EmbAlignedDim_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- [EmbVector position](#)

17.15.1 Member Data Documentation

17.15.1.1 position [EmbVector](#) position

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

- extern/libembroidery/src/[embroidery.h](#)

17.16 EmbAngularDim_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- [EmbVector position](#)

17.16.1 Member Data Documentation

17.16.1.1 position [EmbVector](#) position

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

- [extern/libembroidery/src/embroidery.h](#)

17.17 EmbArc_ Struct Reference

absolute position (not relative)

```
#include <embroidery.h>
```

Public Attributes

- [EmbVector](#) start
- [EmbVector](#) mid
- [EmbVector](#) end

17.17.1 Detailed Description

absolute position (not relative)

17.17.2 Member Data Documentation

17.17.2.1 end [EmbVector](#) end

17.17.2.2 mid [EmbVector](#) mid

17.17.2.3 start [EmbVector](#) start

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

- [extern/libembroidery/src/embroidery.h](#)

17.18 EmbArcLengthDim_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- [EmbVector](#) position

17.18.1 Member Data Documentation

17.18.1.1 position [EmbVector](#) position

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

- extern/libembroidery/src/[embroidery.h](#)

17.19 EmbArray_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- [EmbGeometry](#) * geometry
- [EmbStitch](#) * stitch
- [EmbThread](#) * thread
- int count
- int length
- int type

17.19.1 Member Data Documentation

17.19.1.1 count int count

17.19.1.2 geometry [EmbGeometry](#)* geometry

17.19.1.3 length `int length`

17.19.1.4 stitch `EmbStitch* stitch`

17.19.1.5 thread `EmbThread* thread`

17.19.1.6 type `int type`

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

- `extern/libembroidery/src/embroidery.h`

17.20 EmbBezier_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- `EmbVector start`
- `EmbVector control1`
- `EmbVector control2`
- `EmbVector end`

17.20.1 Member Data Documentation

17.20.1.1 control1 `EmbVector control1`

17.20.1.2 control2 `EmbVector control2`

17.20.1.3 end `EmbVector end`

17.20.1.4 start `EmbVector` start

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

- `extern/libembroidery/src/embroidery.h`

17.21 EmbBlock_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- `EmbVector` position

17.21.1 Member Data Documentation

17.21.1.1 position `EmbVector` position

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

- `extern/libembroidery/src/embroidery.h`

17.22 EmbCircle_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- `EmbVector` center
- `EmbReal` radius

17.22.1 Member Data Documentation

17.22.1.1 center `EmbVector` center

17.22.1.2 radius `EmbReal` radius

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

- `extern/libembroidery/src/embroidery.h`

17.23 EmbColor_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- unsigned char `r`
- unsigned char `g`
- unsigned char `b`

17.23.1 Detailed Description

EmbColor uses the light primaries: red, green, blue in that order.

17.23.2 Member Data Documentation

17.23.2.1 `b` unsigned char `b`

17.23.2.2 `g` unsigned char `g`

17.23.2.3 `r` unsigned char `r`

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

- `extern/libembroidery/src/embroidery.h`

17.24 EmbDetailsDialog Class Reference

```
#include <embroidermodder.h>
```

Public Member Functions

- [EmbDetailsDialog](#) (QGraphicsScene *theScene, QWidget *parent=0)
[EmbDetailsDialog::EmbDetailsDialog.](#)
- [~EmbDetailsDialog](#) ()
[EmbDetailsDialog::~~EmbDetailsDialog.](#)
- void [getInfo](#) ()
[EmbDetailsDialog::getInfo.](#)
- QWidget * [createMainWidget](#) ()
[EmbDetailsDialog::createMainWidget.](#)
- QWidget * [createHistogram](#) ()

Public Attributes

- QWidget * [mainWidget](#)
- QDialogButtonBox * [buttonBox](#)
- uint32_t [stitchesTotal](#)
- uint32_t [stitchesReal](#)
- uint32_t [stitchesJump](#)
- uint32_t [stitchesTrim](#)
- uint32_t [colorTotal](#)
- uint32_t [colorChanges](#)
- QRectF [boundingRect](#)

17.24.1 Detailed Description

17.24.2 Constructor & Destructor Documentation

17.24.2.1 EmbDetailsDialog() [EmbDetailsDialog](#) (
 QGraphicsScene * *theScene*,
 QWidget * *parent* = 0)

[EmbDetailsDialog::EmbDetailsDialog.](#)

Parameters

<i>theScene</i>	
<i>parent</i>	

17.24.2.2 ~EmbDetailsDialog() [~EmbDetailsDialog](#) ()

[EmbDetailsDialog::~~EmbDetailsDialog.](#)

17.24.3 Member Function Documentation

17.24.3.1 createHistogram() `QWidget * createHistogram ()`

17.24.3.2 createMainWidget() `QWidget * createMainWidget ()`

[EmbDetailsDialog::createMainWidget](#).

Returns

17.24.3.3 getInfo() `void getInfo ()`

[EmbDetailsDialog::getInfo](#).

17.24.4 Member Data Documentation

17.24.4.1 boundingRect `QRectF boundingRect`

17.24.4.2 buttonBox `QDialogButtonBox* buttonBox`

17.24.4.3 colorChanges `uint32_t colorChanges`

17.24.4.4 colorTotal `uint32_t colorTotal`

17.24.4.5 mainWidget `QWidget* mainWidget`

17.24.4.6 stitchesJump `uint32_t stitchesJump`

17.24.4.7 stitchesReal `uint32_t stitchesReal`

17.24.4.8 stitchesTotal `uint32_t stitchesTotal`

17.24.4.9 stitchesTrim `uint32_t stitchesTrim`

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

- embroidermodder2/[embroidermodder.h](#)
- embroidermodder2/[embdetails-dialog.cpp](#)

17.25 EmbDiameterDim_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- [EmbVector](#) position

17.25.1 Member Data Documentation

17.25.1.1 position [EmbVector](#) position

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

- extern/libembroidery/src/[embroidery.h](#)

17.26 EmbEllipse_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- [EmbVector](#) center
- [EmbVector](#) radius
- [EmbReal](#) rotation

17.26.1 Member Data Documentation

17.26.1.1 center [EmbVector](#) center

17.26.1.2 radius [EmbVector](#) radius

17.26.1.3 rotation [EmbReal](#) rotation

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

- extern/libembroidery/src/[embroidery.h](#)

17.27 EmbFormatList_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- char [extension](#) [2+EMBFORMAT_MAXEXT]
- char [description](#) [EMBFORMAT_MAXDESC]
- char [reader_state](#)
- char [writer_state](#)
- int [type](#)
- int [color_only](#)
- int [check_for_color_file](#)
- int [write_external_color_file](#)

17.27.1 Member Data Documentation

17.27.1.1 check_for_color_file int check_for_color_file

17.27.1.2 color_only int color_only

17.27.1.3 description `char description[EMBFORMAT_MAXDESC]`

17.27.1.4 extension `char extension[2+EMBFORMAT_MAXEXT]`

17.27.1.5 reader_state `char reader_state`

17.27.1.6 type `int type`

17.27.1.7 write_external_color_file `int write_external_color_file`

17.27.1.8 writer_state `char writer_state`

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

- `extern/libembroidery/src/embroidery.h`

17.28 EmbGeometry_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- union {
 - `EmbArc arc`
 - `EmbCircle circle`
 - `EmbColor color`
 - `EmbEllipse ellipse`
 - `EmbLine line`
 - `EmbPath path`
 - `EmbPoint point`
 - `EmbPolygon polygon`
 - `EmbPolyline polyline`
 - `EmbRect rect`
 - `EmbSpline spline`
 - `EmbVector vector`
- `object`
- `EmbStitch stitch`
- `EmbThread thread`
- `int flag`
- `int type`
- `int lineType`

17.28.1 Member Data Documentation

17.28.1.1 arc `EmbArc` arc

17.28.1.2 circle `EmbCircle` circle

17.28.1.3 color `EmbColor` color

17.28.1.4 ellipse `EmbEllipse` ellipse

17.28.1.5 flag `int` flag

17.28.1.6 line `EmbLine` line

17.28.1.7 lineType `int` lineType

17.28.1.8 `union { ... }` object

17.28.1.9 path `EmbPath` path

17.28.1.10 point `EmbPoint` point

17.28.1.11 **polygon** [EmbPolygon](#) polygon

17.28.1.12 **polyline** [EmbPolyline](#) polyline

17.28.1.13 **rect** [EmbRect](#) rect

17.28.1.14 **spline** [EmbSpline](#) spline

17.28.1.15 **stitch** [EmbStitch](#) stitch

17.28.1.16 **thread** [EmbThread](#) thread

17.28.1.17 **type** int type

17.28.1.18 **vector** [EmbVector](#) vector

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

- [extern/libembroidery/src/embroidery.h](#)

17.29 EmblImage_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- [EmbVector](#) position
- [EmbVector](#) dimensions
- unsigned char * [data](#)
- int [width](#)
- int [height](#)
- char [path](#) [200]
- char [name](#) [200]

17.29.1 Member Data Documentation

17.29.1.1 data `unsigned char* data`

17.29.1.2 dimensions `EmbVector dimensions`

17.29.1.3 height `int height`

17.29.1.4 name `char name[200]`

17.29.1.5 path `char path[200]`

17.29.1.6 position `EmbVector position`

17.29.1.7 width `int width`

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

- `extern/libembroidery/src/embroidery.h`

17.30 EmblInfiniteLine_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- `EmbVector position`

17.30.1 Member Data Documentation

17.30.1.1 position [EmbVector](#) position

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

- [extern/libembroidery/src/embroidery.h](#)

17.31 EmbLayer_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- char [name](#) [100]
- [EmbArray](#) * [geometry](#)

17.31.1 Member Data Documentation

17.31.1.1 geometry [EmbArray](#)* geometry

17.31.1.2 name char name[100]

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

- [extern/libembroidery/src/embroidery.h](#)

17.32 EmbLeaderDim_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- [EmbVector](#) position

17.32.1 Member Data Documentation

17.32.1.1 position `EmbVector` position

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

- `extern/libembroidery/src/embroidery.h`

17.33 EmbLine_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- `EmbVector` start
- `EmbVector` end
- `int` lineType
- `EmbColor` color

17.33.1 Member Data Documentation

17.33.1.1 color `EmbColor` color

17.33.1.2 end `EmbVector` end

17.33.1.3 lineType `int` lineType

17.33.1.4 start `EmbVector` start

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

- `extern/libembroidery/src/embroidery.h`

17.34 EmbLinearDim_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- [EmbVector position](#)

17.34.1 Member Data Documentation

17.34.1.1 position [EmbVector](#) position

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

- extern/libembroidery/src/[embroidery.h](#)

17.35 EmbOrdinateDim_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- [EmbVector position](#)

17.35.1 Member Data Documentation

17.35.1.1 position [EmbVector](#) position

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

- extern/libembroidery/src/[embroidery.h](#)

17.36 EmbPath_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- [EmbArray](#) * [pointList](#)
- [EmbArray](#) * [flagList](#)
- int [lineType](#)
- [EmbColor](#) [color](#)

17.36.1 Member Data Documentation

17.36.1.1 [color](#) [EmbColor](#) [color](#)

17.36.1.2 [flagList](#) [EmbArray](#)* [flagList](#)

17.36.1.3 [lineType](#) int [lineType](#)

17.36.1.4 [pointList](#) [EmbArray](#)* [pointList](#)

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

- extern/libembroidery/src/[embroidery.h](#)

17.37 EmbPattern_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- unsigned int [dstJumpsPerTrim](#)
- [EmbVector](#) [home](#)
- [EmbReal](#) [hoop_width](#)
- [EmbReal](#) [hoop_height](#)
- [EmbArray](#) * [thread_list](#)
- [EmbArray](#) * [stitch_list](#)
- [EmbArray](#) * [geometry](#)
- [EmbLayer](#) [layer](#) [[EMB_MAX_LAYERS](#)]
- int [currentColorIndex](#)

17.37.1 Member Data Documentation

17.37.1.1 `currentColorIndex` `int` `currentColorIndex`

17.37.1.2 `dstJumpsPerTrim` `unsigned int` `dstJumpsPerTrim`

17.37.1.3 `geometry` `EmbArray*` `geometry`

17.37.1.4 `home` `EmbVector` `home`

17.37.1.5 `hoop_height` `EmbReal` `hoop_height`

17.37.1.6 `hoop_width` `EmbReal` `hoop_width`

17.37.1.7 `layer` `EmbLayer` `layer[EMB_MAX_LAYERS]`

17.37.1.8 `stitch_list` `EmbArray*` `stitch_list`

17.37.1.9 `thread_list` `EmbArray*` `thread_list`

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

- `extern/libembroidery/src/embroidery.h`

17.38 EmbPoint_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- [EmbVector](#) position
- int [lineType](#)
- [EmbColor](#) color

17.38.1 Member Data Documentation

17.38.1.1 color [EmbColor](#) color

17.38.1.2 lineType int lineType

17.38.1.3 position [EmbVector](#) position

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

- extern/libembroidery/src/[embroidery.h](#)

17.39 EmbRadiusDim_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- [EmbVector](#) position

17.39.1 Member Data Documentation

17.39.1.1 position [EmbVector](#) position

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

- extern/libembroidery/src/[embroidery.h](#)

17.40 EmbRay_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- [EmbVector](#) position

17.40.1 Member Data Documentation

17.40.1.1 position [EmbVector](#) position

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

- extern/libembroidery/src/[embroidery.h](#)

17.41 EmbRect_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- [EmbReal](#) top
- [EmbReal](#) left
- [EmbReal](#) bottom
- [EmbReal](#) right
- [EmbReal](#) rotation
- [EmbReal](#) radius

17.41.1 Member Data Documentation

17.41.1.1 bottom [EmbReal](#) bottom

17.41.1.2 left [EmbReal](#) left

17.41.1.3 radius [EmbReal](#) radius

17.41.1.4 right [EmbReal](#) right

17.41.1.5 rotation [EmbReal](#) rotation

17.41.1.6 top [EmbReal](#) top

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

- [extern/libembroidery/src/embroidery.h](#)

17.42 EmbSatinOutline_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- [int](#) [length](#)
- [EmbArray](#) * [side1](#)
- [EmbArray](#) * [side2](#)

17.42.1 Member Data Documentation

17.42.1.1 length [int](#) length

17.42.1.2 side1 [EmbArray](#)* side1

17.42.1.3 side2 [EmbArray](#)* side2

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

- [extern/libembroidery/src/embroidery.h](#)

17.43 EmbSpline_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- [EmbArray](#) * [beziers](#)

17.43.1 Member Data Documentation

17.43.1.1 [beziers](#) [EmbArray](#)* [beziers](#)

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

- [extern/libembroidery/src/embroidery.h](#)

17.44 EmbStitch_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- [int](#) [flags](#)
- [EmbReal](#) [x](#)
- [EmbReal](#) [y](#)
- [int](#) [color](#)

17.44.1 Member Data Documentation

17.44.1.1 [color](#) [int](#) [color](#)

positive is up, units are in mm

17.44.1.2 [flags](#) [int](#) [flags](#)

17.44.1.3 `x` `EmbReal` `x`

uses codes defined above

17.44.1.4 `y` `EmbReal` `y`

absolute position (not relative)

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

- `extern/libembroidery/src/embroidery.h`

17.45 EmbTextMulti_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- `EmbVector` `position`
- `char` `text` [200]

17.45.1 Member Data Documentation**17.45.1.1** `position` `EmbVector` `position`**17.45.1.2** `text` `char` `text` [200]

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

- `extern/libembroidery/src/embroidery.h`

17.46 EmbTextSingle_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- `EmbVector` `position`
- `char` `text` [200]

17.46.1 Member Data Documentation

17.46.1.1 position `EmbVector position`

17.46.1.2 text `char text[200]`

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

- `extern/libembroidery/src/embroidery.h`

17.47 EmbThread_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- `EmbColor color`
- `char description[50]`
- `char catalogNumber[30]`

17.47.1 Member Data Documentation

17.47.1.1 catalogNumber `char catalogNumber[30]`

17.47.1.2 color `EmbColor color`

17.47.1.3 description `char description[50]`

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

- `extern/libembroidery/src/embroidery.h`

17.48 EmbTime_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- unsigned int [year](#)
- unsigned int [month](#)
- unsigned int [day](#)
- unsigned int [hour](#)
- unsigned int [minute](#)
- unsigned int [second](#)

17.48.1 Member Data Documentation

17.48.1.1 day unsigned int day

17.48.1.2 hour unsigned int hour

17.48.1.3 minute unsigned int minute

17.48.1.4 month unsigned int month

17.48.1.5 second unsigned int second

17.48.1.6 year unsigned int year

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

- extern/libembroidery/src/[embroidery.h](#)

17.49 EmbVector_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- [EmbReal](#) x
- [EmbReal](#) y

17.49.1 Detailed Description

The basic type to represent points absolutely or represent directions.

Positive y is up, units are in mm.

17.49.2 Member Data Documentation

17.49.2.1 **x** [EmbReal](#) x

17.49.2.2 **y** [EmbReal](#) y

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

- extern/libembroidery/src/[embroidery.h](#)

17.50 Geometry Class Reference

The [Geometry](#) class.

```
#include <embroidermodder.h>
```

Public Types

- enum [ArrowStyle](#) {
 [NoArrow](#) , [Open](#) , [Closed](#) , [Dot](#) ,
 [Box](#) , [Tick](#) }
- enum [lineStyle](#) { [NoLine](#) , [Flared](#) , [Fletching](#) }

Public Member Functions

- virtual int [type](#) ()
- [Geometry](#) (int object_type=[OBJ_TYPE_BASE](#), QGraphicsItem *parent=0)
- [Geometry](#) ([Geometry](#) *obj, QGraphicsItem *parent=0)
- [Geometry](#) ([EmbArc](#) arc, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem *parent=0)
- [Geometry](#) ([EmbCircle](#) circle, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem *parent=0)
- [Geometry](#) ([EmbLine](#) line, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem *parent=0)
- [Geometry](#) ([EmbEllipse](#) ellipse, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem *parent=0)
- [Geometry](#) ([EmbRect](#) rect, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem *parent=0)
- [Geometry](#) (QString str, [EmbVector](#) position, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem *parent=0)
- [Geometry](#) ([EmbLine](#) line, int Type_, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem *parent)
- [Geometry](#) (QPainterPath p, int type_, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem *parent=0)
- [Geometry](#) ([EmbVector](#) pos, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem *parent=0)
- void [init_arc](#) ([EmbArc](#) arc, QRgb rgb, Qt::PenStyle lineType)
- [Geometry::init.](#)
- void [init_circle](#) ([EmbCircle](#) circle, QRgb rgb, Qt::PenStyle lineType)
- void [init_line](#) ([EmbLine](#) line, QRgb rgb, Qt::PenStyle lineType)
- void [init_ellipse](#) ([EmbEllipse](#) ellipse, QRgb rgb, Qt::PenStyle lineType)
- void [init_rect](#) ([EmbRect](#) rect, QRgb rgb, Qt::PenStyle lineType)
- void [init_text_single](#) (QString str, [EmbVector](#) position, QRgb rgb, Qt::PenStyle lineType)
- void [init_path](#) (QPainterPath p, QRgb rgb, Qt::PenStyle lineType)
- void [init_point](#) ([EmbVector](#) pos, QRgb rgb, Qt::PenStyle lineType)
- void [init](#) (void)
- [~Geometry](#) ()
- [Geometry::~~Geometry.](#)
- Qt::PenStyle [objectLineType](#) ()
- [EmbReal](#) [objectLineWeight](#) ()
- QPointF [objectRubberPoint](#) (QString key)
- [Geometry::objectRubberPoint.](#)
- QString [objectRubberText](#) (QString key)
- [Geometry::objectRubberText.](#)
- QPointF [objectCenter](#) ()
- QPointF [objectPos](#) ()
- [EmbReal](#) [objectX](#) ()
- [EmbReal](#) [objectY](#) ()
- QPointF [objectTopLeft](#) ()
- QPointF [objectTopRight](#) ()
- QPointF [objectBottomLeft](#) ()
- QPointF [objectBottomRight](#) ()
- [EmbReal](#) [objectArea](#) ()
- [Geometry::objectArea.](#)
- QPointF [objectStartPoint](#) ()
- [Geometry::objectStartPoint.](#)
- QPointF [objectMidPoint](#) ()
- [Geometry::objectMidPoint.](#)
- QPointF [objectEndPoint](#) ()
- [Geometry::objectEndPoint.](#)
- QRectF [rect](#) ()
- void [circle_click](#) (Dictionary global, [EmbVector](#) v)
- [EmbReal](#) [objectWidth](#) ()
- [EmbReal](#) [objectHeight](#) ()
- [EmbReal](#) [objectRadiusMajor](#) ()

- [EmbReal objectRadiusMinor \(\)](#)
- [EmbReal objectDiameterMajor \(\)](#)
- [EmbReal objectDiameterMinor \(\)](#)
- [QPointF objectEndPoint1 \(\)](#)
DimLeaderObject::objectEndPoint1.
- [QPointF objectEndPoint2 \(\)](#)
Geometry::objectEndPoint2.
- [EmbReal objectStartAngle \(\)](#)
Geometry::objectStartAngle.
- [EmbReal objectEndAngle \(\)](#)
Geometry::objectEndAngle.
- [EmbReal objectArcLength \(\)](#)
Geometry::objectArcLength.
- [EmbReal objectChord \(\)](#)
Geometry::objectChord.
- [EmbReal objectIncludedAngle \(\)](#)
Geometry::objectIncludedAngle.
- [bool objectClockwise \(\)](#)
Geometry::objectClockwise.
- [EmbReal objectX1 \(\)](#)
- [EmbReal objectY1 \(\)](#)
- [EmbReal objectX2 \(\)](#)
- [EmbReal objectY2 \(\)](#)
- [EmbReal objectAngle \(\)](#)
DimLeaderObject::objectAngle.
- [QPointF objectDelta \(\)](#)
- [EmbReal objectLength \(\)](#)
- [EmbReal objectRadius \(\)](#)
- [EmbReal objectDiameter \(\)](#)
- [EmbReal objectCircumference \(\)](#)
- [QPointF objectQuadrant0 \(\)](#)
- [QPointF objectQuadrant90 \(\)](#)
- [QPointF objectQuadrant180 \(\)](#)
- [QPointF objectQuadrant270 \(\)](#)
- [QPainterPath objectCopyPath \(\)](#)
PathObject::objectCopyPath.
- [QPainterPath objectSavePath \(\)](#)
Geometry::objectSavePath.
- [std::vector< QPainterPath > objectSavePathList \(\)](#)
- [std::vector< QPainterPath > subPathList \(\)](#)
- [int findIndex \(const QPointF &point\)](#)
Geometry::findIndex.
- [void setObjectEndPoint1 \(EmbVector endPt1\)](#)
DimLeaderObject::setObjectEndPoint1.
- [void setObjectEndPoint2 \(EmbVector endPt2\)](#)
DimLeaderObject::setObjectEndPoint2.
- [void updatePath \(\)](#)
Geometry::updatePath.
- [void updatePath \(const QPainterPath &p\)](#)
Geometry::updatePath.
- [void updateLeader \(void\)](#)
DimLeaderObject::updateLeader.

- virtual QRectF [boundingRect](#) ()
- void [drawRubberLine](#) (const QLineF &rubLine, QPainter *painter=0, const char *colorFromScene=0)
Geometry::drawRubberLine.
- void [updateRubber](#) (QPainter *painter=0)
DimLeaderObject::updateRubber.
- void [vulcanize](#) (void)
DimLeaderObject::vulcanize.
- QPointF [mouseSnapPoint](#) (const QPointF &mousePoint)
Geometry::mouseSnapPoint.
- std::vector< QPointF > [allGripPoints](#) ()
Geometry::allGripPoints.
- void [gripEdit](#) (const QPointF &before, const QPointF &after)
DimLeaderObject::gripEdit.
- void [realRender](#) (QPainter *painter, const QPainterPath &renderPath)
Geometry::realRender.
- void [paint](#) (QPainter *, const QStyleOptionGraphicsItem *, QWidget *)
Geometry::paint.
- void [calculateArcData](#) (EmbArc arc)
Geometry::calculateArcData.
- void [updateArcRect](#) (EmbReal radius)
Geometry::updateArcRect.
- void [setObjectPos](#) (const QPointF &point)
- void [setObjectX](#) (EmbReal x)
- void [setObjectY](#) (EmbReal y)
- void [setObjectCenter](#) (EmbVector center)
- void [setObjectCenterX](#) (EmbReal centerX)
- void [setObjectCenterY](#) (EmbReal centerY)
- void [setObjectSize](#) (EmbReal width, EmbReal height)
- void [setObjectRect](#) (EmbReal x, EmbReal y, EmbReal w, EmbReal h)
- void [setRect](#) (const QRectF &r)
- void [setRect](#) (EmbReal x, EmbReal y, EmbReal w, EmbReal h)
- void [setLine](#) (const QLineF &li)
- void [setLine](#) (EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2)
- void [setObjectLineWeight](#) (String lineWeight)
Geometry::setObjectLineWeight.
- void [setObjectRadius](#) (EmbReal radius)
Geometry::setObjectRadius.
- void [setObjectStartAngle](#) (EmbReal angle)
Geometry::setObjectStartAngle.
- void [setObjectEndAngle](#) (EmbReal angle)
Geometry::setObjectEndAngle.
- void [setObjectStartPoint](#) (EmbVector point)
Geometry::setObjectStartPoint.
- void [setObjectMidPoint](#) (EmbVector point)
Geometry::setObjectMidPoint.
- void [setObjectEndPoint](#) (EmbVector point)
Geometry::setObjectEndPoint.
- void [setObjectDiameter](#) (EmbReal diameter)
Geometry::setObjectDiameter.
- void [setObjectArea](#) (EmbReal area)
Geometry::setObjectArea.
- void [setObjectCircumference](#) (EmbReal circumference)

Geometry::setObjectCircumference.

- void [setObjectPos](#) ([EmbReal](#) x, [EmbReal](#) y)
 - void [setObjectText](#) ([QString](#) str)
 - void [setObjectTextFont](#) ([QString](#) font)
 - void [setObjectTextJustify](#) ([QString](#) justify)
 - void [setObjectTextSize](#) ([EmbReal](#) size)
 - void [setObjectTextStyle](#) (bool bold, bool italic, bool under, bool strike, bool over)
 - void [setObjectTextBold](#) (bool val)
 - void [setObjectTextItalic](#) (bool val)
 - void [setObjectTextUnderline](#) (bool val)
 - void [setObjectTextStrikeOut](#) (bool val)
 - void [setObjectTextOverline](#) (bool val)
 - void [setObjectTextBackward](#) (bool val)
 - void [setObjectTextUpsideDown](#) (bool val)
 - void [setObjectRadiusMajor](#) ([EmbReal](#) radius)
 - void [setObjectRadiusMinor](#) ([EmbReal](#) radius)
 - void [setObjectDiameterMajor](#) ([EmbReal](#) diameter)
 - void [setObjectDiameterMinor](#) ([EmbReal](#) diameter)
 - void [script_main](#) (void)
 - void [script_click](#) ([EmbVector](#) v)
- circle_click*
- void [script_context](#) ([String](#) str)
 - void [script_prompt](#) ([String](#) str)

Public Attributes

- Dictionary properties
- [QPen](#) [objPen](#)
- [QPen](#) [lwtPen](#)
- [QLineF](#) [objLine](#)
- [String](#) [objRubberMode](#) = "OBJ_RUBBER_OFF"
- [QHash](#)< [QString](#), [QPointF](#) > [objRubberPoints](#)
- [QHash](#)< [QString](#), [QString](#) > [objRubberTexts](#)
- [int64_t](#) [objID](#)
- [QPointF](#) [arcStartPoint](#)
- [QPointF](#) [arcMidPoint](#)
- [QPointF](#) [arcEndPoint](#)
- bool [curved](#)
- bool [filled](#)
- [QPainterPath](#) [lineStylePath](#)
- [QPainterPath](#) [arrowStylePath](#)
- [EmbReal](#) [arrowStyleAngle](#)
- [EmbReal](#) [arrowStyleLength](#)
- [EmbReal](#) [lineStyleAngle](#)
- [EmbReal](#) [lineStyleLength](#)
- [QPainterPath](#) [normalPath](#)
- [QString](#) [objText](#)
- [QString](#) [objTextFont](#)
- [QString](#) [objTextJustify](#)
- bool [objTextBackward](#)
- bool [objTextUpsideDown](#)
- [QPainterPath](#) [objTextPath](#)
- [std::vector](#)< [EmbReal](#) > [x_values](#)
- [std::vector](#)< [EmbReal](#) > [y_values](#)
- [int](#) [gripIndex](#)
- [int](#) [Type](#) = OBJ_TYPE_BASE

17.50.1 Detailed Description

The [Geometry](#) class.

Combine all geometry objects into one class that uses the Type flag to determine the behaviour of overlapping functions and bar the use of nonsensical function calls.

17.50.2 Member Enumeration Documentation

17.50.2.1 ArrowStyle enum [ArrowStyle](#)

Enumerator

NoArrow	
Open	
Closed	
Dot	
Box	
Tick	

17.50.2.2 lineStyle enum [lineStyle](#)

Enumerator

NoLine	
Flared	
Fletching	

17.50.3 Constructor & Destructor Documentation

17.50.3.1 Geometry() [1/11] [Geometry](#) (
 int *object_type* = [OBJ_TYPE_BASE](#),
 QGraphicsItem * *parent* = 0)

17.50.3.2 Geometry() [2/11] [Geometry](#) (
 [Geometry](#) * *obj*,
 QGraphicsItem * *parent* = 0)

17.50.3.3 Geometry() [3/11] `Geometry (`
 `EmbArc arc,`
 `QRgb rgb,`
 `Qt::PenStyle lineType,`
 `QGraphicsItem * parent = 0)`

17.50.3.4 Geometry() [4/11] `Geometry (`
 `EmbCircle circle,`
 `QRgb rgb,`
 `Qt::PenStyle lineType,`
 `QGraphicsItem * parent = 0)`

17.50.3.5 Geometry() [5/11] `Geometry (`
 `EmbLine line,`
 `QRgb rgb,`
 `Qt::PenStyle lineType,`
 `QGraphicsItem * parent = 0)`

17.50.3.6 Geometry() [6/11] `Geometry (`
 `EmbEllipse ellipse,`
 `QRgb rgb,`
 `Qt::PenStyle lineType,`
 `QGraphicsItem * parent = 0)`

17.50.3.7 Geometry() [7/11] `Geometry (`
 `EmbRect rect,`
 `QRgb rgb,`
 `Qt::PenStyle lineType,`
 `QGraphicsItem * parent = 0)`

17.50.3.8 Geometry() [8/11] `Geometry (`
 `QString str,`
 `EmbVector v,`
 `QRgb rgb,`
 `Qt::PenStyle lineType,`
 `QGraphicsItem * parent = 0)`

17.50.3.9 Geometry() [9/11] [Geometry](#) (
[EmbLine](#) *line*,
 int *Type_*,
 QColor *rgb*,
 Qt::PenStyle *lineType*,
 QGraphicsItem * *parent*)

17.50.3.10 Geometry() [10/11] [Geometry](#) (
 QPainterPath *p*,
 int *Type_*,
 QColor *rgb*,
 Qt::PenStyle *lineType*,
 QGraphicsItem * *parent* = 0)

For PATH, POLYLINE and POLYGON, set the *Type_* variable to one of these.

17.50.3.11 Geometry() [11/11] [Geometry](#) (
[EmbVector](#) *vector*,
 QColor *rgb*,
 Qt::PenStyle *lineType*,
 QGraphicsItem * *parent* = 0)

17.50.3.12 ~Geometry() [~Geometry](#) ()

[Geometry::~Geometry](#).

17.50.4 Member Function Documentation

17.50.4.1 allGripPoints() `std::vector< QPointF > allGripPoints ()`

[Geometry::allGripPoints](#).

Returns

17.50.4.2 boundingRect() `QRectF boundingRect ()` [virtual]

If gripped, force this object to be drawn even if it is offscreen.

17.50.4.3 calculateArcData() `void calculateArcData (
EmbArc arc)`

[Geometry::calculateArcData](#).

Parameters

<i>arc</i>	
------------	--

Todo convert this to update and make it Type sensitive.

17.50.4.4 circle_click() `void circle_click (`
 `Dictionary global,`
 `EmbVector v)`

17.50.4.5 drawRubberLine() `void drawRubberLine (`
 `const QLineF & rubLine,`
 `QPainter * painter = 0,`
 `const char * colorFromScene = 0)`

[Geometry::drawRubberLine.](#)

Parameters

<i>rubLine</i>	
<i>painter</i>	
<i>colorFromScene</i>	

17.50.4.6 findIndex() `int findIndex (`
 `const QPointF & point)`

[Geometry::findIndex.](#)

Parameters

<i>point</i>	
--------------	--

Returns

17.50.4.7 gripEdit() `void gripEdit (`
 `const QPointF & before,`
 `const QPointF & after)`

[DimLeaderObject::gripEdit.](#)

Parameters

<i>before</i>	
<i>after</i>	

17.50.4.8 init() `void init (`
 `void)`

17.50.4.9 init_arc() `void init_arc (`
 `EmbArc arc,`
 `QRgb rgb,`
 `Qt::PenStyle lineType)`

[Geometry::init.](#)

Parameters

<i>arc</i>	
<i>rgb</i>	
<i>lineType</i>	

WARNING: DO NOT enable `QGraphicsItem::ItemsMovable`. If it is enabled, WARNING: and the item is double clicked, the scene will erratically move the item while zooming. WARNING: All movement has to be handled explicitly by us, not by the scene.

17.50.4.10 init_circle() `void init_circle (`
 `EmbCircle circle,`
 `QRgb rgb,`
 `Qt::PenStyle lineType)`

WARNING: DO NOT enable `QGraphicsItem::ItemsMovable`. If it is enabled, WARNING: and the item is double clicked, the scene will erratically move the item while zooming. WARNING: All movement has to be handled explicitly by us, not by the scene.

17.50.4.11 init_ellipse() `void init_ellipse (`
 `EmbEllipse ellipse,`
 `QRgb rgb,`
 `Qt::PenStyle lineType)`

Warning

DO NOT enable `QGraphicsItem::ItemsMovable`. If it is enabled, and the item is double clicked, the scene will erratically move the item while zooming. All movement has to be handled explicitly by us, not by the scene.

17.50.4.12 **init_line()** `void init_line (`
 `EmbLine line,`
 `QRgb rgb,`
 `Qt::PenStyle lineType)`

17.50.4.13 **init_path()** `void init_path (`
 `QPainterPath p,`
 `QRgb rgb,`
 `Qt::PenStyle lineType)`

17.50.4.14 **init_point()** `void init_point (`
 `EmbVector pos,`
 `QRgb rgb,`
 `Qt::PenStyle lineType)`

17.50.4.15 **init_rect()** `void init_rect (`
 `EmbRect rect,`
 `QRgb rgb,`
 `Qt::PenStyle lineType)`

17.50.4.16 **init_text_single()** `void init_text_single (`
 `QString str,`
 `EmbVector position,`
 `QRgb rgb,`
 `Qt::PenStyle lineType)`

17.50.4.17 **mouseSnapPoint()** `QPointF mouseSnapPoint (`
 `const QPointF & mousePoint)`

[Geometry::mouseSnapPoint](#).

Parameters

<i>mousePoint</i>	
-------------------	--

Returns

the closest snap point to the mouse point.

17.50.4.18 objectAngle() [EmbReal](#) objectAngle ()

DimLeaderObject::objectAngle.

Returns

17.50.4.19 objectArcLength() [EmbReal](#) objectArcLength (
void)

[Geometry::objectArcLength](#).

Returns

17.50.4.20 objectArea() [EmbReal](#) objectArea ()

[Geometry::objectArea](#).

Returns

17.50.4.21 objectBottomLeft() [QPointF](#) objectBottomLeft ()

17.50.4.22 objectBottomRight() [QPointF](#) objectBottomRight ()

17.50.4.23 objectCenter() [QPointF](#) objectCenter () [inline]

17.50.4.24 objectChord() [EmbReal](#) objectChord (
void)

[Geometry::objectChord](#).

Returns

17.50.4.25 objectCircumference() [EmbReal](#) objectCircumference ()

17.50.4.26 objectClockwise() [bool](#) objectClockwise ()

[Geometry::objectClockwise.](#)

Returns

17.50.4.27 objectCopyPath() [QPainterPath](#) objectCopyPath ()

[PathObject::objectCopyPath.](#)

Returns

17.50.4.28 objectDelta() [QPointF](#) objectDelta () [inline]

17.50.4.29 objectDiameter() [EmbReal](#) objectDiameter ()

17.50.4.30 objectDiameterMajor() [EmbReal](#) objectDiameterMajor ()

17.50.4.31 objectDiameterMinor() [EmbReal](#) objectDiameterMinor ()

17.50.4.32 objectEndAngle() [EmbReal](#) objectEndAngle (
 void)

[Geometry::objectEndAngle.](#)

Returns

17.50.4.33 objectEndPoint() `QPointF objectEndPoint ()`

[Geometry::objectEndPoint.](#)

Returns

17.50.4.34 objectEndPoint1() `QPointF objectEndPoint1 ()`

[DimLeaderObject::objectEndPoint1.](#)

Returns

17.50.4.35 objectEndPoint2() `QPointF objectEndPoint2 ()`

[Geometry::objectEndPoint2.](#)

Returns

17.50.4.36 objectHeight() `EmbReal objectHeight ()`

17.50.4.37 objectIncludedAngle() `EmbReal objectIncludedAngle (void)`

[Geometry::objectIncludedAngle.](#)

Returns

17.50.4.38 objectLength() `EmbReal objectLength () [inline]`

17.50.4.39 objectLineType() `Qt::PenStyle objectLineType () [inline]`

17.50.4.40 objectLineWeight() `EmbReal objectLineWeight () [inline]`

17.50.4.41 objectMidPoint() `QPointF objectMidPoint ()`

[Geometry::objectMidPoint.](#)

Returns

17.50.4.42 objectPos() `QPointF objectPos () [inline]`

17.50.4.43 objectQuadrant0() `QPointF objectQuadrant0 ()`

17.50.4.44 objectQuadrant180() `QPointF objectQuadrant180 ()`

17.50.4.45 objectQuadrant270() `QPointF objectQuadrant270 ()`

17.50.4.46 objectQuadrant90() `QPointF objectQuadrant90 ()`

17.50.4.47 objectRadius() `EmbReal objectRadius ()`

17.50.4.48 objectRadiusMajor() `EmbReal objectRadiusMajor ()`

17.50.4.49 objectRadiusMinor() `EmbReal objectRadiusMinor ()`

17.50.4.50 objectRubberPoint() `QPointF objectRubberPoint (`
`QString key)`

[Geometry::objectRubberPoint.](#)

Parameters

<i>key</i>	
------------	--

Returns

17.50.4.51 objectRubberText() `QString objectRubberText (
 QString key)`

[Geometry::objectRubberText](#).

Parameters

<i>key</i>	
------------	--

Returns

17.50.4.52 objectSavePath() `QPainterPath objectSavePath ()`

[Geometry::objectSavePath](#).

Returns

17.50.4.53 objectSavePathList() `std::vector< QPainterPath > objectSavePathList () [inline]`

17.50.4.54 objectStartAngle() `EmbReal objectStartAngle (
 void)`

[Geometry::objectStartAngle](#).

Returns

17.50.4.55 objectStartPoint() `QPointF objectStartPoint ()`

[Geometry::objectStartPoint](#).

Returns

17.50.4.56 objectTopLeft() `QPointF objectTopLeft ()`

17.50.4.57 objectTopRight() `QPointF objectTopRight ()`

17.50.4.58 objectWidth() `EmbReal objectWidth ()`

17.50.4.59 objectX() `EmbReal objectX () [inline]`

17.50.4.60 objectX1() `EmbReal objectX1 () [inline]`

17.50.4.61 objectX2() `EmbReal objectX2 () [inline]`

17.50.4.62 objectY() `EmbReal objectY () [inline]`

17.50.4.63 objectY1() `EmbReal objectY1 () [inline]`

17.50.4.64 objectY2() `EmbReal objectY2 () [inline]`

17.50.4.65 paint() `void paint (`
 `QPainter * painter,`
 `const QStyleOptionGraphicsItem * option,`
 `QWidget *)`

[Geometry::paint](#).

Parameters

<i>painter</i>	
<i>option</i>	

17.50.4.66 realRender() `void realRender (QPainter * painter, const QPainterPath & renderPath)`

[Geometry::realRender](#).

Parameters

<i>painter</i>	
<i>renderPath</i>	

17.50.4.67 rect() `QRectF rect ()`

17.50.4.68 script_click() `void script_click (EmbVector v)`

circle_click

Returns

17.50.4.68.1 CIRCLE_MODE_1P_RAD mode For the circle object currently focussed, show two rubber points: one for the centre (the anchor) and the other at some point on the radius to adjust the radius.

17.50.4.68.2 CIRCLE_MODE_1P_DIA mode For the circle object currently focussed, show two rubber points: one for the left of the diameter and one for the right. These rubber points can be moved around the circle, but they always oppose one another.

17.50.4.69 script_context() `void script_context (String str)`

17.50.4.70 script_main() void script_main (
 void)

17.50.4.71 script_prompt() void script_prompt (
 String str)

17.50.4.72 setLine() [1/2] void setLine (
 const QLineF & li)

17.50.4.73 setLine() [2/2] void setLine (
 EmbReal x1,
 EmbReal y1,
 EmbReal x2,
 EmbReal y2)

17.50.4.74 setObjectArea() void setObjectArea (
 EmbReal area)

[Geometry::setObjectArea.](#)

Parameters

<i>area</i>	
-------------	--

17.50.4.75 setObjectCenter() void setObjectCenter (
 EmbVector center)

17.50.4.76 setObjectCenterX() void setObjectCenterX (
 EmbReal centerX)

17.50.4.77 setObjectCenterY() void setObjectCenterY (
 EmbReal centerY)

17.50.4.78 setObjectCircumference() `void setObjectCircumference (
 EmbReal circumference)`

[Geometry::setObjectCircumference.](#)

Parameters

<i>circumference</i>	
----------------------	--

17.50.4.79 setObjectDiameter() void setObjectDiameter (
 [EmbReal](#) *diameter*)

[Geometry::setObjectDiameter.](#)

Parameters

<i>diameter</i>	
-----------------	--

17.50.4.80 setObjectDiameterMajor() void setObjectDiameterMajor (
 [EmbReal](#) *diameter*)

17.50.4.81 setObjectDiameterMinor() void setObjectDiameterMinor (
 [EmbReal](#) *diameter*)

17.50.4.82 setObjectEndAngle() void setObjectEndAngle (
 [EmbReal](#) *angle*)

[Geometry::setObjectEndAngle.](#)

Parameters

<i>angle</i>	
--------------	--

17.50.4.83 setObjectEndPoint() void setObjectEndPoint (
 [EmbVector](#) *point*)

[Geometry::setObjectEndPoint.](#)

Parameters

<i>point</i>	
--------------	--

17.50.4.84 setObjectEndPoint1() `void setObjectEndPoint1 (
 EmbVector endPt1)`

`DimLeaderObject::setObjectEndPoint1.`

Parameters

<i>x1</i>	
<i>y1</i>	

17.50.4.85 setObjectEndPoint2() `void setObjectEndPoint2 (
 EmbVector endPt2)`

`DimLeaderObject::setObjectEndPoint2.`

Parameters

<i>x2</i>	
<i>y2</i>	

17.50.4.86 setObjectLineWeight() `void setObjectLineWeight (
 String lineWeight)`

[Geometry::setObjectLineWeight.](#)

Parameters

<i>lineWeight</i>	
-------------------	--

17.50.4.87 setObjectMidPoint() `void setObjectMidPoint (
 EmbVector point)`

[Geometry::setObjectMidPoint.](#)

Parameters

<i>point</i>	
--------------	--

17.50.4.88 setObjectPos() [1/2] void setObjectPos (
const QPointF & *point*) [inline]

17.50.4.89 setObjectPos() [2/2] void setObjectPos (
EmbReal *x*,
EmbReal *y*) [inline]

17.50.4.90 setObjectRadius() void setObjectRadius (
EmbReal *radius*)

[Geometry::setObjectRadius.](#)

Parameters

<i>radius</i>	
---------------	--

17.50.4.91 setObjectRadiusMajor() void setObjectRadiusMajor (
EmbReal *radius*)

17.50.4.92 setObjectRadiusMinor() void setObjectRadiusMinor (
EmbReal *radius*)

17.50.4.93 setObjectRect() void setObjectRect (
EmbReal *x*,
EmbReal *y*,
EmbReal *w*,
EmbReal *h*)

17.50.4.94 setObjectSize() void setObjectSize (
EmbReal *width*,
EmbReal *height*)

17.50.4.95 setObjectStartAngle() void setObjectStartAngle (
EmbReal *angle*)

[Geometry::setObjectStartAngle.](#)

Parameters

<i>angle</i>	
--------------	--

17.50.4.96 setObjectStartPoint() void setObjectStartPoint (
 [EmbVector](#) *point*)

[Geometry::setObjectStartPoint.](#)

Parameters

<i>point</i>	
--------------	--

17.50.4.97 setObjectText() void setObjectText (
 QString *str*)

17.50.4.98 setObjectTextBackward() void setObjectTextBackward (
 bool *val*)

17.50.4.99 setObjectTextBold() void setObjectTextBold (
 bool *val*)

17.50.4.100 setObjectTextFont() void setObjectTextFont (
 QString *font*)

17.50.4.101 setObjectTextItalic() void setObjectTextItalic (
 bool *val*)

17.50.4.102 setObjectTextJustify() void setObjectTextJustify (
 QString *justify*)

Verify the string is a valid option, otherwise default to "Left".

17.50.4.103 setObjectTextOverline() void setObjectTextOverline (
 bool val)

17.50.4.104 setObjectTextSize() void setObjectTextSize (
 EmbReal size)

17.50.4.105 setObjectTextStrikeOut() void setObjectTextStrikeOut (
 bool val)

17.50.4.106 setObjectTextStyle() void setObjectTextStyle (
 bool bold,
 bool italic,
 bool under,
 bool strike,
 bool over)

17.50.4.107 setObjectTextUnderline() void setObjectTextUnderline (
 bool val)

17.50.4.108 setObjectTextUpsideDown() void setObjectTextUpsideDown (
 bool val)

17.50.4.109 setObjectX() void setObjectX (
 EmbReal x) [inline]

17.50.4.110 setObjectY() void setObjectY (
 EmbReal y) [inline]

17.50.4.111 setRect() [1/2] void setRect (
 const QRectF & r)

17.50.4.112 setRect() [2/2] void setRect (
 EmbReal *x*,
 EmbReal *y*,
 EmbReal *w*,
 EmbReal *h*)

17.50.4.113 subPathList() std::vector< QPainterPath > subPathList ()

17.50.4.114 type() virtual int type () [inline], [virtual]

17.50.4.115 updateArcRect() void updateArcRect (
 EmbReal *radius*)

[Geometry::updateArcRect.](#)

Parameters

<i>radius</i>	
---------------	--

17.50.4.116 updateLeader() void updateLeader (
 void)

DimLeaderObject::updateLeader.

17.50.4.117 updatePath() [1/2] void updatePath ()

[Geometry::updatePath.](#)

For path and polyline set normalPath before calling.

17.50.4.118 updatePath() [2/2] void updatePath (
 const QPainterPath & *p*)

[Geometry::updatePath.](#)

Parameters

<i>p</i>	
----------	--

17.50.4.119 updateRubber() `void updateRubber (QPainter * painter = 0)`

DimLeaderObject::updateRubber.

Parameters

<i>painter</i>	
----------------	--

17.50.4.120 vulcanize() `void vulcanize (void)`

DimLeaderObject::vulcanize.

17.50.5 Member Data Documentation

17.50.5.1 arcEndPoint `QPointF arcEndPoint`

17.50.5.2 arcMidPoint `QPointF arcMidPoint`

17.50.5.3 arcStartPoint `QPointF arcStartPoint`

17.50.5.4 arrowStyleAngle `EmbReal arrowStyleAngle`

17.50.5.5 arrowStyleLength `EmbReal arrowStyleLength`

17.50.5.6 arrowStylePath `QPainterPath arrowStylePath`

17.50.5.7 curved `bool curved`

17.50.5.8 filled `bool filled`

17.50.5.9 gripIndex `int gripIndex`

17.50.5.10 lineStyleAngle `EmbReal lineStyleAngle`

17.50.5.11 lineStyleLength `EmbReal lineStyleLength`

17.50.5.12 lineStylePath `QPainterPath lineStylePath`

17.50.5.13 lwtPen `QPen lwtPen`

17.50.5.14 normalPath `QPainterPath normalPath`

17.50.5.15 objID `int64_t objID`

17.50.5.16 objLine `QLineF objLine`

17.50.5.17 objPen `QPen objPen`

17.50.5.18 objRubberMode `String` `objRubberMode = "OBJ_RUBBER_OFF"`

17.50.5.19 objRubberPoints `QHash<QString, QPointF>` `objRubberPoints`

17.50.5.20 objRubberTexts `QHash<QString, QString>` `objRubberTexts`

17.50.5.21 objText `QString` `objText`

17.50.5.22 objTextBackward `bool` `objTextBackward`

17.50.5.23 objTextFont `QString` `objTextFont`

17.50.5.24 objTextJustify `QString` `objTextJustify`

17.50.5.25 objTextPath `QPainterPath` `objTextPath`

17.50.5.26 objTextUpsideDown `bool` `objTextUpsideDown`

17.50.5.27 properties `Dictionary` `properties`

17.50.5.28 Type `int` `Type = OBJ_TYPE_BASE`

17.50.5.29 x_values `std::vector<EmbReal> x_values`

17.50.5.30 y_values `std::vector<EmbReal> y_values`

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

- `embroidermodder2/embroidermodder.h`
- `embroidermodder2/objects.cpp`

17.51 hoop_padding Struct Reference

Public Attributes

- `int left`
- `int right`
- `int top`
- `int bottom`

17.51.1 Member Data Documentation

17.51.1.1 bottom `int bottom`

17.51.1.2 left `int left`

17.51.1.3 right `int right`

17.51.1.4 top `int top`

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

- `extern/libembroidery/src/formats/format_jef.c`

17.52 Huffman Struct Reference

```
#include <embroidery_internal.h>
```

Public Attributes

- int [default_value](#)
- int [lengths](#) [1000]
- int [nlengths](#)
- int [table](#) [1000]
- int [table_width](#)
- int [ntable](#)

17.52.1 Member Data Documentation

17.52.1.1 [default_value](#) `int default_value`

17.52.1.2 [lengths](#) `int lengths[1000]`

17.52.1.3 [nlengths](#) `int nlengths`

17.52.1.4 [ntable](#) `int ntable`

17.52.1.5 [table](#) `int table[1000]`

17.52.1.6 [table_width](#) `int table_width`

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

- `extern/libembroidery/src/embroidery_internal.h`

17.53 ImageWidget Class Reference

```
#include <embroidermodder.h>
```

Public Member Functions

- [ImageWidget](#) (QString filename, QWidget *parent=0)
[ImageWidget::ImageWidget.](#)
- [~ImageWidget](#) ()
[ImageWidget::~~ImageWidget.](#)
- bool [load](#) (QString fileName)
[ImageWidget::load.](#)
- bool [save](#) (QString fileName)
[ImageWidget::save.](#)

Public Attributes

- QImage [img](#)

Protected Member Functions

- void [paintEvent](#) (QPaintEvent *event)
[ImageWidget::paintEvent.](#)

17.53.1 Detailed Description

17.53.2 Constructor & Destructor Documentation

17.53.2.1 ImageWidget() [ImageWidget](#) (
 QString *filename*,
 QWidget * *parent* = 0)

[ImageWidget::ImageWidget.](#)

Parameters

<i>filename</i>	
<i>parent</i>	

17.53.2.2 ~ImageWidget() [~ImageWidget](#) ()

[ImageWidget::~~ImageWidget.](#)

17.53.3 Member Function Documentation

17.53.3.1 load() `bool load (`
 `QString fileName)`

[ImageWidget::load.](#)

Parameters

<code>fileName</code>	
-----------------------	--

Returns

17.53.3.2 paintEvent() `void paintEvent (`
 `QPaintEvent * event) [protected]`

[ImageWidget::paintEvent.](#)

17.53.3.3 save() `bool save (`
 `QString fileName)`

[ImageWidget::save.](#)

Parameters

<code>fileName</code>	
-----------------------	--

Returns

17.53.4 Member Data Documentation

17.53.4.1 img `QImage img`

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

- embroidermodder2/[embroidermodder.h](#)
- embroidermodder2/[imagewidget.cpp](#)

17.54 LayerManager Class Reference

```
#include <embroidermodder.h>
```

Public Member Functions

- [LayerManager](#) (QWidget *parent=0)
[LayerManager::LayerManager](#) mw parent.
- [~LayerManager](#) ()
[LayerManager::~~LayerManager](#).
- void [addLayer](#) (QString name, const bool visible, const bool frozen, const [EmbReal](#) zValue, const QColor color, QString lineType, QString lineWeight, const bool print)
[LayerManager::addLayer](#).

Public Attributes

- QStandardItemModel * [layerModel](#)
- QSortFilterProxyModel * [layerModelSorted](#)
- QTreeView * [treeView](#)

17.54.1 Detailed Description**17.54.2 Constructor & Destructor Documentation**

17.54.2.1 LayerManager() [LayerManager](#) (
 QWidget * parent = 0)

[LayerManager::LayerManager](#) mw parent.

17.54.2.2 ~LayerManager() [~LayerManager](#) ()

[LayerManager::~~LayerManager](#).

17.54.3 Member Function Documentation

17.54.3.1 addLayer() void addLayer (
 QString name,
 const bool visible,
 const bool frozen,
 const [EmbReal](#) zValue,
 const QColor color,
 QString lineType,
 QString lineWeight,
 const bool print)

[LayerManager::addLayer](#).

Parameters

<i>name</i>	
<i>visible</i>	
<i>frozen</i>	
<i>zValue</i>	
<i>color</i>	
<i>lineType</i>	
<i>lineWeight</i>	
<i>print</i>	

17.54.4 Member Data Documentation

17.54.4.1 layerModel `QStandardItem* layerModel`

17.54.4.2 layerModelSorted `QSortFilterProxyModel* layerModelSorted`

17.54.4.3 treeView `QTreeView* treeView`

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

- embroidermodder2/[embroidermodder.h](#)
- embroidermodder2/[layer-manager.cpp](#)

17.55 LSYSTEM Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- char [axiom](#)
- char * [alphabet](#)
- char * [constants](#)
- char ** [rules](#)

17.55.1 Member Data Documentation

17.55.1.1 alphabet `char* alphabet`

17.55.1.2 axiom `char axiom`

17.55.1.3 constants `char* constants`

17.55.1.4 rules `char** rules`

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

- [extern/libembroidery/src/embroidery.h](#)

17.56 MainWindow Class Reference

The [MainWindow](#) class.

```
#include <embroidermodder.h>
```

Public Slots

- void [onCloseWindow](#) ()
MainWindow::onCloseWindow.
- virtual void [onCloseMdiWin](#) (MdiWindow *)
MainWindow::onCloseMdiWin.
- void [recentMenuAboutToShow](#) ()
MainWindow::recentMenuAboutToShow.
- void [onWindowActivated](#) (QMdiSubWindow *w)
MainWindow::onWindowActivated.
- void [windowMenuAboutToShow](#) ()
MainWindow::windowMenuAboutToShow.
- void [windowMenuActivated](#) (bool checked)
MainWindow::windowMenuActivated.
- void [updateAllViewScrollBars](#) (bool val)
MainWindow::updateAllViewScrollBars.
- void [updateAllViewCrossHairColors](#) (QRgb color)
MainWindow::updateAllViewCrossHairColors.
- void [updateAllViewBackgroundColors](#) (QRgb color)
MainWindow::updateAllViewBackgroundColors.
- void [updateAllViewSelectBoxColors](#) (QRgb colorL, QRgb fillL, QRgb colorR, QRgb fillR, int alpha)
MainWindow::updateAllViewSelectBoxColors.
- void [updateAllViewGridColors](#) (QRgb color)
MainWindow::updateAllViewGridColors.

- void `updateAllViewRulerColors` (QRgb color)
MainWindow::updateAllViewRulerColors.
- void `updatePickAddMode` (bool val)
MainWindow::updatePickAddMode.
- void `pickAddModeToggled` ()
MainWindow::pickAddModeToggled.
- void `settingsPrompt` ()
MainWindow::settingsPrompt.
- void `stub_testing` ()
MainWindow::stub_testing.
- void `promptHistoryAppended` (QString txt)
MainWindow::promptHistoryAppended.
- void `logPromptInput` (QString txt)
MainWindow::logPromptInput.
- void `promptInputPrevious` ()
MainWindow::promptInputPrevious.
- void `promptInputNext` ()
- void `about` (void)
about_action
- void `tipOfTheDay` (void)
MainWindow::tipOfTheDay.
- void `newFile` ()
MainWindow::newFile.
- void `openFile` (bool recent=false, String recentFile="")
MainWindow::openFile.
- void `openFilesSelected` (StringList files)
MainWindow::openFilesSelected.
- void `openrecentfile` ()
MainWindow::openrecentfile.
- void `savefile` ()
MainWindow::savefile.
- void `saveasfile` ()
MainWindow::saveasfile.
- void `quit` ()
MainWindow::quit.
- void `checkForUpdates` ()
MainWindow::checkForUpdates.
- void `buttonTipOfTheDayClicked` (int)
- void `closeToolBar` (QAction *)
MainWindow::closeToolBar.
- void `floatingChangedToolBar` (bool)
MainWindow::floatingChangedToolBar.
- void `toggleGrid` ()
MainWindow::toggleGrid.
- void `toggleRuler` ()
MainWindow::toggleRuler.
- void `toggleLwt` ()
MainWindow::toggleLwt.
- void `iconResize` (int iconSize)
- void `layerSelectorIndexChanged` (int index)
MainWindow::layerSelectorIndexChanged.

- void [colorSelectorIndexChanged](#) (int index)
MainWindow::colorSelectorIndexChanged.
- void [linetypeSelectorIndexChanged](#) (int index)
- void [lineweightSelectorIndexChanged](#) (int index)
MainWindow::lineweightSelectorIndexChanged.
- void [textFontSelectorCurrentFontChanged](#) (const QFont &font)
MainWindow::textFontSelectorCurrentFontChanged.
- void [textSizeSelectorIndexChanged](#) (int index)
MainWindow::textSizeSelectorIndexChanged.
- void [setTextFont](#) (QString str)
MainWindow::setTextFont.
- void [setTextSize](#) (EmbReal num)
MainWindow::setTextSize.
- QString [getCurrentLayer](#) ()
MainWindow::getCurrentLayer.
- QColor [getCurrentColor](#) ()
MainWindow::getCurrentColor.
- QString [getCurrentLineType](#) ()
MainWindow::getCurrentLineType.
- QString [getCurrentLineWeight](#) ()
MainWindow::getCurrentLineWeight.
- bool [isShiftPressed](#) ()
- void [setShiftPressed](#) ()
- void [setShiftReleased](#) ()
- void [deletePressed](#) ()
MainWindow::deletePressed.
- void [escapePressed](#) ()
MainWindow::escapePressed.

Public Member Functions

- [MainWindow](#) ()
MainWindow::MainWindow.
- [~MainWindow](#) ()
MainWindow::~~MainWindow.
- [MdiWindow](#) * [activeMdiWindow](#) ()
MainWindow::activeMdiWindow.
- QUndoStack * [activeUndoStack](#) ()
MainWindow::activeUndoStack.
- void [setUndoCleanIcon](#) (bool opened)
MainWindow::setUndoCleanIcon.
- virtual void [updateMenuToolbarStatusbar](#) ()
MainWindow::updateMenuToolbarStatusbar.
- bool [isCommandActive](#) ()
- QString [activeCommand](#) ()
- QIcon [create_icon](#) (QString stub)
MainWindow::create_icon.
- void [create_toolbar](#) (String toolbar, String label, QStringList entries)
MainWindow::create_toolbar.
- QString [platformString](#) ()

Public Attributes

- `std::vector< QGraphicsItem * >` [cutCopyObjectList](#)
- `QString` [formatFilterOpen](#)
- `QString` [formatFilterSave](#)

Protected Member Functions

- virtual void [resizeEvent](#) (`QResizeEvent *`)
[MainWindow::resizeEvent](#).
- void [closeEvent](#) (`QCloseEvent *event`)
[MainWindow::closeEvent](#).
- `QAction *` [getFileSeparator](#) ()
[MainWindow::getFileSeparator](#).
- void [loadFormats](#) ()
[MainWindow::loadFormats](#).
- `QMdiSubWindow *` [findMdiWindow](#) (`String fileName`)
[MainWindow::findMdiWindow](#).
- void [createAllActions](#) ()
[MainWindow::createAllActions](#).
- void [createAllMenus](#) ()
[MainWindow::createAllMenus](#).
- void [createAllToolbars](#) ()
[MainWindow::createAllToolbars](#).

Protected Attributes

- bool [shiftKeyPressedState](#)
- `QByteArray` [layoutState](#)
- int [numOfDocs](#)
- int [docIndex](#)
- `std::vector< MdiWindow * >` [listMdiWin](#)
- `QAction *` [myFileSeparator](#)
- `QComboBox *` [layerSelector](#)
- `QComboBox *` [colorSelector](#)
- `QComboBox *` [linetypeSelector](#)
- `QComboBox *` [lineweightSelector](#)
- `QFontComboBox *` [textFontSelector](#)
- `QComboBox *` [textSizeSelector](#)

Private Slots

- void [hideUnimplemented](#) ()
[MainWindow::hideUnimplemented](#).

17.56.1 Detailed Description

The [MainWindow](#) class.

17.56.2 Constructor & Destructor Documentation

17.56.2.1 MainWindow() `MainWindow ()`

[MainWindow::MainWindow.](#)

17.56.2.2 ~MainWindow() `~MainWindow ()`

[MainWindow::~~MainWindow.](#)

17.56.3 Member Function Documentation

17.56.3.1 **about** `void about (void) [slot]`

`about_action`

Parameters

<i>args</i>	
-------------	--

Returns

17.56.3.2 **activeCommand()** `QString activeCommand () [inline]`

17.56.3.3 **activeMdiWindow()** `MdiWindow * activeMdiWindow ()`

[MainWindow::activeMdiWindow.](#)

Returns

17.56.3.4 activeUndoStack() `QUndoStack * activeUndoStack ()`

[MainWindow::activeUndoStack.](#)

Returns

17.56.3.5 buttonTipOfTheDayClicked `void buttonTipOfTheDayClicked (
int button) [slot]`

17.56.3.6 checkForUpdates `void checkForUpdates () [slot]`

[MainWindow::checkForUpdates.](#)

17.56.3.7 closeEvent() `void closeEvent (
QCloseEvent * event) [protected]`

[MainWindow::closeEvent.](#)

Parameters

<i>event</i>	
--------------	--

17.56.3.8 closeToolBar `void closeToolBar (
QAction * action) [slot]`

[MainWindow::closeToolBar.](#)

Parameters

<i>action</i>	
---------------	--

17.56.3.9 colorSelectorIndexChanged `void colorSelectorIndexChanged (
int index) [slot]`

[MainWindow::colorSelectorIndexChanged.](#)

Parameters

<i>index</i>	
--------------	--

17.56.3.10 create_icon() `QIcon create_icon (`
 `QString stub)`

[MainWindow::create_icon.](#)

Parameters

<i>stub</i>	
-------------	--

Returns

17.56.3.11 create_toolbar() `void create_toolbar (`
 `String toolbar,`
 `String label,`
 `StringList entries)`

[MainWindow::create_toolbar.](#)

Parameters

<i>toolbar</i>	
<i>label</i>	
<i>entries</i>	

17.56.3.12 createAllActions() `void createAllActions () [protected]`

[MainWindow::createAllActions.](#)

Todo Set What's This Context Help to statusTip for now so there is some infos there. Make custom whats this context help popup with more descriptive help than just the status bar/tip one liner(short but not real long) with a hyperlink in the custom popup at the bottom to open full help file description. Ex: like wxPython AGW's SuperToolTip. ACTION->setWhatsThis(statusTip);

Finish All Commands ... <.< If an action calls a script then there will be an entry in config that is a StringList to be interpreted as a script.

An alias is another entry in config that is also a StringList containing just the name of the command it aliases.

icon: The stub used for the icon and the basic command. command: tooltip: The label in the menus and the message that appears when you hover over an icon. statustip: The message that appears at the bottom of the . shortcut: The keyboard shortcut for this action.

17.56.3.13 createAllMenus() `void createAllMenus () [protected]`

[MainWindow::createAllMenus.](#)

17.56.3.14 createAllToolbars() `void createAllToolbars () [protected]`

[MainWindow::createAllToolbars.](#)

17.56.3.15 deletePressed `void deletePressed () [slot]`

[MainWindow::deletePressed.](#)

17.56.3.16 escapePressed `void escapePressed () [slot]`

[MainWindow::escapePressed.](#)

17.56.3.17 findMdiWindow() `QMdiSubWindow * findMdiWindow (
 String fileName) [protected]`

[MainWindow::findMdiWindow.](#)

Parameters

<i>fileName</i>	
-----------------	--

Returns

17.56.3.18 floatingChangedToolBar `void floatingChangedToolBar (
 bool isFloating) [slot]`

[MainWindow::floatingChangedToolBar.](#)

Parameters

<i>isFloating</i>	
-------------------	--

17.56.3.19 getCurrentColor `QRgb getCurrentColor () [slot]`

[MainWindow::getCurrentColor.](#)

Returns

17.56.3.20 getCurrentLayer `QString getCurrentLayer () [slot]`

[MainWindow::getCurrentLayer.](#)

Returns

17.56.3.21 getCurrentLineType `QString getCurrentLineType () [slot]`

[MainWindow::getCurrentLineType.](#)

Returns

17.56.3.22 getCurrentLineWeight `QString getCurrentLineWeight () [slot]`

[MainWindow::getCurrentLineWeight.](#)

Returns

17.56.3.23 getFileSeparator() `QAction * getFileSeparator () [protected]`

[MainWindow::getFileSeparator.](#)

Returns

17.56.3.24 hideUnimplemented `void hideUnimplemented () [private], [slot]`

[MainWindow::hideUnimplemented.](#)

17.56.3.25 iconResize `void iconResize (
int iconSize) [slot]`

17.56.3.26 isCommandActive() `bool isCommandActive () [inline]`

17.56.3.27 isShiftPressed `bool isShiftPressed () [slot]`

17.56.3.28 layerSelectorIndexChanged `void layerSelectorIndexChanged (
int index) [slot]`

[MainWindow::layerSelectorIndexChanged.](#)

Parameters

<i>index</i>	
--------------	--

17.56.3.29 linetypeSelectorIndexChanged `void linetypeSelectorIndexChanged (
int index) [slot]`

17.56.3.30 linewidthSelectorIndexChanged `void linewidthSelectorIndexChanged (
int index) [slot]`

[MainWindow::linewidthSelectorIndexChanged.](#)

Parameters

<i>index</i>	
--------------	--

17.56.3.31 loadFormats() `void loadFormats () [protected]`

[MainWindow::loadFormats.](#)

17.56.3.32 logPromptInput void logPromptInput (
 QString txt) [slot]

[MainWindow::logPromptInput.](#)

Parameters

<i>txt</i>	
------------	--

17.56.3.33 newFile void newFile () [slot]

[MainWindow::newFile.](#)

17.56.3.34 onCloseMdiWin void onCloseMdiWin (
 MdiWindow * theMdiWin) [virtual], [slot]

[MainWindow::onCloseMdiWin.](#)

Parameters

<i>theMdiWin</i>	
------------------	--

17.56.3.35 onCloseWindow void onCloseWindow () [slot]

[MainWindow::onCloseWindow.](#)

17.56.3.36 onWindowActivated void onWindowActivated (
 QMdiSubWindow * w) [slot]

[MainWindow::onWindowActivated.](#)

Parameters

<i>w</i>	
----------	--

17.56.3.37 openFile `void openFile (`
 `bool recent = false,`
 `String recentFile = "") [slot]`

[MainWindow::openFile.](#)

Parameters

<i>recent</i>	
<i>recentFile</i>	

17.56.3.38 openFilesSelected `void openFilesSelected (`
 `StringList filesToOpen) [slot]`

[MainWindow::openFilesSelected.](#)

Parameters

<i>filesToOpen</i>	
--------------------	--

17.56.3.39 openrecentfile `void openrecentfile () [slot]`

[MainWindow::openrecentfile.](#)

17.56.3.40 pickAddModeToggled `void pickAddModeToggled () [slot]`

[MainWindow::pickAddModeToggled.](#)

17.56.3.41 platformString() `QString platformString ()`

17.56.3.42 promptHistoryAppended `void promptHistoryAppended (`
 `QString txt) [slot]`

[MainWindow::promptHistoryAppended.](#)

Parameters

<i>txt</i>	
------------	--

17.56.3.43 promptInputNext void promptInputNext () [slot]

17.56.3.44 promptInputPrevious void promptInputPrevious () [slot]

[MainWindow::promptInputPrevious.](#)

17.56.3.45 quit void quit () [slot]

[MainWindow::quit.](#)

17.56.3.46 recentMenuAboutToShow void recentMenuAboutToShow () [slot]

[MainWindow::recentMenuAboutToShow.](#)

17.56.3.47 resizeEvent() void resizeEvent (
 QResizeEvent * e) [protected], [virtual]

[MainWindow::resizeEvent.](#)

Parameters

<i>e</i>	
----------	--

17.56.3.48 saveasfile void saveasfile () [slot]

[MainWindow::saveasfile.](#)

17.56.3.49 savefile void savefile () [slot]

[MainWindow::savefile.](#)

17.56.3.50 setShiftPressed `void setShiftPressed () [slot]`

17.56.3.51 setShiftReleased `void setShiftReleased () [slot]`

17.56.3.52 setTextFont `void setTextFont (
QString str) [slot]`

[MainWindow::setFont.](#)

Parameters

<i>str</i>	
------------	--

17.56.3.53 setTextSize `void setTextSize (
EmbReal num) [slot]`

[MainWindow::setTextSize.](#)

Parameters

<i>num</i>	
------------	--

17.56.3.54 settingsPrompt `void settingsPrompt () [slot]`

[MainWindow::settingsPrompt.](#)

17.56.3.55 setUndoCleanIcon() `void setUndoCleanIcon (
bool opened)`

[MainWindow::setUndoCleanIcon.](#)

Parameters

<i>opened</i>	
---------------	--

17.56.3.56 stub_testing `void stub_testing () [slot]`

[MainWindow::stub_testing.](#)

17.56.3.57 textFontSelectorCurrentFontChanged void textFontSelectorCurrentFontChanged (
const QFont & font) [slot]

[MainWindow::textFontSelectorCurrentFontChanged.](#)

Parameters

<i>font</i>	
-------------	--

17.56.3.58 textSizeSelectorIndexChanged void textSizeSelectorIndexChanged (
int index) [slot]

[MainWindow::textSizeSelectorIndexChanged.](#)

Parameters

<i>index</i>	
--------------	--

17.56.3.59 tipOfTheDay void tipOfTheDay (
void) [slot]

[MainWindow::tipOfTheDay.](#)

17.56.3.60 toggleGrid void toggleGrid () [slot]

[MainWindow::toggleGrid.](#)

17.56.3.61 toggleLwt void toggleLwt () [slot]

[MainWindow::toggleLwt.](#)

17.56.3.62 toggleRuler void toggleRuler () [slot]

[MainWindow::toggleRuler.](#)

17.56.3.63 updateAllViewBackgroundColors void updateAllViewBackgroundColors (
QRgb color) [slot]

[MainWindow::updateAllViewBackgroundColors.](#)

Parameters

<i>color</i>	
--------------	--

17.56.3.64 updateAllViewCrossHairColors `void updateAllViewCrossHairColors (
QRgb color) [slot]`

[MainWindow::updateAllViewCrossHairColors.](#)

Parameters

<i>color</i>	
--------------	--

17.56.3.65 updateAllViewGridColors `void updateAllViewGridColors (
QRgb color) [slot]`

[MainWindow::updateAllViewGridColors.](#)

Parameters

<i>color</i>	
--------------	--

17.56.3.66 updateAllViewRulerColors `void updateAllViewRulerColors (
QRgb color) [slot]`

[MainWindow::updateAllViewRulerColors.](#)

Parameters

<i>color</i>	
--------------	--

17.56.3.67 updateAllViewScrollBars `void updateAllViewScrollBars (
bool val) [slot]`

[MainWindow::updateAllViewScrollBars.](#)

Parameters

<i>val</i>	
------------	--

17.56.3.68 updateAllViewSelectBoxColors `void updateAllViewSelectBoxColors (`
 `QRgb colorL,`
 `QRgb fillL,`
 `QRgb colorR,`
 `QRgb fillR,`
 `int alpha) [slot]`

[MainWindow::updateAllViewSelectBoxColors.](#)

Parameters

<i>colorL</i>	
<i>fillL</i>	
<i>colorR</i>	
<i>fillR</i>	
<i>alpha</i>	

17.56.3.69 updateMenuToolbarStatusbar() `void updateMenuToolbarStatusbar () [virtual]`

[MainWindow::updateMenuToolbarStatusbar.](#)

17.56.3.70 updatePickAddMode `void updatePickAddMode (`
 `bool val) [slot]`

[MainWindow::updatePickAddMode.](#)

Parameters

<i>val</i>	
------------	--

17.56.3.71 windowMenuAboutToShow `void windowMenuAboutToShow () [slot]`

[MainWindow::windowMenuAboutToShow.](#)

17.56.3.72 windowMenuActivated `void windowMenuActivated (`
 `bool checked) [slot]`

[MainWindow::windowMenuActivated.](#)

Parameters

<i>checked</i>	<input type="checkbox"/>
----------------	--------------------------

17.56.4 Member Data Documentation

17.56.4.1 colorSelector `QComboBox* colorSelector` [protected]

17.56.4.2 cutCopyObjectList `std::vector<QGraphicsItem*> cutCopyObjectList`

17.56.4.3 docIndex `int docIndex` [protected]

17.56.4.4 formatFilterOpen `QString formatFilterOpen`

17.56.4.5 formatFilterSave `QString formatFilterSave`

17.56.4.6 layerSelector `QComboBox* layerSelector` [protected]

17.56.4.7 layoutState `QByteArray layoutState` [protected]

17.56.4.8 linetypeSelector `QComboBox* linetypeSelector` [protected]

17.56.4.9 lineweightSelector `QComboBox* lineweightSelector` [protected]

17.56.4.10 listMdiWin `std::vector<MdiWindow*> listMdiWin` [protected]

17.56.4.11 myFileSeparator `QAction* myFileSeparator` [protected]

17.56.4.12 numOfDocs `int numOfDocs` [protected]

17.56.4.13 shiftKeyPressedState `bool shiftKeyPressedState` [protected]

17.56.4.14 textFontSelector `QFontComboBox* textFontSelector` [protected]

17.56.4.15 textSizeSelector `QComboBox* textSizeSelector` [protected]

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

- embroidermodder2/[embroidermodder.h](#)
- embroidermodder2/[mainwindow-menus.cpp](#)
- embroidermodder2/[mainwindow-toolbars.cpp](#)
- embroidermodder2/[mainwindow.cpp](#)

17.57 MdiArea Class Reference

```
#include <embroidermodder.h>
```

Public Slots

- void [cascade](#) ()
MdiArea::cascade.
- void [tile](#) ()
MdiArea::tile.

Public Member Functions

- void [zoomExtentsAllSubWindows](#) ()
MdiArea::zoomExtentsAllSubWindows.
- void [forceRepaint](#) ()
MdiArea::forceRepaint.
- [MdiArea](#) (QWidget *parent=0)
MdiArea::MdiArea.
- [~MdiArea](#) ()
MdiArea::~~MdiArea.
- void [useBackgroundLogo](#) (bool use)
MdiArea::useBackgroundLogo.
- void [useBackgroundTexture](#) (bool use)
MdiArea::useBackgroundTexture.
- void [useBackgroundColor](#) (bool use)
- void [setBackgroundLogo](#) (QString fileName)
MdiArea::setBackgroundLogo.
- void [setBackgroundTexture](#) (QString fileName)
MdiArea::setBackgroundTexture.
- void [setBackgroundColor](#) (const QColor &color)
MdiArea::setBackgroundColor.

Public Attributes

- bool [useLogo](#)
- bool [useTexture](#)
- bool [useColor](#)
- QPixmap [bgLogo](#)
- QPixmap [bgTexture](#)
- QColor [bgColor](#)

Protected Member Functions

- virtual void [mouseDoubleClickEvent](#) (QMouseEvent *e)
MdiArea::mouseDoubleClickEvent.
- virtual void [paintEvent](#) (QPaintEvent *e)
MdiArea::paintEvent.

17.57.1 Constructor & Destructor Documentation

17.57.1.1 MdiArea() [MdiArea](#) (
 QWidget * parent = 0)

[MdiArea::MdiArea.](#)

Parameters

<i>mw</i>	
<i>parent</i>	

17.57.1.2 `~MdiArea()` `~MdiArea ()`

[MdiArea::~~MdiArea.](#)

17.57.2 Member Function Documentation**17.57.2.1 cascade** `void cascade () [slot]`

[MdiArea::cascade.](#)

17.57.2.2 forceRepaint() `void forceRepaint ()`

[MdiArea::forceRepaint.](#)

17.57.2.3 mouseDoubleClickEvent() `void mouseDoubleClickEvent (
QMouseEvent * e) [protected], [virtual]`

[MdiArea::mouseDoubleClickEvent.](#)

17.57.2.4 paintEvent() `void paintEvent (
QPaintEvent * e) [protected], [virtual]`

[MdiArea::paintEvent.](#)

17.57.2.5 setBackgroundColor() `void setBackgroundColor (
const QColor & color)`

[MdiArea::setBackgroundColor.](#)

Parameters

<i>color</i>	
--------------	--

17.57.2.6 setBackgroundLogo() `void setBackgroundLogo (`
 `QString fileName)`

[MdiArea::setBackgroundLogo.](#)

Parameters

<i>fileName</i>	
-----------------	--

17.57.2.7 setBackgroundTexture() `void setBackgroundTexture (`
 `QString fileName)`

[MdiArea::setBackgroundTexture.](#)

Parameters

<i>fileName</i>	
-----------------	--

17.57.2.8 tile `void tile () [slot]`

[MdiArea::tile.](#)

17.57.2.9 useBackgroundColor() `void useBackgroundColor (`
 `bool use)`

Parameters

<i>use</i>	
------------	--

17.57.2.10 useBackgroundLogo() `void useBackgroundLogo (`
 `bool use)`

[MdiArea::useBackgroundLogo.](#)

Parameters

<i>use</i>	
------------	--

17.57.2.11 useBackgroundTexture() `void useBackgroundTexture (bool use)`

[MdiArea::useBackgroundTexture.](#)

Parameters

<i>use</i>	
------------	--

17.57.2.12 zoomExtentsAllSubWindows() `void zoomExtentsAllSubWindows ()`

[MdiArea::zoomExtentsAllSubWindows.](#)

17.57.3 Member Data Documentation

17.57.3.1 bgColor `QColor bgColor`

17.57.3.2 bgLogo `QPixmap bgLogo`

17.57.3.3 bgTexture `QPixmap bgTexture`

17.57.3.4 useColor `bool useColor`

17.57.3.5 useLogo `bool useLogo`

17.57.3.6 useTexture `bool useTexture`

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

- embroidermodder2/[embroidermodder.h](#)
- embroidermodder2/[mdiarea.cpp](#)

17.58 MdiWindow Class Reference

```
#include <embroidermodder.h>
```

Public Slots

- void [closeEvent](#) (QCloseEvent *e)
MdiWindow::closeEvent.
- void [onWindowActivated](#) ()
MdiWindow::onWindowActivated.
- void [currentLayerChanged](#) (QString layer)
MdiWindow::currentLayerChanged.
- void [currentColorChanged](#) (const QColor &color)
MdiWindow::currentColorChanged.
- void [currentLinetypeChanged](#) (QString type)
MdiWindow::currentLinetypeChanged.
- void [currentLineweightChanged](#) (QString weight)
MdiWindow::currentLineweightChanged.
- void [updateColorLinetypeLineweight](#) ()
- void [deletePressed](#) ()
- void [escapePressed](#) ()
- void [showViewScrollBars](#) (bool val)
- void [setViewCrossHairColor](#) (QColor color)
- void [setViewBackgroundColor](#) (QColor color)
- void [setViewSelectBoxColors](#) (QColor colorL, QColor fillL, QColor colorR, QColor fillR, int alpha)
- void [setViewGridColor](#) (QColor color)
- void [setViewRulerColor](#) (QColor color)
- void [print](#) ()
MdiWindow::print.
- void [saveBMC](#) ()
MdiWindow::saveBMC.
- void [promptHistoryAppended](#) (QString txt)
- void [logPromptInput](#) (QString txt)
- void [promptInputPrevious](#) ()
- void [promptInputNext](#) ()
MdiWindow::promptInputNext.

Signals

- void [sendCloseMdiWin](#) (MdiWindow *)

Public Member Functions

- [MdiWindow](#) (const int theIndex, QMdiArea *parent, Qt::WindowFlags wflags)
- [~MdiWindow](#) ()
MdiWindow::~~MdiWindow.
- void [setCurrentFile](#) (QString fileName)
MdiWindow::setCurrentFile.
- void [promptInputPrevNext](#) (bool prev)
MdiWindow::promptInputPrevNext.
- virtual QSize [sizeHint](#) ()
MdiWindow::sizeHint.
- QString [getShortCurrentFile](#) ()
MdiWindow::getShortCurrentFile.
- void [designDetails](#) ()
- bool [loadFile](#) (String fileName)
MdiWindow::loadFile.
- bool [saveFile](#) (String fileName)
MdiWindow::saveFile.

Public Attributes

- QMdiArea * [mdiArea](#)
- QGraphicsScene * [gscene](#)
- [View](#) * [gview](#)
- bool [fileWasLoaded](#)
- QString [promptHistory](#)
- std::vector< QString > [promptInputList](#)
- int [promptInputNum](#)
- QPrinter [printer](#)
- QString [curFile](#)
- int [myIndex](#)
- QString [curLayer](#)
- QColor [curColor](#)
- QString [curLineType](#)
- QString [curLineWeight](#)

17.58.1 Constructor & Destructor Documentation

17.58.1.1 MdiWindow() [MdiWindow](#) (
const int *theIndex*,
QMdiArea * *parent*,
Qt::WindowFlags *wflags*)

17.58.1.2 ~MdiWindow() [~MdiWindow](#) ()

[MdiWindow::~~MdiWindow.](#)

17.58.2 Member Function Documentation

17.58.2.1 closeEvent `void closeEvent (
 QCloseEvent * e) [slot]`

[MdiWindow::closeEvent.](#)

17.58.2.2 currentColorChanged `void currentColorChanged (
 const QRgb & color) [slot]`

[MdiWindow::currentColorChanged.](#)

Parameters

<i>color</i>	
--------------	--

17.58.2.3 currentLayerChanged `void currentLayerChanged (
 QString layer) [slot]`

[MdiWindow::currentLayerChanged.](#)

Parameters

<i>layer</i>	
--------------	--

17.58.2.4 currentLinetypeChanged `void currentLinetypeChanged (
 QString type) [slot]`

[MdiWindow::currentLinetypeChanged.](#)

Parameters

<i>type</i>	
-------------	--

17.58.2.5 currentLineweightChanged `void currentLineweightChanged (
 QString weight) [slot]`

[MdiWindow::currentLineweightChanged.](#)

Parameters

<i>weight</i>	
---------------	--

17.58.2.6 deletePressed `void deletePressed () [slot]`

17.58.2.7 designDetails() `void designDetails ()`

17.58.2.8 escapePressed `void escapePressed () [slot]`

17.58.2.9 getShortCurrentFile() `QString getShortCurrentFile ()`

[MdiWindow::getShortCurrentFile](#).

Returns

17.58.2.10 loadFile() `bool loadFile (
 String fileName)`

[MdiWindow::loadFile](#).

Parameters

<i>fileName</i>	
-----------------	--

Returns

17.58.2.11 logPromptInput `void logPromptInput (
 QString txt) [slot]`

17.58.2.12 onWindowActivated `void onWindowActivated () [slot]`

[MdiWindow::onWindowActivated.](#)

17.58.2.13 print `void print () [slot]`

[MdiWindow::print.](#)

17.58.2.14 promptHistoryAppended `void promptHistoryAppended (
QString txt) [slot]`

17.58.2.15 promptInputNext `void promptInputNext () [slot]`

[MdiWindow::promptInputNext.](#)

17.58.2.16 promptInputPrevious `void promptInputPrevious () [slot]`

17.58.2.17 promptInputPrevNext() `void promptInputPrevNext (
bool prev)`

[MdiWindow::promptInputPrevNext.](#)

Parameters

<i>prev</i>	
-------------	--

17.58.2.18 saveBMC `void saveBMC () [slot]`

[MdiWindow::saveBMC.](#)

Todo Save a Brother PEL image (An 8bpp, 130x113 pixel monochromatic? bitmap image) Why 8bpp when only 1bpp is needed?

Todo Should BMC be limited to ~32KB or is this a mix up with Bitmap Cache?
Is there/should there be other embedded data in the bitmap besides the image itself?

Note

Can save a Singer BMC image (An 8bpp, 130x113 pixel colored bitmap image)

17.58.2.19 saveFile() `bool saveFile (`
 `String fileName)`

[MdiWindow::saveFile.](#)

Parameters

<i>fileName</i>	
-----------------	--

Returns

17.58.2.20 sendCloseMdiWin `void sendCloseMdiWin (`
 `MdiWindow *) [signal]`

17.58.2.21 setCurrentFile() `void setCurrentFile (`
 `QString fileName)`

[MdiWindow::setCurrentFile.](#)

Parameters

<i>fileName</i>	
-----------------	--

17.58.2.22 setViewBackgroundColor `void setViewBackgroundColor (`
 `QRgb color) [slot]`

17.58.2.23 setViewCrossHairColor `void setViewCrossHairColor (`
 `QRgb color) [slot]`

17.58.2.24 setViewGridColor void setViewGridColor (
 QRgb *color*) [slot]

17.58.2.25 setViewRulerColor void setViewRulerColor (
 QRgb *color*) [slot]

17.58.2.26 setViewSelectBoxColors void setViewSelectBoxColors (
 QRgb *colorL*,
 QRgb *fillL*,
 QRgb *colorR*,
 QRgb *fillR*,
 int *alpha*) [slot]

17.58.2.27 showViewScrollBars void showViewScrollBars (
 bool *val*) [slot]

17.58.2.28 sizeHint() QSize sizeHint () [virtual]

[MdiWindow::sizeHint](#).

Returns

17.58.2.29 updateColorLinetypeLineweight void updateColorLinetypeLineweight () [slot]

17.58.3 Member Data Documentation

17.58.3.1 curColor QRgb curColor

17.58.3.2 curFile QString curFile

17.58.3.3 curLayer `QString curLayer`

17.58.3.4 curLineType `QString curLineType`

17.58.3.5 curLineWeight `QString curLineWeight`

17.58.3.6 fileWasLoaded `bool fileWasLoaded`

17.58.3.7 gscene `QGraphicsScene* gscene`

17.58.3.8 gview `View* gview`

17.58.3.9 mdiArea `QMdiArea* mdiArea`

17.58.3.10 myIndex `int myIndex`

17.58.3.11 printer `QPrinter printer`

17.58.3.12 promptHistory `QString promptHistory`

17.58.3.13 promptInputList `std::vector<QString> promptInputList`

17.58.3.14 promptInputNum `int promptInputNum`

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

- embroidermodder2/[embroidermodder.h](#)
- embroidermodder2/[mdiwindow.cpp](#)

17.59 Node_Struct Reference

```
#include <embroidermodder.h>
```

Public Attributes

- [String](#) `s`
- [EmbReal](#) `r`
- `int32_t` `i`
- `bool` `b`
- [StringList](#) `sl`
- `int` `type`

17.59.1 Member Data Documentation**17.59.1.1 b** `bool b`**17.59.1.2 i** `int32_t i`**17.59.1.3 r** [EmbReal](#) `r`**17.59.1.4 s** [String](#) `s`**17.59.1.5 sl** [StringList](#) `sl`

17.59.1.6 type `int type`

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

- `embroidermodder2/embroidermodder.h`

17.60 PreviewDialog Class Reference

```
#include <embroidermodder.h>
```

Public Member Functions

- [PreviewDialog](#) (`QWidget *parent=0`, `QString caption=QString()`, `QString directory=QString()`, `QString filter=QString()`)
- [~PreviewDialog](#) ()

Public Attributes

- `ImageWidget *imgWidget`

17.60.1 Constructor & Destructor Documentation

17.60.1.1 PreviewDialog() `PreviewDialog (`
 `QWidget * parent = 0,`
 `QString caption = QString(),`
 `QString directory = QString(),`
 `QString filter = QString())`

17.60.1.2 ~PreviewDialog() `~PreviewDialog ()`

17.60.2 Member Data Documentation

17.60.2.1 imgWidget `ImageWidget* imgWidget`

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

- `embroidermodder2/embroidermodder.h`
- `embroidermodder2/preview-dialog.cpp`

17.61 PropertyEditor Class Reference

```
#include <embroidermodder.h>
```

Public Slots

- void [setSelectedItems](#) (std::vector< QGraphicsItem * > itemList)
- void [updatePickAddModeButton](#) (bool pickAddMode)

Signals

- void [pickAddModeToggled](#) ()

Public Member Functions

- [PropertyEditor](#) (QString iconDirectory=QString(), bool pickAddMode=true, QWidget *widgetToFocus=0, QWidget *parent=0)
- [~PropertyEditor](#) ()
- QPushButton * [createToolButton](#) (QString iconName, QString txt)
- QLineEdit * [createLineEdit](#) (QString validatorType=QString(), bool readOnly=false)
- void [updateLineEditStrlfVaries](#) (QLineEdit *lineEdit, QString str)
- void [updateLineEditNumIfVaries](#) (QLineEdit *lineEdit, [EmbReal](#) num, bool useAnglePrecision)
- void [updateFontComboBoxStrlfVaries](#) (QFontComboBox *fontComboBox, QString str)
- void [updateComboBoxStrlfVaries](#) (QComboBox *comboBox, QString str, [StringList](#) strList)
- void [updateComboBoxBoolIfVaries](#) (QComboBox *comboBox, bool val, bool yesOrNoText)
- void [mapSignal](#) (QObject *fieldObj, QString name, QVariant value)
PropertyEditor::mapSignal.
- QComboBox * [createComboBoxSelected](#) ()
- QPushButton * [createToolButtonQSelect](#) ()
- QPushButton * [createToolButtonPickAdd](#) ()
- void [createGroupBox](#) (String group_box_key, const char *title)

Public Attributes

- QWidget * [focusWidget](#)
- QString [iconDir](#)
- int [iconSize](#)
- Qt::ToolButtonStyle [propertyEditorButtonStyle](#)
- bool [pickAdd](#)
- std::vector< QGraphicsItem * > [selectedItemList](#)
- int [precisionAngle](#)
- int [precisionLength](#)
- QSignalMapper * [signalMapper](#)
- QComboBox * [comboBoxSelected](#)
- QPushButton * [toolButtonQSelect](#)
- QPushButton * [toolButtonPickAdd](#)

Protected Member Functions

- bool [eventFilter](#) (QObject *obj, QEvent *event)

Private Slots

- void [fieldEdited](#) (QObject *fieldObj)
- void [showGroups](#) (int objType)
- void [showOneType](#) (int index)
- void [hideAllGroups](#) ()
- void [clearAllFields](#) ()
- void [togglePickAddMode](#) ()

17.61.1 Constructor & Destructor Documentation

17.61.1.1 PropertyEditor() `PropertyEditor (`
 `QString iconDirectory = QString(),`
 `bool pickAddMode = true,`
 `QWidget * widgetToFocus = 0,`
 `QWidget * parent = 0)`

17.61.1.2 ~PropertyEditor() `~PropertyEditor ()`

Todo document this

17.61.2 Member Function Documentation

17.61.2.1 clearAllFields `void clearAllFields () [private], [slot]`

Todo DimAligned
DimAngular
DimArcLength
DimDiameter
DimLeader
DimLinear
DimOrdinate
DimRadius

17.61.2.2 createComboBoxSelected() `QComboBox * createComboBoxSelected ()`

Todo document this

17.61.2.3 createGroupBox() `void createGroupBox (`
 `String group_box_key,`
 `const char * title)`

17.61.2.4 createLineEdit() `QLineEdit * createLineEdit (`
 `QString validatorType = QString(),`
 `bool readOnly = false)`

17.61.2.5 createToolButton() `QToolButton * createToolButton (`
 `QString iconName,`
 `QString txt)`

17.61.2.6 createToolButtonPickAdd() `QToolButton * createToolButtonPickAdd ()`

17.61.2.7 createToolButtonQSelect() `QToolButton * createToolButtonQSelect ()`

Todo document this

17.61.2.8 eventFilter() `bool eventFilter (`
 `QObject * obj,`
 `QEvent * event) [protected]`

Todo document this

17.61.2.9 fieldEdited `void fieldEdited (`
 `QObject * fieldObj) [private], [slot]`

17.61.2.10 hideAllGroups void hideAllGroups () [private], [slot]

Note

General group will never be hidden.

17.61.2.11 mapSignal() void mapSignal (
 QObject * *fieldObj*,
 QString *name*,
 QVariant *value*)

[PropertyEditor::mapSignal.](#)

Parameters

<i>fieldObj</i>	
<i>name</i>	
<i>value</i>	

17.61.2.12 pickAddModeToggled void pickAddModeToggled () [signal]

17.61.2.13 setSelectedItems void setSelectedItems (
 std::vector< QGraphicsItem * > *itemList*) [slot]

17.61.2.14 showGroups void showGroups (
 int *objType*) [private], [slot]

17.61.2.15 showOneType void showOneType (
 int *index*) [private], [slot]

17.61.2.16 togglePickAddMode void togglePickAddMode () [private], [slot]

17.61.2.17 updateComboBoxBoolIfVaries() `void updateComboBoxBoolIfVaries (`
 `QComboBox * comboBox,`
 `bool val,`
 `bool yesOrNoText)`

17.61.2.18 updateComboBoxStrIfVaries() `void updateComboBoxStrIfVaries (`
 `QComboBox * comboBox,`
 `QString str,`
 `StringList strList)`

17.61.2.19 updateFontComboBoxStrIfVaries() `void updateFontComboBoxStrIfVaries (`
 `QFontComboBox * fontComboBox,`
 `QString str)`

17.61.2.20 updateLineEditNumIfVaries() `void updateLineEditNumIfVaries (`
 `QLineEdit * lineEdit,`
 `EmbReal num,`
 `bool useAnglePrecision)`

17.61.2.21 updateLineEditStrIfVaries() `void updateLineEditStrIfVaries (`
 `QLineEdit * lineEdit,`
 `QString str)`

17.61.2.22 updatePickAddModeButton `void updatePickAddModeButton (`
 `bool pickAddMode) [slot]`

17.61.3 Member Data Documentation

17.61.3.1 comboBoxSelected `QComboBox* comboBoxSelected`

17.61.3.2 focusWidget `QWidget* focusWidget`

17.61.3.3 iconDir `QString iconDir`

17.61.3.4 iconSize `int iconSize`

17.61.3.5 pickAdd `bool pickAdd`

17.61.3.6 precisionAngle `int precisionAngle`

17.61.3.7 precisionLength `int precisionLength`

17.61.3.8 propertyEditorButtonStyle `Qt::ToolButtonStyle propertyEditorButtonStyle`

17.61.3.9 selectedItemList `std::vector<QGraphicsItem*> selectedItemList`

17.61.3.10 signalMapper `QSignalMapper* signalMapper`

17.61.3.11 toolButtonPickAdd `QToolButton* toolButtonPickAdd`

17.61.3.12 toolButtonQSelect `QToolButton* toolButtonQSelect`

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

- embroidermodder2/[embroidermodder.h](#)
- embroidermodder2/[property-editor.cpp](#)

17.62 SaveObject Class Reference

```
#include <embroidermodder.h>
```

Public Member Functions

- [SaveObject](#) (QGraphicsScene *theScene, QObject *parent=0)
SaveObject::SaveObject.
- [~SaveObject](#) ()
SaveObject::~~SaveObject.
- bool [save](#) (QString fileName)
- void [addArc](#) (EmbPattern *pattern, QGraphicsItem *item)
SaveObject::addArc.
- void [addBlock](#) (EmbPattern *pattern, QGraphicsItem *item)
SaveObject::addBlock.
- void [addCircle](#) (EmbPattern *pattern, QGraphicsItem *item)
SaveObject::addCircle.
- void [addDimAligned](#) (EmbPattern *pattern, QGraphicsItem *item)
SaveObject::addDimAligned.
- void [addDimAngular](#) (EmbPattern *pattern, QGraphicsItem *item)
SaveObject::addDimAngular.
- void [addDimArcLength](#) (EmbPattern *pattern, QGraphicsItem *item)
SaveObject::addDimArcLength.
- void [addDimDiameter](#) (EmbPattern *pattern, QGraphicsItem *item)
SaveObject::addDimDiameter.
- void [addDimLeader](#) (EmbPattern *pattern, QGraphicsItem *item)
SaveObject::addDimLeader.
- void [addDimLinear](#) (EmbPattern *pattern, QGraphicsItem *item)
SaveObject::addDimLinear.
- void [addDimOrdinate](#) (EmbPattern *pattern, QGraphicsItem *item)
SaveObject::addDimOrdinate.
- void [addDimRadius](#) (EmbPattern *pattern, QGraphicsItem *item)
SaveObject::addDimRadius.
- void [addEllipse](#) (EmbPattern *pattern, QGraphicsItem *item)
SaveObject::addEllipse.
- void [addEllipseArc](#) (EmbPattern *pattern, QGraphicsItem *item)
SaveObject::addEllipseArc.
- void [addGrid](#) (EmbPattern *pattern, QGraphicsItem *item)
SaveObject::addGrid.
- void [addHatch](#) (EmbPattern *pattern, QGraphicsItem *item)
SaveObject::addHatch.
- void [addImage](#) (EmbPattern *pattern, QGraphicsItem *item)
SaveObject::addImage.
- void [addInfiniteLine](#) (EmbPattern *pattern, QGraphicsItem *item)
SaveObject::addInfiniteLine.
- void [addLine](#) (EmbPattern *pattern, QGraphicsItem *item)
SaveObject::addLine.
- void [addPath](#) (EmbPattern *pattern, QGraphicsItem *item)
SaveObject::addPath.
- void [addPoint](#) (EmbPattern *pattern, QGraphicsItem *item)

- [SaveObject::addPoint.](#)
- void [addPolygon](#) ([EmbPattern](#) *pattern, QGraphicsItem *item)
- [SaveObject::addPolygon.](#)
- void [addPolyline](#) ([EmbPattern](#) *pattern, QGraphicsItem *item)
- [SaveObject::addPolyline.](#)
- void [addRay](#) ([EmbPattern](#) *pattern, QGraphicsItem *item)
- [SaveObject::addRay.](#)
- void [addRectangle](#) ([EmbPattern](#) *pattern, QGraphicsItem *item)
- [SaveObject::addRectangle.](#)
- void [addSlot](#) ([EmbPattern](#) *pattern, QGraphicsItem *item)
- [SaveObject::addSlot.](#)
- void [addSpline](#) ([EmbPattern](#) *pattern, QGraphicsItem *item)
- [SaveObject::addSpline.](#)
- void [addTextMulti](#) ([EmbPattern](#) *pattern, QGraphicsItem *item)
- [SaveObject::addTextMulti.](#)
- void [addTextSingle](#) ([EmbPattern](#) *pattern, QGraphicsItem *item)
- [SaveObject::addTextSingle.](#)
- void [toPolyline](#) ([EmbPattern](#) *pattern, const QPointF &objPos, const QPainterPath &objPath, QString layer, const QColor &color, QString lineType, QString lineWeight)
- [SaveObject::toPolyline.](#)

Public Attributes

- QGraphicsScene * [gscene](#)
- int [formatType](#)

17.62.1 Constructor & Destructor Documentation

17.62.1.1 SaveObject() [SaveObject](#) (
 QGraphicsScene * *theScene*,
 QObject * *parent* = 0)

[SaveObject::SaveObject.](#)

Parameters

<i>theScene</i>	
<i>parent</i>	

17.62.1.2 ~SaveObject() [~SaveObject](#) ()

[SaveObject::~~SaveObject.](#)

17.62.2 Member Function Documentation

17.62.2.1 addArc() `void addArc (`
 `EmbPattern * pattern,`
 `QGraphicsItem * item)`

[SaveObject::addArc.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.2 addBlock() `void addBlock (`
 `EmbPattern * pattern,`
 `QGraphicsItem * item)`

[SaveObject::addBlock.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.3 addCircle() `void addCircle (`
 `EmbPattern * pattern,`
 `QGraphicsItem * item)`

[SaveObject::addCircle.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.4 addDimAligned() `void addDimAligned (`
 `EmbPattern * pattern,`
 `QGraphicsItem * item)`

[SaveObject::addDimAligned.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.5 addDimAngular() void addDimAngular (
 [EmbPattern](#) * *pattern*,
 QGraphicsItem * *item*)

[SaveObject::addDimAngular](#).

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.6 addDimArcLength() void addDimArcLength (
 [EmbPattern](#) * *pattern*,
 QGraphicsItem * *item*)

[SaveObject::addDimArcLength](#).

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.7 addDimDiameter() void addDimDiameter (
 [EmbPattern](#) * *pattern*,
 QGraphicsItem * *item*)

[SaveObject::addDimDiameter](#).

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.8 addDimLeader() void addDimLeader (

```
    EmbPattern * pattern,  
    QGraphicsItem * item )
```

[SaveObject::addDimLeader.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.9 addDimLinear() void addDimLinear (
 EmbPattern * pattern,
 QGraphicsItem * item)

[SaveObject::addDimLinear.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.10 addDimOrdinate() void addDimOrdinate (
 EmbPattern * pattern,
 QGraphicsItem * item)

[SaveObject::addDimOrdinate.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.11 addDimRadius() void addDimRadius (
 EmbPattern * pattern,
 QGraphicsItem * item)

[SaveObject::addDimRadius.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.12 addEllipse() void addEllipse (
 [EmbPattern](#) * *pattern*,
 QGraphicsItem * *item*)

[SaveObject::addEllipse.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.13 addEllipseArc() void addEllipseArc (
 [EmbPattern](#) * *pattern*,
 QGraphicsItem * *item*)

[SaveObject::addEllipseArc.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.14 addGrid() void addGrid (
 [EmbPattern](#) * *pattern*,
 QGraphicsItem * *item*)

[SaveObject::addGrid.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.15 addHatch() void addHatch (
 [EmbPattern](#) * *pattern*,
 QGraphicsItem * *item*)

[SaveObject::addHatch.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.16 addImage() void addImage (
 EmbPattern * *pattern*,
 QGraphicsItem * *item*)

[SaveObject::addImage.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.17 addInfiniteLine() void addInfiniteLine (
 EmbPattern * *pattern*,
 QGraphicsItem * *item*)

[SaveObject::addInfiniteLine.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.18 addLine() void addLine (
 EmbPattern * *pattern*,
 QGraphicsItem * *item*)

[SaveObject::addLine.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.19 addPath() void addPath (
 EmbPattern * *pattern*,
 QGraphicsItem * *item*)

```
    EmbPattern * pattern,  
    QGraphicsItem * item )
```

[SaveObject::addPath.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

Todo Reimplement [addPolyline\(\)](#) using the libembroidery C API

17.62.2.20 addPoint() void addPoint (
 EmbPattern * pattern,
 QGraphicsItem * item)

[SaveObject::addPoint.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.21 addPolygon() void addPolygon (
 EmbPattern * pattern,
 QGraphicsItem * item)

[SaveObject::addPolygon.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.22 addPolyline() void addPolyline (
 EmbPattern * pattern,
 QGraphicsItem * item)

[SaveObject::addPolyline.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.23 addRay() `void addRay (`
 `EmbPattern * pattern,`
 `QGraphicsItem * item)`

[SaveObject::addRay.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.24 addRectangle() `void addRectangle (`
 `EmbPattern * pattern,`
 `QGraphicsItem * item)`

[SaveObject::addRectangle.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.25 addSlot() `void addSlot (`
 `EmbPattern * pattern,`
 `QGraphicsItem * item)`

[SaveObject::addSlot.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.26 addSpline() `void addSpline (`
 `EmbPattern * pattern,`
 `QGraphicsItem * item)`

[SaveObject::addSpline.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.27 addTextMulti() `void addTextMulti (`
 `EmbPattern * pattern,`
 `QGraphicsItem * item)`

[SaveObject::addTextMulti.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.28 addTextSingle() `void addTextSingle (`
 `EmbPattern * pattern,`
 `QGraphicsItem * item)`

[SaveObject::addTextSingle.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

17.62.2.29 save() `bool save (`
 `QString fileName)`

Returns whether the save to file process was successful.

Todo Before saving to a stitch only format, Embroidermodder needs to calculate the optimal path to minimize jump stitches. Also based upon which layer needs to be stitched first, the path to the next object needs to be hidden beneath fills that will come later. When finding the optimal path, we need to take into account the color of the thread, as we do not want to try to hide dark colored stitches beneath light colored fills.

```

17.62.2.30 toPolyline() void toPolyline (
    EmbPattern * pattern,
    const QPointF & objPos,
    const QPainterPath & objPath,
    QString layer,
    const QColor & color,
    QString lineType,
    QString lineWeight )

```

[SaveObject::toPolyline.](#)

Parameters

<i>pattern</i>	
<i>objPos</i>	
<i>objPath</i>	
<i>layer</i>	
<i>color</i>	
<i>lineType</i>	
<i>lineWeight</i>	

Note

This function should be used to interpret various object types and save them as polylines for stitchOnly formats.

Todo FIX EmbPolyline* polyObject = embPolyline_init(pointList, color_out, 1); //TODO: proper lineType emb↵
 Pattern_addPolylineAbs(pattern, polyObject);

17.62.3 Member Data Documentation

17.62.3.1 formatType int formatType

17.62.3.2 gscene QGraphicsScene* gscene

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

- embroidermodder2/[embroidermodder.h](#)
- embroidermodder2/[objects.cpp](#)

17.63 SelectBox Class Reference

```
#include <embroidermodder.h>
```


Public Slots

- void [setDirection](#) (int dir)
- void [setColors](#) (const QColor &colorL, const QColor &fillL, const QColor &colorR, const QColor &fillR, int newAlpha)

Public Member Functions

- [SelectBox](#) (Shape s, QWidget *parent=0)
- void [forceRepaint](#) ()

Public Attributes

- QColor [leftBrushColor](#)
- QColor [rightBrushColor](#)
- QColor [leftPenColor](#)
- QColor [rightPenColor](#)
- uint8_t [alpha](#)
- QBrush [dirBrush](#)
- QBrush [leftBrush](#)
- QBrush [rightBrush](#)
- QPen [dirPen](#)
- QPen [leftPen](#)
- QPen [rightPen](#)
- bool [boxDir](#)

Protected Member Functions

- void [paintEvent](#) (QPaintEvent *)

17.63.1 Constructor & Destructor Documentation

17.63.1.1 [SelectBox\(\)](#) `SelectBox (`
 Shape s,
 QWidget * parent = 0)

Embroidermodder 2.

Copyright 2013-2022 The Embroidermodder Team Embroidermodder 2 is Open Source Software. See LICENSE for licensing terms.

Use Python's PEP7 style guide. <https://peps.python.org/pep-0007/>

17.63.2 Member Function Documentation

17.63.2.1 [forceRepaint\(\)](#) `void forceRepaint ()`

17.63.2.2 paintEvent() void paintEvent (
 QPaintEvent *) [protected]

17.63.2.3 setColors void setColors (
 const QColor & *colorL*,
 const QColor & *fillL*,
 const QColor & *colorR*,
 const QColor & *fillR*,
 int *newAlpha*) [slot]

17.63.2.4 setDirection void setDirection (
 int *dir*) [slot]

17.63.3 Member Data Documentation

17.63.3.1 alpha uint8_t alpha

17.63.3.2 boxDir bool boxDir

17.63.3.3 dirBrush QBrush dirBrush

17.63.3.4 dirPen QPen dirPen

17.63.3.5 leftBrush QBrush leftBrush

17.63.3.6 leftBrushColor QColor leftBrushColor

17.63.3.7 leftPen QPen leftPen

17.63.3.8 leftPenColor QColor leftPenColor

17.63.3.9 rightBrush QBrush rightBrush

17.63.3.10 rightBrushColor QColor rightBrushColor

17.63.3.11 rightPen QPen rightPen

17.63.3.12 rightPenColor QColor rightPenColor

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

- embroidermodder2/[embroidermodder.h](#)
- embroidermodder2/[selectbox.cpp](#)

17.64 Settings_Dialog Class Reference

```
#include <embroidermodder.h>
```

Signals

- void [buttonCustomFilterSelectAll](#) (bool)
- void [buttonCustomFilterClearAll](#) (bool)
- void [buttonQSnapSelectAll](#) (bool)
- void [buttonQSnapClearAll](#) (bool)

Public Member Functions

- [Settings_Dialog](#) (QString showTab=QString(), QWidget *parent=0)
- [~Settings_Dialog](#) ()
- QWidget * [createTabGeneral](#) ()
- QWidget * [createTabFilesPaths](#) ()
- QWidget * [createTabDisplay](#) ()
- QWidget * [createTabPrompt](#) ()
- QWidget * [createTabOpenSave](#) ()
- QWidget * [createTabPrinting](#) ()
- QWidget * [createTabSnap](#) ()
- QWidget * [createTabGridRuler](#) ()
- QWidget * [createTabOrthoPolar](#) ()
- QWidget * [createTabQuickSnap](#) ()
- QWidget * [createTabQuickTrack](#) ()
- QWidget * [createTabLineWeight](#) ()
- QWidget * [createTabSelection](#) ()
- void [addColorsToComboBox](#) (QComboBox *comboBox)
- void [create_float_spinbox](#) (QGroupBox *gb, QGridLayout *gridLayout, const char *label_in, [EmbReal](#) single_step, [EmbReal](#) lower, [EmbReal](#) upper, [String](#), int row)
- QCheckBox * [create_checkbox](#) (QGroupBox *groupbox, [String](#) label)

Public Attributes

- QTabWidget * [tabWidget](#)
- QDialogButtonBox * [buttonBox](#)

Private Slots

- void [comboBoxIconSizeCurrentIndexChanged](#) (int)
[Settings_Dialog::comboBoxIconSizeCurrentIndexChanged.](#)
- void [checkBoxGeneralMdiBGUseLogoStateChanged](#) (int)
- void [chooseGeneralMdiBackgroundLogo](#) ()
- void [checkBoxGeneralMdiBGUseTextureStateChanged](#) (int)
[Settings_Dialog::checkBoxGeneralMdiBGUseTextureStateChanged.](#)
- void [chooseGeneralMdiBackgroundTexture](#) ()
- void [checkBoxGeneralMdiBGUseColorStateChanged](#) (int)
- void [chooseGeneralMdiBackgroundColor](#) ()
- void [currentGeneralMdiBackgroundColorChanged](#) (const QColor &)
- void [checkBoxShowScrollBarsStateChanged](#) (int)

- void [comboBoxScrollBarWidgetCurrentIndexChanged](#) (int)
- void [chooseDisplayCrossHairColor](#) ()
- void [currentDisplayCrossHairColorChanged](#) (const QColor &)
- void [chooseDisplayBackgroundColor](#) ()
- void [currentDisplayBackgroundColorChanged](#) (const QColor &)
- void [chooseDisplaySelectBoxLeftColor](#) ()
- void [currentDisplaySelectBoxLeftColorChanged](#) (const QColor &)
- void [chooseDisplaySelectBoxLeftFill](#) ()
- void [currentDisplaySelectBoxLeftFillChanged](#) (const QColor &)
- void [chooseDisplaySelectBoxRightColor](#) ()
- void [currentDisplaySelectBoxRightColorChanged](#) (const QColor &)
- void [chooseDisplaySelectBoxRightFill](#) ()
- void [currentDisplaySelectBoxRightFillChanged](#) (const QColor &)
- void [comboBoxSelectionCoolGripColorCurrentIndexChanged](#) (int index)
- void [comboBoxSelectionHotGripColorCurrentIndexChanged](#) (int index)
- void [spinBoxDisplaySelectBoxAlphaValueChanged](#) (int)
- void [choosePromptTextColor](#) ()
- void [currentPromptTextColorChanged](#) (const QColor &)
- void [choosePromptBackgroundColor](#) ()
- void [currentPromptBackgroundColorChanged](#) (const QColor &)
- void [comboBoxPromptFontFamilyCurrentIndexChanged](#) (QString)
- void [comboBoxPromptFontStyleCurrentIndexChanged](#) (QString)
- void [spinBoxPromptFontSizeValueChanged](#) (int)
- void [checkBoxPromptSaveHistoryAsHtmlStateChanged](#) (int)
- void [checkBoxCustomFilterStateChanged](#) (int)
- void [buttonCustomFilterSelectAllClicked](#) ()
- void [buttonCustomFilterClearAllClicked](#) ()
- void [checkBoxGridColorMatchCrossHairStateChanged](#) (int)
- void [chooseGridColor](#) ()
- void [currentGridColorChanged](#) (const QColor &)
- void [checkBoxGridLoadFromFileStateChanged](#) (int)
- void [comboBoxGridTypeCurrentIndexChanged](#) (QString)
- [Settings_Dialog::comboBoxGridTypeCurrentIndexChanged.](#)
- void [checkBoxGridCenterOnOriginStateChanged](#) (int)
- void [checkBoxRulerShowOnLoadStateChanged](#) (int)
- void [comboBoxRulerMetricCurrentIndexChanged](#) (int)
- void [chooseRulerColor](#) ()
- void [currentRulerColorChanged](#) (const QColor &)
- void [spinBoxRulerPixelSizeValueChanged](#) (double)
- void [buttonQSnapSelectAllClicked](#) ()
- void [buttonQSnapClearAllClicked](#) ()
- void [comboBoxQSnapLocatorColorCurrentIndexChanged](#) (int)
- void [checkBoxLwtShowLwtStateChanged](#) (int)
- void [checkBoxLwtRealRenderStateChanged](#) (int)
- void [acceptChanges](#) ()
- void [rejectChanges](#) ()
- [Settings_Dialog::rejectChanges.](#)

17.64.1 Constructor & Destructor Documentation

17.64.1.1 Settings_Dialog() [Settings_Dialog](#) (
 QString *showTab* = *QString()*,
 QWidget * *parent* = 0)

17.64.1.2 `~Settings_Dialog()` `~Settings_Dialog ()`

17.64.2 Member Function Documentation

17.64.2.1 `acceptChanges` `void acceptChanges ()` [private], [slot]

17.64.2.2 `addColorsToComboBox()` `void addColorsToComboBox (`
`QComboBox * comboBox)`

17.64.2.3 `buttonCustomFilterClearAll` `void buttonCustomFilterClearAll (`
`bool)` [signal]

17.64.2.4 `buttonCustomFilterClearAllClicked` `void buttonCustomFilterClearAllClicked ()` [private], [slot]

17.64.2.5 `buttonCustomFilterSelectAll` `void buttonCustomFilterSelectAll (`
`bool)` [signal]

17.64.2.6 `buttonCustomFilterSelectAllClicked` `void buttonCustomFilterSelectAllClicked ()` [private], [slot]

17.64.2.7 `buttonQSnapClearAll` `void buttonQSnapClearAll (`
`bool)` [signal]

17.64.2.8 `buttonQSnapClearAllClicked` `void buttonQSnapClearAllClicked ()` [private], [slot]

17.64.2.9 `buttonQSnapSelectAll` `void buttonQSnapSelectAll (`
`bool)` [signal]

17.64.2.10 `buttonQSnapSelectAllClicked` `void buttonQSnapSelectAllClicked ()` [private], [slot]

17.64.2.11 `checkBoxCustomFilterStateChanged` `void checkBoxCustomFilterStateChanged (`
`int checked)` [private], [slot]

17.64.2.12 `checkBoxGeneralMdiBGUseColorStateChanged` `void checkBoxGeneralMdiBGUseColor↔`
`StateChanged (`
`int checked)` [private], [slot]

17.64.2.13 `checkBoxGeneralMdiBGUseLogoStateChanged` `void checkBoxGeneralMdiBGUseLogoState↔`
`Changed (`
`int checked)` [private], [slot]

17.64.2.14 checkBoxGeneralMdiBGUseTextureStateChanged void checkBoxGeneralMdiBGUseTexture↔
StateChanged (
 int checked) [private], [slot]
[Settings_Dialog::checkBoxGeneralMdiBGUseTextureStateChanged.](#)

Parameters

<i>checked</i>	<input type="checkbox"/>
----------------	--------------------------

17.64.2.15 checkBoxGridCenterOnOriginStateChanged void checkBoxGridCenterOnOriginStateChanged (
 int checked) [private], [slot]

17.64.2.16 checkBoxGridColorMatchCrossHairStateChanged void checkBoxGridColorMatchCrossHair↔
StateChanged (
 int checked) [private], [slot]

17.64.2.17 checkBoxGridLoadFromFileStateChanged void checkBoxGridLoadFromFileStateChanged (
 int checked) [private], [slot]

17.64.2.18 checkBoxLwtRealRenderStateChanged void checkBoxLwtRealRenderStateChanged (
 int checked) [private], [slot]

17.64.2.19 checkBoxLwtShowLwtStateChanged void checkBoxLwtShowLwtStateChanged (
 int checked) [private], [slot]

17.64.2.20 checkBoxPromptSaveHistoryAsHtmlStateChanged void checkBoxPromptSaveHistoryAs↔
HtmlStateChanged (
 int checked) [private], [slot]

17.64.2.21 checkBoxRulerShowOnLoadStateChanged void checkBoxRulerShowOnLoadStateChanged (
 int checked) [private], [slot]

17.64.2.22 checkBoxShowScrollBarsStateChanged void checkBoxShowScrollBarsStateChanged (
 int checked) [private], [slot]

17.64.2.23 chooseDisplayBackgroundColor void chooseDisplayBackgroundColor () [private],
[slot]

17.64.2.24 chooseDisplayCrossHairColor void chooseDisplayCrossHairColor () [private], [slot]

17.64.2.25 chooseDisplaySelectBoxLeftColor void chooseDisplaySelectBoxLeftColor () [private],
[slot]

17.64.2.26 chooseDisplaySelectBoxLeftFill void chooseDisplaySelectBoxLeftFill () [private], [slot]

17.64.2.27 chooseDisplaySelectBoxRightColor void chooseDisplaySelectBoxRightColor () [private], [slot]

17.64.2.28 chooseDisplaySelectBoxRightFill void chooseDisplaySelectBoxRightFill () [private], [slot]

17.64.2.29 chooseGeneralMdiBackgroundColor void chooseGeneralMdiBackgroundColor () [private], [slot]

17.64.2.30 chooseGeneralMdiBackgroundLogo void chooseGeneralMdiBackgroundLogo () [private], [slot]

17.64.2.31 chooseGeneralMdiBackgroundTexture void chooseGeneralMdiBackgroundTexture () [private], [slot]

17.64.2.32 chooseGridColor void chooseGridColor () [private], [slot]

17.64.2.33 choosePromptBackgroundColor void choosePromptBackgroundColor () [private], [slot]

17.64.2.34 choosePromptTextColor void choosePromptTextColor () [private], [slot]

17.64.2.35 chooseRulerColor void chooseRulerColor () [private], [slot]

17.64.2.36 comboBoxGridTypeCurrentIndexChanged void comboBoxGridTypeCurrentIndexChanged (QString type) [private], [slot]
[Settings_Dialog::comboBoxGridTypeCurrentIndexChanged.](#)

Parameters

<i>type</i>	
-------------	--

17.64.2.37 comboBoxIconSizeCurrentIndexChanged void comboBoxIconSizeCurrentIndexChanged (int index) [private], [slot]
[Settings_Dialog::comboBoxIconSizeCurrentIndexChanged.](#)

Parameters

<i>index</i>	
--------------	--

17.64.2.38 comboBoxPromptFontFamilyCurrentIndexChanged void comboBoxPromptFontFamily↔
CurrentIndexChanged (
 QString *family*) [private], [slot]

17.64.2.39 comboBoxPromptFontStyleCurrentIndexChanged void comboBoxPromptFontStyleCurrent↔
IndexChanged (
 QString *style*) [private], [slot]

17.64.2.40 comboBoxQSnapLocatorColorCurrentIndexChanged void comboBoxQSnapLocatorColor↔
CurrentIndexChanged (
 int *index*) [private], [slot]

17.64.2.41 comboBoxRulerMetricCurrentIndexChanged void comboBoxRulerMetricCurrentIndex↔
Changed (
 int *index*) [private], [slot]

17.64.2.42 comboBoxScrollBarWidgetCurrentIndexChanged void comboBoxScrollBarWidgetCurrent↔
IndexChanged (
 int *index*) [private], [slot]

17.64.2.43 comboBoxSelectionCoolGripColorCurrentIndexChanged void comboBoxSelectionCoolGrip↔
ColorCurrentIndexChanged (
 int *index*) [private], [slot]

17.64.2.44 comboBoxSelectionHotGripColorCurrentIndexChanged void comboBoxSelectionHotGrip↔
ColorCurrentIndexChanged (
 int *index*) [private], [slot]

17.64.2.45 create_checkbox() QCheckBox * create_checkbox (
 QGroupBox * *groupbox*,
 String *label*)

17.64.2.46 create_float_spinbox() void create_float_spinbox (
 QGroupBox * *gb*,
 QGridLayout * *gridLayout*,
 const char * *label_in*,
 EmbReal *single_step*,
 EmbReal *lower*,
 EmbReal *upper*,
 String *key*,
 int *row*)

17.64.2.47 createTabDisplay() QWidget * createTabDisplay ()

17.64.2.48 createTabFilePaths() `QWidget * createTabFilePaths ()`

17.64.2.49 createTabGeneral() `QWidget * createTabGeneral ()`

17.64.2.50 createTabGridRuler() `QWidget * createTabGridRuler ()`

17.64.2.51 createTabLineWeight() `QWidget * createTabLineWeight ()`

17.64.2.52 createTabOpenSave() `QWidget * createTabOpenSave ()`

17.64.2.53 createTabOrthoPolar() `QWidget * createTabOrthoPolar ()`

17.64.2.54 createTabPrinting() `QWidget * createTabPrinting ()`

17.64.2.55 createTabPrompt() `QWidget * createTabPrompt ()`

17.64.2.56 createTabQuickSnap() `QWidget * createTabQuickSnap ()`

17.64.2.57 createTabQuickTrack() `QWidget * createTabQuickTrack ()`

17.64.2.58 createTabSelection() `QWidget * createTabSelection ()`

17.64.2.59 createTabSnap() `QWidget * createTabSnap ()`

17.64.2.60 currentDisplayBackgroundColorChanged `void currentDisplayBackgroundColorChanged (const QColor & color) [private], [slot]`

17.64.2.61 currentDisplayCrossHairColorChanged `void currentDisplayCrossHairColorChanged (const QColor & color) [private], [slot]`

17.64.2.62 currentDisplaySelectBoxLeftColorChanged `void currentDisplaySelectBoxLeftColor↔
Changed (const QColor & color) [private], [slot]`

17.64.2.63 currentDisplaySelectBoxLeftFillChanged `void currentDisplaySelectBoxLeftFillChanged (const QColor & color) [private], [slot]`

17.64.2.64 currentDisplaySelectBoxRightColorChanged void currentDisplaySelectBoxRightColor↔
Changed (
 const QColor & color) [private], [slot]

17.64.2.65 currentDisplaySelectBoxRightFillColorChanged void currentDisplaySelectBoxRightFillColorChanged
(
 const QColor & color) [private], [slot]

17.64.2.66 currentGeneralMdiBackgroundColorChanged void currentGeneralMdiBackgroundColor↔
Changed (
 const QColor & color) [private], [slot]

17.64.2.67 currentGridColorChanged void currentGridColorChanged (
 const QColor & color) [private], [slot]

17.64.2.68 currentPromptBackgroundColorChanged void currentPromptBackgroundColorChanged (
 const QColor & color) [private], [slot]

17.64.2.69 currentPromptTextColorChanged void currentPromptTextColorChanged (
 const QColor & color) [private], [slot]

17.64.2.70 currentRulerColorChanged void currentRulerColorChanged (
 const QColor & color) [private], [slot]

17.64.2.71 rejectChanges void rejectChanges () [private], [slot]
[Settings_Dialog::rejectChanges](#).

17.64.2.72 spinBoxDisplaySelectBoxAlphaValueChanged void spinBoxDisplaySelectBoxAlphaValue↔
Changed (
 int value) [private], [slot]

17.64.2.73 spinBoxPromptFontSizeValueChanged void spinBoxPromptFontSizeValueChanged (
 int value) [private], [slot]

17.64.2.74 spinBoxRulerPixelSizeValueChanged void spinBoxRulerPixelSizeValueChanged (
 double value) [private], [slot]

17.64.3 Member Data Documentation

17.64.3.1 buttonBox QDialogButtonBox* buttonBox

17.64.3.2 tabWidget `QTabWidget* tabWidget`

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

- embroidermodder2/[embroidermodder.h](#)
- embroidermodder2/[settings-dialog.cpp](#)

17.65 StatusBar Class Reference

```
#include <embroidermodder.h>
```

Public Member Functions

- [StatusBar](#) (`QWidget *parent=0`)
- void [setMouseCoord](#) (`EmbReal x`, `EmbReal y`)
- void [context_menu_action](#) (`QToolButton *button`, `const char *icon`, `const char *label`, `QMenu *menu`, [String](#) `setting_page`)
- void [toggle](#) ([String](#) `key`, `bool on`)
- void [context_menu_event](#) (`QContextMenuEvent *event`, `QToolButton *button`)

Public Attributes

- `std::unordered_map< String, QToolButton * >` [buttons](#)
- `QLabel *` [statusBarMouseCoord](#)

17.65.1 Detailed Description**17.65.2 Constructor & Destructor Documentation**

17.65.2.1 StatusBar() `StatusBar (`
`QWidget * parent = 0)`

17.65.3 Member Function Documentation

17.65.3.1 context_menu_action() `void context_menu_action (`
`QToolButton * button,`
`const char * icon,`
`const char * label,`
`QMenu * menu,`
`String setting_page)`

17.65.3.2 context_menu_event() `void context_menu_event (`
`QContextMenuEvent * event,`
`QToolButton * button)`

17.65.3.3 setMouseCoord() `void setMouseCoord (`
`EmbReal x,`
`EmbReal y)`

17.65.3.4 toggle() `void toggle (`
`String key,`
`bool on)`

17.65.4 Member Data Documentation

17.65.4.1 buttons `std::unordered_map<String, QToolButton*> buttons`

17.65.4.2 statusBarMouseCoord `QLabel* statusBarMouseCoord`

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

- embroidermodder2/[embroidermodder.h](#)
- embroidermodder2/[statusbar.cpp](#)

17.66 StxThread_ Struct Reference

```
#include <embroidery_internal.h>
```

Public Attributes

- char * [colorCode](#)
- char * [colorName](#)
- char * [sectionName](#)
- [SubDescriptor](#) * [subDescriptors](#)
- [EmbColor](#) [stxColor](#)

17.66.1 Member Data Documentation

17.66.1.1 colorCode `char* colorCode`

17.66.1.2 colorName `char* colorName`

17.66.1.3 sectionName `char* sectionName`

17.66.1.4 stxColor `EmbColor stxColor`

17.66.1.5 subDescriptors `SubDescriptor* subDescriptors`

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

- extern/libembroidery/src/[embroidery_internal.h](#)

17.67 SubDescriptor_ Struct Reference

```
#include <embroidery_internal.h>
```

Public Attributes

- int [someNum](#)
- int [someInt](#)
- int [someOtherInt](#)
- char * [colorCode](#)
- char * [colorName](#)

17.67.1 Member Data Documentation

17.67.1.1 colorCode `char* colorCode`

Todo better variable naming

17.67.1.2 colorName `char* colorName`

17.67.1.3 someInt `int someInt`

Todo better variable naming

17.67.1.4 someNum `int someNum`

17.67.1.5 someOtherInt `int someOtherInt`

Todo better variable naming

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

- `extern/libembroidery/src/embroidery_internal.h`

17.68 SvgAttribute_ Struct Reference

```
#include <embroidery_internal.h>
```

Public Attributes

- `char * name`
- `char * value`

17.68.1 Member Data Documentation

17.68.1.1 name `char* name`

17.68.1.2 value `char* value`

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

- `extern/libembroidery/src/embroidery_internal.h`

17.69 thread_color_ Struct Reference

```
#include <embroidery.h>
```

Public Attributes

- `char name [22]`
- `unsigned int hex_code`
- `int manufacturer_code`

17.69.1 Member Data Documentation

17.69.1.1 hex_code unsigned int hex_code

17.69.1.2 manufacturer_code int manufacturer_code

17.69.1.3 name char name[22]

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

- extern/libembroidery/src/[embroidery.h](#)

17.70 ThredExtension_ Struct Reference

```
#include <embroidery_internal.h>
```

Public Attributes

- float [hoopX](#)
- float [hoopY](#)
- float [stitchGranularity](#)
- char [creatorName](#) [50]
- char [modifierName](#) [50]
- char [auxFormat](#)
- char [reserved](#) [31]

17.70.1 Member Data Documentation

17.70.1.1 auxFormat char auxFormat

17.70.1.2 creatorName char creatorName[50]

17.70.1.3 hoopX float hoopX

17.70.1.4 hoopY float hoopY

17.70.1.5 modifierName char modifierName[50]

17.70.1.6 reserved char reserved[31]

17.70.1.7 stitchGranularity float stitchGranularity

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

- extern/libembroidery/src/[embroidery_internal.h](#)

17.71 ThredHeader_ Struct Reference

```
#include <embroidery_internal.h>
```

Public Attributes

- unsigned int [sigVersion](#)
- unsigned int [length](#)
- unsigned short [numStiches](#)
- unsigned short [hoopSize](#)
- unsigned short [reserved](#) [7]

17.71.1 Member Data Documentation

17.71.1.1 hoopSize unsigned short hoopSize

17.71.1.2 length unsigned int length

17.71.1.3 numStiches unsigned short numStiches

17.71.1.4 reserved unsigned short reserved[7]

17.71.1.5 sigVersion unsigned int sigVersion

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

- extern/libembroidery/src/[embroidery_internal.h](#)

17.72 UndoableCommand Class Reference

```
#include <embroidermodder.h>
```

Public Member Functions

- [UndoableCommand](#) (String command, QString text, [Geometry](#) *obj, [View](#) *v, QUndoCommand *parent=0)
- [UndoableCommand](#) (EmbVector d, QString text, [Geometry](#) *obj, [View](#) *v, QUndoCommand *parent=0)
- [UndoableCommand](#) (String command, EmbVector pivot, EmbReal angle, QString text, [Geometry](#) *obj, [View](#) *v, QUndoCommand *parent=0)
- [UndoableCommand](#) (QString type, [View](#) *v, QUndoCommand *parent=0)
- [UndoableCommand](#) (const QPointF beforePoint, const QPointF afterPoint, QString text, [Geometry](#) *obj, [View](#) *v, QUndoCommand *parent=0)
- [UndoableCommand](#) (EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, QString text, [Geometry](#) *obj, [View](#) *v, QUndoCommand *parent=0)
- int [id](#) ()
- bool [mergeWith](#) (const QUndoCommand *command)
- void [undo](#) ()
- void [redo](#) ()
- void [mirror](#) ()
- void [rotate](#) (EmbVector pivot, EmbReal rot)

Public Attributes

- [Geometry](#) * [object](#)
- [View](#) * [gview](#)
- [String](#) [command](#)
- [EmbVector](#) [delta](#)
- [EmbVector](#) [pivot](#)
- [QPointF](#) [before](#)
- [QPointF](#) [after](#)
- [EmbReal](#) [angle](#)
- [EmbReal](#) [factor](#)
- [QString](#) [navType](#)
- [QTransform](#) [fromTransform](#)
- [QTransform](#) [toTransform](#)
- [QPointF](#) [fromCenter](#)
- [QPointF](#) [toCenter](#)
- [QLineF](#) [mirrorLine](#)
- [bool](#) [done](#)

17.72.1 Constructor & Destructor Documentation

17.72.1.1 UndoableCommand() [1/6] [UndoableCommand](#) (
 [String](#) *command*,
 [QString](#) *text*,
 [Geometry](#) * *obj*,
 [View](#) * *v*,
 [QUndoCommand](#) * *parent* = 0)

17.72.1.2 UndoableCommand() [2/6] [UndoableCommand](#) (
 [EmbVector](#) *d*,
 [QString](#) *text*,
 [Geometry](#) * *obj*,
 [View](#) * *v*,
 [QUndoCommand](#) * *parent* = 0)

17.72.1.3 UndoableCommand() [3/6] [UndoableCommand](#) (
 [String](#) *command*,
 [EmbVector](#) *pivot*,
 [EmbReal](#) *angle*,
 [QString](#) *text*,
 [Geometry](#) * *obj*,
 [View](#) * *v*,
 [QUndoCommand](#) * *parent* = 0)

17.72.1.4 UndoableCommand() [4/6] [UndoableCommand](#) (
 [QString](#) *type*,
 [View](#) * *v*,
 [QUndoCommand](#) * *parent* = 0)

17.72.1.5 UndoableCommand() [5/6] `UndoableCommand` (

```

    const QPointF beforePoint,
    const QPointF afterPoint,
    QString text,
    Geometry * obj,
    View * v,
    QUndoCommand * parent = 0 )
```

17.72.1.6 UndoableCommand() [6/6] `UndoableCommand` (

```

    EmbReal x1,
    EmbReal y1,
    EmbReal x2,
    EmbReal y2,
    QString text,
    Geometry * obj,
    View * v,
    QUndoCommand * parent = 0 )
```

17.72.2 Member Function Documentation

17.72.2.1 id() `int id ()` [inline]

17.72.2.2 mergeWith() `bool mergeWith (`
`const QUndoCommand * command)`

17.72.2.3 mirror() `void mirror ()`

17.72.2.4 redo() `void redo ()`

17.72.2.5 rotate() `void rotate (`
`EmbVector pivot,`
`EmbReal rot)`

17.72.2.6 undo() `void undo ()`

17.72.3 Member Data Documentation

17.72.3.1 after `QPointF after`

17.72.3.2 angle `EmbReal angle`

17.72.3.3 before `QPointF before`

17.72.3.4 command [String](#) command

17.72.3.5 delta [EmbVector](#) delta

17.72.3.6 done bool done

17.72.3.7 factor [EmbReal](#) factor

17.72.3.8 fromCenter [QPointF](#) fromCenter

17.72.3.9 fromTransform [QTransform](#) fromTransform

17.72.3.10 gview [View*](#) gview

17.72.3.11 mirrorLine [QLineF](#) mirrorLine

17.72.3.12 navType [QString](#) navType

17.72.3.13 object [Geometry*](#) object

17.72.3.14 pivot [EmbVector](#) pivot

17.72.3.15 toCenter [QPointF](#) toCenter

17.72.3.16 toTransform [QTransform](#) toTransform

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

- embroidermodder2/[embroidermodder.h](#)
- embroidermodder2/[undo-commands.cpp](#)

17.73 UndoEditor Class Reference

```
#include <embroidermodder.h>
```

Public Slots

- void [undo](#) ()
- void [redo](#) ()
- void [updateCleanIcon](#) (bool opened)

Public Member Functions

- [UndoEditor](#) (QString iconDirectory=QString(), QWidget *widgetToFocus=0, QWidget *parent=0)
- [~UndoEditor](#) ()
- void [addStack](#) (QUndoStack *stack)
- bool [canUndo](#) ()
- bool [canRedo](#) ()
- QString [undoText](#) ()
- QString [redoText](#) ()

Public Attributes

- QWidget * [focusWidget](#)
- QString [iconDir](#)
- int [iconSize](#)
- QUndoGroup * [undoGroup](#)
- QUndoView * [undoView](#)

17.73.1 Constructor & Destructor Documentation

17.73.1.1 UndoEditor() [UndoEditor](#) (
 QString *iconDirectory* = QString(),
 QWidget * *widgetToFocus* = 0,
 QWidget * *parent* = 0)

17.73.1.2 ~UndoEditor() [~UndoEditor](#) ()

17.73.2 Member Function Documentation

17.73.2.1 addStack() void [addStack](#) (
 QUndoStack * *stack*)

17.73.2.2 canRedo() bool [canRedo](#) ()

17.73.2.3 canUndo() bool [canUndo](#) ()

17.73.2.4 redo void [redo](#) () [slot]

17.73.2.5 redoText() QString [redoText](#) ()

17.73.2.6 undo void [undo](#) () [slot]

17.73.2.7 undoText() QString [undoText](#) ()

17.73.2.8 updateCleanIcon void updateCleanIcon (
bool *opened*) [slot]

17.73.3 Member Data Documentation

17.73.3.1 focusWidget QWidget* focusWidget

17.73.3.2 iconDir QString iconDir

17.73.3.3 iconSize int iconSize

17.73.3.4 undoGroup QUndoGroup* undoGroup

17.73.3.5 undoView QUndoView* undoView

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

- embroidermodder2/[embroidermodder.h](#)
- embroidermodder2/[undo-editor.cpp](#)

17.74 View Class Reference

```
#include <embroidermodder.h>
```

Public Slots

- void [zoomIn](#) ()
- void [zoomOut](#) ()
- void [zoomWindow](#) ()
- void [zoomSelected](#) ()
- void [zoomExtents](#) ()
- void [panRealTime](#) ()
- void [panPoint](#) ()
- void [panLeft](#) ()
- void [panRight](#) ()
- void [panUp](#) ()
- void [panDown](#) ()
- void [selectAll](#) ()
- void [selectionChanged](#) ()
- void [clearSelection](#) ()
- void [deleteSelected](#) ()
- void [moveSelected](#) (EmbReal dx, EmbReal dy)
- void [cut](#) ()
- void [copy](#) ()
- void [paste](#) ()
- void [repeatAction](#) ()
- void [moveAction](#) ()
- void [scaleAction](#) ()
- void [scaleSelected](#) (EmbReal x, EmbReal y, EmbReal factor)
- void [rotateAction](#) ()
- void [rotateSelected](#) (EmbReal x, EmbReal y, EmbReal rot)

- void [mirrorSelected](#) ([EmbReal](#) x1, [EmbReal](#) y1, [EmbReal](#) x2, [EmbReal](#) y2)
- int [numSelected](#) ()
- void [deletePressed](#) ()
- void [escapePressed](#) ()
- void [cornerButtonClicked](#) ()
- void [showScrollBars](#) (bool val)
- void [setCornerButton](#) ()
- void [setCrossHairColor](#) (QRgb color)
- void [setCrossHairSize](#) (uint8_t percent)
- void [setBackgroundColors](#) (QRgb colorL, QRgb fillL, QRgb colorR, QRgb fillR, int alpha)
- void [toggleSnap](#) (bool on)
- void [toggleGrid](#) (bool on)
- void [toggleRuler](#) (bool on)
- void [toggleOrtho](#) (bool on)
- void [togglePolar](#) (bool on)
- void [toggleQSnap](#) (bool on)
- void [toggleQTrack](#) (bool on)
- void [toggleLwt](#) (bool on)
- void [toggleReal](#) (bool on)
- bool [isLwtEnabled](#) ()
- bool [isRealEnabled](#) ()
- void [setGridColor](#) (QRgb color)
- void [createGrid](#) (QString gridType)
- void [setRulerColor](#) (QRgb color)
- void [previewOn](#) (String clone, String mode, [EmbReal](#) x, [EmbReal](#) y, [EmbReal](#) data)
- void [previewOff](#) ()
- bool [allowRubber](#) ()
- void [addToRubberRoom](#) (QGraphicsItem *item)
- void [vulcanizeRubberRoom](#) ()
- void [clearRubberRoom](#) ()
- void [spareRubber](#) (int64_t id)
- void [setRubberMode](#) (String mode)
- void [setRubberPoint](#) (QString key, const QPointF &point)
- void [setRubberText](#) (QString key, QString txt)

Public Member Functions

- [View](#) (QGraphicsScene *theScene, QWidget *parent)
- [~View](#) ()
- std::vector< QGraphicsItem * > [selected_items](#) ()
- bool [allowZoomIn](#) ()
- bool [allowZoomOut](#) ()
- void [updateMouseCoords](#) (int x, int y)
- void [recalculateLimits](#) ()
- void [zoomToPoint](#) (const QPoint &mousePoint, int zoomDir)
- void [centerAt](#) (const QPointF ¢erPoint)
- QPointF [center](#) ()
- QUndoStack * [getUndoStack](#) ()
- void [addObject](#) (Geometry *obj)
- void [deleteObject](#) (Geometry *obj)
- void [vulcanizeObject](#) (Geometry *obj)

Public Attributes

- Dictionary state
- QColor gridColor
- QPainterPath gridPath
- QPainterPath originPath
- bool rulerMetric
- QColor rulerColor
- uint8_t rulerPixelSize
- bool grippingActive
- bool rapidMoveActive
- bool previewActive
- bool pastingActive
- bool movingActive
- bool selectingActive
- bool zoomWindowActive
- bool panningRealTimeActive
- bool panningPointActive
- bool panningActive
- bool qSnapActive
- bool qSnapToggle
- Geometry * gripBaseObj
- Geometry * tempBaseObj
- QGraphicsScene * gscene
- QUndoStack * undoStack
- SelectBox * selectBox
- QPointF scenePressPoint
- QPoint pressPoint
- QPointF sceneMovePoint
- QPoint movePoint
- QPointF sceneReleasePoint
- QPoint releasePoint
- QPointF sceneGripPoint
- QPoint viewMousePoint
- QPointF sceneMousePoint
- QRgb qsnapLocatorColor
- uint8_t qsnapLocatorSize
- uint8_t qsnapApertureSize
- QRgb gripColorCool
- QRgb gripColorHot
- uint8_t gripSize
- uint8_t pickBoxSize
- QRgb crosshairColor
- uint32_t crosshairSize

Protected Member Functions

- void mouseDoubleClickEvent (QMouseEvent *event)
- void mousePressEvent (QMouseEvent *event)
- void mouseMoveEvent (QMouseEvent *event)
- void mouseReleaseEvent (QMouseEvent *event)
- void wheelEvent (QWheelEvent *event)
- void contextMenuEvent (QContextMenuEvent *event)
- void drawBackground (QPainter *painter, const QRectF &rect)
- void drawForeground (QPainter *painter, const QRectF &rect)
- void enterEvent (QEvent *event)

Private Member Functions

- void [createGridRect](#) ()
- void [createGridPolar](#) ()
- void [createGridIso](#) ()
- void [createOrigin](#) ()
- void [loadRulerSettings](#) ()
- bool [willUnderflowInt32](#) (int64_t a, int64_t b)
- bool [willOverflowInt32](#) (int64_t a, int64_t b)
- int [roundToMultiple](#) (bool roundUp, int numToRound, int multiple)
- QPainterPath [createRulerTextPath](#) ([EmbVector](#) position, QString str, [EmbReal](#) height)
- std::vector< QGraphicsItem * > [createObjectList](#) (std::vector< QGraphicsItem * > list)
- void [copySelected](#) ()
- void [startGripping](#) ([Geometry](#) *obj)
- void [stopGripping](#) (bool accept=false)
- void [panStart](#) (const QPoint &point)
- void [alignScenePointWithViewPoint](#) (const QPointF &scenePoint, const QPoint &viewPoint)

Private Attributes

- QHash< int64_t, QGraphicsItem * > [hashDeletedObjects](#)
- [StringList](#) [spareRubberList](#)
- QList< QGraphicsItem * > [previewObjectList](#)
- QGraphicsItemGroup * [previewObjectItemGroup](#)
- QPointF [previewPoint](#)
- [EmbReal](#) [previewData](#)
- [String](#) [previewMode](#)
- QPointF [cutCopyMousePoint](#)
- QGraphicsItemGroup * [pasteObjectItemGroup](#)
- QPointF [pasteDelta](#)
- std::vector< QGraphicsItem * > [rubberRoomList](#)
- int [panDistance](#)
- int [panStartX](#)
- int [panStartY](#)

17.74.1 Constructor & Destructor Documentation

17.74.1.1 View() [View](#) (
 QGraphicsScene * *theScene*,
 QWidget * *parent*)

17.74.1.2 ~View() [~View](#) ()

17.74.2 Member Function Documentation

17.74.2.1 addObject() void [addObject](#) (
 [Geometry](#) * *obj*)

17.74.2.2 addToRubberRoom void [addToRubberRoom](#) (
 QGraphicsItem * *item*) [slot]

17.74.2.3 alignScenePointWithViewPoint() void alignScenePointWithViewPoint (
 const QPointF & *scenePoint*,
 const QPoint & *viewPoint*) [private]

17.74.2.4 allowRubber bool allowRubber () [slot]

17.74.2.5 allowZoomIn() bool allowZoomIn ()

17.74.2.6 allowZoomOut() bool allowZoomOut ()

17.74.2.7 center() QPointF center () [inline]

17.74.2.8 centerAt() void centerAt (
 const QPointF & *centerPoint*)

17.74.2.9 clearRubberRoom void clearRubberRoom () [slot]

17.74.2.10 clearSelection void clearSelection () [slot]

17.74.2.11 contextMenuEvent() void contextMenuEvent (
 QContextMenuEvent * *event*) [protected]

17.74.2.12 copy void copy () [slot]

17.74.2.13 copySelected() void copySelected () [private]

17.74.2.14 cornerButtonClicked void cornerButtonClicked () [slot]

17.74.2.15 createGrid void createGrid (
 QString *gridType*) [slot]

17.74.2.16 createGridIso() void createGridIso () [private]

17.74.2.17 createGridPolar() void createGridPolar () [private]

17.74.2.18 createGridRect() void createGridRect () [private]

17.74.2.19 createObjectList() `std::vector< QGraphicsItem * > createObjectList (`
`std::vector< QGraphicsItem * > list) [private]`

17.74.2.20 createOrigin() `void createOrigin () [private]`

17.74.2.21 createRulerTextPath() `QPainterPath createRulerTextPath (`
`EmbVector position,`
`QString str,`
`EmbReal height) [private]`

17.74.2.22 cut `void cut () [slot]`

17.74.2.23 deleteObject() `void deleteObject (`
`Geometry * obj)`

17.74.2.24 deletePressed `void deletePressed () [slot]`

17.74.2.25 deleteSelected `void deleteSelected () [slot]`

17.74.2.26 drawBackground() `void drawBackground (`
`QPainter * painter,`
`const QRectF & rect) [protected]`

17.74.2.27 drawForeground() `void drawForeground (`
`QPainter * painter,`
`const QRectF & rect) [protected]`

17.74.2.28 enterEvent() `void enterEvent (`
`QEvent * event) [protected]`

17.74.2.29 escapePressed `void escapePressed () [slot]`

17.74.2.30 getUndoStack() `QUndoStack * getUndoStack () [inline]`

17.74.2.31 isLwtEnabled `bool isLwtEnabled () [slot]`

17.74.2.32 isRealEnabled `bool isRealEnabled () [slot]`

17.74.2.33 loadRulerSettings() `void loadRulerSettings () [private]`

17.74.2.34 mirrorSelected void mirrorSelected (
 EmbReal x1,
 EmbReal y1,
 EmbReal x2,
 EmbReal y2) [slot]

17.74.2.35 mouseDoubleClickEvent() void mouseDoubleClickEvent (
 QMouseEvent * event) [protected]

17.74.2.36 mouseMoveEvent() void mouseMoveEvent (
 QMouseEvent * event) [protected]

Todo turn move into an actuator call.

17.74.2.37 mousePressEvent() void mousePressEvent (
 QMouseEvent * event) [protected]

17.74.2.38 mouseReleaseEvent() void mouseReleaseEvent (
 QMouseEvent * event) [protected]

17.74.2.39 moveAction void moveAction () [slot]

17.74.2.40 moveSelected void moveSelected (
 EmbReal dx,
 EmbReal dy) [slot]

17.74.2.41 numSelected int numSelected () [slot]

17.74.2.42 panDown void panDown () [slot]

17.74.2.43 panLeft void panLeft () [slot]

17.74.2.44 panPoint void panPoint () [slot]

17.74.2.45 panRealTime void panRealTime () [slot]

17.74.2.46 panRight void panRight () [slot]

17.74.2.47 panStart() void panStart (
 const QPoint & point) [private]

17.74.2.48 panUp void panUp () [slot]

17.74.2.49 paste void paste () [slot]

17.74.2.50 previewOff void previewOff () [slot]

17.74.2.51 previewOn void previewOn (
 String clone,
 String mode,
 EmbReal x,
 EmbReal y,
 EmbReal data) [slot]

17.74.2.52 recalculateLimits() void recalculateLimits ()

17.74.2.53 repeatAction void repeatAction () [slot]

17.74.2.54 rotateAction void rotateAction () [slot]

17.74.2.55 rotateSelected void rotateSelected (
 EmbReal x,
 EmbReal y,
 EmbReal rot) [slot]

17.74.2.56 roundToMultiple() int roundToMultiple (
 bool roundUp,
 int numToRound,
 int multiple) [private]

Round the number *numToRound* to a multiple of the number *multiple*, rounding up if *roundUp* is true.

First, *multiple* is 0 then we have an invalid input so just return the argument, then if the number is already a multiple of *multiple* then return the argument.

Then take the remainder off the argument and determine which way to round the result.

17.74.2.57 scaleAction void scaleAction () [slot]

17.74.2.58 scaleSelected void scaleSelected (
 EmbReal x,
 EmbReal y,
 EmbReal factor) [slot]

17.74.2.59 selectAll void selectAll () [slot]

17.74.2.60 selected_items() std::vector< QGraphicsItem * > selected_items ()

17.74.2.61 selectionChanged void selectionChanged () [slot]

17.74.2.62 setBackgroundColor void setBackgroundColor (
QRgb *color*) [slot]

17.74.2.63 setCornerButton void setCornerButton () [slot]

17.74.2.64 setCrossHairColor void setCrossHairColor (
QRgb *color*) [slot]

17.74.2.65 setCrossHairSize void setCrossHairSize (
uint8_t *percent*) [slot]

17.74.2.66 setGridColor void setGridColor (
QRgb *color*) [slot]

17.74.2.67 setRubberMode void setRubberMode (
String *mode*) [slot]

17.74.2.68 setRubberPoint void setRubberPoint (
QString *key*,
const QPointF & *point*) [slot]

17.74.2.69 setRubberText void setRubberText (
QString *key*,
QString *txt*) [slot]

17.74.2.70 setRulerColor void setRulerColor (
QRgb *color*) [slot]

17.74.2.71 setSelectBoxColors void setSelectBoxColors (
QRgb *colorL*,
QRgb *fillL*,
QRgb *colorR*,
QRgb *fillR*,
int *alpha*) [slot]

17.74.2.72 showScrollBars void showScrollBars (
bool *val*) [slot]

17.74.2.73 spareRubber void spareRubber (
int64_t *id*) [slot]

- 17.74.2.74 startGripping()** void startGripping (
 `Geometry * obj`) [private]
- 17.74.2.75 stopGripping()** void stopGripping (
 `bool accept = false`) [private]
- 17.74.2.76 toggleGrid** void toggleGrid (
 `bool on`) [slot]
- 17.74.2.77 toggleLwt** void toggleLwt (
 `bool on`) [slot]
- 17.74.2.78 toggleOrtho** void toggleOrtho (
 `bool on`) [slot]
- 17.74.2.79 togglePolar** void togglePolar (
 `bool on`) [slot]
- 17.74.2.80 toggleQSnap** void toggleQSnap (
 `bool on`) [slot]
- 17.74.2.81 toggleQTrack** void toggleQTrack (
 `bool on`) [slot]
- 17.74.2.82 toggleReal** void toggleReal (
 `bool on`) [slot]
- 17.74.2.83 toggleRuler** void toggleRuler (
 `bool on`) [slot]
- 17.74.2.84 toggleSnap** void toggleSnap (
 `bool on`) [slot]
- 17.74.2.85 updateMouseCoords()** void updateMouseCoords (
 `int x,`
 `int y`)
- 17.74.2.86 vulcanizeObject()** void vulcanizeObject (
 `Geometry * obj`)
- 17.74.2.87 vulcanizeRubberRoom** void vulcanizeRubberRoom () [slot]

17.74.2.88 wheelEvent() `void wheelEvent (`
 `QWheelEvent * event) [protected]`

17.74.2.89 willOverflowInt32() `bool willOverflowInt32 (`
 `int64_t a,`
 `int64_t b) [private]`

17.74.2.90 willUnderflowInt32() `bool willUnderflowInt32 (`
 `int64_t a,`
 `int64_t b) [private]`

17.74.2.91 zoomExtents `void zoomExtents () [slot]`

17.74.2.92 zoomIn `void zoomIn () [slot]`

17.74.2.93 zoomOut `void zoomOut () [slot]`

17.74.2.94 zoomSelected `void zoomSelected () [slot]`

17.74.2.95 zoomToPoint() `void zoomToPoint (`
 `const QPoint & mousePoint,`
 `int zoomDir)`

17.74.2.96 zoomWindow `void zoomWindow () [slot]`

17.74.3 Member Data Documentation

17.74.3.1 crosshairColor `QRgb crosshairColor`

17.74.3.2 crosshairSize `uint32_t crosshairSize`

17.74.3.3 cutCopyMousePoint `QPointF cutCopyMousePoint [private]`

17.74.3.4 gridColor `QColor gridColor`

17.74.3.5 gridPath `QPainterPath gridPath`

17.74.3.6 gripBaseObj `Geometry* gripBaseObj`

17.74.3.7 gripColorCool `QRgb gripColorCool`

17.74.3.8 gripColorHot `QRgb gripColorHot`

17.74.3.9 grippingActive `bool grippingActive`

17.74.3.10 gripSize `uint8_t gripSize`

17.74.3.11 gscene `QGraphicsScene* gscene`

17.74.3.12 hashDeletedObjects `QHash<int64_t, QGraphicsItem*> hashDeletedObjects [private]`

17.74.3.13 movePoint `QPoint movePoint`

17.74.3.14 movingActive `bool movingActive`

17.74.3.15 originPath `QPainterPath originPath`

17.74.3.16 panDistance `int panDistance [private]`

17.74.3.17 panningActive `bool panningActive`

17.74.3.18 panningPointActive `bool panningPointActive`

17.74.3.19 panningRealTimeActive `bool panningRealTimeActive`

17.74.3.20 panStartX `int panStartX [private]`

17.74.3.21 panStartY `int panStartY [private]`

17.74.3.22 pasteDelta `QPointF pasteDelta [private]`

17.74.3.23 pasteObjectItemGroup `QGraphicsItemGroup* pasteObjectItemGroup [private]`

17.74.3.24 pastingActive `bool pastingActive`

17.74.3.25 pickBoxSize `uint8_t pickBoxSize`

17.74.3.26 pressPoint `QPoint pressPoint`

17.74.3.27 previewActive `bool previewActive`

17.74.3.28 previewData `EmbReal previewData [private]`

17.74.3.29 previewMode `String previewMode [private]`

17.74.3.30 previewObjectItemGroup `QGraphicsItemGroup* previewObjectItemGroup [private]`

17.74.3.31 previewObjectList `QList<QGraphicsItem*> previewObjectList [private]`

17.74.3.32 previewPoint `QPointF previewPoint [private]`

17.74.3.33 qSnapActive `bool qSnapActive`

17.74.3.34 qsnapApertureSize `uint8_t qsnapApertureSize`

17.74.3.35 qsnapLocatorColor `QRgb qsnapLocatorColor`

17.74.3.36 qsnapLocatorSize `uint8_t qsnapLocatorSize`

17.74.3.37 qSnapToggle `bool qSnapToggle`

17.74.3.38 rapidMoveActive `bool rapidMoveActive`

17.74.3.39 releasePoint `QPoint releasePoint`

17.74.3.40 rubberRoomList `std::vector<QGraphicsItem*> rubberRoomList [private]`

17.74.3.41 rulerColor `QColor rulerColor`

17.74.3.42 rulerMetric `bool rulerMetric`

17.74.3.43 rulerPixelSize `uint8_t rulerPixelSize`

17.74.3.44 sceneGripPoint `QPointF sceneGripPoint`

17.74.3.45 sceneMousePoint `QPointF sceneMousePoint`

17.74.3.46 sceneMovePoint `QPointF sceneMovePoint`

17.74.3.47 scenePressPoint `QPointF scenePressPoint`

17.74.3.48 sceneReleasePoint `QPointF sceneReleasePoint`

17.74.3.49 selectBox `SelectBox* selectBox`

17.74.3.50 selectingActive `bool selectingActive`

17.74.3.51 spareRubberList `StringList spareRubberList [private]`

17.74.3.52 state `Dictionary state`

17.74.3.53 tempBaseObj `Geometry* tempBaseObj`

17.74.3.54 undoStack `QUndoStack* undoStack`

17.74.3.55 viewMousePoint `QPoint viewMousePoint`

17.74.3.56 zoomWindowActive `bool zoomWindowActive`

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

- embroidermodder2/[embroidermodder.h](#)
- embroidermodder2/[view.cpp](#)

17.75 VipHeader_ Struct Reference

```
#include <embroidery_internal.h>
```

Public Attributes

- int [magicCode](#)
- int [numberOfStitches](#)
- int [numberOfColors](#)
- short [positiveXHoopSize](#)
- short [positiveYHoopSize](#)
- short [negativeXHoopSize](#)
- short [negativeYHoopSize](#)
- int [attributeOffset](#)
- int [xOffset](#)
- int [yOffset](#)
- unsigned char [stringVal](#) [8]
- short [unknown](#)
- int [colorLength](#)

17.75.1 Member Data Documentation

17.75.1.1 attributeOffset `int attributeOffset`

17.75.1.2 colorLength `int colorLength`

17.75.1.3 magicCode `int magicCode`

17.75.1.4 negativeXHoopSize `short negativeXHoopSize`

17.75.1.5 negativeYHoopSize `short negativeYHoopSize`

17.75.1.6 numberOfColors `int numberOfColors`

17.75.1.7 numberOfStitches `int numberOfStitches`

17.75.1.8 positiveXHoopSize `short positiveXHoopSize`

17.75.1.9 positiveYHoopSize `short positiveYHoopSize`

17.75.1.10 stringVal `unsigned char stringVal[8]`

17.75.1.11 unknown `short unknown`

17.75.1.12 xOffset `int xOffset`

17.75.1.13 yOffset `int yOffset`

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

- extern/libembroidery/src/[embroidery_internal.h](#)

18 File Documentation

18.1 CODE_OF_CONDUCT.md File Reference

18.2 embroidermodder2/cmdprompt.cpp File Reference

```
#include "embroidermodder.h"
```

18.2.1 Detailed Description

Embroidermodder 2.

Copyright 2013-2022 The Embroidermodder Team Embroidermodder 2 is Open Source Software. See LICENSE for licensing terms.

Use Python's PEP7 style guide. <https://peps.python.org/pep-0007/>

18.3 embroidermodder2/em2_dev_script.py File Reference

Namespaces

- namespace [em2_dev_script](#)

Variables

- string [header](#)
- dictionary `d = {}`
- `s = f.read()`

18.4 embroidermodder2/embdetails-dialog.cpp File Reference

```
#include "embroidermodder.h"
```

18.5 embroidermodder2/embroidermodder.cpp File Reference

```
#include "embroidermodder.h"
```

Functions

- static void [usage](#) (void)
usage
- static void [version](#) ()
version
- int [main](#) (int argc, char *argv[])
qMain

Variables

- static const char * [_appVer_](#) = "v2.0.0-alpha3"
- static bool [exitApp](#) = false

18.5.1 Function Documentation

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

qMain

Parameters

<i>argc</i>	
<i>argv</i>	

Returns

18.5.1.2 usage() `static void usage (`
 `void) [static]`

usage

18.5.1.3 version() `static void version () [static]`
version

18.5.2 Variable Documentation

18.5.2.1 _appVer_ `const char* _appVer_ = "v2.0.0-alpha3" [static]`

18.5.2.2 exitApp `bool exitApp = false [static]`

18.6 embroidermodder2/embroidermodder.h File Reference

```
#include <cstdio>
#include <cmath>
#include <ctime>
#include <cinttypes>
#include <cstdlib>
#include <vector>
#include <unordered_map>
#include <string>
#include <filesystem>
#include "embroidery.h"
#include "toml.h"
#include <QAction>
#include <QApplication>
#include <QtPrintSupport>
```

Classes

- struct [Node_](#)
- class [Geometry](#)
 The [Geometry](#) class.
- class [SaveObject](#)

- class [Application](#)
- class [CmdPromptInput](#)
- class [CmdPromptHistory](#)

The Command Prompt History class.

- class [CmdPromptSplitter](#)
- class [CmdPromptHandle](#)
- class [CmdPrompt](#)
- class [EmbDetailsDialog](#)
- class [ImageWidget](#)
- class [LayerManager](#)
- class [MainWindow](#)

The MainWindow class.

- class [MdiWindow](#)
- class [MdiArea](#)
- class [PreviewDialog](#)
- class [PropertyEditor](#)
- class [SelectBox](#)
- class [Settings_Dialog](#)
- class [StatusBar](#)
- class [UndoEditor](#)
- class [UndoableCommand](#)
- class [View](#)

Macros

- `#define` [STRING_TYPE](#) 0
- `#define` [STRING_LIST_TYPE](#) 1
- `#define` [REAL_TYPE](#) 2
- `#define` [INT_TYPE](#) 3
- `#define` [BOOL_TYPE](#) 4
- `#define` [FUNCTION_TYPE](#) 5
- `#define` [VECTOR_TYPE](#) 6
- `#define` [UNKNOWN_TYPE](#) 7

Typedefs

- `typedef` `std::string` [String](#)
- `typedef` `std::vector< String >` [StringList](#)
- `typedef` `struct` [Node](#) [Node](#)
- `typedef` [String](#)(* [Command](#)) ([String](#))
- `typedef` `std::vector< Node >` [NodeList](#)
- `typedef` `std::unordered_map< String, Node >` [Dictionary](#)

Enumerations

- `enum` [OBJ_TYPE_VALUES](#) {
[OBJ_TYPE_NULL](#) = 0 , [OBJ_TYPE_BASE](#) = 100000 , [OBJ_TYPE_ARC](#) = 100001 , [OBJ_TYPE_BLOCK](#) = 100002 ,
[OBJ_TYPE_CIRCLE](#) = 100003 , [OBJ_TYPE_DIMALIGNED](#) = 100004 , [OBJ_TYPE_DIMANGULAR](#) = 100005 , [OBJ_TYPE_DIMARCLength](#) = 100006 ,
[OBJ_TYPE_DIMDIAMETER](#) = 100007 , [OBJ_TYPE_DIMLEADER](#) = 100008 , [OBJ_TYPE_DIMLINEAR](#) = 100009 , [OBJ_TYPE_DIMORDINATE](#) = 100010 ,
[OBJ_TYPE_DIMRADIUS](#) = 100011 , [OBJ_TYPE_ELLIPSE](#) = 100012 , [OBJ_TYPE_ELLIPSEARC](#) = 100013 ,
[OBJ_TYPE_RUBBER](#) = 100014 ,
[OBJ_TYPE_GRID](#) = 100015 , [OBJ_TYPE_HATCH](#) = 100016 , [OBJ_TYPE_IMAGE](#) = 100017 ,
[OBJ_TYPE_INFINTLINE](#) = 100018 ,

```

OBJ_TYPE_LINE = 100019 , OBJ_TYPE_PATH = 100020 , OBJ_TYPE_POINT = 100021 , OBJ_TYPE_POLYGON
= 100022 ,
OBJ_TYPE_POLYLINE = 100023 , OBJ_TYPE_RAY = 100024 , OBJ_TYPE_RECTANGLE = 100025 ,
OBJ_TYPE_SLOT = 100026 ,
OBJ_TYPE_SPLINE = 100027 , OBJ_TYPE_TEXTMULTI = 100028 , OBJ_TYPE_TEXTSINGLE = 100029 ,
OBJ_TYPE_UNKNOWN = 100030 }
• enum OBJ_KEYS {
  OBJ_TYPE = 0 , OBJ_NAME = 1 , OBJ_LAYER = 2 , OBJ_COLOR = 3 ,
  OBJ_LTYPE = 4 , OBJ_LWT = 5 , OBJ_RUBBER = 6 }

```

Functions

- int [read_configuration](#) (const char *file)
- void [read_settings](#) (void)
 - read_settings*
- void [write_settings](#) (void)
 - MainWindow::writeSettings.*
- [EmbVector rotate_vector](#) ([EmbVector](#) v, [EmbReal](#) alpha)
- [QString translate_str](#) (const char *str)
- bool [contains](#) ([StringList](#), [String](#))
- bool [validFileFormat](#) ([String](#) fileName)
 - MainWindow::validFileFormat.*
- [QString fileExtension](#) ([String](#) fileName)
 - MdiWindow::fileExtension.*
- void [add_polyline](#) ([QPainterPath](#) p, [String](#) rubberMode)
- [String read_string_setting](#) ([toml_table_t](#) *table, const char *key)
- [StringList tokenize](#) ([String](#) str, const char delim)
 - tokenize*
- [String convert_args_to_type](#) ([String](#) label, [StringList](#) args, const char *args_template, [NodeList](#) a)
- [View * activeView](#) (void)
 - activeView*
- [QGraphicsScene * activeScene](#) ()
 - MainWindow::activeScene.*
- void [debug_message](#) ([String](#) msg)
 - debug_message*
- void [set_enabled](#) ([QObject](#) *parent, const char *key, bool enabled)
 - set_enabled*
- void [set_visibility](#) ([QObject](#) *parent, const char *name, bool visibility)
 - set_visibility*
- [String actuator](#) ([String](#) line)
 - MainWindow::actuator.*
- [String run_script_file](#) ([String](#) fname)
 - MainWindow::run_script_file.*
- [String run_script](#) ([StringList](#) script)
 - A basic line-by-line script processor to allow for extensions to the program.*
- [String construct_command](#) ([String](#) command, const char *fmt,...)
 - construct_command*
- void [create_menu](#) ([String](#) menu, [StringList](#) def, bool topLevel)
 - create_menu*
- [QPointF to_QPointF](#) ([EmbVector](#) a)
- [EmbVector to_EmbVector](#) ([QPointF](#) a)
- [EmbVector operator+](#) ([EmbVector](#) a, [EmbVector](#) b)
 - operator + Wrapper for embVector_add to use the syntax a + b.*

- [EmbVector operator-](#) ([EmbVector](#) a, [EmbVector](#) b)
operator - Wrapper for embVector_subtract to use the syntax a - b.
- [EmbVector operator*](#) ([EmbVector](#) v, [EmbReal](#) s)
*operator **
- [EmbReal radians__](#) ([EmbReal](#) degrees)
radians__
- [EmbReal degrees__](#) ([EmbReal](#) radian)
degrees__
- `std::vector< QGraphicsItem * >` [to_vector](#) (`QList< QGraphicsItem * >` list)
to_vector
- `QList< QGraphicsItem * >` [to_qlist](#) (`std::vector< QGraphicsItem * >` list)
to_qlist
- [StringList to_string_vector](#) (`QStringList` list)
to_string_vector
- void [make_ui_element](#) ([String](#) description)
- `QDoubleSpinBox *` [make_spinbox](#) (`QGroupBox *` gb, [String](#) d, `QString` object_name, [EmbReal](#) single_step, [EmbReal](#) lower, [EmbReal](#) upper, [String](#) key)
- `QCheckBox *` [make_checkbox](#) (`QGroupBox *` gb, [String](#) d, const char *label, const char *icon, [String](#) key)
- [Node node_bool](#) (`bool` value)
set_node
- [Node node_int](#) (`int32_t` value)
create_node
- [Node node_uint](#) (`uint32_t` value)
create_node
- [Node node_real](#) ([EmbReal](#) value)
set_node
- [Node node_str](#) ([String](#) value)
set_node
- [Node node_qstr](#) (`QString` value)
set_node
- [Node node_str_list](#) (`StringList` value)
set_node
- `bool` [get_bool](#) (`Dictionary` d, [String](#) key)
- `int32_t` [get_int](#) (`Dictionary` d, [String](#) key)
- `uint32_t` [get_uint](#) (`Dictionary` d, [String](#) key)
- [EmbReal](#) [get_real](#) (`Dictionary` d, [String](#) key)
- [String](#) [get_str](#) (`Dictionary` d, [String](#) key)
- `QString` [get_qstr](#) (`Dictionary` d, [String](#) key)
- [StringList](#) [get_str_list](#) (`Dictionary` d, [String](#) key)

Variables

- static const [EmbReal](#) [emb_constant_pi](#) = 3.14159265358979323846
- [MdiArea](#) * [mdiArea](#)
- [Dictionary](#) [settings](#)
Settings System.
- [Dictionary](#) [dialog](#)
- [Dictionary](#) [config](#)
- `std::unordered_map< String, QStringList >` [scripts](#)
- `std::unordered_map< String, QGroupBox * >` [groupBoxes](#)
- `std::unordered_map< String, QCheckBox * >` [checkboxes](#)
- `std::unordered_map< String, QSpinBox * >` [spinBoxes](#)
- `std::unordered_map< String, QDoubleSpinBox * >` [doubleSpinBoxes](#)

- `std::unordered_map< String, QLabel * >` [labels](#)
- `std::unordered_map< String, QComboBox * >` [comboBoxes](#)
- `std::unordered_map< String, QLineEdit * >` [lineEdits](#)
- `std::unordered_map< String, QPushButton * >` [toolButtons](#)
- `std::unordered_map< String, Dictionary >` [config_tables](#)
- `std::unordered_map< String, QAction * >` [actionHash](#)
- `std::unordered_map< String, QToolBar * >` [toolbarHash](#)
- `std::unordered_map< String, QMenu * >` [menuHash](#)
- `std::unordered_map< String, QMenu * >` [subMenuHash](#)
- `MainWindow * _mainWin`
- `CmdPrompt * prompt`
- `PropertyEditor * dockPropEdit`
- `UndoEditor * dockUndoEdit`
- `StatusBar * statusbar`

18.6.1 Detailed Description

The only header for the GUI part: a good overview of this source code.

18.6.2 Macro Definition Documentation

18.6.2.1 `BOOL_TYPE` `#define BOOL_TYPE 4`

18.6.2.2 `FUNCTION_TYPE` `#define FUNCTION_TYPE 5`

18.6.2.3 `INT_TYPE` `#define INT_TYPE 3`

18.6.2.4 `REAL_TYPE` `#define REAL_TYPE 2`

18.6.2.5 `STRING_LIST_TYPE` `#define STRING_LIST_TYPE 1`

18.6.2.6 `STRING_TYPE` `#define STRING_TYPE 0`

18.6.2.7 `UNKNOWN_TYPE` `#define UNKNOWN_TYPE 7`

18.6.2.8 `VECTOR_TYPE` `#define VECTOR_TYPE 6`

18.6.3 Typedef Documentation

18.6.3.1 `Command` `typedef String(* Command) (String)`

18.6.3.2 `Dictionary` `typedef std::unordered_map<String, Node> Dictionary`

18.6.3.3 Node typedef struct [Node_](#) Node

18.6.3.4 NodeList typedef std::vector<[Node](#)> [NodeList](#)

18.6.3.5 String typedef std::string [String](#)

18.6.3.6 StringList typedef std::vector<[String](#)> [StringList](#)

18.6.4 Enumeration Type Documentation

18.6.4.1 OBJ_KEYS enum [OBJ_KEYS](#)

Custom Data used in QGraphicsItems

```
(      int, const QVariant)
```

I.E. object.setData(OBJ_TYPE, OBJ_TYPE_LINE); I.E. object.setData(OBJ_LAYER, "OUTLINE"); I.E. object.set↵Data(OBJ_COLOR, 123); I.E. object.setData(OBJ_LTYPE, OBJ_LTYPE_CONT);
Keys

Enumerator

OBJ_TYPE	
OBJ_NAME	
OBJ_LAYER	
OBJ_COLOR	
OBJ_LTYPE	value type - int: 0-255 Todo Use color chart in formats/format-dxf.h for this
OBJ_LWT	
OBJ_RUBBER	

18.6.4.2 OBJ_TYPE_VALUES enum [OBJ_TYPE_VALUES](#)

Enumerator

OBJ_TYPE_NULL	
OBJ_TYPE_BASE	
OBJ_TYPE_ARC	
OBJ_TYPE_BLOCK	
OBJ_TYPE_CIRCLE	
OBJ_TYPE_DIMALIGNED	
OBJ_TYPE_DIMANGULAR	
OBJ_TYPE_DIMARCLENGTH	
OBJ_TYPE_DIMDIAMETER	
OBJ_TYPE_DIMLEADER	
OBJ_TYPE_DIMLINEAR	
OBJ_TYPE_DIMORDINATE	
OBJ_TYPE_DIMRADIUS	

Enumerator

OBJ_TYPE_ELLIPSE	
OBJ_TYPE_ELLIPSEARC	
OBJ_TYPE_RUBBER	
OBJ_TYPE_GRID	
OBJ_TYPE_HATCH	
OBJ_TYPE_IMAGE	
OBJ_TYPE_INFINITELINE	
OBJ_TYPE_LINE	
OBJ_TYPE_PATH	
OBJ_TYPE_POINT	
OBJ_TYPE_POLYGON	
OBJ_TYPE_POLYLINE	
OBJ_TYPE_RAY	
OBJ_TYPE_RECTANGLE	
OBJ_TYPE_SLOT	
OBJ_TYPE_SPLINE	
OBJ_TYPE_TEXTMULTI	
OBJ_TYPE_TEXTSINGLE	
OBJ_TYPE_UNKNOWN	

18.6.5 Function Documentation

18.6.5.1 activeScene() `QGraphicsScene * activeScene ()`
MainWindow::activeScene.

Returns

18.6.5.2 activeView() `View * activeView (`
 `void)`
activeView

Returns

18.6.5.3 actuator() `String actuator (`
 `String line)`
MainWindow::actuator.

Parameters

<i>command</i>	
----------------	--

18.6.5.4 RUN COMMAND `QAction* act = qobject_cast<QAction*>(sender()); if (act) { prompt->end←
Command(); prompt->setCurrentText(act->objectName()); prompt->processInput(); }`

18.6.5.5 INIT QString fileName = "commands/" + cmd + "/" + cmd + ".js"; if (!getSettingsSelectionMode() < PickFirst()) { actuator("clear-selection"); } TODO: Uncomment this line when post-selection is available engine->evaluate(cmd + "_main(void)", fileName);

18.6.5.6 PROMPT QString fileName = "commands/" + cmd + "/" + cmd + ".js"; NOTE: Replace any special characters that will cause a syntax error QString safeStr = str; safeStr.replace("\\", "\\"); safeStr.replace("\'", "\'"); if (prompt->promptInput->rapidFireEnabled) { engine->evaluate(cmd + "_prompt(\"" + safeStr + "\")", fileName); } else { engine->evaluate(cmd + "_prompt(\"" + safeStr.toUpper() + "\")", fileName); }

18.6.5.7 add_polyline() void add_polyline (QPainterPath p, String rubberMode)

18.6.5.8 construct_command() String construct_command (String command, const char * fmt, ...)

construct_command

Parameters

<i>command</i>	
<i>fmt</i>	

Returns

18.6.5.9 contains() bool contains (QStringList list, String entry)

18.6.5.10 convert_args_to_type() String convert_args_to_type (String label, QStringList args, const char * args_template, NodeList a)

18.6.5.11 create_menu() void create_menu (std::string menu, QStringList def, bool topLevel)

create_menu

Parameters

<i>menu</i>	
<i>def</i>	
<i>topLevel</i>	

18.6.5.12 debug_message() void debug_message (
std::string msg)
debug_message

Parameters

msg	
-----	--

18.6.5.13 degrees__() EmbReal degrees__ (
EmbReal radian)
degrees__

Parameters

radian	
--------	--

Returns

18.6.5.14 fileExtension() QString fileExtension (
String fileName)
MdiWindow::fileExtension.

Parameters

fileName	
----------	--

Returns

18.6.5.15 get_bool() bool get_bool (
Dictionary d,
String key)

18.6.5.16 get_int() int32_t get_int (
Dictionary d,
String key)

18.6.5.17 get_qstr() QString get_qstr (
Dictionary d,
String key)

18.6.5.18 get_real() EmbReal get_real (
Dictionary d,
String key)

18.6.5.19 `get_str()` `String` `get_str` (
 `Dictionary` `d`,
 `String` `key`)

18.6.5.20 `get_str_list()` `StringList` `get_str_list` (
 `Dictionary` `d`,
 `String` `key`)

18.6.5.21 `get_uint()` `uint32_t` `get_uint` (
 `Dictionary` `d`,
 `String` `key`)

18.6.5.22 `make_checkbox()` `QCheckBox *` `make_checkbox` (
 `QGroupBox *` `gb`,
 `String` `dictionary`,
 `const char *` `label`,
 `const char *` `icon`,
 `String` `key`)

18.6.5.23 `make_spinbox()` `QDoubleSpinBox *` `make_spinbox` (
 `QGroupBox *` `gb`,
 `String` `dictionary`,
 `QString` `object_name`,
 `EmbReal` `single_step`,
 `EmbReal` `lower`,
 `EmbReal` `upper`,
 `String` `key`)

18.6.5.24 `make_ui_element()` `void` `make_ui_element` (
 `String` `description`)

18.6.5.25 `node_bool()` `Node` `node_bool` (
 `bool` `value`)

`set_node`

Parameters

<i>node</i>	
<i>value</i>	

18.6.5.26 `node_int()` `Node` `node_int` (
 `int32_t` `value`)

`create_node`

Parameters

<i>mode</i>	
-------------	--

Returns

18.6.5.27 node_qstr() `Node node_qstr (`
 `QString value)`

set_node

Parameters

<i>node</i>	
<i>value</i>	

18.6.5.28 node_real() `Node node_real (`
 `EmbReal value)`

set_node

Parameters

<i>node</i>	
<i>value</i>	

18.6.5.29 node_str() `Node node_str (`
 `String value)`

set_node

Parameters

<i>node</i>	
<i>value</i>	

18.6.5.30 node_str_list() `Node node_str_list (`
 `StringList value)`

set_node

Parameters

<i>node</i>	
<i>value</i>	

18.6.5.31 node_uint() `Node node_uint (`
 `uint32_t value)`

create_node

Parameters

<i>mode</i>	
-------------	--

Returns

18.6.5.32 operator*() `EmbVector operator* (`
`EmbVector v,`
`EmbReal s)`

operator *

Parameters

<i>v</i>	
<i>s</i>	

Returns

18.6.5.33 operator+() `EmbVector operator+ (`
`EmbVector a,`
`EmbVector b)`

operator + Wrapper for `embVector_add` to use the syntax $a + b$.

18.6.5.34 operator-() `EmbVector operator- (`
`EmbVector a,`
`EmbVector b)`

operator - Wrapper for `embVector_subtract` to use the syntax $a - b$.

18.6.5.35 radians__() `EmbReal radians__ (`
`EmbReal degrees)`

radians__

Parameters

<i>degrees</i>	
----------------	--

Returns

18.6.5.36 read_configuration() `int read_configuration (`
`const char * file)`

18.6.5.37 read_settings() `void read_settings (`
`void)`

read_settings

This file needs to be read from the users home directory to ensure it is writable.

18.6.5.38 read_string_setting() `String` read_string_setting (
 `toml_table_t * table,`
 `const char * key`)

18.6.5.39 rotate_vector() `EmbVector` rotate_vector (
 `EmbVector v,`
 `EmbReal alpha`)

Returns

18.6.5.40 run_script() `String` run_script (
 `StringList script`)

A basic line-by-line script processor to allow for extensions to the program.

Since the actuator uses command line style parsing, a script is just a text file with each line a compatible command. It should be stressed that this has no control flow on purpose. We don't want this to be hacked into a full scripting language that could cause havoc on the user's system.

However, it may be useful to set and get variables and define macros: neither of these will allow for endless loops, stack overflow or other problems that third-party scripts could introduce.

```
example.sh
-----
# Save characters by defining functions.
# The syntax features
# Semi-colon ';' seperates out lines like in bash.
# The line ending is the end of the function, but the style
# is a shell function, so we need to write the end brace.

donut() { circle $1 $2 $3 $5 ; circle $1 $2 $4 $5 }

donut 10 20 20 black
donut 20 40 20 black
-----
```

18.6.5.41 run_script_file() `String` run_script_file (
 `String fname`)

MainWindow::run_script_file.

Parameters

<i>fname</i>	The path of the script to run.
--------------	--------------------------------

18.6.5.42 set_enabled() `void` set_enabled (
 `QObject * parent,`
 `const char * key,`
 `bool enabled`)

set_enabled

Parameters

<i>parent</i>	
<i>key</i>	
<i>enabled</i>	

Todo error reporting.

18.6.5.43 set_visibility() void set_visibility (
 QObject * *parent*,
 const char * *key*,
 bool *visibility*)
 set_visibility

Parameters

<i>parent</i>	
<i>key</i>	
<i>visibility</i>	

Todo error reporting.

18.6.5.44 to_EmbVector() EmbVector to_EmbVector (
 QPointF *a*)

18.6.5.45 to_qlist() QList< QGraphicsItem * > to_qlist (
 std::vector< QGraphicsItem * > *list*)
 to_qlist

Parameters

<i>list</i>	
-------------	--

Returns

18.6.5.46 to_QPointF() QPointF to_QPointF (
 EmbVector *a*)

18.6.5.47 to_string_vector() QStringList to_string_vector (
 QStringList *list*)
 to_string_vector

Parameters

<i>list</i>	
-------------	--

Returns

18.6.5.48 to_vector() `std::vector< QGraphicsItem * > to_vector (`
 `QList< QGraphicsItem * > list)`
`to_vector`

Parameters

<i>list</i>	
-------------	--

Returns

18.6.5.49 tokenize() `StringList tokenize (`
 `String str,`
 `const char delim)`
`tokenize`

Parameters

<i>str</i>	
<i>delim</i>	

Returns

18.6.5.50 translate_str() `QString translate_str (`
 `const char * str)`

18.6.5.51 validFileFormat() `bool validFileFormat (`
 `String fileName)`

`MainWindow::validFileFormat.`

Parameters

<i>fileName</i>	
-----------------	--

Returns

Todo check the file exists on the system, rename to validFile?

18.6.5.52 write_settings() `void write_settings (`
 `void)`

`MainWindow::writeSettings.`

This file needs to be read from the users home directory to ensure it is writable

18.6.6 Variable Documentation

- 18.6.6.1** `_mainWin` `MainWindow*` `_mainWin` [extern]
- 18.6.6.2** `actionHash` `std::unordered_map<String, QAction*>` `actionHash` [extern]
- 18.6.6.3** `checkBoxes` `std::unordered_map<String, QCheckBox *>` `checkBoxes` [extern]
- 18.6.6.4** `comboBoxes` `std::unordered_map<String, QComboBox *>` `comboBoxes` [extern]
- 18.6.6.5** `config` `Dictionary` `config`
- 18.6.6.6** `config_tables` `std::unordered_map<String, Dictionary>` `config_tables` [extern]
- 18.6.6.7** `dialog` `Dictionary` `dialog`
- 18.6.6.8** `dockPropEdit` `PropertyEditor*` `dockPropEdit` [extern]
- 18.6.6.9** `dockUndoEdit` `UndoEditor*` `dockUndoEdit` [extern]
- 18.6.6.10** `doubleSpinBoxes` `std::unordered_map<String, QDoubleSpinBox *>` `doubleSpinBoxes` [extern]
- 18.6.6.11** `emb_constant_pi` `const EmbReal` `emb_constant_pi = 3.14159265358979323846` [static]
- 18.6.6.12** `groupBoxes` `std::unordered_map<String, QGroupBox *>` `groupBoxes` [extern]
- 18.6.6.13** `labels` `std::unordered_map<String, QLabel *>` `labels` [extern]
- 18.6.6.14** `lineEdits` `std::unordered_map<String, QLineEdit *>` `lineEdits` [extern]
- 18.6.6.15** `mdiArea` `MdiArea*` `mdiArea` [extern]
- 18.6.6.16** `menuHash` `std::unordered_map<String, QMenu*>` `menuHash` [extern]
- 18.6.6.17** `prompt` `CmdPrompt*` `prompt` [extern]
- 18.6.6.18** `scripts` `std::unordered_map<String, QStringList>` `scripts` [extern]

18.6.6.19 settings `Dictionary settings [extern]`

Settings System.

Rather than pollute the global namespace, we collect together all the global settings into a structure that stores them. This also allows us to create a complete copy of the settings for the purpose of restoring them if the user cancels out of the Settings Dialog.

18.6.6.20 spinBoxes `std::unordered_map<String, QSpinBox *> spinBoxes [extern]`**18.6.6.21 statusbar** `StatusBar* statusbar [extern]`**18.6.6.22 subMenuHash** `std::unordered_map<String, QMenu*> subMenuHash [extern]`**18.6.6.23 toolbarHash** `std::unordered_map<String, QToolBar*> toolbarHash [extern]`**18.6.6.24 toolButtons** `std::unordered_map<String, QPushButton *> toolButtons [extern]`**18.7 embroidermodder.h**

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

```

1 /*
2  * Embroidermodder 2.
3  *
4  * -----
5  *
6  * Copyright 2013-2023 The Embroidermodder Team
7  * Embroidermodder 2 is Open Source Software.
8  * See LICENSE for licensing terms.
9  *
10 * -----
11 *
12 * Use Python's PEP7 style guide.
13 *     https://peps.python.org/pep-0007/
14 */
15
21 #ifndef __EMBROIDERMODDER_UTILITY_H__
22 #define __EMBROIDERMODDER_UTILITY_H__
23
24 /*
25 * C/C++ Standard Libraries.
26 */
27 #include <cstdio>
28 #include <cmath>
29 #include <ctime>
30 #include <stdint.h>
31 #include <string>
32 #include <vector>
33 #include <unordered_map>
34 #include <string>
35 #include <filesystem>
36
37 /*
38 * Libraries included in "extern/".
39 */
40 #include "embroidery.h"
41 #include "toml.h"
42
43 /*
44 * Qt 6.0+ libraries.
45 */
46 #include <QAction>
47 #include <QApplication>
48
49 #include <QtPrintSupport>
50
51 #define STRING_TYPE 0
52 #define STRING_LIST_TYPE 1
53 #define REAL_TYPE 2
54 #define INT_TYPE 3
55 #define BOOL_TYPE 4
56 #define FUNCTION_TYPE 5

```

```

57 #define VECTOR_TYPE          6
58 #define UNKNOWN_TYPE        7
59
60 class ImageWidget;
61 class MdiArea;
62 class MdiWindow;
63 class View;
64 class StatusBar;
65 class CmdPrompt;
66 class PropertyEditor;
67 class UndoEditor;
68 class MainWindow;
69 class Geometry;
70
71 typedef std::string String;
72 typedef std::vector<String> StringList;
73
74 typedef struct Node_ {
75     String s;
76     EmbReal r;
77     int32_t i;
78     bool b;
79     StringList sl;
80     int type;
81 } Node;
82
83 typedef String (*Command)(String);
84 typedef std::vector<Node> NodeList;
85 typedef std::unordered_map<String, Node> Dictionary;
86
87 //Values
88 enum OBJ_TYPE_VALUES {
89     OBJ_TYPE_NULL = 0,
90     /*< NOTE: Allow this enum to evaluate false */
91     OBJ_TYPE_BASE = 100000,
92     /*< NOTE: Values >= 65536 ensure compatibility with qgraphicsitem_cast() */
93     OBJ_TYPE_ARC = 100001,
94     OBJ_TYPE_BLOCK = 100002,
95     /*< For the block type, that has to exist for SVG. */
96     OBJ_TYPE_CIRCLE = 100003,
97     OBJ_TYPE_DIMALIGNED = 100004,
98     /*< For the Aligned Dimension, that has to exist for DXF drawings. */
99     OBJ_TYPE_DIMANGULAR = 100005,
100    /*< For the Angular Dimension, that has to exist for DXF drawings. */
101    OBJ_TYPE_DIMARCLENGTH = 100006,
102    /*< For the Arc Length Dimension, that has to exist for DXF drawings. */
103    OBJ_TYPE_DIMDIAMETER = 100007,
104    OBJ_TYPE_DIMLEADER = 100008,
105    OBJ_TYPE_DIMLINEAR = 100009,
106    /*< For the Linear Dimension, that has to exist for DXF drawings. */
107    OBJ_TYPE_DIMORDINATE = 100010,
108    /*< For the Ordinate Dimension, that has to exist for DXF drawings. */
109    OBJ_TYPE_DIMRADIUS = 100011,
110    /*< For the Radial Dimension, that has to exist for DXF drawings. */
111    OBJ_TYPE_ELLIPSE = 100012,
112    OBJ_TYPE_ELLIPSEARC = 100013,
113    OBJ_TYPE_RUBBER = 100014,
114    OBJ_TYPE_GRID = 100015,
115    OBJ_TYPE_HATCH = 100016,
116    OBJ_TYPE_IMAGE = 100017,
117    OBJ_TYPE_INFINITELINE = 100018,
118    /*< For the Infinite Line object. Which should be removed from output as it exists
119    for drafting reasons. */
120    OBJ_TYPE_LINE = 100019,
121    OBJ_TYPE_PATH = 100020,
122    OBJ_TYPE_POINT = 100021,
123    OBJ_TYPE_POLYGON = 100022,
124    OBJ_TYPE_POLYLINE = 100023,
125    OBJ_TYPE_RAY = 100024,
126    /*< For the Ray object. */
127    OBJ_TYPE_RECTANGLE = 100025,
128    OBJ_TYPE_SLOT = 100026,
129    OBJ_TYPE_SPLINE = 100027,
130    OBJ_TYPE_TEXTMULTI = 100028,
131    OBJ_TYPE_TEXTSINGLE = 100029,
132    OBJ_TYPE_UNKNOWN = 100030
133 };
134
135 enum OBJ_KEYS {
136     OBJ_TYPE = 0,
137     /*< value type - int: See OBJ_TYPE_VALUES */
138     OBJ_NAME = 1,
139     /*< value type - str: See OBJ_NAME_VALUES */
140     OBJ_LAYER = 2,
141     /*< value type - str: "USER", "DEFINED", "STRINGS", etc... */
142     OBJ_COLOR = 3,
143     OBJ_LTYPE = 4,

```

```

159     /*< value type - int: See OBJ_LTYPE_VALUES */
160     OBJ_LWT = 5, //value type - int: 0-27
161     OBJ_RUBBER = 6 //value type - int: See OBJ_RUBBER_VALUES
162 };
163
164 static const EmbReal emb_constant_pi = 3.14159265358979323846;
165
166 /* Global variables
167 * -----
168 */
169 extern MdiArea* mdiArea;
170
171 extern Dictionary settings, dialog, config;
172 extern std::unordered_map<String, StringList> scripts;
173 extern std::unordered_map<String, QGroupBox *> groupBoxes;
174 extern std::unordered_map<String, QCheckBox *> checkBoxes;
175 extern std::unordered_map<String, QSpinBox *> spinBoxes;
176 extern std::unordered_map<String, QDoubleSpinBox *> doubleSpinBoxes;
177 extern std::unordered_map<String, QLabel *> labels;
178 extern std::unordered_map<String, QComboBox *> comboBoxes;
179 extern std::unordered_map<String, QLineEdit *> lineEdits;
180 extern std::unordered_map<String, QToolButton *> toolButtons;
181 extern std::unordered_map<String, Dictionary> config_tables;
182 extern std::unordered_map<String, QAction*> actionHash;
183 extern std::unordered_map<String, QToolBar*> toolbarHash;
184 extern std::unordered_map<String, QMenu*> menuHash;
185 extern std::unordered_map<String, QMenu*> subMenuHash;
186
187 extern MainWindow* _mainWin;
188 extern CmdPrompt* prompt;
189 extern PropertyEditor* dockPropEdit;
190 extern UndoEditor* dockUndoEdit;
191 extern StatusBar* statusBar;
192
193 /* Functions in the global namespace
194 * -----
195 */
196 int read_configuration(const char *file);
197 void read_settings(void);
198 void write_settings(void);
199 EmbVector rotate_vector(EmbVector v, EmbReal alpha);
200
201 QString translate_str(const char *str);
202 bool contains(StringList, String);
203 bool validFileFormat(String fileName);
204 QString fileExtension(String fileName);
205
206 void add_polyline(QPainterPath p, String rubberMode);
207
208 String read_string_setting(toml_table_t *table, const char *key);
209 StringList tokenize(String str, const char delim);
210 String convert_args_to_type(String label, StringList args,
211     const char *args_template, NodeList a);
212
213 View *activeView(void);
214 QGraphicsScene* activeScene();
215
216 void debug_message(String msg);
217 void set_enabled(QObject *parent, const char *key, bool enabled);
218 void set_visibility(QObject *parent, const char *name, bool visibility);
219
220 String actuator(String line);
221 String run_script_file(String fname);
222 String run_script(StringList script);
223 String construct_command(String command, const char *fmt, ...);
224
225 void create_menu(String menu, StringList def, bool topLevel);
226
227 QPointF to_QPointF(EmbVector a);
228 EmbVector to_EmbVector(QPointF a);
229 EmbVector operator+(EmbVector a, EmbVector b);
230 EmbVector operator-(EmbVector a, EmbVector b);
231 EmbVector operator*(EmbVector v, EmbReal s);
232 EmbReal radians__(EmbReal degrees);
233 EmbReal degrees__(EmbReal radian);
234
235 std::vector<QGraphicsItem*> to_vector(QList<QGraphicsItem*> list);
236 QList<QGraphicsItem*> to_qlist(std::vector<QGraphicsItem*> list);
237
238 StringList to_string_vector(QStringList list);
239
240 /* Interface creation functions.
241 */
242 void make_ui_element(String description);
243 QDoubleSpinBox *make_spinbox(QGroupBox *gb, String d,
244     QString object_name, EmbReal single_step, EmbReal lower, EmbReal upper, String key);
245 QCheckBox *make_checkbox(QGroupBox *gb, String d,

```

```

254     const char *label, const char *icon, String key);
255
256 /* Dictionary management functions.
257 */
258 Node node_bool(bool value);
259 Node node_int(int32_t value);
260 Node node_uint(uint32_t value);
261 Node node_real(EmbReal value);
262 Node node_str(String value);
263 Node node_qstr(QString value);
264 Node node_str_list(StringList value);
265
266 bool get_bool(Dictionary d, String key);
267 int32_t get_int(Dictionary d, String key);
268 uint32_t get_uint(Dictionary d, String key);
269 EmbReal get_real(Dictionary d, String key);
270 String get_str(Dictionary d, String key);
271 QString get_qstr(Dictionary d, String key);
272 StringList get_str_list(Dictionary d, String key);
273
274 class Geometry : public QGraphicsPathItem
275 {
276 public:
277     enum ArrowStyle {
278         NoArrow, //NOTE: Allow this enum to evaluate false
279         Open,
280         Closed,
281         Dot,
282         Box,
283         Tick
284     };
285
286     enum lineStyle {
287         NoLine, //NOTE: Allow this enum to evaluate false
288         Flared,
289         Fletching
290     };
291
292     Dictionary properties;
293
294     QPen objPen;
295     QPen lwtPen;
296     QLineF objLine;
297     String objRubberMode = "OBJ_RUBBER_OFF";
298     QHash<QString, QPointF> objRubberPoints;
299     QHash<QString, QString> objRubberTexts;
300     int64_t objID;
301
302     QPointF arcStartPoint;
303     QPointF arcMidPoint;
304     QPointF arcEndPoint;
305
306     bool curved;
307     bool filled;
308     QPainterPath lineStylePath;
309     QPainterPath arrowStylePath;
310     EmbReal arrowStyleAngle;
311     EmbReal arrowStyleLength;
312     EmbReal lineStyleAngle;
313     EmbReal lineStyleLength;
314
315     QPainterPath normalPath;
316
317     QString objText;
318     QString objTextFont;
319     QString objTextJustify;
320     bool objTextBackward;
321     bool objTextUpsideDown;
322     QPainterPath objTextPath;
323
324     std::vector<EmbReal> x_values;
325     std::vector<EmbReal> y_values;
326
327     int gripIndex;
328
329     int Type = OBJ_TYPE_BASE;
330     virtual int type() { return Type; }
331
332     Geometry(int object_type = OBJ_TYPE_BASE, QGraphicsItem* parent = 0);
333     Geometry(Geometry *obj, QGraphicsItem* parent = 0);
334     Geometry(EmbArc arc, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
335     Geometry(EmbCircle circle, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
336     Geometry(EmbLine line, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
337     Geometry(EmbEllipse ellipse, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
338     Geometry(EmbRect rect, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
339     Geometry(QString str, EmbVector position, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent =
340 0);

```

```

347     Geometry(EmbLine line, int Type_, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent);
348     Geometry(QPainterPath p, int type_, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
349     Geometry(EmbVector pos, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
350
351     void init_arc(EmbArc arc, QRgb rgb, Qt::PenStyle lineType);
352     void init_circle(EmbCircle circle, QRgb rgb, Qt::PenStyle lineType);
353     void init_line(EmbLine line, QRgb rgb, Qt::PenStyle lineType);
354     void init_ellipse(EmbEllipse ellipse, QRgb rgb, Qt::PenStyle lineType);
355     void init_rect(EmbRect rect, QRgb rgb, Qt::PenStyle lineType);
356     void init_text_single(QString str, EmbVector position, QRgb rgb, Qt::PenStyle lineType);
357     void init_path(QPainterPath p, QRgb rgb, Qt::PenStyle lineType);
358     void init_point(EmbVector pos, QRgb rgb, Qt::PenStyle lineType);
359
360     void init(void);
361
362     ~Geometry();
363
364     /* Getters */
365     Qt::PenStyle objectLineType() { return objPen.style(); }
366     EmbReal objectLineWeight() { return lwtPen.widthF(); }
367     QPointF objectRubberPoint(QString key);
368     QString objectRubberText(QString key);
369
370     QPointF objectCenter() { return scenePos(); }
371     QPointF objectPos() { return scenePos(); }
372     EmbReal objectX() { return scenePos().x(); }
373     EmbReal objectY() { return scenePos().y(); }
374
375     QPointF objectTopLeft();
376     QPointF objectTopRight();
377     QPointF objectBottomLeft();
378     QPointF objectBottomRight();
379     EmbReal objectArea();
380     QPointF objectStartPoint();
381     QPointF objectMidPoint();
382     QPointF objectEndPoint();
383
384     QRectF rect();
385     void circle_click(Dictionary global, EmbVector v);
386     EmbReal objectWidth();
387     EmbReal objectHeight();
388     EmbReal objectRadiusMajor();
389     EmbReal objectRadiusMinor();
390     EmbReal objectDiameterMajor();
391     EmbReal objectDiameterMinor();
392     QPointF objectEndPoint1();
393     QPointF objectEndPoint2();
394     EmbReal objectStartAngle();
395     EmbReal objectEndAngle();
396     EmbReal objectArcLength();
397     EmbReal objectChord();
398     EmbReal objectIncludedAngle();
399     bool objectClockwise();
400     EmbReal objectX1() { return objectEndPoint1().x(); }
401     EmbReal objectY1() { return objectEndPoint1().y(); }
402     EmbReal objectX2() { return objectEndPoint2().x(); }
403     EmbReal objectY2() { return objectEndPoint2().y(); }
404     EmbReal objectAngle();
405     QPointF objectDelta() { return objectEndPoint2() - objectEndPoint1(); }
406     EmbReal objectLength() { return objLine.length()*scale(); }
407     EmbReal objectRadius();
408     EmbReal objectDiameter();
409     EmbReal objectCircumference();
410     QPointF objectQuadrant0();
411     QPointF objectQuadrant90();
412     QPointF objectQuadrant180();
413     QPointF objectQuadrant270();
414     QPainterPath objectCopyPath();
415     QPainterPath objectSavePath();
416
417     std::vector<QPainterPath> objectSavePathList() { return subPathList(); }
418     std::vector<QPainterPath> subPathList();
419
420     int findIndex(const QPointF& point);
421
422     void setObjectEndPoint1(EmbVector endPt1);
423     void setObjectEndPoint2(EmbVector endPt2);
424
425     void updatePath();
426     void updatePath(const QPainterPath& p);
427     void updateLeader(void);
428
429     virtual QRectF boundingRect();
430
431     void drawRubberLine(const QLineF& rubLine, QPainter* painter = 0, const char* colorFromScene = 0);
432
433     void updateRubber(QPainter* painter = 0);

```



```

434 void vulcanize(void);
435 QPointF mouseSnapPoint(const QPointF& mousePoint);
436 std::vector<QPointF> allGripPoints();
437 void gripEdit(const QPointF& before, const QPointF& after);
438
439 void realRender(QPainter* painter, const QPainterPath& renderPath);
440 void paint(QPainter*, const QStyleOptionGraphicsItem*, QWidget*);
441
442 /* Updaters, todo: combine */
443 void calculateArcData(EmbArc arc);
444 void updateArcRect(EmbReal radius);
445
446 /* Setters */
447 void setObjectPos(const QPointF& point) { setPos(point.x(), point.y()); }
448 void setObjectX(EmbReal x) { setPos(x, objectY()); }
449 void setObjectY(EmbReal y) { setPos(objectX(), y); }
450 void setObjectCenter(EmbVector center);
451 void setObjectCenterX(EmbReal centerX);
452 void setObjectCenterY(EmbReal centerY);
453 void setObjectSize(EmbReal width, EmbReal height);
454 void setObjectRect(EmbReal x, EmbReal y, EmbReal w, EmbReal h);
455 void setRect(const QRectF& r);
456 void setRect(EmbReal x, EmbReal y, EmbReal w, EmbReal h);
457 void setLine(const QLineF& li);
458 void setLine(EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2);
459 void setObjectLineWeight(QString lineWeight);
460 void setObjectRadius(EmbReal radius);
461 void setObjectStartAngle(EmbReal angle);
462 void setObjectEndAngle(EmbReal angle);
463 void setObjectStartPoint(EmbVector point);
464 void setObjectMidPoint(EmbVector point);
465 void setObjectEndPoint(EmbVector point);
466 void setObjectDiameter(EmbReal diameter);
467 void setObjectArea(EmbReal area);
468 void setObjectCircumference(EmbReal circumference);
469 void setObjectPos(EmbReal x, EmbReal y) { setPos(x, y); }
470 void setObjectText(QString str);
471 void setObjectTextFont(QString font);
472 void setObjectTextJustify(QString justify);
473 void setObjectTextSize(EmbReal size);
474 void setObjectTextStyle(bool bold, bool italic, bool under, bool strike, bool over);
475 void setObjectTextBold(bool val);
476 void setObjectTextItalic(bool val);
477 void setObjectTextUnderline(bool val);
478 void setObjectTextStrikeOut(bool val);
479 void setObjectTextOverline(bool val);
480 void setObjectTextBackward(bool val);
481 void setObjectTextUpsideDown(bool val);
482 void setObjectRadiusMajor(EmbReal radius);
483 void setObjectRadiusMinor(EmbReal radius);
484 void setObjectDiameterMajor(EmbReal diameter);
485 void setObjectDiameterMinor(EmbReal diameter);
486
487 /* Scripted commands, uses the script string in */
488 void script_main(void);
489 void script_click(EmbVector v);
490 void script_context(QString str);
491 void script_prompt(QString str);
492 };
493
494 class SaveObject : public QObject
495 {
496     Q_OBJECT
497
498 public:
499     SaveObject(QGraphicsScene* theScene, QObject* parent = 0);
500     ~SaveObject();
501
502     bool save(QString fileName);
503
504     void addArc(EmbPattern* pattern, QGraphicsItem* item);
505     void addBlock(EmbPattern* pattern, QGraphicsItem* item);
506     void addCircle(EmbPattern* pattern, QGraphicsItem* item);
507     void addDimAligned(EmbPattern* pattern, QGraphicsItem* item);
508     void addDimAngular(EmbPattern* pattern, QGraphicsItem* item);
509     void addDimArcLength(EmbPattern* pattern, QGraphicsItem* item);
510     void addDimDiameter(EmbPattern* pattern, QGraphicsItem* item);
511     void addDimLeader(EmbPattern* pattern, QGraphicsItem* item);
512     void addDimLinear(EmbPattern* pattern, QGraphicsItem* item);
513     void addDimOrdinate(EmbPattern* pattern, QGraphicsItem* item);
514     void addDimRadius(EmbPattern* pattern, QGraphicsItem* item);
515     void addEllipse(EmbPattern* pattern, QGraphicsItem* item);
516     void addEllipseArc(EmbPattern* pattern, QGraphicsItem* item);
517     void addGrid(EmbPattern* pattern, QGraphicsItem* item);
518     void addHatch(EmbPattern* pattern, QGraphicsItem* item);
519     void addImage(EmbPattern* pattern, QGraphicsItem* item);
520     void addInfiniteLine(EmbPattern* pattern, QGraphicsItem* item);

```

```

524 void addLine(EmbPattern* pattern, QGraphicsItem* item);
525 void addPath(EmbPattern* pattern, QGraphicsItem* item);
526 void addPoint(EmbPattern* pattern, QGraphicsItem* item);
527 void addPolygon(EmbPattern* pattern, QGraphicsItem* item);
528 void addPolyline(EmbPattern* pattern, QGraphicsItem* item);
529 void addRay(EmbPattern* pattern, QGraphicsItem* item);
530 void addRectangle(EmbPattern* pattern, QGraphicsItem* item);
531 void addSlot(EmbPattern* pattern, QGraphicsItem* item);
532 void addSpline(EmbPattern* pattern, QGraphicsItem* item);
533 void addTextMulti(EmbPattern* pattern, QGraphicsItem* item);
534 void addTextSingle(EmbPattern* pattern, QGraphicsItem* item);
535
536 QGraphicsScene* gscene;
537 int formatType;
538
539 void toPolyline(EmbPattern* pattern, const QPointF& objPos, const QPainterPath& objPath, QString
layer, const QColor& color, QString lineType, QString lineWeight);
540 };
541
542 class Application : public QApplication
543 {
544     Q_OBJECT
545 public:
546     Application(int argc, char **argv);
547     void setMainWin(MainWindow* mainWin) { __mainWin = __mainWin; }
548     MainWindow* __mainWin;
549 protected:
550     virtual bool event(QEvent *e);
551 };
552
553 class CmdPromptInput : public QLineEdit
554 {
555     Q_OBJECT
556 public:
557     CmdPromptInput(QWidget* parent = 0);
558     ~CmdPromptInput() {}
559
560     QString curText;
561     QString defaultPrefix;
562     QString prefix;
563
564     QString lastCmd;
565     QString curCmd;
566     bool cmdActive;
567
568     bool rapidFireEnabled;
569     bool isBlinking;
570
571     void changeFormatting(std::vector<QTextLayout::FormatRange> formats);
572     void clearFormatting();
573     void applyFormatting();
574
575 protected:
576     void contextMenuEvent(QContextMenuEvent *event);
577     bool eventFilter(QObject *obj, QEvent *event);
578
579 signals:
580     void appendHistory(QString txt, int prefixLength);
581
582     //These connect to the CmdPrompt signals
583     void startCommand(QString cmd);
584     void runCommand(QString cmd, QString cmdtxt);
585     void deletePressed();
586     void tabPressed();
587     void escapePressed();
588     void upPressed();
589     void downPressed();
590     void F1Pressed();
591     void F2Pressed();
592     void F3Pressed();
593     void F4Pressed();
594     void F5Pressed();
595     void F6Pressed();
596     void F7Pressed();
597     void F8Pressed();
598     void F9Pressed();
599     void F10Pressed();
600     void F11Pressed();
601     void F12Pressed();
602     void cutPressed();
603     void copyPressed();
604     void pastePressed();
605     void selectAllPressed();
606     void undoPressed();
607     void redoPressed();

```

```

618
619     void shiftPressed();
620     void shiftReleased();
621
622     void showSettings();
623
624     void stopBlinking();
625
626 public slots:
627     void endCommand();
628     void processInput(void);
629     void checkSelection();
630     void updateCurrentText(QString txt);
631     void checkEditedText(QString txt);
632     void checkChangedText(QString txt);
633     void checkCursorPosition(int oldpos, int newpos);
634 private slots:
635     void copyClip();
636     void pasteClip();
637 };
638
639 class CmdPromptHistory : public QTextBrowser
640 {
641     Q_OBJECT
642
643 public:
644     CmdPromptHistory(QWidget* parent = 0);
645     ~CmdPromptHistory();
646
647     int tmpHeight;
648     QString applyFormatting(QString txt, int prefixLength);
649
650 protected:
651     void contextMenuEvent(QContextMenuEvent* event);
652
653 public slots:
654     void appendHistory(QString txt, int prefixLength);
655     void startResizeHistory(int y);
656     void stopResizeHistory(int y);
657     void resizeHistory(int y);
658
659 signals:
660     void historyAppended(QString txt);
661 };
662
663 class CmdPromptSplitter : public QSplitter
664 {
665     Q_OBJECT
666
667 public:
668     CmdPromptSplitter(QWidget* parent = 0);
669     ~CmdPromptSplitter();
670
671 protected:
672     QSplitterHandle* createHandle();
673
674 signals:
675     void pressResizeHistory(int y);
676     void releaseResizeHistory(int y);
677     void moveResizeHistory(int y);
678 };
679
680 class CmdPromptHandle : public QSplitterHandle
681 {
682     Q_OBJECT
683
684 public:
685     CmdPromptHandle(Qt::Orientation orientation, QSplitter* parent);
686     ~CmdPromptHandle();
687
688     int pressY;
689     int releaseY;
690     int moveY;
691
692 protected:
693     void mousePressEvent(QMouseEvent* e);
694     void mouseReleaseEvent(QMouseEvent* e);
695     void mouseMoveEvent(QMouseEvent* e);
696
697 signals:
698     void handlePressed(int y);
699     void handleReleased(int y);
700     void handleMoved(int y);
701 };
702
703 class CmdPrompt : public QWidget
704 {

```

```

717     Q_OBJECT
718
719 public:
720     CmdPrompt(QWidget* parent = 0);
721     ~CmdPrompt();
722
723     CmdPromptInput* promptInput;
724     CmdPromptHistory* promptHistory;
725     QVBoxLayout* promptVBoxLayout;
726     QFrame* promptDivider;
727
728     CmdPromptSplitter* promptSplitter;
729
730     QHash<QString, QString>* styleHash;
731     void updateStyle();
732     QTimer* blinkTimer;
733     bool blinkState;
734
735 public slots:
736     void setCurrentText(QString txt) {
737         promptInput->curText = promptInput->prefix + txt;
738         promptInput->setText(promptInput->curText);
739     }
740     void setHistory(QString txt) {
741         promptHistory->setHtml(txt);
742         promptHistory->moveCursor(QTextCursor::End, QTextCursor::MoveAnchor);
743     }
744     void setPrefix(QString txt);
745     void appendHistory(QString txt);
746
747     void alert(QString txt);
748
749     void startBlinking();
750     void stopBlinking();
751     void blink();
752
753     void setPromptTextColor(const QColor&);
754     void setPromptBackgroundColor(const QColor&);
755     void setPromptFontFamily(QString );
756     void setPromptFontStyle(QString );
757     void setPromptFontSize(int);
758
759     void floatingChanged(bool);
760
761     void saveHistory(QString fileName, bool html);
762
763 signals:
764     void appendTheHistory(QString txt, int prefixLength);
765
766     //For connecting outside of command prompt
767     void startCommand(QString cmd);
768     void runCommand(QString cmd, QString cmdtxt);
769     void deletePressed();
770     void tabPressed();
771     void escapePressed();
772     void upPressed();
773     void downPressed();
774     void F1Pressed();
775     void F2Pressed();
776     void F3Pressed();
777     void F4Pressed();
778     void F5Pressed();
779     void F6Pressed();
780     void F7Pressed();
781     void F8Pressed();
782     void F9Pressed();
783     void F10Pressed();
784     void F11Pressed();
785     void F12Pressed();
786     void cutPressed();
787     void copyPressed();
788     void pastePressed();
789     void selectAllPressed();
790     void undoPressed();
791     void redoPressed();
792
793     void shiftPressed();
794     void shiftReleased();
795
796     void showSettings();
797
798     void historyAppended(QString txt);
799 };
800
801 class EmbDetailsDialog : public QDialog
802 {
803     Q_OBJECT

```

```

807
808 public:
809     EmbDetailsDialog(QGraphicsScene* theScene, QWidget *parent = 0);
810     ~EmbDetailsDialog();
811
812     QWidget* mainWindow;
813
814     void getInfo();
815     QWidget* createMainWindow();
816     QWidget* createHistogram();
817
818     QDialogButtonBox* buttonBox;
819
820     uint32_t stitchesTotal;
821     uint32_t stitchesReal;
822     uint32_t stitchesJump;
823     uint32_t stitchesTrim;
824     uint32_t colorTotal;
825     uint32_t colorChanges;
826
827     QRectF boundingRect;
828 };
829
830 class ImageWidget : public QWidget
831 {
832     Q_OBJECT
833
834 public:
835     QImage img;
836     ImageWidget(QString filename, QWidget* parent = 0);
837     ~ImageWidget();
838
839     bool load(QString fileName);
840     bool save(QString fileName);
841
842 protected:
843     void paintEvent(QPaintEvent* event);
844 };
845
846 class LayerManager : public QDialog
847 {
848     Q_OBJECT
849
850 public:
851     QStandardItemModel* layerModel;
852     QSortFilterProxyModel* layerModelSorted;
853     QTreeView* treeView;
854
855     LayerManager(QWidget *parent = 0);
856     ~LayerManager();
857
858     void addLayer(QString name, const bool visible, const bool frozen,
859                  const EmbReal zValue, const QColor color, QString lineType,
860                  QString lineWeight, const bool print);
861 };
862
863 class MainWindow: public QMainWindow
864 {
865     Q_OBJECT
866
867 public:
868     MainWindow();
869     ~MainWindow();
870
871     MdiWindow* activeMdiWindow();
872     QUndoStack* activeUndoStack();
873
874     void setUndoCleanIcon(bool opened);
875
876     virtual void updateMenuToolbarStatusbar();
877
878     std::vector<QGraphicsItem*> cutCopyObjectList;
879
880     QString formatFilterOpen;
881     QString formatFilterSave;
882
883     bool isCommandActive() { return prompt->promptInput->cmdActive; }
884     QString activeCommand() { return prompt->promptInput->curCmd; }
885     QIcon create_icon(QString stub);
886     void create_toolbar(String toolbar, String label, QStringList entries);
887
888     QString platformString();
889
890 public slots:
891
892     void onCloseWindow();
893     virtual void onCloseMdiWin(MdiWindow*);

```

```

903
904 void recentMenuAboutToShow();
905
906 void onWindowActivated(QMdiSubWindow* w);
907 void windowMenuAboutToShow();
908 void windowMenuActivated( bool checked/*int id*/ );
909
910 void updateAllViewScrollBars(bool val);
911 void updateAllViewCrossHairColors(QRgb color);
912 void updateAllViewBackgroundColors(QRgb color);
913 void updateAllViewSelectBoxColors(QRgb colorL, QRgb fillL, QRgb colorR, QRgb fillR, int alpha);
914 void updateAllViewGridColors(QRgb color);
915 void updateAllViewRulerColors(QRgb color);
916
917 void updatePickAddMode(bool val);
918 void pickAddModeToggled();
919
920 void settingsPrompt();
921
922 protected:
923 virtual void resizeEvent(QResizeEvent*);
924 void closeEvent(QCloseEvent *event);
925 QAction* getFileSeparator();
926 void loadFormats();
927
928 bool shiftKeyPressedState;
929
930 QByteArray layoutState;
931
932 int numOfDocs;
933 int docIndex;
934
935 std::vector<MdiWindow*> listMdiWin;
936 QMdiSubWindow* findMdiWindow(String fileName);
937
938 QAction* myFileSeparator;
939
940 void createAllActions();
941 void createAllMenus();
942 void createAllToolbars();
943
944 // Selectors
945 QComboBox* layerSelector;
946 QComboBox* colorSelector;
947 QComboBox* linetypeSelector;
948 QComboBox* linewidthSelector;
949 QFontComboBox* textFontSelector;
950 QComboBox* textSizeSelector;
951
952 private slots:
953 void hideUnimplemented();
954
955 public slots:
956 void stub_testing();
957
958 void promptHistoryAppended(QString txt);
959 void logPromptInput(QString txt);
960 void promptInputPrevious();
961 void promptInputNext();
962
963 void about(void);
964 void tipOfTheDay(void);
965
966 void newFile();
967 void openFile(bool recent = false, String recentFile = "");
968 void openFilesSelected(StringList files);
969 void openrecentfile();
970 void savefile();
971 void saveasfile();
972 void quit();
973 void checkForUpdates();
974 // Help Menu
975 void buttonTipOfTheDayClicked(int);
976
977 void closeToolBar(QAction*);
978 void floatingChangedToolBar(bool);
979
980 void toggleGrid();
981 void toggleRuler();
982 void toggleLwt();
983
984 // Icons
985 void iconResize(int iconSize);
986
987 //Selectors
988 void layerSelectorIndexChanged(int index);
989 void colorSelectorIndexChanged(int index);

```

```

990 void linetypeSelectorIndexChanged(int index);
991 void linewidthSelectorIndexChanged(int index);
992 void textFontSelectorCurrentFontChanged(const QFont& font);
993 void textSizeSelectorIndexChanged(int index);
994
995 void setTextFont(QString str);
996 void setTextSize(EmbReal num);
997
998 QString getCurrentLayer();
999 QRgb getCurrentColor();
1000 QString getCurrentLineType();
1001 QString getCurrentLineWeight();
1002
1003 bool isShiftPressed();
1004 void setShiftPressed();
1005 void setShiftReleased();
1006
1007 void deletePressed();
1008 void escapePressed();
1009 };
1010
1011 class MdiWindow: public QMdiSubWindow
1012 {
1013     Q_OBJECT
1014
1015 public:
1016     MdiWindow(const int theIndex, QMdiArea* parent, Qt::WindowFlags wflags);
1017     ~MdiWindow();
1018
1019     QMdiArea* mdiArea;
1020     QGraphicsScene* gscene;
1021     View* gview;
1022
1023     bool fileWasLoaded;
1024
1025     QString promptHistory;
1026     std::vector<QString> promptInputList;
1027     int promptInputNum;
1028
1029     QPrinter printer;
1030
1031     QString curFile;
1032     void setCurrentFile(QString fileName);
1033
1034     int myIndex;
1035
1036     QString curLayer;
1037     QRgb curColor;
1038     QString curLineType;
1039     QString curLineWeight;
1040
1041     void promptInputPrevNext(bool prev);
1042
1043     virtual QSize sizeHint();
1044     QString getShortCurrentFile();
1045     void designDetails();
1046     bool loadFile(QString fileName);
1047     bool saveFile(QString fileName);
1048 signals:
1049     void sendCloseMdiWin(MdiWindow*);
1050
1051 public slots:
1052     void closeEvent(QCloseEvent* e);
1053     void onWindowActivated();
1054
1055     void currentLayerChanged(QString layer);
1056     void currentColorChanged(const QRgb& color);
1057     void currentLinetypeChanged(QString type);
1058     void currentLineweightChanged(QString weight);
1059
1060     void updateColorLinetypeLineweight();
1061     void deletePressed();
1062     void escapePressed();
1063
1064     void showViewScrollBars(bool val);
1065     void setViewCrossHairColor(QRgb color);
1066     void setViewBackgroundColor(QRgb color);
1067     void setViewSelectBoxColors(QRgb colorL, QRgb fillL, QRgb colorR, QRgb fillR, int alpha);
1068     void setViewGridColor(QRgb color);
1069     void setViewRulerColor(QRgb color);
1070
1071     void print();
1072     void saveBMC();
1073
1074     void promptHistoryAppended(QString txt);
1075     void logPromptInput(QString txt);
1076     void promptInputPrevious();

```

```

1077     void promptInputNext ();
1078 };
1079
1083 class MdiArea : public QMdiArea
1084 {
1085     Q_OBJECT
1086
1087 public:
1088     bool useLogo;
1089     bool useTexture;
1090     bool useColor;
1091
1092     QPixmap bgLogo;
1093     QPixmap bgTexture;
1094     QColor bgColor;
1095
1096     void zoomExtentsAllSubWindows ();
1097     void forceRepaint ();
1098
1099     MdiArea(QWidget* parent = 0);
1100     ~MdiArea ();
1101
1102     void useBackgroundLogo(bool use);
1103     void useBackgroundTexture(bool use);
1104     void useBackgroundColor(bool use);
1105
1106     void setBackgroundLogo(QString fileName);
1107     void setBackgroundTexture(QString fileName);
1108     void setBackgroundColor(const QColor& color);
1109
1110 public slots:
1111     void cascade();
1112     void tile();
1113 protected:
1114     virtual void mouseDoubleClickEvent(QMouseEvent* e);
1115     virtual void paintEvent(QPaintEvent* e);
1116 };
1117
1121 class PreviewDialog : public QFileDialog
1122 {
1123     Q_OBJECT
1124
1125 public:
1126     PreviewDialog(QWidget* parent = 0,
1127         QString caption = QString(),
1128         QString directory = QString(),
1129         QString filter = QString());
1130     ~PreviewDialog();
1131
1132     ImageWidget* imgWidget;
1133 };
1134
1135
1136 class PropertyEditor : public QDockWidget
1137 {
1138     Q_OBJECT
1139
1140 public:
1141     PropertyEditor(QString iconDirectory = QString(), bool pickAddMode = true, QWidget* widgetToFocus
1142         = 0, QWidget* parent = 0); //, Qt::WindowFlags flags = 0);
1143     ~PropertyEditor();
1144
1145     QWidget* focusWidget;
1146
1147     QString iconDir;
1148     int iconSize;
1149     Qt::ToolButtonStyle propertyEditorButtonStyle;
1150
1151     bool pickAdd;
1152
1153     std::vector<QGraphicsItem*> selectedItemList;
1154
1155     QToolButton* createToolButton(QString iconName, QString txt);
1156     QLineEdit* createLineEdit(QString validatorType = QString(), bool readOnly = false);
1157
1158     int precisionAngle;
1159     int precisionLength;
1160
1161     void updateLineEditStrIfVaries(QLineEdit* lineEdit, QString str);
1162     void updateLineEditNumIfVaries(QLineEdit* lineEdit, EmbReal num, bool useAnglePrecision);
1163     void updateFontComboBoxStrIfVaries(QFontComboBox* fontComboBox, QString str);
1164     void updateComboBoxStrIfVaries(QComboBox* comboBox, QString str, QStringList strList);
1165     void updateComboBoxBoolIfVaries(QComboBox* comboBox, bool val, bool yesOrNoText);
1166
1167     QSignalMapper* signalMapper;
1168     void mapSignal(QObject* fieldObj, QString name, QVariant value);
1169

```



```

1169     // Selection
1170     // =====
1171     QComboBox*   createComboBoxSelected();
1172     QToolButton* createToolButtonQSelect();
1173     QToolButton* createToolButtonPickAdd();
1174
1175     QComboBox*   comboBoxSelected;
1176     QToolButton* toolButtonQSelect;
1177     QToolButton* toolButtonPickAdd;
1178
1179     //TODO: Alphabetic/Categorized TabWidget
1180
1181     void createGroupBox(String group_box_key, const char *title);
1182
1183 protected:
1184     bool eventFilter(QObject *obj, QEvent *event);
1185
1186 signals:
1187     void pickAddModeToggled();
1188
1189 public slots:
1190     void setSelectedItems(std::vector<QGraphicsItem*> itemList);
1191     void updatePickAddModeButton(bool pickAddMode);
1192
1193 private slots:
1194     void fieldEdited(QObject* fieldObj);
1195     void showGroups(int objType);
1196     void showOneType(int index);
1197     void hideAllGroups();
1198     void clearAllFields();
1199     void togglePickAddMode();
1200 };
1201
1202
1203 class SelectBox : public QRubberBand
1204 {
1205     Q_OBJECT
1206
1207 public:
1208     SelectBox(Shape s, QWidget* parent = 0);
1209
1210     QColor leftBrushColor;
1211     QColor rightBrushColor;
1212     QColor leftPenColor;
1213     QColor rightPenColor;
1214     uint8_t alpha;
1215
1216     QBrush dirBrush;
1217     QBrush leftBrush;
1218     QBrush rightBrush;
1219
1220     QPen dirPen;
1221     QPen leftPen;
1222     QPen rightPen;
1223
1224     bool boxDir;
1225
1226     void forceRepaint();
1227
1228 public slots:
1229     void setDirection(int dir);
1230     void setColors(const QColor& colorL, const QColor& fillL, const QColor& colorR, const QColor&
        fillR, int newAlpha);
1231
1232 protected:
1233     void paintEvent(QPaintEvent*);
1234 };
1235
1236
1237 class Settings_Dialog : public QDialog
1238 {
1239     Q_OBJECT
1240
1241 public:
1242     Settings_Dialog(QString showTab = QString(), QWidget *parent = 0);
1243     ~Settings_Dialog();
1244
1245     QTabWidget* tabWidget;
1246
1247     QWidget* createTabGeneral();
1248     QWidget* createTabFilesPaths();
1249     QWidget* createTabDisplay();
1250     QWidget* createTabPrompt();
1251     QWidget* createTabOpenSave();
1252     QWidget* createTabPrinting();
1253     QWidget* createTabSnap();
1254     QWidget* createTabGridRuler();
1255     QWidget* createTabOrthoPolar();

```

```

1258     QWidget* createTabQuickSnap();
1259     QWidget* createTabQuickTrack();
1260     QWidget* createTabLineWeight();
1261     QWidget* createTabSelection();
1262
1263     QDialogButtonBox* buttonBox;
1264
1265     void addColorsToComboBox(QComboBox* comboBox);
1266
1267     void create_float_spinbox(
1268         QGroupBox *gb,
1269         QGridLayout* gridLayout,
1270         const char *label_in,
1271         EmbReal single_step,
1272         EmbReal lower,
1273         EmbReal upper,
1274         String,
1275         int row);
1276     QCheckBox* create_checkbox(QGroupBox *groupbox, String label);
1277
1278 private slots:
1279     void comboBoxIconSizeCurrentIndexChanged(int);
1280     void checkBoxGeneralMdiBGUseLogoStateChanged(int);
1281     void chooseGeneralMdiBackgroundLogo();
1282     void checkBoxGeneralMdiBGUseTextureStateChanged(int);
1283     void chooseGeneralMdiBackgroundTexture();
1284     void checkBoxGeneralMdiBGUseColorStateChanged(int);
1285     void chooseGeneralMdiBackgroundColor();
1286     void currentGeneralMdiBackgroundColorChanged(const QColor&);
1287     void checkBoxShowScrollBarsStateChanged(int);
1288     void comboBoxScrollBarWidgetCurrentIndexChanged(int);
1289     void chooseDisplayCrossHairColor();
1290     void currentDisplayCrossHairColorChanged(const QColor&);
1291     void chooseDisplayBackgroundColor();
1292     void currentDisplayBackgroundColorChanged(const QColor&);
1293     void chooseDisplaySelectBoxLeftColor();
1294     void currentDisplaySelectBoxLeftColorChanged(const QColor&);
1295     void chooseDisplaySelectBoxLeftFill();
1296     void currentDisplaySelectBoxLeftFillChanged(const QColor&);
1297     void chooseDisplaySelectBoxRightColor();
1298     void currentDisplaySelectBoxRightColorChanged(const QColor&);
1299     void chooseDisplaySelectBoxRightFill();
1300     void currentDisplaySelectBoxRightFillChanged(const QColor&);
1301     void comboBoxSelectionCoolGripColorCurrentIndexChanged(int index);
1302     void comboBoxSelectionHotGripColorCurrentIndexChanged(int index);
1303     void spinBoxDisplaySelectBoxAlphaValueChanged(int);
1304     void choosePromptTextColor();
1305     void currentPromptTextColorChanged(const QColor&);
1306     void choosePromptBackgroundColor();
1307     void currentPromptBackgroundColorChanged(const QColor&);
1308     void comboBoxPromptFontFamilyCurrentIndexChanged(QString );
1309     void comboBoxPromptFontStyleCurrentIndexChanged(QString );
1310     void spinBoxPromptFontSizeValueChanged(int);
1311     void checkBoxPromptSaveHistoryAsHtmlStateChanged(int);
1312     void checkBoxCustomFilterStateChanged(int);
1313     void buttonCustomFilterSelectAllClicked();
1314     void buttonCustomFilterClearAllClicked();
1315     void checkBoxGridColorMatchCrossHairStateChanged(int);
1316     void chooseGridColor();
1317     void currentGridColorChanged(const QColor&);
1318     void checkBoxGridLoadFromFileStateChanged(int);
1319     void comboBoxGridTypeCurrentIndexChanged(QString );
1320     void checkBoxGridCenterOnOriginStateChanged(int);
1321     void checkBoxRulerShowOnLoadStateChanged(int);
1322     void comboBoxRulerMetricCurrentIndexChanged(int);
1323     void chooseRulerColor();
1324     void currentRulerColorChanged(const QColor&);
1325     void spinBoxRulerPixelSizeValueChanged(double);
1326     void buttonQSnapSelectAllClicked();
1327     void buttonQSnapClearAllClicked();
1328     void comboBoxQSnapLocatorColorCurrentIndexChanged(int);
1329     void checkBoxLwtShowLwtStateChanged(int);
1330     void checkBoxLwtRealRenderStateChanged(int);
1331
1332     void acceptChanges();
1333     void rejectChanges();
1334
1335 signals:
1336     void buttonCustomFilterSelectAll(bool);
1337     void buttonCustomFilterClearAll(bool);
1338     void buttonQSnapSelectAll(bool);
1339     void buttonQSnapClearAll(bool);
1340 };
1341
1342 class StatusBar : public QStatusBar
1343 {
1344     Q_OBJECT

```

```

1348
1349 public:
1350     StatusBar(QWidget* parent = 0);
1351     std::unordered_map<String, QToolButton*> buttons;
1352     QLabel* statusBarMouseCoord;
1353     void setMouseCoord(EmbReal x, EmbReal y);
1354     void context_menu_action(QToolButton *button, const char *icon, const char *label, QMenu *menu,
1355                               String setting_page);
1356     void toggle(String key, bool on);
1357     void context_menu_event(QContextMenuEvent *event, QToolButton *button);
1358 };
1359
1360 class UndoEditor : public QDockWidget
1361 {
1362     Q_OBJECT
1363 public:
1364     UndoEditor(QString iconDirectory = QString(), QWidget* widgetToFocus = 0, QWidget* parent = 0);
1365     //, Qt::WindowFlags flags = 0);
1366     ~UndoEditor();
1367
1368     void addStack(QUndoStack* stack);
1369
1370     bool canUndo();
1371     bool canRedo();
1372
1373     QWidget* focusWidget;
1374
1375     QString iconDir;
1376     int iconSize;
1377
1378     QUndoGroup* undoGroup;
1379     QUndoView* undoView;
1380
1381     QString undoText();
1382     QString redoText();
1383 protected:
1384 public slots:
1385     void undo();
1386     void redo();
1387
1388     void updateCleanIcon(bool opened);
1389 };
1390
1391 class UndoableCommand : public QUndoCommand
1392 {
1393 public:
1394     UndoableCommand(String command, QString text, Geometry* obj, View* v, QUndoCommand* parent = 0);
1395     UndoableCommand(EmbVector d, QString text, Geometry* obj, View* v, QUndoCommand* parent = 0);
1396     UndoableCommand(String command, EmbVector pivot, EmbReal angle, QString text, Geometry* obj, View*
1397 v, QUndoCommand* parent = 0);
1398     UndoableCommand(QString type, View* v, QUndoCommand* parent = 0);
1399     UndoableCommand(const QPointF beforePoint, const QPointF afterPoint, QString text, Geometry* obj,
1400 View* v, QUndoCommand* parent = 0);
1401     UndoableCommand(EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, QString text, Geometry* obj, View*
1402 v, QUndoCommand* parent = 0);
1403
1404     int id(){ return 1234; }
1405     bool mergeWith(const QUndoCommand* command);
1406     void undo();
1407     void redo();
1408     void mirror();
1409     void rotate(EmbVector pivot, EmbReal rot);
1410
1411     Geometry* object;
1412     View* gview;
1413     String command;
1414     EmbVector delta;
1415     EmbVector pivot;
1416     QPointF before;
1417     QPointF after;
1418     EmbReal angle;
1419     EmbReal factor;
1420     QString navType;
1421     QTransform fromTransform;
1422     QTransform toTransform;
1423     QPointF fromCenter;
1424     QPointF toCenter;
1425     QLineF mirrorLine;
1426     bool done;
1427 };
1428
1429 class View : public QGraphicsView
1430 {
1431     Q_OBJECT
1432
1433

```

```

1439 public:
1440     View(QGraphicsScene* theScene, QWidget* parent);
1441     ~View();
1442
1443     Dictionary state;
1444
1445     std::vector<QGraphicsItem*> selected_items();
1446
1447     bool allowZoomIn();
1448     bool allowZoomOut();
1449
1450     QColor gridColor;
1451     QPainterPath gridPath;
1452     QPainterPath originPath;
1453     bool rulerMetric;
1454     QColor rulerColor;
1455     uint8_t rulerPixelSize;
1456
1457     bool grippingActive;
1458     bool rapidMoveActive;
1459     bool previewActive;
1460     bool pastingActive;
1461     bool movingActive;
1462     bool selectingActive;
1463     bool zoomWindowActive;
1464     bool panningRealTimeActive;
1465     bool panningPointActive;
1466     bool panningActive;
1467     bool qSnapActive;
1468     bool qSnapToggle;
1469
1470     Geometry* gripBaseObj;
1471     Geometry* tempBaseObj;
1472
1473     QGraphicsScene* gscene;
1474     QUndoStack* undoStack;
1475
1476     SelectBox* selectBox;
1477     QPointF scenePressPoint;
1478     QPoint pressPoint;
1479     QPointF sceneMovePoint;
1480     QPoint movePoint;
1481     QPointF sceneReleasePoint;
1482     QPoint releasePoint;
1483     QPointF sceneGripPoint;
1484
1485     void updateMouseCoords(int x, int y);
1486     QPoint viewMousePoint;
1487     QPointF sceneMousePoint;
1488     QRgb qsnapLocatorColor;
1489     uint8_t qsnapLocatorSize;
1490     uint8_t qsnapApertureSize;
1491     QRgb gripColorCool;
1492     QRgb gripColorHot;
1493     uint8_t gripSize;
1494     uint8_t pickBoxSize;
1495     QRgb crosshairColor;
1496     uint32_t crosshairSize;
1497
1498     void recalculateLimits();
1499     void zoomToPoint(const QPoint& mousePoint, int zoomDir);
1500     void centerAt(const QPointF& centerPoint);
1501     QPointF center() { return mapToScene(rect().center()); }
1502
1503     QUndoStack* getUndoStack() { return undoStack; }
1504     void addObject(Geometry* obj);
1505     void deleteObject(Geometry* obj);
1506     void vulcanizeObject(Geometry* obj);
1507
1508 public slots:
1509     void zoomIn();
1510     void zoomOut();
1511     void zoomWindow();
1512     void zoomSelected();
1513     void zoomExtents();
1514     void panRealTime();
1515     void panPoint();
1516     void panLeft();
1517     void panRight();
1518     void panUp();
1519     void panDown();
1520     void selectAll();
1521     void selectionChanged();
1522     void clearSelection();
1523     void deleteSelected();
1524     void moveSelected(EmbReal dx, EmbReal dy);
1525     void cut();

```

```

1526 void copy();
1527 void paste();
1528 void repeatAction();
1529 void moveAction();
1530 void scaleAction();
1531 void scaleSelected(EmbReal x, EmbReal y, EmbReal factor);
1532 void rotateAction();
1533 void rotateSelected(EmbReal x, EmbReal y, EmbReal rot);
1534 void mirrorSelected(EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2);
1535 int numSelected();
1536
1537 void deletePressed();
1538 void escapePressed();
1539
1540 void cornerButtonClicked();
1541
1542 void showScrollBars(bool val);
1543 void setCornerButton();
1544 void setCrossHairColor(QRgb color);
1545 void setCrossHairSize(uint8_t percent);
1546 void setBackgroundColor(QRgb color);
1547 void setSelectBoxColors(QRgb colorL, QRgb fillL, QRgb colorR, QRgb fillR, int alpha);
1548 void toggleSnap(bool on);
1549 void toggleGrid(bool on);
1550 void toggleRuler(bool on);
1551 void toggleOrtho(bool on);
1552 void togglePolar(bool on);
1553 void toggleQSnap(bool on);
1554 void toggleQTrack(bool on);
1555 void toggleLwt(bool on);
1556 void toggleReal(bool on);
1557 bool isLwtEnabled();
1558 bool isRealEnabled();
1559
1560 void setGridColor(QRgb color);
1561 void createGrid(QString gridType);
1562 void setRulerColor(QRgb color);
1563
1564 void previewOn(QString clone, QString mode, EmbReal x, EmbReal y, EmbReal data);
1565 void previewOff();
1566
1567 bool allowRubber();
1568 void addToRubberRoom(QGraphicsItem* item);
1569 void vulcanizeRubberRoom();
1570 void clearRubberRoom();
1571 void spareRubber(int64_t id);
1572 void setRubberMode(QString mode);
1573 void setRubberPoint(QString key, const QPointF& point);
1574 void setRubberText(QString key, QString txt);
1575
1576 protected:
1577 void mouseDoubleClickEvent(QMouseEvent* event);
1578 void mousePressEvent(QMouseEvent* event);
1579 void mouseMoveEvent(QMouseEvent* event);
1580 void mouseReleaseEvent(QMouseEvent* event);
1581 void wheelEvent(QWheelEvent* event);
1582 void contextMenuEvent(QContextMenuEvent* event);
1583 void drawBackground(QPainter* painter, const QRectF& rect);
1584 void drawForeground(QPainter* painter, const QRectF& rect);
1585 void enterEvent(QEvent* event);
1586
1587 private:
1588 QHash<int64_t, QGraphicsItem*> hashDeletedObjects;
1589
1590 QStringList spareRubberList;
1591
1592 void createGridRect();
1593 void createGridPolar();
1594 void createGridIso();
1595 void createOrigin();
1596
1597 void loadRulerSettings();
1598
1599 bool willUnderflowInt32(int64_t a, int64_t b);
1600 bool willOverflowInt32(int64_t a, int64_t b);
1601 int roundToMultiple(bool roundUp, int numToRound, int multiple);
1602 QPainterPath createRulerTextPath(EmbVector position, QString str, EmbReal height);
1603
1604 QList<QGraphicsItem*> previewObjectList;
1605 QGraphicsItemGroup* previewObjectItemGroup;
1606 QPointF previewPoint;
1607 EmbReal previewData;
1608 String previewMode;
1609
1610 std::vector<QGraphicsItem*> createObjectList(std::vector<QGraphicsItem*> list);
1611 QPointF cutCopyMousePoint;
1612 QGraphicsItemGroup* pasteObjectItemGroup;

```

```

1613     QPointF pasteDelta;
1614
1615     std::vector<QGraphicsItem*> rubberRoomList;
1616
1617     void copySelected();
1618
1619     void startGripping(Geometry* obj);
1620     void stopGripping(bool accept = false);
1621
1622     void panStart(const QPoint& point);
1623     int panDistance;
1624     int panStartX;
1625     int panStartY;
1626
1627     void alignScenePointWithViewPoint(const QPointF& scenePoint, const QPoint& viewPoint);
1628 };
1629
1630 #endif

```

18.8 embroidermodder2/imagewidget.cpp File Reference

```
#include "embroidermodder.h"
```

18.9 embroidermodder2/interface.cpp File Reference

```
#include "embroidermodder.h"
```

Functions

- QString [translate_str](#) (const char *str)
- Node [node_bool](#) (bool value)
 - set_node*
- Node [node_int](#) (int32_t value)
 - create_node*
- Node [node_uint](#) (uint32_t value)
 - create_node*
- Node [node_real](#) (EmbReal value)
 - set_node*
- Node [node_str](#) (String value)
 - set_node*
- Node [node_qstr](#) (QString value)
 - set_node*
- Node [node_str_list](#) (StringList value)
 - set_node*
- bool [get_bool](#) (Dictionary d, String key)
- int [get_int](#) (Dictionary d, String key)
- uint32_t [get_uint](#) (Dictionary d, String key)
- EmbReal [get_real](#) (Dictionary d, String key)
- String [get_str](#) (Dictionary d, String key)
- QString [get_qstr](#) (Dictionary d, String key)
- StringList [get_str_list](#) (Dictionary d, String key)
- StringList [to_string_vector](#) (QStringList list)
 - to_string_vector*
- StringList [tokenize](#) (String str, const char delim)
 - tokenize*
- QPointF [to_QPointF](#) (EmbVector a)
- EmbVector [to_EmbVector](#) (QPointF a)
- EmbVector [operator+](#) (EmbVector a, EmbVector b)

- operator +* Wrapper for `embVector_add` to use the syntax `a + b`.
- [EmbVector operator-](#) ([EmbVector](#) a, [EmbVector](#) b)
 - operator -* Wrapper for `embVector_subtract` to use the syntax `a - b`.
- [EmbVector operator*](#) ([EmbVector](#) v, [EmbReal](#) s)
 - operator **
- [EmbReal radians__](#) ([EmbReal](#) degrees)
 - radians__*
- [EmbReal degrees__](#) ([EmbReal](#) radian)
 - degrees__*
- `std::vector< QGraphicsItem * >` [to_vector](#) (`QList< QGraphicsItem * >` list)
 - to_vector*
- `QList< QGraphicsItem * >` [to_qlist](#) (`std::vector< QGraphicsItem * >` list)
 - to_qlist*
- void [debug_message](#) (`std::string` msg)
 - debug_message*
- `std::vector< float >` [get_n_reals](#) (`StringList` list, int n, int *offset)
- void [add_to_path](#) (`QPainterPath` *path, [EmbVector](#) scale, [String](#) command)
- void [set_enabled](#) (`QObject` *parent, const char *key, bool enabled)
 - set_enabled*
- void [set_visibility](#) (`QObject` *parent, const char *key, bool visibility)
 - set_visibility*
- void [make_ui_element](#) ([Dictionary](#) description)
- `QCheckBox` * [make_checkbox](#) (`QGroupBox` *gb, [String](#) dictionary, const char *label, const char *icon, [String](#) key)
- `QDoubleSpinBox` * [make_spinbox](#) (`QGroupBox` *gb, [String](#) dictionary, `QString` object_name, [EmbReal](#) single_step, [EmbReal](#) lower, [EmbReal](#) upper, [String](#) key)

18.9.1 Detailed Description

For wrappers to the Qt internals.

To help reduce reliance on Qt, only the functions wrap the Qt functions have a wrapper here. Ideally we could move some of the Qt headers here.

18.9.2 Function Documentation

18.9.2.1 [add_to_path\(\)](#) void [add_to_path](#) (
 [QPainterPath](#) * path,
 [EmbVector](#) scale,
 [String](#) command)

18.9.2.2 [debug_message\(\)](#) void [debug_message](#) (
 std::string msg)

[debug_message](#)

Parameters

<i>msg</i>	
------------	--

18.9.2.3 [degrees__\(\)](#) [EmbReal](#) [degrees__](#) (

`EmbReal radian)`

degrees__

Parameters

<code>radian</code>	
---------------------	--

Returns

18.9.2.4 get_bool() `bool get_bool (`
 `Dictionary d,`
 `String key)`

18.9.2.5 get_int() `int get_int (`
 `Dictionary d,`
 `String key)`

18.9.2.6 get_n_reals() `std::vector< float > get_n_reals (`
 `StringList list,`
 `int n,`
 `int * offset)`

Utility function for add_to_path.

18.9.2.7 get_qstr() `QString get_qstr (`
 `Dictionary d,`
 `String key)`

18.9.2.8 get_real() `EmbReal get_real (`
 `Dictionary d,`
 `String key)`

18.9.2.9 get_str() `String get_str (`
 `Dictionary d,`
 `String key)`

18.9.2.10 get_str_list() `StringList get_str_list (`
 `Dictionary d,`
 `String key)`

18.9.2.11 get_uint() `uint32_t get_uint (`
 `Dictionary d,`
 `String key)`

18.9.2.12 make_checkbox() `QCheckBox * make_checkbox (`
`QGroupBox * gb,`
`String dictionary,`
`const char * label,`
`const char * icon,`
`String key)`

18.9.2.13 make_spinbox() `QDoubleSpinBox * make_spinbox (`
`QGroupBox * gb,`
`String dictionary,`
`QString object_name,`
`EmbReal single_step,`
`EmbReal lower,`
`EmbReal upper,`
`String key)`

18.9.2.14 make_ui_element() `void make_ui_element (`
`Dictionary description)`

18.9.2.15 node_bool() `Node node_bool (`
`bool value)`

set_node

Parameters

<i>node</i>	
<i>value</i>	

18.9.2.16 node_int() `Node node_int (`
`int32_t value)`

create_node

Parameters

<i>mode</i>	
-------------	--

Returns

18.9.2.17 node_qstr() `Node node_qstr (`
`QString value)`

set_node

Parameters

<i>node</i>	
<i>value</i>	

18.9.2.18 node_real() `Node node_real (`
 `EmbReal value)`
`set_node`

Parameters

<i>node</i>	
<i>value</i>	

18.9.2.19 node_str() `Node node_str (`
 `String value)`
`set_node`

Parameters

<i>node</i>	
<i>value</i>	

18.9.2.20 node_str_list() `Node node_str_list (`
 `StringList value)`
`set_node`

Parameters

<i>node</i>	
<i>value</i>	

18.9.2.21 node_uint() `Node node_uint (`
 `uint32_t value)`
`create_node`

Parameters

<i>mode</i>	
-------------	--

Returns

18.9.2.22 operator*() `EmbVector operator* (`
 `EmbVector v,`
 `EmbReal s)`
`operator *`

Parameters

<i>v</i>	
<i>s</i>	

Returns

18.9.2.23 operator+() `EmbVector operator+ (`
`EmbVector a,`
`EmbVector b)`

operator + Wrapper for `embVector_add` to use the syntax `a + b`.

18.9.2.24 operator-() `EmbVector operator- (`
`EmbVector a,`
`EmbVector b)`

operator - Wrapper for `embVector_subtract` to use the syntax `a - b`.

18.9.2.25 radians__() `EmbReal radians__ (`
`EmbReal degrees)`

`radians__`

Parameters

<i>degrees</i>	
----------------	--

Returns

18.9.2.26 set_enabled() `void set_enabled (`
`QObject * parent,`
`const char * key,`
`bool enabled)`

`set_enabled`

Parameters

<i>parent</i>	
<i>key</i>	
<i>enabled</i>	

Todo error reporting.

18.9.2.27 set_visibility() `void set_visibility (`
`QObject * parent,`
`const char * key,`
`bool visibility)`

`set_visibility`

Parameters

<i>parent</i>	
---------------	--

Parameters

<i>key</i>	
<i>visibility</i>	

Todo error reporting.

18.9.2.28 to_EmbVector() `EmbVector to_EmbVector (`
`QPointF a)`

18.9.2.29 to_qlist() `QList< QGraphicsItem * > to_qlist (`
`std::vector< QGraphicsItem * > list)`

`to_qlist`

Parameters

<i>list</i>	
-------------	--

Returns

18.9.2.30 to_QPointF() `QPointF to_QPointF (`
`EmbVector a)`

18.9.2.31 to_string_vector() `StringList to_string_vector (`
`QStringList list)`

`to_string_vector`

Parameters

<i>list</i>	
-------------	--

Returns

18.9.2.32 to_vector() `std::vector< QGraphicsItem * > to_vector (`
`QList< QGraphicsItem * > list)`

`to_vector`

Parameters

<i>list</i>	
-------------	--

Returns

18.9.2.33 tokenize() `StringList tokenize (`
 `String str,`
 `const char delim)`

tokenize

Parameters

<i>str</i>	
<i>delim</i>	

Returns

18.9.2.34 translate_str() `QString translate_str (`
 `const char * str)`

18.10 embroidermodder2/layer-manager.cpp File Reference

```
#include "embroidermodder.h"
```

18.10.1 Detailed Description

Embroidermodder 2.

Copyright 2013-2022 The Embroidermodder Team Embroidermodder 2 is Open Source Software. See LICENSE for licensing terms.

Use Python's PEP7 style guide. <https://peps.python.org/pep-0007/>

18.11 embroidermodder2/mainwindow-menus.cpp File Reference

```
#include "embroidermodder.h"
```

Functions

- void `create_menu` (std::string menu, `StringList` def, bool topLevel)
 create_menu

18.11.1 Function Documentation

18.11.1.1 create_menu() `void create_menu (`
 `std::string menu,`
 `StringList def,`
 `bool topLevel)`

create_menu

Parameters

<i>menu</i>	
-------------	--

Parameters

<i>def</i>	
<i>topLevel</i>	

18.12 embroidermodder2/mainwindow-toolbars.cpp File Reference

```
#include "embroidermodder.h"
```

18.13 embroidermodder2/mainwindow.cpp File Reference

```
#include "embroidermodder.h"
#include <cerrno>
#include <iostream>
#include <fstream>
#include <string>
```

Enumerations

- enum [OBJ_LTYPE_VALUES](#) {
[OBJ_LTYPE_CONT](#) = 0 , [OBJ_LTYPE_CENTER](#) = 1 , [OBJ_LTYPE_DOT](#) = 2 , [OBJ_LTYPE_HIDDEN](#) = 3 ,
[OBJ_LTYPE_PHANTOM](#) = 4 , [OBJ_LTYPE_ZIGZAG](#) = 5 , [OBJ_LTYPE_RUNNING](#) = 6 , [OBJ_LTYPE_SATIN](#)
= 7 ,
[OBJ_LTYPE_FISHBONE](#) = 8 }
- enum [OBJ_LWT_VALUES](#) {
[OBJ_LWT_BYLAYER](#) = -2 , [OBJ_LWT_BYBLOCK](#) = -1 , [OBJ_LWT_DEFAULT](#) = 0 , [OBJ_LWT_01](#) = 1 ,
[OBJ_LWT_02](#) = 2 , [OBJ_LWT_03](#) = 3 , [OBJ_LWT_04](#) = 4 , [OBJ_LWT_05](#) = 5 ,
[OBJ_LWT_06](#) = 6 , [OBJ_LWT_07](#) = 7 , [OBJ_LWT_08](#) = 8 , [OBJ_LWT_09](#) = 9 ,
[OBJ_LWT_10](#) = 10 , [OBJ_LWT_11](#) = 11 , [OBJ_LWT_12](#) = 12 , [OBJ_LWT_13](#) = 13 ,
[OBJ_LWT_14](#) = 14 , [OBJ_LWT_15](#) = 15 , [OBJ_LWT_16](#) = 16 , [OBJ_LWT_17](#) = 17 ,
[OBJ_LWT_18](#) = 18 , [OBJ_LWT_19](#) = 19 , [OBJ_LWT_20](#) = 20 , [OBJ_LWT_21](#) = 21 ,
[OBJ_LWT_22](#) = 22 , [OBJ_LWT_23](#) = 23 , [OBJ_LWT_24](#) = 24 }
- enum [OBJ_SNAP_VALUES](#) {
[OBJ_SNAP_NULL](#) = 0 , [OBJ_SNAP_ENDPOINT](#) = 1 , [OBJ_SNAP_MIDPOINT](#) = 2 , [OBJ_SNAP_CENTER](#)
= 3 ,
[OBJ_SNAP_NODE](#) = 4 , [OBJ_SNAP_QUADRANT](#) = 5 , [OBJ_SNAP_INTERSECTION](#) = 6 , [OBJ_SNAP_EXTENSION](#)
= 7 ,
[OBJ_SNAP_INSERTION](#) = 8 , [OBJ_SNAP_PERPENDICULAR](#) = 9 , [OBJ_SNAP_TANGENT](#) = 10 ,
[OBJ_SNAP_NEAREST](#) = 11 ,
[OBJ_SNAP_APPINTERSECTION](#) = 12 , [OBJ_SNAP_PARALLEL](#) = 13 }

Functions

- static [String about_action](#) ([String](#) args)
- static [String add_arc_action](#) ([String](#) args)
[add_arc_action](#)
- static [String add_circle_action](#) ([String](#) args)
[add_circle_action](#)
- static [String add_dim_leader_action](#) ([String](#) args)
- static [String add_ellipse_action](#) ([String](#) args)
[AddEllipse.](#)
- static [String add_geometry_action](#) ([String](#) args)
[add_geometry_action](#)
- static [String add_horizontal_dimension_action](#) ([String](#) args)

- static [String add_image_action](#) ([String](#) args)
- static [String add_infinite_line_action](#) ([String](#) args)
- static [String add_line_action](#) ([String](#) args)
- static [String add_path_action](#) ([String](#) args)
- static [String add_point_action](#) ([String](#) args)
add_point_action
- static [String add_polygon_action](#) ([String](#) args)
add_polygon_action
- static [String add_polyline_action](#) ([String](#) args)
- static [String add_ray_action](#) ([String](#) args)
- static [String add_rectangle_action](#) ([String](#) args)
add_rectangle_action
- static [String add_regular_polygon_action](#) ([String](#) args)
AddRegularPolygon.
- static [String add_rounded_rectangle_action](#) ([String](#) args)
add_rounded_rectangle_action
- static [String add_rubber_action](#) ([String](#) args)
add_rubber_action
- static [String add_slot_action](#) ([String](#) args)
add_slot_action
- static [String add_text_multi_action](#) ([String](#) args)
add_text_multi_action
- static [String add_text_single_action](#) ([String](#) args)
add_text_single_action
- static [String add_to_selection_action](#) ([String](#) args)
add_to_selection_action
- static [String add_triangle_action](#) ([String](#) args)
add_triangle_action
- static [String add_vertical_dimension_action](#) ([String](#) args)
- static [String alert_action](#) ([String](#) args)
alert_action
- static [String allow_rubber_action](#) ([String](#) args)
AllowRubber.
- static [String append_history_action](#) ([String](#) args)
append_history_action
- static [String append_prompt_history_action](#) ([String](#) args)
AppendPromptHistory.
- static [String calculate_angle_action](#) ([String](#) args)
calculate_angle_action
- static [String calculate_distance_action](#) ([String](#) args)
calculate_distance
- static [String changelog_action](#) ([String](#) args)
changelog_action
- static [String clear_rubber_action](#) ([String](#) args)
ClearRubber.
- static [String copy_action](#) ([String](#) args)
copy_action
- static [String copy_selected_action](#) ([String](#) args)
CopySelected x y.
- static [String cut_action](#) ([String](#) args)
cut_action

- static [String cut_selected_action](#) ([String](#) args)
CutSelected x y.
- static [String day_vision_action](#) ([String](#) args)
MainWindow::dayVision.
- static [String delete_selected_action](#) ([String](#) args)
DeleteSelected.
- static [String design_details_action](#) ([String](#) args)
- static [String do_nothing_action](#) ([String](#) args)
do_nothing_action This action intensionally does nothing.
- static [String end_action](#) ([String](#) args)
end_action
- static [String error_action](#) ([String](#) args)
Error.
- static [String help_action](#) ([String](#) args)
help_action
- static [String icon_action](#) ([String](#) command)
icon_action
- static [String init_action](#) ([String](#) args)
init_action
- static [String messagebox_action](#) ([String](#) args)
MessageBox type title text.
- static [String mirror_selected_action](#) ([String](#) args)
MirrorSelected x1 y1 x2 y2.
- static [String mouse_x_action](#) ([String](#) args)
MouseX.
- static [String mouse_y_action](#) ([String](#) args)
MouseY.
- static [String move_selected_action](#) ([String](#) args)
MoveSelected dx dy.
- static [String new_action](#) ([String](#) args)
new_action
- static [String night_vision_action](#) ([String](#) args)
MainWindow::nightVision.
- static [String num_selected_action](#) ([String](#) args)
NumSelected.
- static [String open_action](#) ([String](#) args)
open_action
- static [String pan_action](#) ([String](#) mode)
pan_action
- static [String paste_action](#) ([String](#) args)
paste_action
- static [String paste_selected_action](#) ([String](#) args)
PasteSelected x y.
- static [String perpendicular_distance_action](#) ([String](#) args)
- static [String platform_action](#) ([String](#) args)
platform_action
- static [String preview_off_action](#) ([String](#) args)
PreviewOff.
- static [String preview_on_action](#) ([String](#) args)
preview_on_action
- static [String print_action](#) ([String](#) args)

- print_action*
- static [String print_area_action](#) ([String](#) args)
 - PrintArea x y w h.*
- static [String qsnap_x_action](#) ([String](#) args)
 - QSnapX.*
- static [String qsnap_y_action](#) ([String](#) args)
 - QSnapY.*
- static [String quit_action](#) ([String](#) args)
 - quit_action*
- static [String redo_action](#) ([String](#) args)
 - redo_action*
- static [String rotate_selected_action](#) ([String](#) args)
 - RotateSelected x y rot.*
- static [String scale_selected_action](#) ([String](#) args)
 - ScaleSelected x y factor.*
- static [String select_all_action](#) ([String](#) args)
 - select_all_action*
- static [String set_background_color_action](#) ([String](#) args)
 - set_background_color_action*
- static [String set_crosshair_color_action](#) ([String](#) args)
- static [String set_cursor_shape_action](#) ([String](#) args)
- static [String set_grid_color_action](#) ([String](#) args)
- static [String set_prompt_prefix_action](#) ([String](#) args)
 - set_prompt_prefix_action*
- static [String set_rubber_filter_action](#) ([String](#) args)
- static [String set_rubber_mode_action](#) ([String](#) args)
- static [String set_rubber_point_action](#) ([String](#) args)
- static [String set_rubber_text_action](#) ([String](#) args)
 - set_rubber_text_action*
- static [String settings_dialog_action](#) ([String](#) showTab)
 - settings_dialog*
- static [String spare_rubber_action](#) ([String](#) args)
 - SpareRubber.*
- static [String tip_of_the_day_action](#) ([String](#) args)
 - tip_of_the_day_action*
- static [String todo_action](#) ([String](#) args)
 - Todo.*
- static [String undo_action](#) ([String](#) args)
 - undo_action*
- static [String version_action](#) ([String](#) args)
 - version_action*
- static [String whats_this_action](#) ([String](#) args)
 - whats_this_action*
- static [String window_action](#) ([String](#) args)
 - window_action*
- static [String zoom_action](#) ([String](#) mode)
 - zoom_action*
- void [no_argument_debug](#) ([String](#) function_name, [String](#) args)
 - no_argument_debug*
- [String platformString](#) (void)
 - platformString*

- `View * activeView` (void)
activeView
- `QGraphicsScene * activeScene` ()
MainWindow::activeScene.
- `String make_layer_active_action` (String args)
MainWindow::makeLayerActive.
- `String layer_manager_action` (String args)
layer_manager_action
- `String layer_previous_action` (String args)
layer_previous_action
- static `String set_crosshair_color_action` (uint8_t r, uint8_t g, uint8_t b)
SetCrossHairColor.
- static `String set_grid_color_action` (uint8_t r, uint8_t g, uint8_t b)
set_grid_color
- static `String preview_on_action` (String clone, String mode, EmbReal x, EmbReal y, EmbReal data)
PreviewOn.
- static `String SetRubberText` (QString key, QString txt)
- static `String add_point_action` (EmbReal x, EmbReal y)
AddPoint.
- `String construct_command` (String command, const char *fmt,...)
construct_command
- `String read_string_setting` (toml_table_t *table, const char *key)
- `std::vector< String > read_string_list_setting` (toml_table_t *table, const char *key)
- int `read_configuration` (void)
Read the settings from file which aren't editable by the user. These files need to be placed in the install folder.
- bool `validRGB` (int r, int g, int b)
- `String disable_action` (String variable)
disable_action
- `String run_script_file` (String fname)
MainWindow::run_script_file.
- `String run_script` (StringList script)
A basic line-by-line script processor to allow for extensions to the program.
- `String actuator` (String line)
MainWindow::actuator.
- static `String clear_selection_action` (String args)
- static `String debug_action` (String args)
- static `String vulcanize_action` (String args)
- static `String rubber_action` (String command)
- static `String blink_prompt_action` (String args)
- `String convert_args_to_type` (String label, std::vector< String > args, const char *args_template, NodeList a)
Inspired by PyArg_ParseTupleAndKeywords allowing a uniform argument parsing framework.
- `String include_action` (NodeList a)
Include.
- `String is_int_action` (String args)
- `String SetTextAngle_action` (String args)
- bool `validFileFormat` (String fileName)
MainWindow::validFileFormat.

Variables

- [MainWindow](#) * [_mainWin](#) = 0
- [MdiArea](#) * [mdiArea](#) = 0
- [CmdPrompt](#) * [prompt](#) = 0
- [PropertyEditor](#) * [dockPropEdit](#) = 0
- [UndoEditor](#) * [dockUndoEdit](#) = 0
- [StatusBar](#) * [statusbar](#) = 0
- [QWizard](#) * [wizardTipOfTheDay](#)
- [QLabel](#) * [labelTipOfTheDay](#)
- [QCheckBox](#) * [checkBoxTipOfTheDay](#)
- [Dictionary](#) settings
 - Settings System.*
 - [Dictionary](#) dialog
 - [Dictionary](#) config
 - [std::unordered_map](#)< [String](#), [StringList](#) > [scripts](#)
 - [std::unordered_map](#)< [String](#), [QGroupBox](#) * > [groupBoxes](#)
 - [std::unordered_map](#)< [String](#), [QCheckBox](#) * > [checkBoxes](#)
 - [std::unordered_map](#)< [String](#), [QSpinBox](#) * > [spinBoxes](#)
 - [std::unordered_map](#)< [String](#), [QDoubleSpinBox](#) * > [doubleSpinBoxes](#)
 - [std::unordered_map](#)< [String](#), [QLabel](#) * > [labels](#)
 - [std::unordered_map](#)< [String](#), [QComboBox](#) * > [comboBoxes](#)
 - [std::unordered_map](#)< [String](#), [QLineEdit](#) * > [lineEdits](#)
 - [std::unordered_map](#)< [String](#), [QToolButton](#) * > [toolButtons](#)
 - [std::unordered_map](#)< [String](#), [Dictionary](#) > [config_tables](#)
 - [std::unordered_map](#)< [String](#), [QAction](#) * > [actionHash](#)
 - [std::unordered_map](#)< [String](#), [QToolBar](#) * > [toolbarHash](#)
 - [std::unordered_map](#)< [String](#), [QMenu](#) * > [menuHash](#)
 - [std::unordered_map](#)< [String](#), [QMenu](#) * > [subMenuHash](#)
 - [std::unordered_map](#)< [String](#), [Command](#) > [command_map](#)
 - [StringList](#) [rubber_modes](#)

18.13.1 Enumeration Type Documentation**18.13.1.1 OBJ_LTYPE_VALUES** enum [OBJ_LTYPE_VALUES](#)

Enumerator

OBJ_LTYPE_CONT	
OBJ_LTYPE_CENTER	
OBJ_LTYPE_DOT	
OBJ_LTYPE_HIDDEN	
OBJ_LTYPE_PHANTOM	
OBJ_LTYPE_ZIGZAG	
OBJ_LTYPE_RUNNING	
OBJ_LTYPE_SATIN	
OBJ_LTYPE_FISHBONE	

18.13.1.2 OBJ_LWT_VALUES enum [OBJ_LWT_VALUES](#)

Enumerator

OBJ_LWT_BYLAYER	
OBJ_LWT_BYBLOCK	
OBJ_LWT_DEFAULT	
OBJ_LWT_01	
OBJ_LWT_02	
OBJ_LWT_03	
OBJ_LWT_04	
OBJ_LWT_05	
OBJ_LWT_06	
OBJ_LWT_07	
OBJ_LWT_08	
OBJ_LWT_09	
OBJ_LWT_10	
OBJ_LWT_11	
OBJ_LWT_12	
OBJ_LWT_13	
OBJ_LWT_14	
OBJ_LWT_15	
OBJ_LWT_16	
OBJ_LWT_17	
OBJ_LWT_18	
OBJ_LWT_19	
OBJ_LWT_20	
OBJ_LWT_21	
OBJ_LWT_22	
OBJ_LWT_23	
OBJ_LWT_24	

18.13.1.3 OBJ_SNAP_VALUES enum [OBJ_SNAP_VALUES](#)

Enumerator

OBJ_SNAP_NULL	
OBJ_SNAP_ENDPOINT	
OBJ_SNAP_MIDPOINT	
OBJ_SNAP_CENTER	
OBJ_SNAP_NODE	
OBJ_SNAP_QUADRANT	
OBJ_SNAP_INTERSECTION	
OBJ_SNAP_EXTENSION	
OBJ_SNAP_INSERTION	
OBJ_SNAP_PERPENDICULAR	
OBJ_SNAP_TANGENT	
OBJ_SNAP_NEAREST	
OBJ_SNAP_APPINTERSECTION	
OBJ_SNAP_PARALLEL	

18.13.2 Function Documentation

18.13.2.1 about_action() `String about_action (`
`String args) [static]`

ACTIONS

Todo these should all be static, since other files use the actuator to call them.

18.13.2.2 activeScene() `QGraphicsScene * activeScene ()`
 MainWindow::activeScene.

Returns

18.13.2.3 activeView() `View * activeView (`
`void)`

activeView

Returns

18.13.2.4 actuator() `String actuator (`
`String line)`

MainWindow::actuator.

Parameters

<i>command</i>	
----------------	--

18.13.2.5 RUN COMMAND `QAction* act = qobject_cast<QAction*>(sender()); if (act) { prompt->endCommand(); prompt->setCurrentText(act->objectName()); prompt->processInput(); }`

18.13.2.6 INIT `QString fileName = "commands/" + cmd + "/" + cmd + ".js"; if (!getSettingsSelectionModePickFirst()) { actuator("clear-selection"); } TODO: Uncomment this line when post-selection is available engine->evaluate(cmd + "_main(void)", fileName);`

18.13.2.7 PROMPT `QString fileName = "commands/" + cmd + "/" + cmd + ".js"; NOTE: Replace any special characters that will cause a syntax error QString safeStr = str; safeStr.replace("\\", "\\"); safeStr.replace("'", "'"); if (prompt->promptInput->rapidFireEnabled) { engine->evaluate(cmd + "_prompt(" + safeStr + ")", fileName); } else { engine->evaluate(cmd + "_prompt(" + safeStr.toUpper() + ")", fileName); }`

18.13.2.8 add_arc_action() `static String add_arc_action (`
`String args) [static]`

add_arc_action

Parameters

<i>args</i>	
-------------	--

Returns

EmbReal startX, EmbReal startY, EmbReal midX, EmbReal midY, EmbReal endX, EmbReal endY, String rubberMode

18.13.2.9 add_circle_action() static `String` add_circle_action (
 `String` args) [static]
 add_circle_action

Returns

EmbReal centerX, EmbReal centerY, EmbReal radius, bool fill, String rubberMode

18.13.2.10 add_dim_leader_action() static `String` add_dim_leader_action (
 `String` args) [static]
 EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, EmbReal rot, String rubberMode

18.13.2.11 add_ellipse_action() static `String` add_ellipse_action (
 `String` args) [static]
 AddEllipse.

Parameters

<code>args</code>	
-------------------	--

Returns

EmbReal centerX, EmbReal centerY, EmbReal width, EmbReal height, EmbReal rot, bool fill, String rubberMode

18.13.2.12 add_geometry_action() static `String` add_geometry_action (
 `String` args) [static]
 add_geometry_action

Parameters

<code>args</code>	
-------------------	--

Returns

18.13.2.13 add_horizontal_dimension_action() static `String` add_horizontal_dimension_action (
 `String` args) [static]
 EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, EmbReal legHeight

18.13.2.14 add_image_action() static `String` add_image_action (
 `String` args) [static]
 QString img, EmbReal x, EmbReal y, EmbReal w, EmbReal h, EmbReal rot

18.13.2.15 add_infinite_line_action() static `String` add_infinite_line_action (
 `String` args) [static]
 EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, EmbReal rot

18.13.2.16 add_line_action() static `String` add_line_action (
`String args`) [static]
 EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, EmbReal rot, String rubberMode

18.13.2.17 add_path_action() static `String` add_path_action (
`String args`) [static]

Note

This native is different than the rest in that the Y+ is down (scripters need not worry about this).

EmbReal startX, EmbReal startY, const QPainterPath& p, String rubberMode

18.13.2.18 add_point_action() [1/2] static `String` add_point_action (
`EmbReal x`,
`EmbReal y`) [static]

AddPoint.

Parameters

<code>args</code>	
-------------------	--

Returns

18.13.2.19 add_point_action() [2/2] static `String` add_point_action (
`String args`) [static]
 add_point_action

Parameters

<code>args</code>	
-------------------	--

Returns

18.13.2.20 add_polygon_action() static `String` add_polygon_action (
`String args`) [static]
 add_polygon_action

Parameters

<code>args</code>	
-------------------	--

Returns

NOTE: This native is different than the rest in that the Y+ is down (scripters need not worry about this) Emb↔
 Real startX, EmbReal startY, const QPainterPath& p, String rubberMode

18.13.2.21 add_polyline_action() static `String` add_polyline_action (
`String args`) [static]

18.13.2.22 add_ray_action() static `String` add_ray_action (
 `String` args) [static]
 EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, EmbReal rot

18.13.2.23 add_rectangle_action() static `String` add_rectangle_action (
 `String` args) [static]
 add_rectangle_action

Parameters

args	
------	--

Returns

18.13.2.24 add_regular_polygon_action() static `String` add_regular_polygon_action (
 `String` args) [static]
 AddRegularPolygon.

Returns

EmbReal centerX, EmbReal centerY, quint16 sides, uint8_t mode, EmbReal rad, EmbReal rot, bool fill

18.13.2.25 add_rounded_rectangle_action() static `String` add_rounded_rectangle_action (
 `String` args) [static]
 add_rounded_rectangle_action

Parameters

args	
------	--

Returns

EmbReal x, EmbReal y, EmbReal w, EmbReal h, EmbReal rad, EmbReal rot, bool fill

18.13.2.26 add_rubber_action() `String` add_rubber_action (
 `String` args) [static]
 add_rubber_action

Parameters

args	
------	--

Returns

18.13.2.27 add_slot_action() static `String` add_slot_action (
 `String` args) [static]
add_slot_action

Parameters

args	
------	--

Returns

EmbReal centerX, EmbReal centerY, EmbReal diameter, EmbReal length, EmbReal rot, bool fill, String rubberMode

18.13.2.28 add_text_multi_action() static `String` add_text_multi_action (
 `String` args) [static]
add_text_multi_action

Returns

QString str, EmbReal x, EmbReal y, EmbReal rot, bool fill, String rubberMode

18.13.2.29 add_text_single_action() static `String` add_text_single_action (
 `String` args) [static]
add_text_single_action

Returns

QString str, EmbReal x, EmbReal y, EmbReal rot, bool fill, String rubberMode

18.13.2.30 add_to_selection_action() static `String` add_to_selection_action (
 `String` args) [static]
add_to_selection_action

Parameters

args	
------	--

Returns

18.13.2.31 add_triangle_action() static `String` add_triangle_action (
 `String` args) [static]
add_triangle_action

Parameters

args	
------	--

Returns

18.13.2.32 add_vertical_dimension_action() static `String` add_vertical_dimension_action (
 `String args`) [static]
EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, EmbReal legHeight

18.13.2.33 alert_action() static `String` alert_action (
 `String args`) [static]
alert_action

Parameters

<code>args</code>	
-------------------	--

Returns

18.13.2.34 allow_rubber_action() static `String` allow_rubber_action (
 `String args`) [static]
AllowRubber.

Returns

18.13.2.35 append_history_action() static `String` append_history_action (
 `String args`) [static]
append_history_action

Parameters

<code>args</code>	
-------------------	--

Returns

18.13.2.36 append_prompt_history_action() `String` append_prompt_history_action (
 `String args`) [static]
AppendPromptHistory.

Parameters

<code>a</code>	
----------------	--

Returns

18.13.2.37 blink_prompt_action() static `String` blink_prompt_action (
 `String args`) [static]

18.13.2.38 calculate_angle_action() static `String` calculate_angle_action (
 `String args`) [static]
calculate_angle_action

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.39 calculate_distance_action() static `String` calculate_distance_action (
 `String args`) [static]
calculate_distance

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.40 changelog_action() static `String` changelog_action (
 `String args`) [static]
changelog_action

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.41 clear_rubber_action() static `String` clear_rubber_action (
 `String args`) [static]
ClearRubber.

18.13.2.42 clear_selection_action() static `String` clear_selection_action (
 `String args`) [static]

18.13.2.43 construct_command() `String` construct_command (
 `String` command,
 const char * fmt,
 ...)

construct_command

Parameters

<i>command</i>	
<i>fmt</i>	

Returns

18.13.2.44 convert_args_to_type() `String` convert_args_to_type (
 `String` label,
 std::vector< `String` > args,
 const char * args_template,
 `NodeList` a)

Inspired by PyArg_ParseTupleAndKeywords allowing a uniform argument parsing framework.

Parameters

<i>label</i>	The caller's name.
<i>args</i>	The list of strings passed from the user.
<i>args_template</i>	The string of characters describing the types of the output.
<i>result</i>	The fixed length array of results.

Returns

An error message if an error occurred or an empty string if it passes.

18.13.2.45 copy_action() static `String` copy_action (
 `String` args) [static]

copy_action

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.46 copy_selected_action() static `String` copy_selected_action (
 `String` args) [static]

CopySelected x y.

18.13.2.47 cut_action() static `String` cut_action (
 `String` args) [static]
cut_action

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.48 cut_selected_action() static `String` cut_selected_action (
 `String` args) [static]
CutSelected x y.

18.13.2.49 day_vision_action() `String` day_vision_action (
 `String` args) [static]
MainWindow::dayVision.

Todo Make day vision color settings.

18.13.2.50 debug_action() static `String` debug_action (
 `String` args) [static]

18.13.2.51 delete_selected_action() static `String` delete_selected_action (
 `String` args) [static]
DeleteSelected.

18.13.2.52 design_details_action() `String` design_details_action (
 `String` args) [static]

18.13.2.53 disable_action() `String` disable_action (
 `String` variable)
disable_action

Parameters

<i>variable</i>	
-----------------	--

Returns

18.13.2.54 do_nothing_action() `String` do_nothing_action (
 `String` args) [static]
do_nothing_action This action intensionally does nothing.

Parameters

<i>args</i>	This is ignored, it's present to make it a Command.
-------------	---

Returns

An empty string.

18.13.2.55 end_action() static `String` end_action (
 `String args`) [static]
end_action

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.56 error_action() `String` error_action (
 `String args`) [static]
Error.

Parameters

<i>a</i>	
----------	--

Returns

18.13.2.57 help_action() static `String` help_action (
 `String args`) [static]
help_action

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.58 icon_action() static `String` icon_action (
 `String command`) [static]
icon_action

Parameters

<i>command</i>	
----------------	--

Returns

18.13.2.59 include_action() `String include_action (`
`NodeList a)`

Include.

Parameters

<i>a</i>	
----------	--

Returns

18.13.2.60 init_action() `static String init_action (`
`String args) [static]`

init_action

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.61 is_int_action() `String is_int_action (`
`String args)`

argument string "i"

18.13.2.62 layer_manager_action() `String layer_manager_action (`
`String args)`

layer_manager_action

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.63 layer_previous_action() `String layer_previous_action (`
`String args)`

layer_previous_action

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.64 make_layer_active_action() `String` make_layer_active_action (
 `String` args)

MainWindow::makeLayerActive.

Returns

18.13.2.65 messagebox_action() `static String` messagebox_action (
 `String` args) [static]

MessageBox *type title text*.

18.13.2.66 mirror_selected_action() `static String` mirror_selected_action (
 `String` args) [static]

MirrorSelected *x1 y1 x2 y2*.

18.13.2.67 mouse_x_action() `static String` mouse_x_action (
 `String` args) [static]

MouseX.

Returns

18.13.2.68 mouse_y_action() `static String` mouse_y_action (
 `String` args) [static]

MouseY.

Returns

18.13.2.69 move_selected_action() `static String` move_selected_action (
 `String` args) [static]

MoveSelected *dx dy*.

18.13.2.70 new_action() `static String` new_action (
 `String` args) [static]

new_action

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.71 night_vision_action() `String` night_vision_action (
 `String` *args*) [static]
MainWindow::nightVision.

Todo Make night vision color settings.

18.13.2.72 no_argument_debug() `void` no_argument_debug (
 `String` *function_name*,
 `String` *args*)
no_argument_debug

Parameters

<i>function_name</i>	
<i>args</i>	

18.13.2.73 num_selected_action() `static String` num_selected_action (
 `String` *args*) [static]
NumSelected.

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.74 open_action() `static String` open_action (
 `String` *args*) [static]
open_action

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.75 `pan_action()` `String` pan_action (
 `String` mode) [static]
pan_action

Parameters

mode	
------	--

Returns

18.13.2.76 `paste_action()` static `String` paste_action (
 `String` args) [static]
paste_action

Parameters

args	
------	--

Returns

18.13.2.77 `paste_selected_action()` static `String` paste_selected_action (
 `String` args) [static]
PasteSelected x y.

18.13.2.78 `perpendicular_distance_action()` static `String` perpendicular_distance_action (
 `String` args) [static]

18.13.2.79 `platform_action()` static `String` platform_action (
 `String` args) [static]
platform_action

Parameters

args	
------	--

Returns

18.13.2.80 `platformString()` `String` platformString (
 void)
platformString

Returns

18.13.2.81 preview_off_action() static `String` preview_off_action (
`String args`) [static]
 PreviewOff.

18.13.2.82 preview_on_action() [1/2] `String` preview_on_action (
`String args`) [static]
 preview_on_action

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.83 preview_on_action() [2/2] static `String` preview_on_action (
`String clone`,
`String mode`,
`EmbReal x`,
`EmbReal y`,
`EmbReal data`) [static]
 PreviewOn.

Parameters

<i>clone</i>	
<i>mode</i>	
<i>x</i>	
<i>y</i>	
<i>data</i>	

18.13.2.84 print_action() `String` print_action (
`String args`) [static]
 print_action

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.85 print_area_action() static `String` print_area_action (
`String args`) [static]
 PrintArea *x y w h*.
 EmbReal *x*, EmbReal *y*, EmbReal *w*, EmbReal *h*

18.13.2.86 qsnap_x_action() static `String` qsnap_x_action (
`String args`) [static]

QSnapX.

Returns

18.13.2.87 qsnap_y_action() static `String` qsnap_y_action (
 `String` args) [static]

QSnapY.

Returns

18.13.2.88 quit_action() static `String` quit_action (
 `String` args) [static]

quit_action

Parameters

<code>args</code>	
-------------------	--

Returns

18.13.2.89 read_configuration() int read_configuration (
 void)

Read the settings from file which aren't editable by the user. These files need to be placed in the install folder. Expected Keys for actions String icon; The stub used for the icon and the basic command. String command; String tooltip; The label in the menus and the message that appears when you hover over an icon. String statustip; The message that appears at the bottom of the . String shortcut; The keyboard shortcut for this action. StringList aliases; A list of all alternative commands, if empty only the icon string will be . StringList script; If this is a compound action this will be a list of commands or it can allow for command line style command aliases. For example: icon16 would become the string list {"iconResize 16"}.

18.13.2.90 read_string_list_setting() std::vector< `String` > read_string_list_setting (
 toml_table_t * table,
 const char * key)

18.13.2.91 read_string_setting() `String` read_string_setting (
 toml_table_t * table,
 const char * key)

18.13.2.92 redo_action() static `String` redo_action (
 `String` args) [static]

redo_action

Parameters

<code>args</code>	
-------------------	--

Returns

18.13.2.93 rotate_selected_action() static `String` rotate_selected_action (`String` args) [static]

RotateSelected *x y rot*.

18.13.2.94 rubber_action() static `String` rubber_action (`String` command) [static]

18.13.2.95 run_script() `String` run_script (`StringList` script)

A basic line-by-line script processor to allow for extensions to the program.

Since the actuator uses command line style parsing, a script is just a text file with each line a compatible command. It should be stressed that this has no control flow on purpose. We don't want this to be hacked into a full scripting language that could cause havoc on the user's system.

However, it may be useful to set and get variables and define macros: neither of these will allow for endless loops, stack overflow or other problems that third-party scripts could introduce.

example.sh

```
-----
# Save characters by defining functions.
# The syntax features
# Semi-colon ';' seperates out lines like in bash.
# The line ending is the end of the function, but the style
# is a shell function, so we need to write the end brace.

donut() { circle $1 $2 $3 $5 ; circle $1 $2 $4 $5 }

donut 10 20 20 black
donut 20 40 20 black
-----
```

18.13.2.96 run_script_file() `String` run_script_file (`String` fname)

MainWindow::run_script_file.

Parameters

<i>fname</i>	The path of the script to run.
--------------	--------------------------------

18.13.2.97 scale_selected_action() static `String` scale_selected_action (`String` args) [static]

ScaleSelected *x y factor*.

18.13.2.98 select_all_action() static `String` select_all_action (`String` args) [static]

select_all_action

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.99 set_background_color_action() static `String` set_background_color_action (
 `String args`) [static]
set_background_color_action

Parameters

<i>r</i>	
<i>g</i>	
<i>b</i>	

uint8_t r, uint8_t g, uint8_t b

18.13.2.100 set_crosshair_color_action() [1/2] static `String` set_crosshair_color_action (
 `String args`) [static]
argument string "iii"

18.13.2.101 set_crosshair_color_action() [2/2] static `String` set_crosshair_color_action (
 uint8_t r,
 uint8_t g,
 uint8_t b) [static]
SetCrossHairColor.

Parameters

<i>r</i>	
<i>g</i>	
<i>b</i>	

18.13.2.102 set_cursor_shape_action() static `String` set_cursor_shape_action (
 `String str`) [static]

18.13.2.103 set_grid_color_action() [1/2] `String` set_grid_color_action (
 `String args`) [static]
argument string "iii"

18.13.2.104 set_grid_color_action() [2/2] static `String` set_grid_color_action (
 uint8_t r,
 uint8_t g,
 uint8_t b) [static]
set_grid_color

Parameters

<i>r</i>	
----------	--

Parameters

<i>g</i>	
<i>b</i>	

18.13.2.105 set_prompt_prefix_action() static `String` set_prompt_prefix_action (
 `String args`) [static]
set_prompt_prefix_action

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.106 set_rubber_filter_action() static `String` set_rubber_filter_action (
 `String args`) [static]

18.13.2.107 set_rubber_mode_action() static `String` set_rubber_mode_action (
 `String args`) [static]

18.13.2.108 set_rubber_point_action() static `String` set_rubber_point_action (
 `String args`) [static]
QString key, EmbReal x, EmbReal y

18.13.2.109 set_rubber_text_action() `String` set_rubber_text_action (
 `String args`) [static]
set_rubber_text_action

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.110 SetRubberText() static `String` SetRubberText (
 `QString key`,
 `QString txt`) [static]

18.13.2.111 SetTextAngle_action() `String` SetTextAngle_action (
 `String args`)

18.13.2.112 settings_dialog_action() `String settings_dialog_action (`
 `String showTab) [static]`
`settings_dialog`

Parameters

<code>showTab</code>	
----------------------	--

18.13.2.113 spare_rubber_action() `static String spare_rubber_action (`
 `String args) [static]`
`SpareRubber.`

Parameters

<code>qint64</code>	<code>id</code>
---------------------	-----------------

18.13.2.114 tip_of_the_day_action() `String tip_of_the_day_action (`
 `String args) [static]`
`tip_of_the_day_action`

Parameters

<code>args</code>	
-------------------	--

Returns

18.13.2.115 todo_action() `String todo_action (`
 `String args) [static]`
`Todo.`

Parameters

<code>a</code>	
----------------	--

Returns

18.13.2.116 undo_action() `static String undo_action (`
 `String args) [static]`
`undo_action`

Parameters

<code>args</code>	
-------------------	--

Returns

18.13.2.117 validFileFormat() `bool validFileFormat (
 String fileName)`

MainWindow::validFileFormat.

Parameters

<i>fileName</i>	
-----------------	--

Returns

Todo check the file exists on the system, rename to validFile?

18.13.2.118 validRGB() `bool validRGB (
 int r,
 int g,
 int b)`

18.13.2.119 version_action() `static String version_action (
 String args) [static]`

version_action

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.120 vulcanize_action() `static String vulcanize_action (
 String args) [static]`

18.13.2.121 whats_this_action() `String whats_this_action (
 String args) [static]`

whats_this_action

Parameters

<i>args</i>	
-------------	--

Returns

18.13.2.122 window_action() static `String` window_action (
 `String` args) [static]
window_action

Parameters

args	
------	--

Returns

18.13.2.123 zoom_action() `String` zoom_action (
 `String` mode) [static]
zoom_action

Parameters

mode	
------	--

Returns

18.13.3 Variable Documentation

18.13.3.1 _mainWin `MainWindow*` _mainWin = 0

18.13.3.2 actionHash `std::unordered_map<String, QAction*>` actionHash

18.13.3.3 checkBoxes `std::unordered_map<String, QCheckBox *>` checkBoxes

18.13.3.4 checkBoxTipOfTheDay `QCheckBox*` checkBoxTipOfTheDay

18.13.3.5 comboBoxes `std::unordered_map<String, QComboBox *>` comboBoxes

18.13.3.6 command_map `std::unordered_map<String, Command>` command_map

18.13.3.7 config `Dictionary` config

18.13.3.8 config_tables `std::unordered_map<String, Dictionary>` config_tables

18.13.3.9 dialog `Dictionary` dialog

18.13.3.10 dockPropEdit `PropertyEditor* dockPropEdit = 0`

18.13.3.11 dockUndoEdit `UndoEditor* dockUndoEdit = 0`

18.13.3.12 doubleSpinBoxes `std::unordered_map<String, QDoubleSpinBox *> doubleSpinBoxes`

18.13.3.13 groupBoxes `std::unordered_map<String, QGroupBox *> groupBoxes`

18.13.3.14 labels `std::unordered_map<String, QLabel *> labels`

18.13.3.15 labelTipOfTheDay `QLabel* labelTipOfTheDay`

18.13.3.16 lineEdits `std::unordered_map<String, QLineEdit *> lineEdits`

18.13.3.17 mdiArea `MdiArea* mdiArea = 0`

18.13.3.18 menuHash `std::unordered_map<String, QMenu*> menuHash`

18.13.3.19 prompt `CmdPrompt* prompt = 0`

18.13.3.20 rubber_modes `StringList rubber_modes`

Initial value:

```
= {
    "CIRCLE_1P_RAD",
    "CIRCLE_1P_DIA",
    "CIRCLE_2P",
    "CIRCLE_3P",
    "CIRCLE_TTR",
    "CIRCLE_TTT",
    "DIMLEADER_LINE",
    "ELLIPSE_LINE",
    "ELLIPSE_MAJORDIAMETER_MINORRADIUS",
    "ELLIPSE_MAJORRADIUS_MINORRADIUS",
    "ELLIPSE_ROTATION",
    "LINE",
    "POLYGON",
    "POLYGON_INSCRIBE",
    "POLYGON_CIRCUMSCRIBE",
    "POLYLINE",
    "RECTANGLE",
    "TEXTSINGLE"
}
```

18.13.3.21 scripts `std::unordered_map<String, StringList> scripts`

18.13.3.22 settings `Dictionary settings`

Settings System.

Rather than pollute the global namespace, we collect together all the global settings into a structure that stores them. This also allows us to create a complete copy of the settings for the purpose of restoring them if the user cancels out of the Settings Dialog.

18.13.3.23 spinBoxes `std::unordered_map<String, QSpinBox *> spinBoxes`

18.13.3.24 statusbar `StatusBar* statusbar = 0`

18.13.3.25 subMenuHash `std::unordered_map<String, QMenu*> subMenuHash`

18.13.3.26 toolbarHash `std::unordered_map<String, QToolBar*> toolbarHash`

18.13.3.27 toolButtons `std::unordered_map<String, QToolButton *> toolButtons`

18.13.3.28 wizardTipOfTheDay `QWizard* wizardTipOfTheDay`

18.14 embroidermodder2/mdiarea.cpp File Reference

```
#include "embroidermodder.h"
```

18.15 embroidermodder2/mdiwindow.cpp File Reference

```
#include "embroidermodder.h"
```

Functions

- `QString fileExtension (String fileName)`
MdiWindow::fileExtension.

18.15.1 Function Documentation

18.15.1.1 fileExtension() `QString fileExtension (String fileName)`

MdiWindow::fileExtension.

Parameters

<i>fileName</i>	
-----------------	--

Returns

18.16 embroidermodder2/objects.cpp File Reference

```
#include "embroidermodder.h"
```

Functions

- `QPointF closest_point (QPointF position, std::vector< QPointF > points)`
mouse_snap_point

- [EmbReal](#) `fourier_series` ([EmbReal](#) arg, `std::vector< EmbReal >` terms)
- void `add_polyline` ([QPainterPath](#) p, [String](#) rubberMode)
- [EmbVector](#) `rotate_vector` ([EmbVector](#) v, [EmbReal](#) alpha)

18.16.1 Function Documentation

18.16.1.1 `add_polyline()` `void add_polyline (`
 [QPainterPath](#) p,
 [String](#) rubberMode)

18.16.1.2 `closest_point()` `QPointF closest_point (`
 [QPointF](#) position,
 `std::vector< QPointF >` points)
[mouse_snap_point](#)

Parameters

points	
------------------------	--

Returns

18.16.1.3 `fourier_series()` [EmbReal](#) `fourier_series (`
 [EmbReal](#) arg,
 `std::vector< EmbReal >` terms)

18.16.1.4 `rotate_vector()` [EmbVector](#) `rotate_vector (`
 [EmbVector](#) v,
 [EmbReal](#) alpha)

Returns

18.17 embroidermodder2/preview-dialog.cpp File Reference

```
#include "embroidermodder.h"
```

18.18 embroidermodder2/property-editor.cpp File Reference

```
#include "embroidermodder.h"
```

Functions

- `std::vector< Dictionary >` `load_group_box_data_from_table` ([String](#) key)

Variables

- [QString](#) `fieldOldText`

- QString [fieldNewText](#)
- QString [fieldVariesText](#)
- QString [fieldYesText](#)
- QString [fieldNoText](#)
- QString [fieldOnText](#)
- QString [fieldOffText](#)
- [StringList](#) [object_names](#)
- std::vector< std::pair< [String](#), int > > [group_box_types](#)
- QFontComboBox * [comboBoxTextSingleFont](#)
- std::unordered_map< [String](#), [Dictionary](#) > [group_box_data](#)

18.18.1 Function Documentation

18.18.1.1 [load_group_box_data_from_table\(\)](#) std::vector< [Dictionary](#) > [load_group_box_data_from_table](#) (
 [String](#) *key*)

18.18.2 Variable Documentation

18.18.2.1 [comboBoxTextSingleFont](#) QFontComboBox* [comboBoxTextSingleFont](#)

18.18.2.2 [fieldNewText](#) QString [fieldNewText](#)

18.18.2.3 [fieldNoText](#) QString [fieldNoText](#)

18.18.2.4 [fieldOffText](#) QString [fieldOffText](#)

18.18.2.5 [fieldOldText](#) QString [fieldOldText](#)

18.18.2.6 [fieldOnText](#) QString [fieldOnText](#)

18.18.2.7 [fieldVariesText](#) QString [fieldVariesText](#)

18.18.2.8 [fieldYesText](#) QString [fieldYesText](#)

18.18.2.9 [group_box_data](#) std::unordered_map<[String](#), [Dictionary](#)> [group_box_data](#)

18.18.2.10 [group_box_types](#) std::vector<std::pair<[String](#), int> > [group_box_types](#)

18.18.2.11 object_names [StringList](#) object_names**Initial value:**

```
= {
    "Base",
    "Arc",
    "Block",
    "Circle",
    "Aligned Dimension",
    "Angular Dimension",
    "Arclength Dimension",
    "Diameter Dimension",
    "Leader Dimension",
    "Linear Dimension",
    "Ordinate Dimension",
    "Radius Dimension",
    "Ellipse",
    "Image",
    "Infinite Line",
    "Line",
    "Path",
    "Point",
    "Polygon",
    "Polyline",
    "Ray",
    "Rectangle",
    "Multiline Text",
    "Text",
    "Unknown"
}
```

18.19 embroidermodder2/README.md File Reference**18.20 embroidermodder2/selectbox.cpp File Reference**

```
#include "embroidermodder.h"
```

18.21 embroidermodder2/settings-dialog.cpp File Reference

```
#include "embroidermodder.h"
#include <fstream>
```

Functions

- void [make_editing_copy](#) ([StringList](#) props)
- void [read_settings](#) (void)
 - read_settings*
- void [write_settings](#) (void)
 - MainWindow::writeSettings.*

Variables

- [Dictionary preview](#)
- [Dictionary accept_](#)
- [StringList extensions](#)
- [StringList general_props](#)
- [StringList display_props](#)
- [StringList prompt_props](#)
- [StringList quick_snap_props](#)
- [StringList opensave_props](#)

18.21.1 Function Documentation

18.21.1.1 make_editing_copy() void make_editing_copy (
 StringList props)

18.21.1.2 read_settings() void read_settings (
 void)

read_settings

This file needs to be read from the users home directory to ensure it is writable.

18.21.1.3 write_settings() void write_settings (
 void)

MainWindow::writeSettings.

This file needs to be read from the users home directory to ensure it is writable

18.21.2 Variable Documentation

18.21.2.1 accept_ Dictionary accept_

18.21.2.2 display_props StringList display_props

Initial value:

```
= {
    "display_use_opengl",
    "display_renderhint_aa",
    "display_renderhint_text_aa",
    "display_renderhint_smooth_pix",
    "display_renderhint_high_aa",
    "display_renderhint_noncosmetic",
    "display_show_scrollbars",
    "display_scrollbar_widget_num",
    "display_crosshair_color",
    "display_bg_color",
    "display_selectbox_left_color",
    "display_selectbox_left_fill",
    "display_selectbox_right_color",
    "display_selectbox_right_fill",
    "display_selectbox_alpha",
    "display_zoomscale_in",
    "display_zoomscale_out"
}
```

18.21.2.3 extensions StringList extensions

Initial value:

```
= {
    "100", "10o", "ART", "BMC", "BRO",
    "CND", "COL", "CSD", "CSV", "DAT",
    "DEM", "DSB", "DST", "DSZ", "DXF",
    "EDR", "EMD", "EXP", "EXY", "EYS",
    "FXY", "GNC", "GT", "HUS", "INB",
    "JEF", "KSM", "PCD", "PCM", "PCQ",
    "PCS", "PEC", "PEL", "PEM", "PES",
    "PHB", "PHC", "RGB", "SEW", "SHV",
    "SST", "STX", "SVG", "T09", "TAP",
    "THR", "TXT", "U00", "U01", "VIP",
    "VP3", "XXX", "ZSK"
}
```

18.21.2.4 general_props StringList general_props

Initial value:

```
= {
    "general_icon_theme",
    "general_icon_size",
    "general_mdi_bg_use_logo",
    "general_mdi_bg_logo",
    "general_mdi_bg_use_texture",
    "general_mdi_bg_texture",
}
```



```

    "general_mdi_bg_use_color",
    "general_mdi_bg_color",
    "general_tip_of_the_day"
}

```

18.21.2.5 opensave_props `StringList` opensave_props

Initial value:

```

= {
    "opensave_custom_filter"
}

```

18.21.2.6 preview `Dictionary` preview

Embroidermodder 2

Copyright 2013-2022 The Embroidermodder Team Embroidermodder 2 is Open Source Software. See LICENSE for licensing terms.

Use Python's PEP7 style guide. <https://peps.python.org/pep-0007/> The actuator changes the program state via these global variables.

These copies of the settings struct are for restoring the state if the user doesn't want to accept their changes in the settings dialog.

18.21.2.7 prompt_props `StringList` prompt_props

Initial value:

```

= {
    "prompt_text_color",
    "prompt_background_color",
    "prompt_font_family",
    "prompt_font_size",
    "prompt_save_history",
    "prompt_save_history_as_html"
}

```

18.21.2.8 quick_snap_props `StringList` quick_snap_props

Initial value:

```

= {
    "quicksnap_endpoint",
    "quicksnap_midpoint",
    "quicksnap_center",
    "quicksnap_node",
    "quicksnap_quadrant",
    "quicksnap_intersection",
    "quicksnap_extension",
    "quicksnap_insertion",
    "quicksnap_perpendicular",
    "quicksnap_tangent",
    "quicksnap_nearest",
    "quicksnap_apparent",
    "quicksnap_parallel",
    "quicksnap_locator_color",
    "quicksnap_locator_size",
    "quicksnap_aperture_size"
}

```

18.22 embroidermodder2/statusbar.cpp File Reference

```
#include "embroidermodder.h"
```

18.23 embroidermodder2/undo-commands.cpp File Reference

```
#include "embroidermodder.h"
```

18.24 embroidermodder2/undo-editor.cpp File Reference

```
#include "embroidermodder.h"
```

18.24.1 Detailed Description

Embroidermodder 2.

Copyright 2013-2022 The Embroidermodder Team Embroidermodder 2 is Open Source Software. See LICENSE for licensing terms.

Use Python's PEP7 style guide. <https://peps.python.org/pep-0007/>

18.25 embroidermodder2/view.cpp File Reference

```
#include "embroidermodder.h"
#include <cassert>
```

Functions

- bool `contains` (`StringList` list, `String` entry)

18.25.1 Detailed Description

Embroidermodder 2.

Copyright 2013-2022 The Embroidermodder Team Embroidermodder 2 is Open Source Software. See LICENSE for licensing terms.

Use Python's PEP7 style guide. <https://peps.python.org/pep-0007/>

18.25.2 Function Documentation

18.25.2.1 `contains()` `bool contains (`
 `StringList list,`
 `String entry)`

18.26 extern/libembroidery/src/array.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "embroidery_internal.h"
```

Functions

- `EmbArray *` `embArray_create` (`int` type)
- `int` `embArray_resize` (`EmbArray *`a)
- `void` `embArray_copy` (`EmbArray *`dst, `EmbArray *`src)
- `int` `embArray_addArc` (`EmbArray *`a, `EmbArc` b)
- `int` `embArray_addCircle` (`EmbArray *`a, `EmbCircle` b)
- `int` `embArray_addEllipse` (`EmbArray *`a, `EmbEllipse` b)
- `int` `embArray_addFlag` (`EmbArray *`a, `EmbFlag` b)
- `int` `embArray_addLine` (`EmbArray *`a, `EmbLine` b)
- `int` `embArray_addPath` (`EmbArray *`a, `EmbPath` b)
- `int` `embArray_addPoint` (`EmbArray *`a, `EmbPoint` b)
- `int` `embArray_addPolyline` (`EmbArray *`a, `EmbPolyline` b)
- `int` `embArray_addPolygon` (`EmbArray *`a, `EmbPolygon` b)
- `int` `embArray_addRect` (`EmbArray *`a, `EmbRect` b)
- `int` `embArray_addStitch` (`EmbArray *`a, `EmbStitch` b)
- `int` `embArray_addVector` (`EmbArray *`a, `EmbVector` b)
- `void` `embArray_free` (`EmbArray *`a)

18.26.1 Function Documentation

18.26.1.1 `embArray_addArc()` `int embArray_addArc (`
 `EmbArray * a,`
 `EmbArc b)`

Add an arc *b* to the EmbArray *a* and it returns if the element was successfully added.

18.26.1.2 `embArray_addCircle()` `int embArray_addCircle (`
 `EmbArray * a,`
 `EmbCircle b)`

Add a circle *b* to the EmbArray *a* and it returns if the element was successfully added.

18.26.1.3 `embArray_addEllipse()` `int embArray_addEllipse (`
 `EmbArray * a,`
 `EmbEllipse b)`

Add an ellipse *b* to the EmbArray *a* and it returns if the element was successfully added.

18.26.1.4 `embArray_addFlag()` `int embArray_addFlag (`
 `EmbArray * a,`
 `EmbFlag b)`

Add a flag *b* to the EmbArray *a* and it returns if the element was successfully added.

18.26.1.5 `embArray_addLine()` `int embArray_addLine (`
 `EmbArray * a,`
 `EmbLine b)`

Add a line *b* to the EmbArray *a* and it returns if the element was successfully added.

18.26.1.6 `embArray_addPath()` `int embArray_addPath (`
 `EmbArray * a,`
 `EmbPath b)`

Add a path *b* to the EmbArray *a* and it returns if the element was successfully added.

18.26.1.7 `embArray_addPoint()` `int embArray_addPoint (`
 `EmbArray * a,`
 `EmbPoint b)`

Add a point *b* to the EmbArray *a* and it returns if the element was successfully added.

18.26.1.8 `embArray_addPolygon()` `int embArray_addPolygon (`
 `EmbArray * a,`
 `EmbPolygon b)`

Add a polygon *b* to the EmbArray *a* and it returns if the element was successfully added.

18.26.1.9 `embArray_addPolyline()` `int embArray_addPolyline (`
 `EmbArray * a,`
 `EmbPolyline b)`

Add a polyline *b* to the EmbArray *a* and it returns if the element was successfully added.

18.26.1.10 `embArray_addRect()` `int embArray_addRect (`
 `EmbArray * a,`
 `EmbRect b)`

Add a rectangle *b* to the EmbArray *a* and it returns if the element was successfully added.

18.26.1.11 `embArray_addStitch()` `int embArray_addStitch (`
 `EmbArray * a,`
 `EmbStitch b)`

Add a stitch *b* to the EmbArray *a* and it returns if the element was successfully added.

18.26.1.12 `embArray_addVector()` `int embArray_addVector (`
 `EmbArray * a,`
 `EmbVector b)`

Add a vector *b* to the EmbArray *a* and it returns if the element was successfully added.

18.26.1.13 `embArray_copy()` `void embArray_copy (`
 `EmbArray * dst,`
 `EmbArray * src)`

Copies all entries in the EmbArray struct from *src* to *dst*.

18.26.1.14 `embArray_create()` `EmbArray * embArray_create (`
 `int type)`

Allocates memory for an EmbArray of the type determined by the argument *type*.

18.26.1.15 `embArray_free()` `void embArray_free (`
 `EmbArray * a)`

Free the memory of EmbArray *a*, recursively if necessary.

18.26.1.16 `embArray_resize()` `int embArray_resize (`
 `EmbArray * a)`

Resizes the array *a* to be CHUNK_SIZE entries longer if and only if the amount of room left is less than 3 entries.

18.27 extern/libembroidery/src/compress.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "embroidery_internal.h"
```

Functions

- int [hus_compress](#) (char *data, int length, char *output, int *output_length)
- void [huffman_build_table](#) (huffman *h)
- int * [huffman_lookup](#) (huffman h, int byte_lookup)
- void [compress_init](#) ()
- int [compress_get_bits](#) (compress *c, int length)
- int [compress_pop](#) (compress *c, int bit_count)
- int [compress_peek](#) (compress *c, int bit_count)
- int [compress_read_variable_length](#) (compress *c)
- void [compress_load_character_length_huffman](#) (compress *c)
- void [compress_load_character_huffman](#) (compress *c)
- void [compress_load_distance_huffman](#) (compress *c)
- void [compress_load_block](#) (compress *c)
- int [compress_get_token](#) (compress *c)
- int [compress_get_position](#) (compress *c)
- int [hus_decompress](#) (char *data, int length, char *output, int *output_length)

Variables

- int [huffman_lookup_data](#) [2]

18.27.1 Detailed Description

This is a work in progress.

Thanks to Jason Weiler for describing the binary formats of the HUS and VIP formats at:

<http://www.jasonweiler.com/HUSandVIPFileInfo.html>

Further thanks to github user tatarize for solving the mystery of the compression in:

<https://github.com/EmbroiderPy/pyembroidery>

with a description of that work here:

<https://stackoverflow.com/questions/7852670/greenleaf-archive-library>

This is based on their work.

18.27.2 Function Documentation

18.27.2.1 compress_get_bits() int compress_get_bits (
 compress * c,
 int length)

c length Returns .

18.27.2.2 compress_get_position() int compress_get_position (
 compress * c)

c . Returns the position as an int.

18.27.2.3 compress_get_token() int compress_get_token (
 compress * c)

c . Returns the token as an int.

18.27.2.4 compress_init() void compress_init ()

18.27.2.5 compress_load_block() void compress_load_block (
 compress * c)

c . Returns nothing.

18.27.2.6 compress_load_character_huffman() void compress_load_character_huffman (
 compress * c)

Load character table to compress struct *c*. Returns nothing.

18.27.2.7 compress_load_character_length_huffman() void compress_load_character_length_huffman (
 compress * c)

c . Returns.

18.27.2.8 compress_load_distance_huffman() void compress_load_distance_huffman (
 compress * c)

c . Returns nothing.

18.27.2.9 compress_peek() int compress_peek (
 compress * c,
 int bit_count)

c bit_count. Returns.

18.27.2.10 compress_pop() int compress_pop (
 compress * c,
 int bit_count)

c bit_count . Returns.

18.27.2.11 compress_read_variable_length() `int compress_read_variable_length (compress * c)`

c. Returns.

18.27.2.12 huffman_build_table() `void huffman_build_table (huffman * h)`

These next 2 functions represent the [Huffman](#) class in tartarize's code. *h*

18.27.2.13 huffman_lookup() `int * huffman_lookup (huffman h, int byte_lookup)`

Lookup *byte_lookup* in huffman table *h* return result as two bytes using the memory `huffman_lookup_data`.

18.27.2.14 hus_compress() `int hus_compress (char * data, int length, char * output, int * output_length)`

data length output *output_length* . Returns whether it was successful as an int.

This avoids the now unnecessary compression by placing a minimal header of 6 bytes and using only literals in the huffman compressed part (see the sources above).

18.27.2.15 hus_decompress() `int hus_decompress (char * data, int length, char * output, int * output_length)`

data length output *output_length* . Returns whether the decompression was successful.

18.27.3 Variable Documentation

18.27.3.1 huffman_lookup_data `int huffman_lookup_data[2]`

18.28 extern/libembroidery/src/embroidery.h File Reference

Classes

- struct [EmbColor_](#)
- struct [EmbVector_](#)
- struct [EmbImage_](#)
- struct [EmbBlock_](#)
- struct [EmbAlignedDim_](#)
- struct [EmbAngularDim_](#)
- struct [EmbArcLengthDim_](#)
- struct [EmbDiameterDim_](#)
- struct [EmbLeaderDim_](#)
- struct [EmbLinearDim_](#)
- struct [EmbOrdinateDim_](#)
- struct [EmbRadiusDim_](#)
- struct [EmbInfiniteLine_](#)
- struct [EmbRay_](#)
- struct [EmbTextMulti_](#)
- struct [EmbTextSingle_](#)
- struct [EmbTime_](#)

- struct [EmbPoint_](#)
- struct [EmbLine_](#)
- struct [EmbPath_](#)
- struct [EmbStitch_](#)
- struct [EmbThread_](#)
- struct [thread_color_](#)
- struct [EmbArc_](#)
- *absolute position (not relative)*
- struct [EmbRect_](#)
- struct [EmbCircle_](#)
- struct [EmbSatinOutline_](#)
- struct [EmbEllipse_](#)
- struct [EmbBezier_](#)
- struct [EmbSpline_](#)
- struct [LSYSTEM](#)
- struct [EmbGeometry_](#)
- struct [EmbArray_](#)
- struct [EmbLayer_](#)
- struct [EmbPattern_](#)
- struct [EmbFormatList_](#)

Macros

- #define [LIBEMBROIDERY_EMBEDDED_VERSION](#) 0
- #define [NORMAL](#) 0 /*! stitch to (x, y) */
- #define [JUMP](#) 1 /*! move to (x, y) */
- #define [TRIM](#) 2 /*! trim + move to (x, y) */
- #define [STOP](#) 4 /*! pause machine for thread change */
- #define [SEQUIN](#) 8 /*! sequin */
- #define [END](#) 16 /*! end of program */
- #define [EMB_FORMAT_100](#) 0
- #define [EMB_FORMAT_100](#) 1
- #define [EMB_FORMAT_ART](#) 2
- #define [EMB_FORMAT_BMC](#) 3
- #define [EMB_FORMAT_BRO](#) 4
- #define [EMB_FORMAT_CND](#) 5
- #define [EMB_FORMAT_COL](#) 6
- #define [EMB_FORMAT_CSD](#) 7
- #define [EMB_FORMAT_CSV](#) 8
- #define [EMB_FORMAT_DAT](#) 9
- #define [EMB_FORMAT_DEM](#) 10
- #define [EMB_FORMAT_DSB](#) 11
- #define [EMB_FORMAT_DST](#) 12
- #define [EMB_FORMAT_DSZ](#) 13
- #define [EMB_FORMAT_DXF](#) 14
- #define [EMB_FORMAT_EDR](#) 15
- #define [EMB_FORMAT_EMD](#) 16
- #define [EMB_FORMAT_EXP](#) 17
- #define [EMB_FORMAT_EXY](#) 18
- #define [EMB_FORMAT_EYS](#) 19
- #define [EMB_FORMAT_FXY](#) 20
- #define [EMB_FORMAT_GC](#) 21
- #define [EMB_FORMAT_GNC](#) 22
- #define [EMB_FORMAT_GT](#) 23
- #define [EMB_FORMAT_HUS](#) 24

- `#define EMB_FORMAT_INB` 25
- `#define EMB_FORMAT_INF` 26
- `#define EMB_FORMAT_JEF` 27
- `#define EMB_FORMAT_KSM` 28
- `#define EMB_FORMAT_MAX` 29
- `#define EMB_FORMAT_MIT` 30
- `#define EMB_FORMAT_NEW` 31
- `#define EMB_FORMAT_OFM` 32
- `#define EMB_FORMAT_PCD` 33
- `#define EMB_FORMAT_PCM` 34
- `#define EMB_FORMAT_PCQ` 35
- `#define EMB_FORMAT_PCS` 36
- `#define EMB_FORMAT_PEC` 37
- `#define EMB_FORMAT_PEL` 38
- `#define EMB_FORMAT_PEM` 39
- `#define EMB_FORMAT_PES` 40
- `#define EMB_FORMAT_PHB` 41
- `#define EMB_FORMAT_PHC` 42
- `#define EMB_FORMAT_PLT` 43
- `#define EMB_FORMAT_RGB` 44
- `#define EMB_FORMAT_SEW` 45
- `#define EMB_FORMAT_SHV` 46
- `#define EMB_FORMAT_SST` 47
- `#define EMB_FORMAT_STX` 48
- `#define EMB_FORMAT_SVG` 49
- `#define EMB_FORMAT_T01` 50
- `#define EMB_FORMAT_T09` 51
- `#define EMB_FORMAT_TAP` 52
- `#define EMB_FORMAT_THR` 53
- `#define EMB_FORMAT_TXT` 54
- `#define EMB_FORMAT_U00` 55
- `#define EMB_FORMAT_U01` 56
- `#define EMB_FORMAT_VIP` 57
- `#define EMB_FORMAT_VP3` 58
- `#define EMB_FORMAT_XXX` 59
- `#define EMB_FORMAT_ZSK` 60
- `#define Arc_Polyester` 0
- `#define Arc_Rayon` 1
- `#define CoatsAndClark_Rayon` 2
- `#define Exquisite_Polyester` 3
- `#define Fufu_Polyester` 4
- `#define Fufu_Rayon` 5
- `#define Hemingworth_Polyester` 6
- `#define Isacord_Polyester` 7
- `#define Isafil_Rayon` 8
- `#define Marathon_Polyester` 9
- `#define Marathon_Rayon` 10
- `#define Madeira_Polyester` 11
- `#define Madeira_Rayon` 12
- `#define Metro_Polyester` 13
- `#define Pantone` 14
- `#define RobisonAnton_Polyester` 15
- `#define RobisonAnton_Rayon` 16
- `#define Sigma_Polyester` 17
- `#define Sulky_Rayon` 18

- `#define ThreadArt_Rayon` 19
- `#define ThreadArt_Polyester` 20
- `#define ThreaDelight_Polyester` 21
- `#define Z102_Isacord_Polyester` 22
- `#define SVG_Colors` 23
- `#define hus_thread` 24
- `#define jef_thread` 25
- `#define pcm_thread` 26
- `#define pec_thread` 27
- `#define shv_thread` 28
- `#define dxf_color` 29
- `#define EMB_ARRAY` 0
- `#define EMB_ARC` 1
- `#define EMB_CIRCLE` 2
- `#define EMB_DIM_DIAMETER` 3
- `#define EMB_DIM_LEADER` 4
- `#define EMB_ELLIPSE` 5
- `#define EMB_FLAG` 6
- `#define EMB_LINE` 7
- `#define EMB_IMAGE` 8
- `#define EMB_PATH` 9
- `#define EMB_POINT` 10
- `#define EMB_POLYGON` 11
- `#define EMB_POLYLINE` 12
- `#define EMB_RECT` 13
- `#define EMB_SPLINE` 14
- `#define EMB_STITCH` 15
- `#define EMB_TEXT_SINGLE` 16
- `#define EMB_TEXT_MULTI` 17
- `#define EMB_VECTOR` 18
- `#define EMB_THREAD` 19
- `#define EMBFORMAT_UNSUPPORTED` 0
- `#define EMBFORMAT_STITCHONLY` 1
- `#define EMBFORMAT_OBJECTONLY` 2
- `#define EMBFORMAT_STCHANDOBJ` 3 /* binary operation: 1+2=3 */
- `#define numberOfFormats` 61
- `#define CHUNK_SIZE` 128
- `#define EMB_MAX_LAYERS` 10
- `#define MAX_THREADS` 256
- `#define EMBFORMAT_MAXEXT` 3
- `#define EMBFORMAT_MAXDESC` 50
- `#define MAX_STITCHES` 1000000
- `#define EMB_PUBLIC`

Typedefs

- `typedef float EmbReal`
- `typedef struct EmbColor_ EmbColor`
- `typedef struct EmbVector_ EmbVector`
- `typedef struct EmbArray_ EmbArray`
- `typedef struct EmbImage_ EmbImage`
- `typedef struct EmbBlock_ EmbBlock`
- `typedef struct EmbAlignedDim_ EmbAlignedDim`
- `typedef struct EmbAngularDim_ EmbAngularDim`
- `typedef struct EmbArcLengthDim_ EmbArcLengthDim`

- typedef struct [EmbDiameterDim_](#) [EmbDiameterDim](#)
- typedef struct [EmbLeaderDim_](#) [EmbLeaderDim](#)
- typedef struct [EmbLinearDim_](#) [EmbLinearDim](#)
- typedef struct [EmbOrdinateDim_](#) [EmbOrdinateDim](#)
- typedef struct [EmbRadiusDim_](#) [EmbRadiusDim](#)
- typedef struct [EmbInfiniteLine_](#) [EmbInfiniteLine](#)
- typedef struct [EmbRay_](#) [EmbRay](#)
- typedef struct [EmbTextMulti_](#) [EmbTextMulti](#)
- typedef struct [EmbTextSingle_](#) [EmbTextSingle](#)
- typedef struct [EmbTime_](#) [EmbTime](#)
- typedef struct [EmbPoint_](#) [EmbPoint](#)
- typedef struct [EmbLine_](#) [EmbLine](#)
- typedef struct [EmbPath_](#) [EmbPath](#)
- typedef struct [EmbStitch_](#) [EmbStitch](#)
- typedef struct [EmbThread_](#) [EmbThread](#)
- typedef struct [thread_color_](#) [thread_color](#)
- typedef struct [EmbArc_](#) [EmbArc](#)

absolute position (not relative)

- typedef struct [EmbRect_](#) [EmbRect](#)
- typedef struct [EmbCircle_](#) [EmbCircle](#)
- typedef [EmbPath](#) [EmbPolygon](#)
- typedef [EmbPath](#) [EmbPolyline](#)
- typedef int [EmbFlag](#)
- typedef struct [EmbSatinOutline_](#) [EmbSatinOutline](#)
- typedef struct [EmbEllipse_](#) [EmbEllipse](#)
- typedef struct [EmbBezier_](#) [EmbBezier](#)
- typedef struct [EmbSpline_](#) [EmbSpline](#)
- typedef struct [LSYSTEM L_](#) [system](#)
- typedef struct [EmbGeometry_](#) [EmbGeometry](#)
- typedef struct [EmbLayer_](#) [EmbLayer](#)
- typedef struct [EmbPattern_](#) [EmbPattern](#)
- typedef struct [EmbFormatList_](#) [EmbFormatList](#)

Functions

- [EMB_PUBLIC](#) int [lindenmayer_system](#) ([L_system](#) L, char *state, int iteration, int complete)
- [EMB_PUBLIC](#) int [hilbert_curve](#) ([EmbPattern](#) *pattern, int iterations)
- [EMB_PUBLIC](#) int [emb_identify_format](#) (const char *ending)
fileName
- [EMB_PUBLIC](#) void [testMain](#) (int level)
- [EMB_PUBLIC](#) int [convert](#) (const char *inf, const char *outf)
- [EMB_PUBLIC](#) [EmbColor](#) [embColor_make](#) (unsigned char r, unsigned char g, unsigned char b)
- [EMB_PUBLIC](#) [EmbColor](#) * [embColor_create](#) (unsigned char r, unsigned char g, unsigned char b)
- [EMB_PUBLIC](#) [EmbColor](#) [embColor_fromHexStr](#) (char *val)
Converts a 6 digit hex string (I.E. "00FF00") into an EmbColor and returns it.
- [EMB_PUBLIC](#) int [embColor_distance](#) ([EmbColor](#) a, [EmbColor](#) b)
a b
- [EMB_PUBLIC](#) [EmbArray](#) * [embArray_create](#) (int type)
- [EMB_PUBLIC](#) int [embArray_resize](#) ([EmbArray](#) *g)
- [EMB_PUBLIC](#) void [embArray_copy](#) ([EmbArray](#) *dst, [EmbArray](#) *src)
- [EMB_PUBLIC](#) int [embArray_addArc](#) ([EmbArray](#) *g, [EmbArc](#) arc)
- [EMB_PUBLIC](#) int [embArray_addCircle](#) ([EmbArray](#) *g, [EmbCircle](#) circle)
- [EMB_PUBLIC](#) int [embArray_addEllipse](#) ([EmbArray](#) *g, [EmbEllipse](#) ellipse)
- [EMB_PUBLIC](#) int [embArray_addFlag](#) ([EmbArray](#) *g, int flag)

- EMB_PUBLIC int embArray_addLine (EmbArray *g, EmbLine line)
 - EMB_PUBLIC int embArray_addRect (EmbArray *g, EmbRect rect)
 - EMB_PUBLIC int embArray_addPath (EmbArray *g, EmbPath p)
 - EMB_PUBLIC int embArray_addPoint (EmbArray *g, EmbPoint p)
 - EMB_PUBLIC int embArray_addPolygon (EmbArray *g, EmbPolygon p)
 - EMB_PUBLIC int embArray_addPolyline (EmbArray *g, EmbPolyline p)
 - EMB_PUBLIC int embArray_addStitch (EmbArray *g, EmbStitch st)
 - EMB_PUBLIC int embArray_addThread (EmbArray *g, EmbThread p)
 - EMB_PUBLIC int embArray_addVector (EmbArray *g, EmbVector)
 - EMB_PUBLIC void embArray_free (EmbArray *p)
 - EMB_PUBLIC EmbLine embLine_make (EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2)
 - EMB_PUBLIC void embLine_normalVector (EmbLine line, EmbVector *result, int clockwise)
 - EMB_PUBLIC EmbVector embLine_intersectionPoint (EmbLine line1, EmbLine line2)
 - EMB_PUBLIC int embThread_findNearestColor (EmbColor color, EmbColor *colors, int n_colors)
 - EMB_PUBLIC int embThread_findNearestThread (EmbColor color, EmbThread *threads, int n_threads)
- color thread_list n_threads*
- EMB_PUBLIC EmbThread embThread_getRandom (void)
 - EMB_PUBLIC void embVector_normalize (EmbVector vector, EmbVector *result)
 - EMB_PUBLIC void embVector_multiply (EmbVector vector, EmbReal magnitude, EmbVector *result)
 - EMB_PUBLIC EmbVector embVector_add (EmbVector v1, EmbVector v2)
 - EMB_PUBLIC EmbVector embVector_average (EmbVector v1, EmbVector v2)
 - EMB_PUBLIC EmbVector embVector_subtract (EmbVector v1, EmbVector v2)
 - EMB_PUBLIC EmbReal embVector_dot (EmbVector v1, EmbVector v2)
 - EMB_PUBLIC EmbReal embVector_cross (EmbVector v1, EmbVector v2)
- The "cross product" as vectors a and b returned as a real value.*
- EMB_PUBLIC void embVector_transpose_product (EmbVector v1, EmbVector v2, EmbVector *result)
 - EMB_PUBLIC EmbReal embVector_length (EmbVector vector)
 - EMB_PUBLIC EmbReal embVector_relativeX (EmbVector a1, EmbVector a2, EmbVector a3)
 - EMB_PUBLIC EmbReal embVector_relativeY (EmbVector a1, EmbVector a2, EmbVector a3)
 - EMB_PUBLIC EmbReal embVector_angle (EmbVector v)
 - EMB_PUBLIC EmbReal embVector_distance (EmbVector a, EmbVector b)
 - EMB_PUBLIC EmbVector embVector_unit (EmbReal angle)
 - EMB_PUBLIC EmbArc embArc_init (void)
 - EMB_PUBLIC char embArc_clockwise (EmbArc arc)
 - EMB_PUBLIC void getArcCenter (EmbArc arc, EmbVector *arcCenter)
 - EMB_PUBLIC char getArcDataFromBulge (EmbReal bulge, EmbArc *arc, EmbReal *arcCenterX, EmbReal *arcCenterY, EmbReal *radius, EmbReal *diameter, EmbReal *chord, EmbReal *chordMidX, EmbReal *chordMidY, EmbReal *sagitta, EmbReal *apothem, EmbReal *incAngleInDegrees, char *clockwise)
 - EMB_PUBLIC EmbCircle embCircle_init (void)
 - EMB_PUBLIC int getCircleCircleIntersections (EmbCircle c0, EmbCircle c1, EmbVector *v0, EmbVector *v1)
 - EMB_PUBLIC int getCircleTangentPoints (EmbCircle c, EmbVector p, EmbVector *v0, EmbVector *v1)
 - EMB_PUBLIC EmbEllipse embEllipse_init (void)
 - EMB_PUBLIC EmbEllipse embEllipse_make (EmbReal cx, EmbReal cy, EmbReal rx, EmbReal ry)
 - EMB_PUBLIC EmbReal embEllipse_diameterX (EmbEllipse ellipse)
 - EMB_PUBLIC EmbReal embEllipse_diameterY (EmbEllipse ellipse)
 - EMB_PUBLIC EmbReal embEllipse_width (EmbEllipse ellipse)
 - EMB_PUBLIC EmbReal embEllipse_height (EmbEllipse ellipse)
 - EMB_PUBLIC EmbReal embEllipse_area (EmbEllipse ellipse)
 - EMB_PUBLIC EmbReal embEllipse_perimeter (EmbEllipse ellipse)
 - EMB_PUBLIC EmbImage embImage_create (int, int)
 - EMB_PUBLIC void embImage_read (EmbImage *image, char *fname)
 - EMB_PUBLIC int embImage_write (EmbImage *image, char *fname)
 - EMB_PUBLIC void embImage_free (EmbImage *image)
 - EMB_PUBLIC EmbRect embRect_init (void)

- EMB_PUBLIC EmbReal embRect_area (EmbRect)
- EMB_PUBLIC int threadColor (const char *, int brand)
- EMB_PUBLIC int threadColorNum (unsigned int color, int brand)
- EMB_PUBLIC const char * threadColorName (unsigned int color, int brand)
- EMB_PUBLIC void embTime_initNow (EmbTime *t)
- EMB_PUBLIC EmbTime embTime_time (EmbTime *t)
- EMB_PUBLIC void embSatinOutline_generateSatinOutline (EmbArray *lines, EmbReal thickness, EmbSatinOutline *result)
lines thickness result
- EMB_PUBLIC EmbArray * embSatinOutline_renderStitches (EmbSatinOutline *result, EmbReal density)
result density
- EMB_PUBLIC EmbGeometry * embGeometry_init (int type_in)
Our generic object interface backends to each individual type.
- EMB_PUBLIC void embGeometry_free (EmbGeometry *obj)
Free the memory occupied by a non-stitch geometry object.
- EMB_PUBLIC void embGeometry_move (EmbGeometry *obj, EmbVector delta)
Translate obj by the vector delta.
- EMB_PUBLIC EmbRect embGeometry_boundingRect (EmbGeometry *obj)
Calculate the bounding box of geometry obj based on what kind of geometric object it is.
- EMB_PUBLIC void embGeometry_vulcanize (EmbGeometry *obj)
Toggle the rubber mode of the object.
- EMB_PUBLIC EmbPattern * embPattern_create (void)
- EMB_PUBLIC void embPattern_hideStitchesOverLength (EmbPattern *p, int length)
- EMB_PUBLIC void embPattern_fixColorCount (EmbPattern *p)
- EMB_PUBLIC int embPattern_addThread (EmbPattern *p, EmbThread thread)
- EMB_PUBLIC void embPattern_addStitchAbs (EmbPattern *p, EmbReal x, EmbReal y, int flags, int isAuto↔ ColorIndex)
- EMB_PUBLIC void embPattern_addStitchRel (EmbPattern *p, EmbReal dx, EmbReal dy, int flags, int isAuto↔ AutoColorIndex)
- EMB_PUBLIC void embPattern_changeColor (EmbPattern *p, int index)
- EMB_PUBLIC void embPattern_free (EmbPattern *p)
- EMB_PUBLIC void embPattern_scale (EmbPattern *p, EmbReal scale)
- EMB_PUBLIC EmbReal embPattern_totalStitchLength (EmbPattern *pattern)
- EMB_PUBLIC EmbReal embPattern_minimumStitchLength (EmbPattern *pattern)
- EMB_PUBLIC EmbReal embPattern_maximumStitchLength (EmbPattern *pattern)
- EMB_PUBLIC void embPattern_lengthHistogram (EmbPattern *pattern, int *bin, int NUMBINS)
- EMB_PUBLIC int embPattern_realStitches (EmbPattern *pattern)
- EMB_PUBLIC int embPattern_jumpStitches (EmbPattern *pattern)
- EMB_PUBLIC int embPattern_trimStitches (EmbPattern *pattern)
- EMB_PUBLIC EmbRect embPattern_calcBoundingBox (EmbPattern *p)
- EMB_PUBLIC void embPattern_flipHorizontal (EmbPattern *p)
- EMB_PUBLIC void embPattern_flipVertical (EmbPattern *p)
- EMB_PUBLIC void embPattern_flip (EmbPattern *p, int horz, int vert)
- EMB_PUBLIC void embPattern_combineJumpStitches (EmbPattern *p)
- EMB_PUBLIC void embPattern_correctForMaxStitchLength (EmbPattern *p, EmbReal maxStitchLength, EmbReal maxJumpLength)
- EMB_PUBLIC void embPattern_center (EmbPattern *p)
- EMB_PUBLIC void embPattern_loadExternalColorFile (EmbPattern *p, const char *fileName)
- EMB_PUBLIC void embPattern_convertGeometry (EmbPattern *p)
- EMB_PUBLIC void embPattern_designDetails (EmbPattern *p)
- EMB_PUBLIC EmbPattern * embPattern_combine (EmbPattern *p1, EmbPattern *p2)

- `EMB_PUBLIC` int `embPattern_color_count` (`EmbPattern` *pattern, `EmbColor` startColor)
- `EMB_PUBLIC` void `embPattern_end` (`EmbPattern` *p)
- `EMB_PUBLIC` void `embPattern_crossstitch` (`EmbPattern` *pattern, `EmbImage` *, int threshold)
- `EMB_PUBLIC` void `embPattern_horizontal_fill` (`EmbPattern` *pattern, `EmbImage` *, int threshold)
- `EMB_PUBLIC` int `embPattern_render` (`EmbPattern` *pattern, char *fname)
- `EMB_PUBLIC` int `embPattern_simulate` (`EmbPattern` *pattern, char *fname)
- `EMB_PUBLIC` void `embPattern_addCircleAbs` (`EmbPattern` *p, `EmbCircle` obj)
- `EMB_PUBLIC` void `embPattern_addEllipseAbs` (`EmbPattern` *p, `EmbEllipse` obj)
- `EMB_PUBLIC` void `embPattern_addLineAbs` (`EmbPattern` *p, `EmbLine` obj)
- `EMB_PUBLIC` void `embPattern_addPathAbs` (`EmbPattern` *p, `EmbPath` obj)
- `EMB_PUBLIC` void `embPattern_addPointAbs` (`EmbPattern` *p, `EmbPoint` obj)
- `EMB_PUBLIC` void `embPattern_addPolygonAbs` (`EmbPattern` *p, `EmbPolygon` obj)
- `EMB_PUBLIC` void `embPattern_addPolylineAbs` (`EmbPattern` *p, `EmbPolyline` obj)
- `EMB_PUBLIC` void `embPattern_addRectAbs` (`EmbPattern` *p, `EmbRect` obj)
- `EMB_PUBLIC` void `embPattern_copyStitchListToPolylines` (`EmbPattern` *pattern)
- `EMB_PUBLIC` void `embPattern_copyPolylinesToStitchList` (`EmbPattern` *pattern)
- `EMB_PUBLIC` void `embPattern_moveStitchListToPolylines` (`EmbPattern` *pattern)
- `EMB_PUBLIC` void `embPattern_movePolylinesToStitchList` (`EmbPattern` *pattern)
- `EMB_PUBLIC` char `embPattern_read` (`EmbPattern` *pattern, const char *fileName, int format)
pattern fileName format
- `EMB_PUBLIC` char `embPattern_write` (`EmbPattern` *pattern, const char *fileName, int format)
pattern fileName format
- `EMB_PUBLIC` char `embPattern_readAuto` (`EmbPattern` *pattern, const char *fileName)
pattern fileName
- `EMB_PUBLIC` char `embPattern_writeAuto` (`EmbPattern` *pattern, const char *fileName)
pattern fileName
- `EMB_PUBLIC` void `report` (int result, char *label)
- `EMB_PUBLIC` int `full_test_matrix` (char *fname)
- `EMB_PUBLIC` int `emb_round` (`EmbReal` x)
- `EMB_PUBLIC` `EmbReal` `radians` (`EmbReal` degree)
- `EMB_PUBLIC` `EmbReal` `degrees` (`EmbReal` radian)

Variables

- `EmbFormatList` `formatTable` [`numberOfFormats`]
- const int `pecThreadCount`
- const int `shvThreadCount`
- const `EmbReal` `embConstantPi`
- const `EmbThread` `husThreads` []
- const `EmbThread` `jefThreads` []
- const `EmbThread` `shvThreads` []
- const `EmbThread` `pcmThreads` []
- const `EmbThread` `pecThreads` []
- const unsigned char `_dxfColorTable` [][3]
- `EmbThread` `black_thread`
- const unsigned char `vipDecodingTable` []
- int `emb_error`
Error code storage for optional control flow blocking.
- int `emb_verbose`
Verbosity level.

18.28.1 Macro Definition Documentation

18.28.1.1 Arc_Polyester `#define Arc_Polyester 0`

18.28.1.2 Arc_Rayon `#define Arc_Rayon 1`

18.28.1.3 CHUNK_SIZE `#define CHUNK_SIZE 128`

18.28.1.4 CoatsAndClark_Rayon `#define CoatsAndClark_Rayon 2`

18.28.1.5 dxf_color `#define dxf_color 29`

18.28.1.6 EMB_ARC `#define EMB_ARC 1`

18.28.1.7 EMB_ARRAY `#define EMB_ARRAY 0`

18.28.1.8 EMB_CIRCLE `#define EMB_CIRCLE 2`

18.28.1.9 EMB_DIM_DIAMETER `#define EMB_DIM_DIAMETER 3`

18.28.1.10 EMB_DIM_LEADER `#define EMB_DIM_LEADER 4`

18.28.1.11 EMB_ELLIPSE `#define EMB_ELLIPSE 5`

18.28.1.12 EMB_FLAG `#define EMB_FLAG 6`

18.28.1.13 EMB_FORMAT_100 `#define EMB_FORMAT_100 0`
Format identifiers

18.28.1.14 EMB_FORMAT_100 `#define EMB_FORMAT_100 1`

18.28.1.15 EMB_FORMAT_ART `#define EMB_FORMAT_ART 2`

18.28.1.16 EMB_FORMAT_BMC `#define EMB_FORMAT_BMC 3`

18.28.1.17 EMB_FORMAT_BRO `#define EMB_FORMAT_BRO 4`

18.28.1.18 EMB_FORMAT_CND `#define EMB_FORMAT_CND 5`

- 18.28.1.19 **EMB_FORMAT_COL** `#define EMB_FORMAT_COL 6`
- 18.28.1.20 **EMB_FORMAT_CSD** `#define EMB_FORMAT_CSD 7`
- 18.28.1.21 **EMB_FORMAT_CSV** `#define EMB_FORMAT_CSV 8`
- 18.28.1.22 **EMB_FORMAT_DAT** `#define EMB_FORMAT_DAT 9`
- 18.28.1.23 **EMB_FORMAT_DEM** `#define EMB_FORMAT_DEM 10`
- 18.28.1.24 **EMB_FORMAT_DSB** `#define EMB_FORMAT_DSB 11`
- 18.28.1.25 **EMB_FORMAT_DST** `#define EMB_FORMAT_DST 12`
- 18.28.1.26 **EMB_FORMAT_DSZ** `#define EMB_FORMAT_DSZ 13`
- 18.28.1.27 **EMB_FORMAT_DXF** `#define EMB_FORMAT_DXF 14`
- 18.28.1.28 **EMB_FORMAT_EDR** `#define EMB_FORMAT_EDR 15`
- 18.28.1.29 **EMB_FORMAT_EMD** `#define EMB_FORMAT_EMD 16`
- 18.28.1.30 **EMB_FORMAT_EXP** `#define EMB_FORMAT_EXP 17`
- 18.28.1.31 **EMB_FORMAT_EXY** `#define EMB_FORMAT_EXY 18`
- 18.28.1.32 **EMB_FORMAT_EYS** `#define EMB_FORMAT_EYS 19`
- 18.28.1.33 **EMB_FORMAT_FXY** `#define EMB_FORMAT_FXY 20`
- 18.28.1.34 **EMB_FORMAT_GC** `#define EMB_FORMAT_GC 21`
- 18.28.1.35 **EMB_FORMAT_GNC** `#define EMB_FORMAT_GNC 22`
- 18.28.1.36 **EMB_FORMAT_GT** `#define EMB_FORMAT_GT 23`

18.28.1.37 EMB_FORMAT_HUS #define EMB_FORMAT_HUS 24

18.28.1.38 EMB_FORMAT_INB #define EMB_FORMAT_INB 25

18.28.1.39 EMB_FORMAT_INF #define EMB_FORMAT_INF 26

18.28.1.40 EMB_FORMAT_JEF #define EMB_FORMAT_JEF 27

18.28.1.41 EMB_FORMAT_KSM #define EMB_FORMAT_KSM 28

18.28.1.42 EMB_FORMAT_MAX #define EMB_FORMAT_MAX 29

18.28.1.43 EMB_FORMAT_MIT #define EMB_FORMAT_MIT 30

18.28.1.44 EMB_FORMAT_NEW #define EMB_FORMAT_NEW 31

18.28.1.45 EMB_FORMAT_OFM #define EMB_FORMAT_OFM 32

18.28.1.46 EMB_FORMAT_PCD #define EMB_FORMAT_PCD 33

18.28.1.47 EMB_FORMAT_PCM #define EMB_FORMAT_PCM 34

18.28.1.48 EMB_FORMAT_PCQ #define EMB_FORMAT_PCQ 35

18.28.1.49 EMB_FORMAT_PCS #define EMB_FORMAT_PCS 36

18.28.1.50 EMB_FORMAT_PEC #define EMB_FORMAT_PEC 37

18.28.1.51 EMB_FORMAT_PEL #define EMB_FORMAT_PEL 38

18.28.1.52 EMB_FORMAT_PEM #define EMB_FORMAT_PEM 39

18.28.1.53 EMB_FORMAT_PES #define EMB_FORMAT_PES 40

18.28.1.54 EMB_FORMAT_PHB #define EMB_FORMAT_PHB 41

- 18.28.1.55 EMB_FORMAT_PHC** #define EMB_FORMAT_PHC 42
- 18.28.1.56 EMB_FORMAT_PLT** #define EMB_FORMAT_PLT 43
- 18.28.1.57 EMB_FORMAT_RGB** #define EMB_FORMAT_RGB 44
- 18.28.1.58 EMB_FORMAT_SEW** #define EMB_FORMAT_SEW 45
- 18.28.1.59 EMB_FORMAT_SHV** #define EMB_FORMAT_SHV 46
- 18.28.1.60 EMB_FORMAT_SST** #define EMB_FORMAT_SST 47
- 18.28.1.61 EMB_FORMAT_STX** #define EMB_FORMAT_STX 48
- 18.28.1.62 EMB_FORMAT_SVG** #define EMB_FORMAT_SVG 49
- 18.28.1.63 EMB_FORMAT_T01** #define EMB_FORMAT_T01 50
- 18.28.1.64 EMB_FORMAT_T09** #define EMB_FORMAT_T09 51
- 18.28.1.65 EMB_FORMAT_TAP** #define EMB_FORMAT_TAP 52
- 18.28.1.66 EMB_FORMAT_THR** #define EMB_FORMAT_THR 53
- 18.28.1.67 EMB_FORMAT_TXT** #define EMB_FORMAT_TXT 54
- 18.28.1.68 EMB_FORMAT_U00** #define EMB_FORMAT_U00 55
- 18.28.1.69 EMB_FORMAT_U01** #define EMB_FORMAT_U01 56
- 18.28.1.70 EMB_FORMAT_VIP** #define EMB_FORMAT_VIP 57
- 18.28.1.71 EMB_FORMAT_VP3** #define EMB_FORMAT_VP3 58
- 18.28.1.72 EMB_FORMAT_XXX** #define EMB_FORMAT_XXX 59

18.28.1.73 EMB_FORMAT_ZSK `#define EMB_FORMAT_ZSK 60`

18.28.1.74 EMB_IMAGE `#define EMB_IMAGE 8`

18.28.1.75 EMB_LINE `#define EMB_LINE 7`

18.28.1.76 EMB_MAX_LAYERS `#define EMB_MAX_LAYERS 10`

18.28.1.77 EMB_PATH `#define EMB_PATH 9`

18.28.1.78 EMB_POINT `#define EMB_POINT 10`

18.28.1.79 EMB_POLYGON `#define EMB_POLYGON 11`

18.28.1.80 EMB_POLYLINE `#define EMB_POLYLINE 12`

18.28.1.81 EMB_PUBLIC `#define EMB_PUBLIC`

18.28.1.82 EMB_RECT `#define EMB_RECT 13`

18.28.1.83 EMB_SPLINE `#define EMB_SPLINE 14`

18.28.1.84 EMB_STITCH `#define EMB_STITCH 15`

18.28.1.85 EMB_TEXT_MULTI `#define EMB_TEXT_MULTI 17`

18.28.1.86 EMB_TEXT_SINGLE `#define EMB_TEXT_SINGLE 16`

18.28.1.87 EMB_THREAD `#define EMB_THREAD 19`

18.28.1.88 EMB_VECTOR `#define EMB_VECTOR 18`

18.28.1.89 EMBFORMAT_MAXDESC `#define EMBFORMAT_MAXDESC 50`

18.28.1.90 EMBFORMAT_MAXEXT `#define EMBFORMAT_MAXEXT 3`

18.28.1.91 EMBFORMAT_OBJECTONLY #define EMBFORMAT_OBJECTONLY 2

18.28.1.92 EMBFORMAT_STCHANDOBJ #define EMBFORMAT_STCHANDOBJ 3 /* binary operation↵
: 1+2=3 */

18.28.1.93 EMBFORMAT_STITCHONLY #define EMBFORMAT_STITCHONLY 1

18.28.1.94 EMBFORMAT_UNSUPPORTED #define EMBFORMAT_UNSUPPORTED 0

18.28.1.95 END #define END 16 /*! end of program */

18.28.1.96 Exquisite_Polyester #define Exquisite_Polyester 3

18.28.1.97 Fufu_Polyester #define Fufu_Polyester 4

18.28.1.98 Fufu_Rayon #define Fufu_Rayon 5

18.28.1.99 Hemingworth_Polyester #define Hemingworth_Polyester 6

18.28.1.100 hus_thread #define hus_thread 24

18.28.1.101 Isacord_Polyester #define Isacord_Polyester 7

18.28.1.102 Isafil_Rayon #define Isafil_Rayon 8

18.28.1.103 jef_thread #define jef_thread 25

18.28.1.104 JUMP #define JUMP 1 /*! move to (x, y) */

18.28.1.105 LIBEMBROIDERY_EMBEDDED_VERSION #define LIBEMBROIDERY_EMBEDDED_VERSION 0

18.28.1.106 Madeira_Polyester #define Madeira_Polyester 11

18.28.1.107 Madeira_Rayon #define Madeira_Rayon 12

18.28.1.108 Marathon_Polyester #define Marathon_Polyester 9

18.28.1.109 Marathon_Rayon `#define Marathon_Rayon 10`

18.28.1.110 MAX_STITCHES `#define MAX_STITCHES 1000000`

18.28.1.111 MAX_THREADS `#define MAX_THREADS 256`

18.28.1.112 Metro_Polyester `#define Metro_Polyester 13`

18.28.1.113 NORMAL `#define NORMAL 0 /*! stitch to (x, y) */`
Machine codes for stitch flags

18.28.1.114 numberOfFormats `#define numberOfFormats 61`

18.28.1.115 Pantone `#define Pantone 14`

18.28.1.116 pcm_thread `#define pcm_thread 26`

18.28.1.117 pec_thread `#define pec_thread 27`

18.28.1.118 RobisonAnton_Polyester `#define RobisonAnton_Polyester 15`

18.28.1.119 RobisonAnton_Rayon `#define RobisonAnton_Rayon 16`

18.28.1.120 SEQUIN `#define SEQUIN 8 /*! sequin */`

18.28.1.121 shv_thread `#define shv_thread 28`

18.28.1.122 Sigma_Polyester `#define Sigma_Polyester 17`

18.28.1.123 STOP `#define STOP 4 /*! pause machine for thread change */`

18.28.1.124 Sulky_Rayon `#define Sulky_Rayon 18`

18.28.1.125 SVG_Colors `#define SVG_Colors 23`

18.28.1.126 ThreadArt_Polyester `#define ThreadArt_Polyester 20`

18.28.1.127 ThreadArt_Rayon `#define ThreadArt_Rayon 19`

18.28.1.128 ThreaDelight_Polyester `#define ThreaDelight_Polyester 21`

18.28.1.129 TRIM `#define TRIM 2 /*! trim + move to (x, y) */`

18.28.1.130 Z102_Isacord_Polyester `#define Z102_Isacord_Polyester 22`

18.28.2 Typedef Documentation

18.28.2.1 EmbAlignedDim `typedef struct EmbAlignedDim_ EmbAlignedDim`

18.28.2.2 EmbAngularDim `typedef struct EmbAngularDim_ EmbAngularDim`

18.28.2.3 EmbArc `typedef struct EmbArc_ EmbArc`
absolute position (not relative)

18.28.2.4 EmbArcLengthDim `typedef struct EmbArcLengthDim_ EmbArcLengthDim`

18.28.2.5 EmbArray `typedef struct EmbArray_ EmbArray`
The basic array type.

18.28.2.6 EmbBezier `typedef struct EmbBezier_ EmbBezier`

18.28.2.7 EmbBlock `typedef struct EmbBlock_ EmbBlock`

18.28.2.8 EmbCircle `typedef struct EmbCircle_ EmbCircle`

18.28.2.9 EmbColor `typedef struct EmbColor_ EmbColor`
EmbColor uses the light primaries: red, green, blue in that order.

18.28.2.10 EmbDiameterDim `typedef struct EmbDiameterDim_ EmbDiameterDim`

18.28.2.11 EmbEllipse `typedef struct EmbEllipse_ EmbEllipse`

18.28.2.12 EmbFlag `typedef int EmbFlag`

18.28.2.13 EmbFormatList `typedef struct EmbFormatList_ EmbFormatList`

18.28.2.14 EmbGeometry typedef struct [EmbGeometry_](#) EmbGeometry

18.28.2.15 EmbImage typedef struct [EmbImage_](#) EmbImage

18.28.2.16 EmbInfiniteLine typedef struct [EmbInfiniteLine_](#) EmbInfiniteLine

18.28.2.17 EmbLayer typedef struct [EmbLayer_](#) EmbLayer

18.28.2.18 EmbLeaderDim typedef struct [EmbLeaderDim_](#) EmbLeaderDim

18.28.2.19 EmbLine typedef struct [EmbLine_](#) EmbLine

18.28.2.20 EmbLinearDim typedef struct [EmbLinearDim_](#) EmbLinearDim

18.28.2.21 EmbOrdinateDim typedef struct [EmbOrdinateDim_](#) EmbOrdinateDim

18.28.2.22 EmbPath typedef struct [EmbPath_](#) EmbPath

18.28.2.23 EmbPattern typedef struct [EmbPattern_](#) EmbPattern

18.28.2.24 EmbPoint typedef struct [EmbPoint_](#) EmbPoint

18.28.2.25 EmbPolygon typedef [EmbPath](#) EmbPolygon

18.28.2.26 EmbPolyline typedef [EmbPath](#) EmbPolyline

18.28.2.27 EmbRadiusDim typedef struct [EmbRadiusDim_](#) EmbRadiusDim

18.28.2.28 EmbRay typedef struct [EmbRay_](#) EmbRay

18.28.2.29 EmbReal typedef float [EmbReal](#)

18.28.2.30 EmbRect typedef struct [EmbRect_](#) EmbRect

18.28.2.31 EmbSatinOutline typedef struct [EmbSatinOutline_](#) EmbSatinOutline

18.28.2.32 EmbSpline typedef struct [EmbSpline_](#) EmbSpline

18.28.2.33 EmbStitch typedef struct [EmbStitch_](#) EmbStitch

18.28.2.34 EmbTextMulti typedef struct [EmbTextMulti_](#) EmbTextMulti

18.28.2.35 EmbTextSingle typedef struct [EmbTextSingle_](#) EmbTextSingle

18.28.2.36 EmbThread typedef struct [EmbThread_](#) EmbThread

18.28.2.37 EmbTime typedef struct [EmbTime_](#) EmbTime

18.28.2.38 EmbVector typedef struct [EmbVector_](#) EmbVector

The basic type to represent points absolutely or represent directions.
Positive y is up, units are in mm.

18.28.2.39 L_system typedef struct [LSYSTEM](#) L_system

18.28.2.40 thread_color typedef struct [thread_color_](#) thread_color

18.28.3 Function Documentation

18.28.3.1 convert() [EMB_PUBLIC](#) int convert (
const char * *inf*,
const char * *outf*)

18.28.3.2 degrees() [EMB_PUBLIC](#) EmbReal degrees (
[EmbReal](#) *radian*)

18.28.3.3 emb_identify_format() [EMB_PUBLIC](#) int emb_identify_format (
const char * *fileName*)

fileName

Returns

int

18.28.3.4 emb_round() [EMB_PUBLIC](#) int emb_round (
[EmbReal](#) *x*)

18.28.3.5 embArc_clockwise() [EMB_PUBLIC](#) char embArc_clockwise (
[EmbArc](#) *arc*)

18.28.3.6 `embArc_init()` `EMB_PUBLIC EmbArc embArc_init (`
`void)`

18.28.3.7 `embArray_addArc()` `EMB_PUBLIC int embArray_addArc (`
`EmbArray * a,`
`EmbArc b)`

Add an arc *b* to the EmbArray *a* and it returns if the element was successfully added.

18.28.3.8 `embArray_addCircle()` `EMB_PUBLIC int embArray_addCircle (`
`EmbArray * a,`
`EmbCircle b)`

Add a circle *b* to the EmbArray *a* and it returns if the element was successfully added.

18.28.3.9 `embArray_addEllipse()` `EMB_PUBLIC int embArray_addEllipse (`
`EmbArray * a,`
`EmbEllipse b)`

Add an ellipse *b* to the EmbArray *a* and it returns if the element was successfully added.

18.28.3.10 `embArray_addFlag()` `EMB_PUBLIC int embArray_addFlag (`
`EmbArray * a,`
`EmbFlag b)`

Add a flag *b* to the EmbArray *a* and it returns if the element was successfully added.

18.28.3.11 `embArray_addLine()` `EMB_PUBLIC int embArray_addLine (`
`EmbArray * a,`
`EmbLine b)`

Add a line *b* to the EmbArray *a* and it returns if the element was successfully added.

18.28.3.12 `embArray_addPath()` `EMB_PUBLIC int embArray_addPath (`
`EmbArray * a,`
`EmbPath b)`

Add a path *b* to the EmbArray *a* and it returns if the element was successfully added.

18.28.3.13 `embArray_addPoint()` `EMB_PUBLIC int embArray_addPoint (`
`EmbArray * a,`
`EmbPoint b)`

Add a point *b* to the EmbArray *a* and it returns if the element was successfully added.

18.28.3.14 `embArray_addPolygon()` `EMB_PUBLIC int embArray_addPolygon (`
`EmbArray * a,`
`EmbPolygon b)`

Add a polygon *b* to the EmbArray *a* and it returns if the element was successfully added.

18.28.3.15 `embArray_addPolyline()` `EMB_PUBLIC int embArray_addPolyline (`
`EmbArray * a,`
`EmbPolyline b)`

Add a polyline *b* to the EmbArray *a* and it returns if the element was successfully added.

18.28.3.16 `embArray_addRect()` `EMB_PUBLIC int embArray_addRect (`
`EmbArray * a,`
`EmbRect b)`

Add a rectangle *b* to the EmbArray *a* and it returns if the element was successfully added.

18.28.3.17 `embArray_addStitch()` `EMB_PUBLIC` `int` `embArray_addStitch` (
`EmbArray` * `a`,
`EmbStitch` `b`)

Add a stitch `b` to the `EmbArray` `a` and it returns if the element was successfully added.

18.28.3.18 `embArray_addThread()` `EMB_PUBLIC` `int` `embArray_addThread` (
`EmbArray` * `g`,
`EmbThread` `p`)

18.28.3.19 `embArray_addVector()` `EMB_PUBLIC` `int` `embArray_addVector` (
`EmbArray` * `a`,
`EmbVector` `b`)

Add a vector `b` to the `EmbArray` `a` and it returns if the element was successfully added.

18.28.3.20 `embArray_copy()` `EMB_PUBLIC` `void` `embArray_copy` (
`EmbArray` * `dst`,
`EmbArray` * `src`)

Copies all entries in the `EmbArray` struct from `src` to `dst`.

18.28.3.21 `embArray_create()` `EMB_PUBLIC` `EmbArray` * `embArray_create` (
`int` `type`)

Allocates memory for an `EmbArray` of the type determined by the argument `type`.

18.28.3.22 `embArray_free()` `EMB_PUBLIC` `void` `embArray_free` (
`EmbArray` * `a`)

Free the memory of `EmbArray` `a`, recursively if necessary.

18.28.3.23 `embArray_resize()` `EMB_PUBLIC` `int` `embArray_resize` (
`EmbArray` * `a`)

Resizes the array `a` to be `CHUNK_SIZE` entries longer if and only if the amount of room left is less than 3 entries.

18.28.3.24 `embCircle_init()` `EMB_PUBLIC` `EmbCircle` `embCircle_init` (
`void`)

18.28.3.25 `embColor_create()` `EMB_PUBLIC` `EmbColor` * `embColor_create` (
`unsigned char` `r`,
`unsigned char` `g`,
`unsigned char` `b`)

18.28.3.26 `embColor_distance()` `EMB_PUBLIC` `int` `embColor_distance` (
`EmbColor` `a`,
`EmbColor` `b`)

`a b`

Returns

`int`

18.28.3.27 `embColor_fromHexStr()` `EMB_PUBLIC` `EmbColor` `embColor_fromHexStr` (
`char` * `val`)

Converts a 6 digit hex string (I.E. "00FF00") into an `EmbColor` and returns it.

`val` 6 byte code describing the color as a hex string, doesn't require null termination.

Returns

EmbColor the same color as our internal type.

18.28.3.28 embColor_make() `EMB_PUBLIC EmbColor embColor_make (`
 `unsigned char r,`
 `unsigned char g,`
 `unsigned char b)`

18.28.3.29 embEllipse_area() `EMB_PUBLIC EmbReal embEllipse_area (`
 `EmbEllipse ellipse)`

18.28.3.30 embEllipse_diameterX() `EMB_PUBLIC EmbReal embEllipse_diameterX (`
 `EmbEllipse ellipse)`

18.28.3.31 embEllipse_diameterY() `EMB_PUBLIC EmbReal embEllipse_diameterY (`
 `EmbEllipse ellipse)`

18.28.3.32 embEllipse_height() `EMB_PUBLIC EmbReal embEllipse_height (`
 `EmbEllipse ellipse)`

18.28.3.33 embEllipse_init() `EMB_PUBLIC EmbEllipse embEllipse_init (`
 `void)`

18.28.3.34 embEllipse_make() `EMB_PUBLIC EmbEllipse embEllipse_make (`
 `EmbReal cx,`
 `EmbReal cy,`
 `EmbReal rx,`
 `EmbReal ry)`

18.28.3.35 embEllipse_perimeter() `EMB_PUBLIC EmbReal embEllipse_perimeter (`
 `EmbEllipse ellipse)`

18.28.3.36 embEllipse_width() `EMB_PUBLIC EmbReal embEllipse_width (`
 `EmbEllipse ellipse)`

18.28.3.37 embGeometry_boundingRect() `EMB_PUBLIC EmbRect embGeometry_boundingRect (`
 `EmbGeometry * obj)`

Calculate the bounding box of geometry *obj* based on what kind of geometric object it is.
obj A pointer to the geometry memory.

Returns

EmbRect The bounding box in the same scale as the input geometry.

In the case of a failure the bounding box returned is always the unit square with top left corner at (0, 0).

18.28.3.38 embGeometry_free() `EMB_PUBLIC void embGeometry_free (EmbGeometry * obj)`

Free the memory occupied by a non-stitch geometry object.
obj Pointer to geometry memory.

18.28.3.39 embGeometry_init() `EMB_PUBLIC EmbGeometry * embGeometry_init (int type_in)`

Our generic object interface backends to each individual type.
type_in

Returns

EmbGeometry*

18.28.3.40 embGeometry_move() `EMB_PUBLIC void embGeometry_move (EmbGeometry * obj, EmbVector delta)`

Translate *obj* by the vector *delta*.
obj A pointer to the geometry memory. *delta* A vector in the 0.1mm scale to offset the geometry by.

18.28.3.41 embGeometry_vulcanize() `EMB_PUBLIC void embGeometry_vulcanize (EmbGeometry * obj)`

Toggle the rubber mode of the object.
obj

Todo Review. This could be controlled by a simple flag.

18.28.3.42 embImage_create() `EMB_PUBLIC EmbImage embImage_create (int , int)`

18.28.3.43 embImage_free() `EMB_PUBLIC void embImage_free (EmbImage * image)`

18.28.3.44 embImage_read() `EMB_PUBLIC void embImage_read (EmbImage * image, char * fname)`

18.28.3.45 embImage_write() `EMB_PUBLIC int embImage_write (EmbImage * image, char * fname)`

18.28.3.46 embLine_intersectionPoint() `EMB_PUBLIC EmbVector embLine_intersectionPoint (EmbLine line1, EmbLine line2)`

18.28.3.47 embLine_make() `EMB_PUBLIC EmbLine embLine_make (`
`EmbReal x1,`
`EmbReal y1,`
`EmbReal x2,`
`EmbReal y2)`

18.28.3.48 embLine_normalVector() `EMB_PUBLIC void embLine_normalVector (`
`EmbLine line,`
`EmbVector * result,`
`int clockwise)`

Finds the normalized vector perpendicular (clockwise) to the line given by v1->v2 (normal to the line)

18.28.3.49 embPattern_addCircleAbs() `EMB_PUBLIC void embPattern_addCircleAbs (`
`EmbPattern * p,`
`EmbCircle circle)`

Adds a circle object to pattern (*p*) with its center at the absolute position (*cx,cy*) with a radius of (*r*). Positive y is up. Units are in millimeters.

18.28.3.50 embPattern_addEllipseAbs() `EMB_PUBLIC void embPattern_addEllipseAbs (`
`EmbPattern * p,`
`EmbEllipse ellipse)`

Adds an ellipse object to pattern (*p*) with its center at the absolute position (*cx,cy*) with radii of (*rx,ry*). Positive y is up. Units are in millimeters.

18.28.3.51 embPattern_addLineAbs() `EMB_PUBLIC void embPattern_addLineAbs (`
`EmbPattern * p,`
`EmbLine line)`

Adds a line object to pattern (*p*) starting at the absolute position (*x1,y1*) and ending at the absolute position (*x2,y2*). Positive y is up. Units are in millimeters.

18.28.3.52 embPattern_addPathAbs() `EMB_PUBLIC void embPattern_addPathAbs (`
`EmbPattern * p,`
`EmbPath obj)`

18.28.3.53 embPattern_addPointAbs() `EMB_PUBLIC void embPattern_addPointAbs (`
`EmbPattern * p,`
`EmbPoint obj)`

Adds a point object to pattern (*p*) at the absolute position (*x,y*). Positive y is up. Units are in millimeters.

18.28.3.54 embPattern_addPolygonAbs() `EMB_PUBLIC void embPattern_addPolygonAbs (`
`EmbPattern * p,`
`EmbPolygon obj)`

18.28.3.55 embPattern_addPolylineAbs() `EMB_PUBLIC void embPattern_addPolylineAbs (`
`EmbPattern * p,`
`EmbPolyline obj)`

18.28.3.56 embPattern_addRectAbs() `EMB_PUBLIC void embPattern_addRectAbs (`
`EmbPattern * p,`
`EmbRect rect)`

Adds a rectangle object to pattern (*p*) at the absolute position (*x,y*) with a width of (*w*) and a height of (*h*). Positive y is up. Units are in millimeters.

18.28.3.57 embPattern_addStitchAbs() `EMB_PUBLIC void embPattern_addStitchAbs (`
`EmbPattern * p,`
`EmbReal x,`
`EmbReal y,`
`int flags,`
`int isAutoColorIndex)`

Adds a stitch to the pattern (*p*) at the absolute position (*x,y*). Positive *y* is up. Units are in millimeters.

18.28.3.58 embPattern_addStitchRel() `EMB_PUBLIC void embPattern_addStitchRel (`
`EmbPattern * p,`
`EmbReal dx,`
`EmbReal dy,`
`int flags,`
`int isAutoColorIndex)`

Adds a stitch to the pattern (*p*) at the relative position (*dx,dy*) to the previous stitch. Positive *y* is up. Units are in millimeters.

18.28.3.59 embPattern_addThread() `EMB_PUBLIC int embPattern_addThread (`
`EmbPattern * pattern,`
`EmbThread thread)`

pattern thread

Returns

int

18.28.3.60 embPattern_calcBoundingBox() `EMB_PUBLIC EmbRect embPattern_calcBoundingBox (`
`EmbPattern * p)`

Returns an EmbRect that encapsulates all stitches and objects in the pattern (*p*).

18.28.3.61 embPattern_center() `EMB_PUBLIC void embPattern_center (`
`EmbPattern * p)`

Center the pattern *p*.

18.28.3.62 embPattern_changeColor() `EMB_PUBLIC void embPattern_changeColor (`
`EmbPattern * p,`
`int index)`

Change the currentColorIndex of pattern *p* to *index*.

18.28.3.63 embPattern_color_count() `EMB_PUBLIC int embPattern_color_count (`
`EmbPattern * pattern,`
`EmbColor startColor)`

18.28.3.64 embPattern_combine() `EMB_PUBLIC EmbPattern * embPattern_combine (`
`EmbPattern * p1,`
`EmbPattern * p2)`

p1 p2

Returns

EmbPattern*

18.28.3.65 embPattern_combineJumpStitches() `EMB_PUBLIC void embPattern_combineJumpStitches (`
`EmbPattern * p)`

p

18.28.3.66 `embPattern_convertGeometry()` `EMB_PUBLIC` void `embPattern_convertGeometry` (
`EmbPattern * p`)
p

18.28.3.67 `embPattern_copyPolylinesToStitchList()` `EMB_PUBLIC` void `embPattern_copyPolylinesTo↵`
`StitchList` (
`EmbPattern * pattern`)

18.28.3.68 `embPattern_copyStitchListToPolylines()` `EMB_PUBLIC` void `embPattern_copyStitchListTo↵`
`Polylines` (
`EmbPattern * pattern`)

18.28.3.69 `embPattern_correctForMaxStitchLength()` `EMB_PUBLIC` void `embPattern_correctForMax↵`
`StitchLength` (
`EmbPattern * p`,
`EmbReal maxStitchLength`,
`EmbReal maxJumpLength`)

Todo The params determine the max XY movement rather than the length. They need renamed or clarified further.

18.28.3.70 `embPattern_create()` `EMB_PUBLIC` `EmbPattern *` `embPattern_create` (
`void`)

Returns a pointer to an `EmbPattern`. It is created on the heap. The caller is responsible for freeing the allocated memory with `embPattern_free()`.

Returns

`EmbPattern*`

18.28.3.71 `embPattern_crossstitch()` `EMB_PUBLIC` void `embPattern_crossstitch` (
`EmbPattern * pattern`,
`EmbImage * image`,
`int threshold`)

pattern image threshold

Uses a threshold method to determine where to put crosses in the fill.

To improve this, we can remove the vertical stitches when two crosses neighbour. Currently the simple way to do this is to chain crosses that are neighbours exactly one ahead.

18.28.3.72 `embPattern_designDetails()` `EMB_PUBLIC` void `embPattern_designDetails` (
`EmbPattern * p`)

18.28.3.73 `embPattern_end()` `EMB_PUBLIC` void `embPattern_end` (
`EmbPattern * p`)

18.28.3.74 `embPattern_fixColorCount()` `EMB_PUBLIC` void `embPattern_fixColorCount` (
`EmbPattern * p`)

p

18.28.3.75 embPattern_flip() `EMB_PUBLIC void embPattern_flip (`
`EmbPattern * p,`
`int horz,`
`int vert)`

Flips the entire pattern (*p*) horizontally about the x-axis if (*horz*) is true. Flips the entire pattern (*p*) vertically about the y-axis if (*vert*) is true.

18.28.3.76 embPattern_flipHorizontal() `EMB_PUBLIC void embPattern_flipHorizontal (`
`EmbPattern * p)`

Flips the entire pattern (*p*) horizontally about the y-axis.

18.28.3.77 embPattern_flipVertical() `EMB_PUBLIC void embPattern_flipVertical (`
`EmbPattern * p)`

Flips the entire pattern (*p*) vertically about the x-axis.

18.28.3.78 embPattern_free() `EMB_PUBLIC void embPattern_free (`
`EmbPattern * p)`

Frees all memory allocated in the pattern (*p*).

18.28.3.79 embPattern_hideStitchesOverLength() `EMB_PUBLIC void embPattern_hideStitchesOver↵`
`Length (`
`EmbPattern * p,`
`int length)`

p length

18.28.3.80 embPattern_horizontal_fill() `EMB_PUBLIC void embPattern_horizontal_fill (`
`EmbPattern * pattern,`
`EmbImage * image,`
`int threshold)`

pattern image threshold

Uses a threshold method to determine where to put lines in the fill.

Needs to pass a "donut test", i.e. an image with black pixels where: $10 < x*x + y*y < 20$ over the area $(-30, 30) \times (-30, 30)$.

Use render then image difference to see how well it passes.

18.28.3.81 embPattern_jumpStitches() `EMB_PUBLIC int embPattern_jumpStitches (`
`EmbPattern * pattern)`

18.28.3.82 embPattern_lengthHistogram() `EMB_PUBLIC void embPattern_lengthHistogram (`
`EmbPattern * pattern,`
`int * bin,`
`int NUMBINS)`

18.28.3.83 embPattern_loadExternalColorFile() `EMB_PUBLIC void embPattern_loadExternalColorFile (`
`EmbPattern * p,`
`const char * fileName)`

TODO: Description needed.

18.28.3.84 embPattern_maximumStitchLength() `EMB_PUBLIC EmbReal embPattern_maximumStitchLength`
`(`
`EmbPattern * pattern)`

18.28.3.85 `embPattern_minimumStitchLength()` `EMB_PUBLIC` `EmbReal` `embPattern_minimumStitchLength` (
 `EmbPattern` * *pattern*)

18.28.3.86 `embPattern_movePolylinesToStitchList()` `EMB_PUBLIC` `void` `embPattern_movePolylinesToStitchList` (
 `EmbPattern` * *pattern*)

18.28.3.87 `embPattern_moveStitchListToPolylines()` `EMB_PUBLIC` `void` `embPattern_moveStitchListToPolylines` (
 `EmbPattern` * *pattern*)

18.28.3.88 `embPattern_read()` `EMB_PUBLIC` `char` `embPattern_read` (
 `EmbPattern` * *pattern*,
 const `char` * *fileName*,
 int *format*)

pattern fileName format

Returns

`char`

18.28.3.89 `embPattern_readAuto()` `EMB_PUBLIC` `char` `embPattern_readAuto` (
 `EmbPattern` * *pattern*,
 const `char` * *fileName*)

pattern fileName

Returns

`char`

18.28.3.90 `embPattern_realStitches()` `EMB_PUBLIC` `int` `embPattern_realStitches` (
 `EmbPattern` * *pattern*)

18.28.3.91 `embPattern_render()` `EMB_PUBLIC` `int` `embPattern_render` (
 `EmbPattern` * *pattern*,
 `char` * *fname*)

18.28.3.92 `embPattern_scale()` `EMB_PUBLIC` `void` `embPattern_scale` (
 `EmbPattern` * *p*,
 `EmbReal` *scale*)

Very simple scaling of the x and y axis for every point. Doesn't insert or delete stitches to preserve density.

18.28.3.93 `embPattern_simulate()` `EMB_PUBLIC` `int` `embPattern_simulate` (
 `EmbPattern` * *pattern*,
 `char` * *fname*)

18.28.3.94 `embPattern_totalStitchLength()` `EMB_PUBLIC` `EmbReal` `embPattern_totalStitchLength` (
`EmbPattern * pattern`)

pattern

Returns

float

18.28.3.95 `embPattern_trimStitches()` `EMB_PUBLIC` `int` `embPattern_trimStitches` (
`EmbPattern * pattern`)

18.28.3.96 `embPattern_write()` `EMB_PUBLIC` `char` `embPattern_write` (
`EmbPattern * pattern`,
`const char * fileName`,
`int format`)

pattern fileName format

Returns

char

18.28.3.97 `embPattern_writeAuto()` `EMB_PUBLIC` `char` `embPattern_writeAuto` (
`EmbPattern * pattern`,
`const char * fileName`)

pattern fileName

Returns

char

18.28.3.98 `embRect_area()` `EMB_PUBLIC` `EmbReal` `embRect_area` (
`EmbRect rect`)

18.28.3.99 `embRect_init()` `EMB_PUBLIC` `EmbRect` `embRect_init` (
`void`)

18.28.3.100 `embSatinOutline_generateSatinOutline()` `EMB_PUBLIC` `void` `embSatinOutline_generate`↵
`SatinOutline` (
`EmbArray * lines`,
`EmbReal thickness`,
`EmbSatinOutline * result`)

lines thickness result

18.28.3.101 `embSatinOutline_renderStitches()` `EMB_PUBLIC` `EmbArray *` `embSatinOutline_render`↵
`Stitches` (
`EmbSatinOutline * result`,
`EmbReal density`)

result density

Returns

EmbArray*

18.28.3.102 embThread_findNearestColor() `EMB_PUBLIC int embThread_findNearestColor (`
`EmbColor color,`
`EmbColor * color_list,`
`int n_colors)`

Returns the closest color to the required color based on a list of available threads. The algorithm is a simple least squares search against the list. If the (square of) Euclidean 3-dimensional distance between the points in (red, green, blue) space is smaller then the index is saved and the remaining index is returned to the caller.

color The EmbColor color to match. *colors* The EmbThreadList pointer to start the search at. *mode* Is the argument an array of threads (0) or colors (1)?

Returns

closestIndex The entry in the ThreadList that matches.

18.28.3.103 embThread_findNearestThread() `EMB_PUBLIC int embThread_findNearestThread (`
`EmbColor color,`
`EmbThread * thread_list,`
`int n_threads)`

color thread_list n_threads

Returns

int

18.28.3.104 embThread_getRandom() `EMB_PUBLIC EmbThread embThread_getRandom (`
`void)`

Returns a random thread color, useful in filling in cases where the actual color of the thread doesn't matter but one needs to be declared to test or render a pattern.

Returns

c The resulting color.

18.28.3.105 embTime_initNow() `EMB_PUBLIC void embTime_initNow (`
`EmbTime * t)`

t

18.28.3.106 embTime_time() `EMB_PUBLIC EmbTime embTime_time (`
`EmbTime * t)`

t

Returns

EmbTime

18.28.3.107 embVector_add() `EMB_PUBLIC EmbVector embVector_add (`
`EmbVector a,`
`EmbVector b)`

The sum of vectors *a* and *b* returned as a vector.

Equivalent to:

$$\mathbf{c} = \mathbf{a} + \mathbf{b} = \begin{pmatrix} a_x + b_x \\ a_y + b_y \end{pmatrix}$$

18.28.3.108 embVector_angle() `EMB_PUBLIC EmbReal embVector_angle (EmbVector v)`

The angle, measured anti-clockwise from the x-axis, of a vector *v*.

18.28.3.109 embVector_average() `EMB_PUBLIC EmbVector embVector_average (EmbVector a, EmbVector b)`

The average of vectors *v1* and *v2* returned as a vector.

Equivalent to:

$$\mathbf{c} = \frac{\mathbf{a} + \mathbf{b}}{2} = \begin{pmatrix} \frac{a_x + b_x}{2} \\ \frac{a_y + b_y}{2} \end{pmatrix}$$

18.28.3.110 embVector_cross() `EMB_PUBLIC EmbReal embVector_cross (EmbVector a, EmbVector b)`

The "cross product" as vectors *a* and *b* returned as a real value.

Technically, this is the magnitude of the cross product when the embroidery is placed in the *z*=0 plane (since the cross product is defined for 3-dimensional vectors). That is:

$$|c| = \left| \begin{pmatrix} a_x \\ a_y \\ 0 \end{pmatrix} \times \begin{pmatrix} b_x \\ b_y \\ 0 \end{pmatrix} \right| = \left| \begin{pmatrix} 0 \\ 0 \\ a_x b_y - a_y b_x \end{pmatrix} \right| = a_x b_y - a_y b_x$$

18.28.3.111 embVector_distance() `EMB_PUBLIC EmbReal embVector_distance (EmbVector a, EmbVector b)`

The distance between *a* and *b* returned as a real value.

$$d = |\mathbf{a} - \mathbf{b}| = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}$$

18.28.3.112 embVector_dot() `EMB_PUBLIC EmbReal embVector_dot (EmbVector a, EmbVector b)`

The dot product as vectors *v1* and *v2* returned as a *EmbReal*.

Equivalent to:

$$c = \mathbf{a} \cdot \mathbf{b} = a_x b_x + a_y b_y$$

18.28.3.113 embVector_length() `EMB_PUBLIC EmbReal embVector_length (EmbVector vector)`

The length or absolute value of the vector *vector*.

Equivalent to:

$$|v| = \sqrt{v_x^2 + v_y^2}$$

18.28.3.114 embVector_multiply() `EMB_PUBLIC void embVector_multiply (EmbVector vector, EmbReal magnitude, EmbVector * result)`

The scalar multiple *magnitude* of a vector *vector*. Returned as *result*.

Todo make result return argument.

18.28.3.115 embVector_normalize() `EMB_PUBLIC void embVector_normalize (`
`EmbVector vector,`
`EmbVector * result)`

Finds the unit length vector *result* in the same direction as *vector*.

Equivalent to:

$$\mathbf{u} = \frac{\mathbf{v}}{|\mathbf{v}|}$$

Todo make result return argument.

18.28.3.116 embVector_relativeX() `EMB_PUBLIC EmbReal embVector_relativeX (`
`EmbVector a1,`
`EmbVector a2,`
`EmbVector a3)`

The x-component of the vector

18.28.3.117 embVector_relativeY() `EMB_PUBLIC EmbReal embVector_relativeY (`
`EmbVector a1,`
`EmbVector a2,`
`EmbVector a3)`

The y-component of the vector

18.28.3.118 embVector_subtract() `EMB_PUBLIC EmbVector embVector_subtract (`
`EmbVector v1,`
`EmbVector v2)`

The difference between vectors *v1* and *v2* returned as *result*.

Equivalent to:

$$\mathbf{c} = \mathbf{a} - \mathbf{b} = \begin{pmatrix} a_x - b_x \\ a_y - b_y \end{pmatrix}$$

18.28.3.119 embVector_transpose_product() `EMB_PUBLIC void embVector_transpose_product (`
`EmbVector v1,`
`EmbVector v2,`
`EmbVector * result)`

Since we aren't using full vector algebra here, all vectors are "vertical". so this is like the product $\mathbf{v1}^T \mathbf{v2}$ for our vectors *v1* and *v2* so a "component-wise product". The result is stored at the pointer *result*.

That is $\begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}$

18.28.3.120 embVector_unit() `EMB_PUBLIC EmbVector embVector_unit (`
`EmbReal alpha)`

The unit vector in the direction *angle*.

$$\mathbf{a}_\alpha = \begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \end{pmatrix}$$

18.28.3.121 full_test_matrix() `EMB_PUBLIC int full_test_matrix (`
`char * fname)`

18.28.3.122 getArcCenter() `EMB_PUBLIC void getArcCenter (`
`EmbArc arc,`
`EmbVector * arcCenter)`

18.28.3.123 getArcDataFromBulge() `EMB_PUBLIC char getArcDataFromBulge (`
`EmbReal bulge,`
`EmbArc * arc,`
`EmbReal * arcCenterX,`
`EmbReal * arcCenterY,`
`EmbReal * radius,`
`EmbReal * diameter,`
`EmbReal * chord,`
`EmbReal * chordMidX,`
`EmbReal * chordMidY,`
`EmbReal * sagitta,`
`EmbReal * apothem,`
`EmbReal * incAngleInDegrees,`
`char * clockwise)`

18.28.3.124 getCircleCircleIntersections() `EMB_PUBLIC int getCircleCircleIntersections (`
`EmbCircle c0,`
`EmbCircle c1,`
`EmbVector * v0,`
`EmbVector * v1)`

18.28.3.125 getCircleTangentPoints() `EMB_PUBLIC int getCircleTangentPoints (`
`EmbCircle c,`
`EmbVector p,`
`EmbVector * v0,`
`EmbVector * v1)`

18.28.3.126 hilbert_curve() `EMB_PUBLIC int hilbert_curve (`
`EmbPattern * pattern,`
`int iterations)`

pattern iterations

https://en.wikipedia.org/wiki/Hilbert_curve

Using the Lindenmayer System, so we can save work across different functions.

18.28.3.127 lindenmayer_system() `EMB_PUBLIC int lindenmayer_system (`
`L_system L,`
`char * state,`
`int iterations,`
`int complete)`

L state iterations complete

Returns

int

This is a slow generation algorithm.

18.28.3.128 radians() `EMB_PUBLIC EmbReal radians (`
`EmbReal degree)`

18.28.3.129 report() `EMB_PUBLIC` void report (
 int *result*,
 char * *label*)

18.28.3.130 testMain() `EMB_PUBLIC` void testMain (
 int *level*)

18.28.3.131 threadColor() `EMB_PUBLIC` int threadColor (
 const char * *name*,
 int *brand*)

18.28.3.132 threadColorName() `EMB_PUBLIC` const char * threadColorName (
 unsigned int *color*,
 int *brand*)

18.28.3.133 threadColorNum() `EMB_PUBLIC` int threadColorNum (
 unsigned int *color*,
 int *brand*)

18.28.4 Variable Documentation

18.28.4.1 _dxfColorTable const unsigned char _dxfColorTable[][3] [extern]

18.28.4.2 black_thread `EmbThread` black_thread [extern]

18.28.4.3 emb_error int emb_error [extern]
Error code storage for optional control flow blocking.

18.28.4.4 emb_verbose int emb_verbose [extern]
Verbosity level.

18.28.4.5 embConstantPi const `EmbReal` embConstantPi [extern]

18.28.4.6 formatTable `EmbFormatList` formatTable[`numberOfFormats`] [extern]

This file is part of libembroidery.

Copyright 2018-2022 The Embroidermodder Team Licensed under the terms of the zlib license.

This file contains all the read and write functions for the library.

Todo This list needs reviewed in case some stitch formats also can contain object data (EMBFORMAT_↵
STCHANDOBJ). *

18.28.4.7 husThreads const `EmbThread` husThreads[] [extern]

18.28.4.8 jefThreads `const EmbThread jefThreads[] [extern]`

18.28.4.9 pcmThreads `const EmbThread pcmThreads[] [extern]`

18.28.4.10 pecThreadCount `const int pecThreadCount [extern]`

18.28.4.11 pecThreads `const EmbThread pecThreads[] [extern]`

18.28.4.12 shvThreadCount `const int shvThreadCount [extern]`

18.28.4.13 shvThreads `const EmbThread shvThreads[] [extern]`

18.28.4.14 vipDecodingTable `const unsigned char vipDecodingTable[] [extern]`

18.28.4.15 Embroidery Format (.pcq) The Pfaff vip format is stitch-only.

18.29 embroidery.h

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

```
1 #ifndef LIBEMBROIDERY_HEADER__
2 #define LIBEMBROIDERY_HEADER__
3
4 #ifdef __cplusplus
5 extern "C" {
6 #endif
7
16 #ifndef LIBEMBROIDERY_EMBEDDED_VERSION
17 #define LIBEMBROIDERY_EMBEDDED_VERSION 0
18 #endif
19
20 /* MACROS
21 *****/
22
26 #define NORMAL 0
27 #define JUMP 1
28 #define TRIM 2
29 #define STOP 4
30 #define SEQUIN 8
31 #define END 16
36 #define EMB_FORMAT_100 0
37 #define EMB_FORMAT_100 1
38 #define EMB_FORMAT_ART 2
39 #define EMB_FORMAT_BMC 3
40 #define EMB_FORMAT_BRO 4
41 #define EMB_FORMAT_CND 5
42 #define EMB_FORMAT_COL 6
43 #define EMB_FORMAT_CSD 7
44 #define EMB_FORMAT_CSV 8
45 #define EMB_FORMAT_DAT 9
46 #define EMB_FORMAT_DEM 10
47 #define EMB_FORMAT_DSB 11
48 #define EMB_FORMAT_DST 12
49 #define EMB_FORMAT_DSZ 13
50 #define EMB_FORMAT_DXF 14
51 #define EMB_FORMAT_EDR 15
52 #define EMB_FORMAT_EMD 16
53 #define EMB_FORMAT_EXP 17
54 #define EMB_FORMAT_EXY 18
55 #define EMB_FORMAT_EYS 19
56 #define EMB_FORMAT_FXY 20
57 #define EMB_FORMAT_GC 21
58 #define EMB_FORMAT_GNC 22
59 #define EMB_FORMAT_GT 23
60 #define EMB_FORMAT_HUS 24
61 #define EMB_FORMAT_INB 25
62 #define EMB_FORMAT_INF 26
```

```
63 #define EMB_FORMAT_JEF          27
64 #define EMB_FORMAT_KSM          28
65 #define EMB_FORMAT_MAX          29
66 #define EMB_FORMAT_MIT          30
67 #define EMB_FORMAT_NEW          31
68 #define EMB_FORMAT_OFM          32
69 #define EMB_FORMAT_PCD          33
70 #define EMB_FORMAT_PCM          34
71 #define EMB_FORMAT_PCQ          35
72 #define EMB_FORMAT_PCS          36
73 #define EMB_FORMAT_PEC          37
74 #define EMB_FORMAT_PEL          38
75 #define EMB_FORMAT_PEM          39
76 #define EMB_FORMAT_PES          40
77 #define EMB_FORMAT_PHB          41
78 #define EMB_FORMAT_PHC          42
79 #define EMB_FORMAT_PLT          43
80 #define EMB_FORMAT_RGB          44
81 #define EMB_FORMAT_SEW          45
82 #define EMB_FORMAT_SHV          46
83 #define EMB_FORMAT_SST          47
84 #define EMB_FORMAT_STX          48
85 #define EMB_FORMAT_SVG          49
86 #define EMB_FORMAT_T01          50
87 #define EMB_FORMAT_T09          51
88 #define EMB_FORMAT_TAP          52
89 #define EMB_FORMAT_THR          53
90 #define EMB_FORMAT_TXT          54
91 #define EMB_FORMAT_U00          55
92 #define EMB_FORMAT_U01          56
93 #define EMB_FORMAT_VIP          57
94 #define EMB_FORMAT_VP3          58
95 #define EMB_FORMAT_XXX          59
96 #define EMB_FORMAT_ZSK          60
97
98 /* Thread color */
99 #define Arc_Polyester            0
100 #define Arc_Rayon                1
101 #define CoatsAndClark_Rayon      2
102 #define Exquisite_Polyester      3
103 #define Fufu_Polyester           4
104 #define Fufu_Rayon               5
105 #define Hemingworth_Polyester    6
106 #define Isacord_Polyester        7
107 #define Isafil_Rayon             8
108 #define Marathon_Polyester       9
109 #define Marathon_Rayon          10
110 #define Madeira_Polyester        11
111 #define Madeira_Rayon            12
112 #define Metro_Polyester          13
113 #define Pantone                  14
114 #define RobisonAnton_Polyester   15
115 #define RobisonAnton_Rayon       16
116 #define Sigma_Polyester          17
117 #define Sulky_Rayon              18
118 #define ThreadArt_Rayon          19
119 #define ThreadArt_Polyester      20
120 #define ThreaDelight_Polyester   21
121 #define Z102_Isacord_Polyester   22
122 #define SVG_Colors               23
123 #define hus_thread               24
124 #define jef_thread               25
125 #define pcm_thread               26
126 #define pec_thread               27
127 #define shv_thread               28
128 #define dxf_color                29
129
130 #define EMB_ARRAY                0
131 #define EMB_ARC                  1
132 #define EMB_CIRCLE               2
133 #define EMB_DIM_DIAMETER         3
134 #define EMB_DIM_LEADER           4
135 #define EMB_ELLIPSE              5
136 #define EMB_FLAG                 6
137 #define EMB_LINE                 7
138 #define EMB_IMAGE                8
139 #define EMB_PATH                 9
140 #define EMB_POINT                10
141 #define EMB_POLYGON              11
142 #define EMB_POLYLINE             12
143 #define EMB_RECT                 13
144 #define EMB_SPLINE               14
145 #define EMB_STITCH               15
146 #define EMB_TEXT_SINGLE          16
147 #define EMB_TEXT_MULTI           17
148 #define EMB_VECTOR               18
149 #define EMB_THREAD               19
```



```

150
151 #define EMBFORMAT_UNSUPPORTED      0
152 #define EMBFORMAT_STITCHONLY      1
153 #define EMBFORMAT_OBJECTONLY      2
154 #define EMBFORMAT_STCHANDOBJ      3 /* binary operation: 1+2=3 */
155
156 #define numberOfFormats             61
157
158 #define CHUNK_SIZE                  128
159
160 #define EMB_MAX_LAYERS              10
161 #define MAX_THREADS                 256
162 #define EMBFORMAT_MAXEXT           3
163 /* maximum length of extension without dot */
164 #define EMBFORMAT_MAXDESC          50
165 /* the longest possible description string length */
166 #define MAX_STITCHES               1000000
167
168
169
170 #if defined(_WIN32) && !defined(WIN32)
171 #define WIN32
172 #endif
173
174 /* When building a shared library,
175 * use the proper export keyword depending on the compiler */
176 #define EMB_PUBLIC
177 #if defined(LIBEMBROIDERY_SHARED)
178 #undef EMB_PUBLIC
179 #if defined(__WIN32__) || defined(WIN32)
180 #define EMB_PUBLIC __declspec(dllexport)
181 #else
182 #define EMB_PUBLIC __attribute__((visibility("default")))
183 #endif
184 #endif
185
186 /* TYPEDEFS AND STRUCTS
187 *****/
188
189 typedef float EmbReal;
190
191 typedef struct EmbColor_
192 {
193     unsigned char r;
194     unsigned char g;
195     unsigned char b;
196 } EmbColor;
197
198 typedef struct EmbVector_
199 {
200     EmbReal x;
201     EmbReal y;
202 } EmbVector;
203
204 typedef struct EmbArray_ EmbArray;
205
206 typedef struct EmbImage_ {
207     EmbVector position;
208     EmbVector dimensions;
209     unsigned char* data;
210     int width;
211     int height;
212     char path[200];
213     char name[200];
214 } EmbImage;
215
216 typedef struct EmbBlock_ {
217     EmbVector position;
218 } EmbBlock;
219
220 typedef struct EmbAlignedDim_ {
221     EmbVector position;
222 } EmbAlignedDim;
223
224 typedef struct EmbAngularDim_ {
225     EmbVector position;
226 } EmbAngularDim;
227
228 typedef struct EmbArcLengthDim_ {
229     EmbVector position;
230 } EmbArcLengthDim;
231
232 typedef struct EmbDiameterDim_ {
233     EmbVector position;
234 } EmbDiameterDim;
235
236 typedef struct EmbLeaderDim_ {

```

```
272     EmbVector position;
273 } EmbLeaderDim;
274
275 typedef struct EmbLinearDim_ {
276     EmbVector position;
277 } EmbLinearDim;
278
279 typedef struct EmbOrdinateDim_ {
280     EmbVector position;
281 } EmbOrdinateDim;
282
283 typedef struct EmbRadiusDim_ {
284     EmbVector position;
285 } EmbRadiusDim;
286
287 typedef struct EmbInfiniteLine_ {
288     EmbVector position;
289 } EmbInfiniteLine;
290
291 typedef struct EmbRay_ {
292     EmbVector position;
293 } EmbRay;
294
295 typedef struct EmbTextMulti_ {
296     EmbVector position;
297     char text[200];
298 } EmbTextMulti;
299
300 typedef struct EmbTextSingle_ {
301     EmbVector position;
302     char text[200];
303 } EmbTextSingle;
304
305 typedef struct EmbTime_
306 {
307     unsigned int year;
308     unsigned int month;
309     unsigned int day;
310     unsigned int hour;
311     unsigned int minute;
312     unsigned int second;
313 } EmbTime;
314
315 typedef struct EmbPoint_
316 {
317     EmbVector position;
318     int lineType;
319     EmbColor color;
320 } EmbPoint;
321
322 typedef struct EmbLine_
323 {
324     EmbVector start;
325     EmbVector end;
326     int lineType;
327     EmbColor color;
328 } EmbLine;
329
330 typedef struct EmbPath_
331 {
332     EmbArray* pointList;
333     EmbArray* flagList;
334     int lineType;
335     EmbColor color;
336 } EmbPath;
337
338 typedef struct EmbStitch_
339 {
340     int flags;
341     EmbReal x;
342     EmbReal y;
343     int color;
344 } EmbStitch;
345
346 typedef struct EmbThread_
347 {
348     EmbColor color;
349     char description[50];
350     char catalogNumber[30];
351 } EmbThread;
352
353 typedef struct thread_color_ {
354     char name[22];
355     unsigned int hex_code;
356     int manufacturer_code;
357 } thread_color;
358
```

```
420 typedef struct EmbArc_  
421 {  
422     EmbVector start;  
423     EmbVector mid;  
424     EmbVector end;  
425 } EmbArc;  
426  
431 typedef struct EmbRect_  
432 {  
433     EmbReal top;  
434     EmbReal left;  
435     EmbReal bottom;  
436     EmbReal right;  
437     EmbReal rotation;  
438     EmbReal radius;  
439 } EmbRect;  
440  
445 typedef struct EmbCircle_  
446 {  
447     EmbVector center;  
448     EmbReal radius;  
449 } EmbCircle;  
450  
455 typedef EmbPath EmbPolygon;  
456  
461 typedef EmbPath EmbPolyline;  
462  
467 typedef int EmbFlag;  
468  
473 typedef struct EmbSatinOutline_  
474 {  
475     int length;  
476     EmbArray* side1;  
477     EmbArray* side2;  
478 } EmbSatinOutline;  
479  
484 typedef struct EmbEllipse_  
485 {  
486     EmbVector center;  
487     EmbVector radius;  
488     EmbReal rotation;  
489 } EmbEllipse;  
490  
495 typedef struct EmbBezier_ {  
496     EmbVector start;  
497     EmbVector control1;  
498     EmbVector control2;  
499     EmbVector end;  
500 } EmbBezier;  
501  
506 typedef struct EmbSpline_ {  
507     EmbArray *beziers;  
508 } EmbSpline;  
509  
514 typedef struct LSYSTEM {  
515     char axiom;  
516     char *alphabet;  
517     char *constants;  
518     char **rules;  
519 } L_system;  
520  
525 typedef struct EmbGeometry_ {  
526     union {  
527         EmbArc arc;  
528         EmbCircle circle;  
529         EmbColor color;  
530         EmbEllipse ellipse;  
531         EmbLine line;  
532         EmbPath path;  
533         EmbPoint point;  
534         EmbPolygon polygon;  
535         EmbPolyline polyline;  
536         EmbRect rect;  
537         EmbSpline spline;  
538         EmbVector vector;  
539     } object;  
540     EmbStitch stitch;  
541     EmbThread thread;  
542     int flag;  
543     int type;  
544     int lineType;  
545     EmbColor color;  
546 } EmbGeometry;  
547  
552 struct EmbArray_ {  
553     EmbGeometry *geometry;  
554     EmbStitch *stitch;
```

```

555     EmbThread *thread;
556     int count;
557     int length;
558     int type;
559 };
560
565 typedef struct EmbLayer_
566 {
567     char name[100];
568     EmbArray *geometry;
569 } EmbLayer;
570
575 typedef struct EmbPattern_
576 {
577     unsigned int dstJumpsPerTrim;
578     EmbVector home;
579     EmbReal hoop_width;
580     EmbReal hoop_height;
581     EmbArray *thread_list;
582     EmbArray *stitch_list;
583     EmbArray *geometry;
584     EmbLayer layer[EMB_MAX_LAYERS];
585     int currentColorIndex;
586 } EmbPattern;
587
592 typedef struct EmbFormatList_
593 {
594     char extension[2 + EMBFORMAT_MAXEXT];
595     char description[EMBFORMAT_MAXDESC];
596     char reader_state;
597     char writer_state;
598     int type;
599     int color_only;
600     int check_for_color_file;
601     int write_external_color_file;
602 } EmbFormatList;
603
604 /* Function Declarations
605 *****/
606 EMB_PUBLIC int lindenmayer_system(L_system L, char* state, int iteration, int complete);
607 EMB_PUBLIC int hilbert_curve(EmbPattern *pattern, int iterations);
608
609 EMB_PUBLIC int emb_identify_format(const char *ending);
610 EMB_PUBLIC void testMain(int level);
611 EMB_PUBLIC int convert(const char *inf, const char *outf);
612
613 EMB_PUBLIC EmbColor embColor_make(unsigned char r, unsigned char g, unsigned char b);
614 EMB_PUBLIC EmbColor* embColor_create(unsigned char r, unsigned char g, unsigned char b);
615 EMB_PUBLIC EmbColor embColor_fromHexStr(char* val);
616 EMB_PUBLIC int embColor_distance(EmbColor a, EmbColor b);
617
618 EMB_PUBLIC EmbArray* embArray_create(int type);
619 EMB_PUBLIC int embArray_resize(EmbArray *g);
620 EMB_PUBLIC void embArray_copy(EmbArray *dst, EmbArray *src);
621 EMB_PUBLIC int embArray_addArc(EmbArray* g, EmbArc arc);
622 EMB_PUBLIC int embArray_addCircle(EmbArray* g, EmbCircle circle);
623 EMB_PUBLIC int embArray_addEllipse(EmbArray* g, EmbEllipse ellipse);
624 EMB_PUBLIC int embArray_addFlag(EmbArray* g, int flag);
625 EMB_PUBLIC int embArray_addLine(EmbArray* g, EmbLine line);
626 EMB_PUBLIC int embArray_addRect(EmbArray* g, EmbRect rect);
627 EMB_PUBLIC int embArray_addPath(EmbArray* g, EmbPath p);
628 EMB_PUBLIC int embArray_addPoint(EmbArray* g, EmbPoint p);
629 EMB_PUBLIC int embArray_addPolygon(EmbArray* g, EmbPolygon p);
630 EMB_PUBLIC int embArray_addPolyline(EmbArray* g, EmbPolyline p);
631 /* EMB_PUBLIC int embArray_addSpline(EmbArray* g, EmbSpline p); */
632 EMB_PUBLIC int embArray_addStitch(EmbArray* g, EmbStitch st);
633 EMB_PUBLIC int embArray_addThread(EmbArray* g, EmbThread p);
634 EMB_PUBLIC int embArray_addVector(EmbArray* g, EmbVector);
635 EMB_PUBLIC void embArray_free(EmbArray* p);
636
637 EMB_PUBLIC EmbLine embLine_make(EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2);
638
639 EMB_PUBLIC void embLine_normalVector(EmbLine line, EmbVector* result, int clockwise);
640 EMB_PUBLIC EmbVector embLine_intersectionPoint(EmbLine line1, EmbLine line2);
641
642 EMB_PUBLIC int embThread_findNearestColor(EmbColor color, EmbColor* colors, int n_colors);
643 EMB_PUBLIC int embThread_findNearestThread(EmbColor color, EmbThread* threads, int n_threads);
644 EMB_PUBLIC EmbThread embThread_getRandom(void);
645
646 EMB_PUBLIC void embVector_normalize(EmbVector vector, EmbVector* result);
647 EMB_PUBLIC void embVector_multiply(EmbVector vector, EmbReal magnitude, EmbVector* result);
648 EMB_PUBLIC EmbVector embVector_add(EmbVector v1, EmbVector v2);
649 EMB_PUBLIC EmbVector embVector_average(EmbVector v1, EmbVector v2);
650 EMB_PUBLIC EmbVector embVector_subtract(EmbVector v1, EmbVector v2);
651 EMB_PUBLIC EmbReal embVector_dot(EmbVector v1, EmbVector v2);
652 EMB_PUBLIC EmbReal embVector_cross(EmbVector v1, EmbVector v2);
653 EMB_PUBLIC void embVector_transpose_product(EmbVector v1, EmbVector v2, EmbVector* result);

```

```

654 EMB_PUBLIC EmbReal embVector_length(EmbVector vector);
655 EMB_PUBLIC EmbReal embVector_relativeX(EmbVector a1, EmbVector a2, EmbVector a3);
656 EMB_PUBLIC EmbReal embVector_relativeY(EmbVector a1, EmbVector a2, EmbVector a3);
657 EMB_PUBLIC EmbReal embVector_angle(EmbVector v);
658 EMB_PUBLIC EmbReal embVector_distance(EmbVector a, EmbVector b);
659 EMB_PUBLIC EmbVector embVector_unit(EmbReal angle);
660
661 EMB_PUBLIC EmbArc embArc_init(void);
662 EMB_PUBLIC char embArc_clockwise(EmbArc arc);
663
664 EMB_PUBLIC void getArcCenter(EmbArc arc, EmbVector *arcCenter);
665 EMB_PUBLIC char getArcDataFromBulge(EmbReal bulge,
666                                     EmbArc *arc,
667                                     EmbReal* arcCenterX,      EmbReal* arcCenterY,
668                                     EmbReal* radius,           EmbReal* diameter,
669                                     EmbReal* chord,            EmbReal* chordMidX,
670                                     EmbReal* chordMidY,         EmbReal* chordMidY,
671                                     EmbReal* sagitta,           EmbReal* apothem,
672                                     EmbReal* incAngleInDegrees, char* clockwise);
673
674 EMB_PUBLIC EmbCircle embCircle_init(void);
675 EMB_PUBLIC int getCircleCircleIntersections(
676     EmbCircle c0, EmbCircle c1, EmbVector *v0, EmbVector *v1);
677 EMB_PUBLIC int getCircleTangentPoints(
678     EmbCircle c, EmbVector p, EmbVector *v0, EmbVector *v1);
679
680 EMB_PUBLIC EmbEllipse embEllipse_init(void);
681 EMB_PUBLIC EmbEllipse embEllipse_make(EmbReal cx, EmbReal cy, EmbReal rx, EmbReal ry);
682 EMB_PUBLIC EmbReal embEllipse_diameterX(EmbEllipse ellipse);
683 EMB_PUBLIC EmbReal embEllipse_diameterY(EmbEllipse ellipse);
684 EMB_PUBLIC EmbReal embEllipse_width(EmbEllipse ellipse);
685 EMB_PUBLIC EmbReal embEllipse_height(EmbEllipse ellipse);
686 EMB_PUBLIC EmbReal embEllipse_area(EmbEllipse ellipse);
687 EMB_PUBLIC EmbReal embEllipse_perimeter(EmbEllipse ellipse);
688
689 EMB_PUBLIC EmbImage embImage_create(int, int);
690 EMB_PUBLIC void embImage_read(EmbImage *image, char *fname);
691 EMB_PUBLIC int embImage_write(EmbImage *image, char *fname);
692 EMB_PUBLIC void embImage_free(EmbImage *image);
693
694 EMB_PUBLIC EmbRect embRect_init(void);
695 EMB_PUBLIC EmbReal embRect_area(EmbRect);
696
697 EMB_PUBLIC int threadColor(const char*, int brand);
698 EMB_PUBLIC int threadColorNum(unsigned int color, int brand);
699 EMB_PUBLIC const char* threadColorName(unsigned int color, int brand);
700
701 EMB_PUBLIC void embTime_initNow(EmbTime* t);
702 EMB_PUBLIC EmbTime embTime_time(EmbTime* t);
703
704 EMB_PUBLIC void embSatinOutline_generateSatinOutline(EmbArray* lines, EmbReal thickness,
705     EmbSatinOutline* result);
706 EMB_PUBLIC EmbArray* embSatinOutline_renderStitches(EmbSatinOutline* result, EmbReal density);
707
708 EMB_PUBLIC EmbGeometry *embGeometry_init(int type_in);
709 EMB_PUBLIC void embGeometry_free(EmbGeometry *obj);
710 EMB_PUBLIC void embGeometry_move(EmbGeometry *obj, EmbVector delta);
711 EMB_PUBLIC EmbRect embGeometry_boundingRect(EmbGeometry *obj);
712 EMB_PUBLIC void embGeometry_vulcanize(EmbGeometry *obj);
713
714 EMB_PUBLIC EmbPattern* embPattern_create(void);
715 EMB_PUBLIC void embPattern_hideStitchesOverLength(EmbPattern* p, int length);
716 EMB_PUBLIC void embPattern_fixColorCount(EmbPattern* p);
717 EMB_PUBLIC int embPattern_addThread(EmbPattern* p, EmbThread thread);
718 EMB_PUBLIC void embPattern_addStitchAbs(EmbPattern* p, EmbReal x, EmbReal y, int flags, int
719     isAutoColorIndex);
719 EMB_PUBLIC void embPattern_addStitchRel(EmbPattern* p, EmbReal dx, EmbReal dy, int flags, int
720     isAutoColorIndex);
721 EMB_PUBLIC void embPattern_changeColor(EmbPattern* p, int index);
722 EMB_PUBLIC void embPattern_free(EmbPattern* p);
723 EMB_PUBLIC void embPattern_scale(EmbPattern* p, EmbReal scale);
724 EMB_PUBLIC EmbReal embPattern_totalStitchLength(EmbPattern *pattern);
725 EMB_PUBLIC EmbReal embPattern_minimumStitchLength(EmbPattern *pattern);
726 EMB_PUBLIC EmbReal embPattern_maximumStitchLength(EmbPattern *pattern);
727 EMB_PUBLIC void embPattern_lengthHistogram(EmbPattern *pattern, int *bin, int NUMBINS);
728 EMB_PUBLIC int embPattern_realStitches(EmbPattern *pattern);
729 EMB_PUBLIC int embPattern_jumpStitches(EmbPattern *pattern);
730 EMB_PUBLIC int embPattern_trimStitches(EmbPattern *pattern);
731 EMB_PUBLIC EmbRect embPattern_calcBoundingBox(EmbPattern* p);
732 EMB_PUBLIC void embPattern_flipHorizontal(EmbPattern* p);
733 EMB_PUBLIC void embPattern_flipVertical(EmbPattern* p);
734 EMB_PUBLIC void embPattern_flip(EmbPattern* p, int horz, int vert);
735 EMB_PUBLIC void embPattern_combineJumpStitches(EmbPattern* p);
736 EMB_PUBLIC void embPattern_correctForMaxStitchLength(EmbPattern* p, EmbReal maxStitchLength, EmbReal
737     maxJumpLength);
738 EMB_PUBLIC void embPattern_center(EmbPattern* p);
739 EMB_PUBLIC void embPattern_loadExternalColorFile(EmbPattern* p, const char* fileName);

```

```

737 EMB_PUBLIC void embPattern_convertGeometry(EmbPattern* p);
738 EMB_PUBLIC void embPattern_designDetails(EmbPattern *p);
739 EMB_PUBLIC EmbPattern *embPattern_combine(EmbPattern *p1, EmbPattern *p2);
740 EMB_PUBLIC int embPattern_color_count(EmbPattern *pattern, EmbColor startColor);
741 EMB_PUBLIC void embPattern_end(EmbPattern* p);
742 EMB_PUBLIC void embPattern_crossstitch(EmbPattern *pattern, EmbImage *, int threshold);
743 EMB_PUBLIC void embPattern_horizontal_fill(EmbPattern *pattern, EmbImage *, int threshold);
744 EMB_PUBLIC int embPattern_render(EmbPattern *pattern, char *fname);
745 EMB_PUBLIC int embPattern_simulate(EmbPattern *pattern, char *fname);
746
747 EMB_PUBLIC void embPattern_addCircleAbs(EmbPattern* p, EmbCircle obj);
748 EMB_PUBLIC void embPattern_addEllipseAbs(EmbPattern* p, EmbEllipse obj);
749 EMB_PUBLIC void embPattern_addLineAbs(EmbPattern* p, EmbLine obj);
750 EMB_PUBLIC void embPattern_addPathAbs(EmbPattern* p, EmbPath obj);
751 EMB_PUBLIC void embPattern_addPointAbs(EmbPattern* p, EmbPoint obj);
752 EMB_PUBLIC void embPattern_addPolygonAbs(EmbPattern* p, EmbPolygon obj);
753 EMB_PUBLIC void embPattern_addPolylineAbs(EmbPattern* p, EmbPolyline obj);
754 EMB_PUBLIC void embPattern_addRectAbs(EmbPattern* p, EmbRect obj);
755
756 EMB_PUBLIC void embPattern_copyStitchListToPolylines(EmbPattern* pattern);
757 EMB_PUBLIC void embPattern_copyPolylinesToStitchList(EmbPattern* pattern);
758 EMB_PUBLIC void embPattern_moveStitchListToPolylines(EmbPattern* pattern);
759 EMB_PUBLIC void embPattern_movePolylinesToStitchList(EmbPattern* pattern);
760
761 EMB_PUBLIC char embPattern_read(EmbPattern *pattern, const char* fileName, int format);
762 EMB_PUBLIC char embPattern_write(EmbPattern *pattern, const char* fileName, int format);
763
764 EMB_PUBLIC char embPattern_readAuto(EmbPattern *pattern, const char* fileName);
765 EMB_PUBLIC char embPattern_writeAuto(EmbPattern *pattern, const char* fileName);
766
767 EMB_PUBLIC void report(int result, char *label);
768 EMB_PUBLIC int full_test_matrix(char *fname);
769
770 EMB_PUBLIC int emb_round(EmbReal x);
771 EMB_PUBLIC EmbReal radians(EmbReal degree);
772 EMB_PUBLIC EmbReal degrees(EmbReal radian);
773
774 /* NON-MACRO CONSTANTS
775 *****/
776
777 extern EmbFormatList formatTable[numberOfFormats];
778 extern const int pecThreadCount;
779 extern const int shvThreadCount;
780 extern const EmbReal embConstantPi;
781 extern const EmbThread husThreads[];
782 extern const EmbThread jefThreads[];
783 extern const EmbThread shvThreads[];
784 extern const EmbThread pcmThreads[];
785 extern const EmbThread pecThreads[];
786 extern const unsigned char _dxfColorTable[][3];
787 extern EmbThread black_thread;
788 extern const unsigned char vipDecodingTable[];
789
790 /* VARIABLES
791 *****/
792
793 extern int emb_error;
794
795 extern int emb_verbose;
796
797 #ifdef __cplusplus
798 }
799 #endif /* __cplusplus */
800
801 #endif /* LIBEMBROIDERY_HEADER__ */
802

```

18.30 extern/libembroidery/src/embroidery_internal.h File Reference

```

#include "embroidery.h"
#include <stdio.h>

```

Classes

- struct [_bcf_file_difat](#)
- struct [_bcf_file_fat](#)
- struct [_bcf_directory_entry](#)
- struct [_bcf_directory](#)
- struct [_bcf_file_header](#)

- struct [_bcf_file](#)
- struct [_vp3Hoop](#)
- struct [ThredHeader_](#)
- struct [ThredExtension_](#)
- struct [SubDescriptor_](#)
- struct [StxThread_](#)
- struct [VipHeader_](#)
- struct [SvgAttribute_](#)
- struct [Huffman](#)
- struct [Compress](#)

Macros

- #define [CompoundFileSector_MaxRegSector](#) 0xFFFFFFFFFA
- #define [CompoundFileSector_DIFAT_Sector](#) 0xFFFFFFFFFC
- #define [CompoundFileSector_FAT_Sector](#) 0xFFFFFFFFFD
- #define [CompoundFileSector_EndOfChain](#) 0xFFFFFFFFFE
- #define [CompoundFileSector_FreeSector](#) 0xFFFFFFFFFF
- #define [ObjectTypeUnknown](#) 0x00
- #define [ObjectTypeStorage](#) 0x01
- #define [ObjectTypeStream](#) 0x02
- #define [ObjectTypeRootEntry](#) 0x05
- #define [CompoundFileStreamId_MaxRegularStreamId](#) 0xFFFFFFFFFA
- #define [CompoundFileStreamId_NoStream](#) 0xFFFFFFFFFF
- #define [ELEMENT_XML](#) 0
- #define [ELEMENT_A](#) 1
- #define [ELEMENT_ANIMATE](#) 2
- #define [ELEMENT_ANIMATECOLOR](#) 3
- #define [ELEMENT_ANIMATEMOTION](#) 4
- #define [ELEMENT_ANIMATETRANSFORM](#) 5
- #define [ELEMENT_ANIMATION](#) 6
- #define [ELEMENT_AUDIO](#) 7
- #define [ELEMENT_CIRCLE](#) 8
- #define [ELEMENT_DEFS](#) 9
- #define [ELEMENT_DESC](#) 10
- #define [ELEMENT_DISCARD](#) 11
- #define [ELEMENT_ELLIPSE](#) 12
- #define [ELEMENT_FONT](#) 13
- #define [ELEMENT_FONT_FACE](#) 14
- #define [ELEMENT_FONT_FACE_SRC](#) 15
- #define [ELEMENT_FONT_FACE_URI](#) 16
- #define [ELEMENT_FOREIGN_OBJECT](#) 17
- #define [ELEMENT_G](#) 18
- #define [ELEMENT_GLYPH](#) 19
- #define [ELEMENT_HANDLER](#) 20
- #define [ELEMENT_HKERN](#) 21
- #define [ELEMENT_IMAGE](#) 22
- #define [ELEMENT_LINE](#) 23
- #define [ELEMENT_LINEAR_GRADIENT](#) 24
- #define [ELEMENT_LISTENER](#) 25
- #define [ELEMENT_METADATA](#) 26
- #define [ELEMENT_MISSING_GLYPH](#) 27
- #define [ELEMENT_MPATH](#) 28
- #define [ELEMENT_PATH](#) 29
- #define [ELEMENT_POLYGON](#) 30

- #define ELEMENT_POLYLINE 31
- #define ELEMENT_PREFETCH 32
- #define ELEMENT_RADIAL_GRADIENT 33
- #define ELEMENT_RECT 34
- #define ELEMENT_SCRIPT 35
- #define ELEMENT_SET 36
- #define ELEMENT_SOLID_COLOR 37
- #define ELEMENT_STOP 38
- #define ELEMENT_SVG 39
- #define ELEMENT_SWITCH 40
- #define ELEMENT_TBREAK 41
- #define ELEMENT_TEXT 42
- #define ELEMENT_TEXT_AREA 43
- #define ELEMENT_TITLE 44
- #define ELEMENT_TSPAN 45
- #define ELEMENT_USE 46
- #define ELEMENT_VIDEO 47
- #define RED_TERM_COLOR "\x1B[0;31m"
- #define GREEN_TERM_COLOR "\x1B[0;32m"
- #define YELLOW_TERM_COLOR "\x1B[1;33m"
- #define RESET_TERM_COLOR "\033[0m"
- #define HOOP_126X110 0
- #define HOOP_110X110 1
- #define HOOP_50X50 2
- #define HOOP_140X200 3
- #define HOOP_230X200 4
- #define EMB_MIN(A, B) (((A) < (B)) ? (A) : (B))
- #define EMB_MAX(A, B) (((A) > (B)) ? (A) : (B))
- #define EMB_BIG_ENDIAN 0
- #define EMB_LITTLE_ENDIAN 1
- #define ENDIAN_HOST EMB_LITTLE_ENDIAN
- #define EMB_INT16_BIG 2
- #define EMB_INT16_LITTLE 3
- #define EMB_INT32_BIG 4
- #define EMB_INT32_LITTLE 5
- #define PES0001 0
- #define PES0020 1
- #define PES0022 2
- #define PES0030 3
- #define PES0040 4
- #define PES0050 5
- #define PES0055 6
- #define PES0056 7
- #define PES0060 8
- #define PES0070 9
- #define PES0080 10
- #define PES0090 11
- #define PES0100 12
- #define N_PES_VERSIONS 13
- #define DXF_VERSION_R10 "AC1006"
- #define DXF_VERSION_R11 "AC1009"
- #define DXF_VERSION_R12 "AC1009"
- #define DXF_VERSION_R13 "AC1012"
- #define DXF_VERSION_R14 "AC1014"
- #define DXF_VERSION_R15 "AC1015"

- `#define DXF_VERSION_R18` "AC1018"
- `#define DXF_VERSION_R21` "AC1021"
- `#define DXF_VERSION_R24` "AC1024"
- `#define DXF_VERSION_R27` "AC1027"
- `#define DXF_VERSION_2000` "AC1015"
- `#define DXF_VERSION_2002` "AC1015"
- `#define DXF_VERSION_2004` "AC1018"
- `#define DXF_VERSION_2006` "AC1018"
- `#define DXF_VERSION_2007` "AC1021"
- `#define DXF_VERSION_2009` "AC1021"
- `#define DXF_VERSION_2010` "AC1024"
- `#define DXF_VERSION_2013` "AC1027"
- `#define SVG_CREATOR_NULL` 0
- `#define SVG_CREATOR_EMBROIDERMODDER` 1
- `#define SVG_CREATOR_ILLUSTRATOR` 2
- `#define SVG_CREATOR_INKSCAPE` 3
- `#define SVG_EXPECT_NULL` 0
- `#define SVG_EXPECT_ELEMENT` 1
- `#define SVG_EXPECT_ATTRIBUTE` 2
- `#define SVG_EXPECT_VALUE` 3
- `#define SVG_NULL` 0
- `#define SVG_ELEMENT` 1
- `#define SVG_PROPERTY` 2
- `#define SVG_MEDIA_PROPERTY` 3
- `#define SVG_ATTRIBUTE` 4
- `#define SVG_CATCH_ALL` 5
- `#define LINETO` 0
- `#define MOVETO` 1
- `#define BULGETOCONTROL` 2
- `#define BULGETOEND` 4
- `#define ELLIPSETORAD` 8
- `#define ELLIPSETOEND` 16
- `#define CUBICTOCONTROL1` 32
- `#define CUBICTOCONTROL2` 64
- `#define CUBICTOEND` 128
- `#define QUADTOCONTROL` 256
- `#define QUADTOEND` 512

Typedefs

- `typedef struct _bcf_file_difat` `bcf_file_difat`
- `typedef struct _bcf_file_fat` `bcf_file_fat`
- `typedef struct _bcf_directory_entry` `bcf_directory_entry`
- `typedef struct _bcf_directory` `bcf_directory`
- `typedef struct _bcf_file_header` `bcf_file_header`
- `typedef struct _bcf_file` `bcf_file`
- `typedef struct _vp3Hoop` `vp3Hoop`
- `typedef struct ThredHeader_` `ThredHeader`
- `typedef struct ThredExtension_` `ThredExtension`
- `typedef struct SubDescriptor_` `SubDescriptor`
- `typedef struct StxThread_` `StxThread`
- `typedef struct VipHeader_` `VipHeader`
- `typedef struct SvgAttribute_` `SvgAttribute`
- `typedef struct Huffman` `huffman`
- `typedef struct Compress` `compress`

Enumerations

- enum `CSV_EXPECT` { `CSV_EXPECT_NULL` , `CSV_EXPECT_QUOTE1` , `CSV_EXPECT_QUOTE2` , `CSV_EXPECT_COMMA` }
- enum `CSV_MODE` { `CSV_MODE_NULL` , `CSV_MODE_COMMENT` , `CSV_MODE_VARIABLE` , `CSV_MODE_THREAD` , `CSV_MODE_STITCH` }

Functions

- void `huffman_build_table` (`huffman` *h)
- int * `huffman_table_lookup` (`huffman` *h, int byte_lookup, int *lengths)
- int `compress_get_bits` (`compress` *c, int length)
- int `compress_pop` (`compress` *c, int bit_count)
- int `compress_read_variable_length` (`compress` *c)
- void `compress_load_character_length_huffman` (`compress` *c)
- void `compress_load_character_huffman` (`compress` *c)
- void `compress_load_distance_huffman` (`compress` *c)
- void `compress_load_block` (`compress` *c)
- int `compress_get_token` (`compress` *c)
- int `compress_get_position` (`compress` *c)
- void `readPecStitches` (`EmbPattern` *pattern, FILE *file)
- void `writePecStitches` (`EmbPattern` *pattern, FILE *file, const char *filename)
- int `decodeNewStitch` (unsigned char value)
value
- void `pfaffEncode` (FILE *file, int x, int y, int flags)
- `EmbReal` `pfaffDecode` (unsigned char a1, unsigned char a2, unsigned char a3)
- unsigned char `mitEncodeStitch` (`EmbReal` value)
value
- int `mitDecodeStitch` (unsigned char value)
value
- int `encode_tajima_ternary` (unsigned char b[3], int x, int y)
- void `decode_tajima_ternary` (unsigned char b[3], int *x, int *y)
- void `encode_t01_record` (unsigned char b[3], int x, int y, int flags)
- int `decode_t01_record` (unsigned char b[3], int *x, int *y, int *flags)
- void `readPESHeaderV5` (FILE *file, `EmbPattern` *pattern)
- void `readPESHeaderV6` (FILE *file, `EmbPattern` *pattern)
- void `readPESHeaderV7` (FILE *file, `EmbPattern` *pattern)
- void `readPESHeaderV8` (FILE *file, `EmbPattern` *pattern)
- void `readPESHeaderV9` (FILE *file, `EmbPattern` *pattern)
- void `readPESHeaderV10` (FILE *file, `EmbPattern` *pattern)
- void `readDescriptions` (FILE *file, `EmbPattern` *pattern)
- void `readHoopName` (FILE *file, `EmbPattern` *pattern)
- void `readImageString` (FILE *file, `EmbPattern` *pattern)
- void `readProgrammableFills` (FILE *file, `EmbPattern` *pattern)
- void `readMotifPatterns` (FILE *file, `EmbPattern` *pattern)
- void `readFeatherPatterns` (FILE *file, `EmbPattern` *pattern)
- void `readThreads` (FILE *file, `EmbPattern` *pattern)
- void `emblnt_read` (FILE *f, char *label, void *b, int mode)
- void `emblnt_write` (FILE *f, char *label, void *b, int mode)
- int `embl_readline` (FILE *file, char *line, int maxLength)
file line maxLength
- int `bcfFile_read` (FILE *file, `bcf_file` *bcfFile)
file bcfFile
- FILE * `GetFile` (`bcf_file` *bcfFile, FILE *file, char *fileToFind)

- Get the File object.
- void `bcf_file_free` (`bcf_file` *bcfFile)
 - `bcfFile`
- void `binaryReadString` (FILE *file, char *buffer, int maxLength)
 - `file` `buffer` `maxLength`
- void `binaryReadUnicodeString` (FILE *file, char *buffer, const int stringLength)
 - `file` `buffer` `stringLength`
- int `stringInArray` (const char *s, const char **array)
- void `fpad` (FILE *f, char c, int n)
 - `f`
- char * `copy_trim` (char const *s)
 - `s`
- char * `emb_optOut` (EmbReal num, char *str)
 - Optimizes the number (num) for output to a text file and returns it as a string (str).
- void `write_24bit` (FILE *file, int)
 - `file` `x`
- int `check_header_present` (FILE *file, int minimum_header_length)
 - `file` `minimum_header_length`
- unsigned short `fread_uint16` (FILE *file)
 - `f`
- short `fread_int16` (FILE *f)
 - `f`
- int `fread_int32_be` (FILE *f)
 - `f`
- void `safe_free` (void *data)
 - `data`
- void `binaryWriteUIntBE` (FILE *f, unsigned int data)
 - `f` `data`
- void `binaryWriteUInt` (FILE *f, unsigned int data)
 - `f` `data`
- void `binaryWriteIntBE` (FILE *f, int data)
 - `f` `data`
- void `binaryWriteInt` (FILE *f, int data)
 - `f` `data`
- void `binaryWriteUShort` (FILE *f, unsigned short data)
 - `f` `data`
- void `binaryWriteUShortBE` (FILE *f, unsigned short data)
 - `f` `data`
- void `binaryWriteShort` (FILE *f, short data)
 - `f` `data`
- `bcf_file_difat` * `bcf_difat_create` (FILE *file, unsigned int fatSectors, const unsigned int `sectorSize`)
 - `file` `fatSectors` `sectorSize`
- unsigned int `readFullSector` (FILE *file, `bcf_file_difat` *bcfFile, unsigned int *numberOfDifatEntriesStillToRead)
 - `file` `bcfFile` `difatEntriesToRead`
- unsigned int `numberOfEntriesInDifatSector` (`bcf_file_difat` *fat)
- void `bcf_file_difat_free` (`bcf_file_difat` *difat)
- unsigned int `entriesInDifatSector` (`bcf_file_difat` *fat)
 - `fat`
- `bcf_file_fat` * `bcfFileFat_create` (const unsigned int `sectorSize`)
 - `sectorSize`

- void `loadFatFromSector` (`bcf_file_fat` *fat, FILE *file)
 - fat file*
- void `bcf_file_fat_free` (`bcf_file_fat` **fat)
- `bcf_directory_entry` * `CompoundFileDirectoryEntry` (FILE *file)
 - file*
- `bcf_directory` * `CompoundFileDirectory` (const unsigned int maxNumberOfDirectoryEntries)
 - maxNumberOfDirectoryEntries*
- void `readNextSector` (FILE *file, `bcf_directory` *dir)
 - file dir*
- void `bcf_directory_free` (`bcf_directory` **dir)
 - dir*
- `bcf_file_header` `bcfFileHeader_read` (FILE *file)
 - file*
- int `bcfFileHeader_isValid` (`bcf_file_header` header)
- int `hus_compress` (char *input, int size, char *output, int *out_size)
- int `hus_decompress` (char *input, int size, char *output, int *out_size)
- void `testTangentPoints` (`EmbCircle` c, `EmbVector` p, `EmbVector` *t0, `EmbVector` *t1)
- void `printArcResults` (`EmbReal` bulge, `EmbArc` arc, `EmbReal` centerX, `EmbReal` centerY, `EmbReal` radius, `EmbReal` diameter, `EmbReal` chord, `EmbReal` chordMidX, `EmbReal` chordMidY, `EmbReal` sagitta, `EmbReal` apothem, `EmbReal` incAngle, char clockwise)
- int `create_test_file_1` (const char *outf)
- int `create_test_file_2` (const char *outf)
- int `create_test_file_3` (const char *outf)
- int `testEmbCircle` (void)
- int `testEmbCircle_2` (void)
- int `testGeomArc` (void)
- int `testThreadColor` (void)
- int `testEmbFormat` (void)
- void `embColor_read` (FILE *f, `EmbColor` *c, int toRead)
 - f c toRead*
- void `embColor_write` (FILE *f, `EmbColor` c, int toWrite)
 - f c toWrite*
- char `read100` (`EmbPattern` *pattern, FILE *file)
- char `write100` (`EmbPattern` *pattern, FILE *file)
- char `read10o` (`EmbPattern` *pattern, FILE *file)
- char `write10o` (`EmbPattern` *pattern, FILE *file)
- char `readArt` (`EmbPattern` *pattern, FILE *file)
- char `writeArt` (`EmbPattern` *pattern, FILE *file)
- char `readBmc` (`EmbPattern` *pattern, FILE *file)
- char `writeBmc` (`EmbPattern` *pattern, FILE *file)
- char `readBro` (`EmbPattern` *pattern, FILE *file)
- char `writeBro` (`EmbPattern` *pattern, FILE *file)
- char `readCnd` (`EmbPattern` *pattern, FILE *file)
- char `writeCnd` (`EmbPattern` *pattern, FILE *file)
- char `readCol` (`EmbPattern` *pattern, FILE *file)
- char `writeCol` (`EmbPattern` *pattern, FILE *file)
- char `readCsd` (`EmbPattern` *pattern, FILE *file)
- char `writeCsd` (`EmbPattern` *pattern, FILE *file)
- char `readCsv` (`EmbPattern` *pattern, FILE *file)
- char `writeCsv` (`EmbPattern` *pattern, FILE *file)
- char `readDat` (`EmbPattern` *pattern, FILE *file)
- char `writeDat` (`EmbPattern` *pattern, FILE *file)
- char `readDem` (`EmbPattern` *pattern, FILE *file)

- char [writeDem](#) (EmbPattern *pattern, FILE *file)
- char [readDsb](#) (EmbPattern *pattern, FILE *file)
- char [writeDsb](#) (EmbPattern *pattern, FILE *file)
- char [readDst](#) (EmbPattern *pattern, FILE *file)
- char [writeDst](#) (EmbPattern *pattern, FILE *file)
- char [readDsz](#) (EmbPattern *pattern, FILE *file)
- char [writeDsz](#) (EmbPattern *pattern, FILE *file)
- char [readDxf](#) (EmbPattern *pattern, FILE *file)
- char [writeDxf](#) (EmbPattern *pattern, FILE *file)
- char [readEdr](#) (EmbPattern *pattern, FILE *file)
- char [writeEdr](#) (EmbPattern *pattern, FILE *file)
- char [readEmd](#) (EmbPattern *pattern, FILE *file)
- char [writeEmd](#) (EmbPattern *pattern, FILE *file)
- char [readExp](#) (EmbPattern *pattern, FILE *file)
- char [writeExp](#) (EmbPattern *pattern, FILE *file)
- char [readExy](#) (EmbPattern *pattern, FILE *file)
- char [writeExy](#) (EmbPattern *pattern, FILE *file)
- char [readEys](#) (EmbPattern *pattern, FILE *file)
- char [writeEys](#) (EmbPattern *pattern, FILE *file)
- char [readFxy](#) (EmbPattern *pattern, FILE *file)
- char [writeFxy](#) (EmbPattern *pattern, FILE *file)
- char [readGc](#) (EmbPattern *pattern, FILE *file)
- char [writeGc](#) (EmbPattern *pattern, FILE *file)
- char [readGnc](#) (EmbPattern *pattern, FILE *file)
- char [writeGnc](#) (EmbPattern *pattern, FILE *file)
- char [readGt](#) (EmbPattern *pattern, FILE *file)
- char [writeGt](#) (EmbPattern *pattern, FILE *file)
- char [readHus](#) (EmbPattern *pattern, FILE *file)
- char [writeHus](#) (EmbPattern *pattern, FILE *file)
- char [readInb](#) (EmbPattern *pattern, FILE *file)
- char [writeInb](#) (EmbPattern *pattern, FILE *file)
- char [readInf](#) (EmbPattern *pattern, FILE *file)
- char [writeInf](#) (EmbPattern *pattern, FILE *file)
- char [readJef](#) (EmbPattern *pattern, FILE *file)
- char [writeJef](#) (EmbPattern *pattern, FILE *file)
- char [readKsm](#) (EmbPattern *pattern, FILE *file)
- char [writeKsm](#) (EmbPattern *pattern, FILE *file)
- char [readMax](#) (EmbPattern *pattern, FILE *file)
- char [writeMax](#) (EmbPattern *pattern, FILE *file)
- char [readMit](#) (EmbPattern *pattern, FILE *file)
- char [writeMit](#) (EmbPattern *pattern, FILE *file)
- char [readNew](#) (EmbPattern *pattern, FILE *file)
- char [writeNew](#) (EmbPattern *pattern, FILE *file)
- char [readOfm](#) (EmbPattern *pattern, FILE *file)
- char [writeOfm](#) (EmbPattern *pattern, FILE *file)
- char [readPcd](#) (EmbPattern *pattern, const char *fileName, FILE *file)
- char [writePcd](#) (EmbPattern *pattern, FILE *file)
- char [readPcm](#) (EmbPattern *pattern, FILE *file)
- char [writePcm](#) (EmbPattern *pattern, FILE *file)
- char [readPcq](#) (EmbPattern *pattern, const char *fileName, FILE *file)
- char [writePcq](#) (EmbPattern *pattern, FILE *file)
- char [readPcs](#) (EmbPattern *pattern, const char *fileName, FILE *file)
- char [writePcs](#) (EmbPattern *pattern, FILE *file)
- char [readPec](#) (EmbPattern *pattern, const char *fileName, FILE *file)
- char [writePec](#) (EmbPattern *pattern, const char *fileName, FILE *file)

- char [readPel](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writePel](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readPem](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writePem](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readPes](#) ([EmbPattern](#) *pattern, const char *fileName, FILE *file)
- char [writePes](#) ([EmbPattern](#) *pattern, const char *fileName, FILE *file)
- char [readPhb](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writePhb](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readPhc](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writePhc](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readPlt](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writePlt](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readRgb](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeRgb](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readSew](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeSew](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readShv](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeShv](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readSst](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeSst](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readStx](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeStx](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readSvg](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeSvg](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readT01](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeT01](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readT09](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeT09](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readTap](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeTap](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readThr](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeThr](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readTxt](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeTxt](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readU00](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeU00](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readU01](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeU01](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readVip](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeVip](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readVp3](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeVp3](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readXxx](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeXxx](#) ([EmbPattern](#) *pattern, FILE *file)
- char [readZsk](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeZsk](#) ([EmbPattern](#) *pattern, FILE *file)

Variables

- const char [imageWithFrame](#) [38][48]

18.30.1 Macro Definition Documentation

18.30.1.1 BULGETOCONTROL `#define BULGETOCONTROL 2`

18.30.1.2 BULGETOEND `#define BULGETOEND 4`

18.30.1.3 CompoundFileSector_DIFAT_Sector `#define CompoundFileSector_DIFAT_Sector 0xFFFFFFFFC`

18.30.1.4 CompoundFileSector_EndOfChain `#define CompoundFileSector_EndOfChain 0xFFFFFFFFE`

18.30.1.5 CompoundFileSector_FAT_Sector `#define CompoundFileSector_FAT_Sector 0xFFFFFFFFD`

18.30.1.6 CompoundFileSector_FreeSector `#define CompoundFileSector_FreeSector 0xFFFFFFFFF`

18.30.1.7 CompoundFileSector_MaxRegSector `#define CompoundFileSector_MaxRegSector 0xFFFFFFFFFA`
Type of sector

18.30.1.8 CompoundFileStreamId_MaxRegularStreamId `#define CompoundFileStreamId_MaxRegular↵
StreamId 0xFFFFFFFFFA`

Special values for Stream Identifiers All real stream Ids are less than this

18.30.1.9 CompoundFileStreamId_NoStream `#define CompoundFileStreamId_NoStream 0xFFFFFFFFF`
There is no valid stream Id

18.30.1.10 CUBICTOCONTROL1 `#define CUBICTOCONTROL1 32`

18.30.1.11 CUBICTOCONTROL2 `#define CUBICTOCONTROL2 64`

18.30.1.12 CUBICTOEND `#define CUBICTOEND 128`

18.30.1.13 DXF_VERSION_2000 `#define DXF_VERSION_2000 "AC1015"`

18.30.1.14 DXF_VERSION_2002 `#define DXF_VERSION_2002 "AC1015"`

18.30.1.15 DXF_VERSION_2004 `#define DXF_VERSION_2004 "AC1018"`

18.30.1.16 DXF_VERSION_2006 `#define DXF_VERSION_2006 "AC1018"`

18.30.1.17 DXF_VERSION_2007 `#define DXF_VERSION_2007 "AC1021"`

18.30.1.18 DXF_VERSION_2009 `#define DXF_VERSION_2009 "AC1021"`

18.30.1.19 DXF_VERSION_2010 `#define DXF_VERSION_2010 "AC1024"`

18.30.1.20 DXF_VERSION_2013 `#define DXF_VERSION_2013 "AC1027"`

18.30.1.21 DXF_VERSION_R10 `#define DXF_VERSION_R10 "AC1006"`

18.30.1.22 DXF_VERSION_R11 `#define DXF_VERSION_R11 "AC1009"`

18.30.1.23 DXF_VERSION_R12 `#define DXF_VERSION_R12 "AC1009"`

18.30.1.24 DXF_VERSION_R13 `#define DXF_VERSION_R13 "AC1012"`

18.30.1.25 DXF_VERSION_R14 `#define DXF_VERSION_R14 "AC1014"`

18.30.1.26 DXF_VERSION_R15 `#define DXF_VERSION_R15 "AC1015"`

18.30.1.27 DXF_VERSION_R18 `#define DXF_VERSION_R18 "AC1018"`

18.30.1.28 DXF_VERSION_R21 `#define DXF_VERSION_R21 "AC1021"`

18.30.1.29 DXF_VERSION_R24 `#define DXF_VERSION_R24 "AC1024"`

18.30.1.30 DXF_VERSION_R27 `#define DXF_VERSION_R27 "AC1027"`

18.30.1.31 ELEMENT_A `#define ELEMENT_A 1`

18.30.1.32 ELEMENT_ANIMATE `#define ELEMENT_ANIMATE 2`

18.30.1.33 ELEMENT_ANIMATECOLOR `#define ELEMENT_ANIMATECOLOR 3`

18.30.1.34 ELEMENT_ANIMATEMOTION `#define ELEMENT_ANIMATEMOTION 4`

18.30.1.35 ELEMENT_ANIMATETRANSFORM `#define ELEMENT_ANIMATETRANSFORM 5`

- 18.30.1.36 **ELEMENT_ANIMATION** `#define ELEMENT_ANIMATION 6`
- 18.30.1.37 **ELEMENT_AUDIO** `#define ELEMENT_AUDIO 7`
- 18.30.1.38 **ELEMENT_CIRCLE** `#define ELEMENT_CIRCLE 8`
- 18.30.1.39 **ELEMENT_DEFS** `#define ELEMENT_DEFS 9`
- 18.30.1.40 **ELEMENT_DESC** `#define ELEMENT_DESC 10`
- 18.30.1.41 **ELEMENT_DISCARD** `#define ELEMENT_DISCARD 11`
- 18.30.1.42 **ELEMENT_ELLIPSE** `#define ELEMENT_ELLIPSE 12`
- 18.30.1.43 **ELEMENT_FONT** `#define ELEMENT_FONT 13`
- 18.30.1.44 **ELEMENT_FONT_FACE** `#define ELEMENT_FONT_FACE 14`
- 18.30.1.45 **ELEMENT_FONT_FACE_SRC** `#define ELEMENT_FONT_FACE_SRC 15`
- 18.30.1.46 **ELEMENT_FONT_FACE_URI** `#define ELEMENT_FONT_FACE_URI 16`
- 18.30.1.47 **ELEMENT_FOREIGN_OBJECT** `#define ELEMENT_FOREIGN_OBJECT 17`
- 18.30.1.48 **ELEMENT_G** `#define ELEMENT_G 18`
- 18.30.1.49 **ELEMENT_GLYPH** `#define ELEMENT_GLYPH 19`
- 18.30.1.50 **ELEMENT_HANDLER** `#define ELEMENT_HANDLER 20`
- 18.30.1.51 **ELEMENT_HKERN** `#define ELEMENT_HKERN 21`
- 18.30.1.52 **ELEMENT_IMAGE** `#define ELEMENT_IMAGE 22`
- 18.30.1.53 **ELEMENT_LINE** `#define ELEMENT_LINE 23`

- 18.30.1.54 ELEMENT_LINEAR_GRADIENT** #define ELEMENT_LINEAR_GRADIENT 24

- 18.30.1.55 ELEMENT_LISTENER** #define ELEMENT_LISTENER 25

- 18.30.1.56 ELEMENT_METADATA** #define ELEMENT_METADATA 26

- 18.30.1.57 ELEMENT_MISSING_GLYPH** #define ELEMENT_MISSING_GLYPH 27

- 18.30.1.58 ELEMENT_MPATH** #define ELEMENT_MPATH 28

- 18.30.1.59 ELEMENT_PATH** #define ELEMENT_PATH 29

- 18.30.1.60 ELEMENT_POLYGON** #define ELEMENT_POLYGON 30

- 18.30.1.61 ELEMENT_POLYLINE** #define ELEMENT_POLYLINE 31

- 18.30.1.62 ELEMENT_PREFETCH** #define ELEMENT_PREFETCH 32

- 18.30.1.63 ELEMENT_RADIAL_GRADIENT** #define ELEMENT_RADIAL_GRADIENT 33

- 18.30.1.64 ELEMENT_RECT** #define ELEMENT_RECT 34

- 18.30.1.65 ELEMENT_SCRIPT** #define ELEMENT_SCRIPT 35

- 18.30.1.66 ELEMENT_SET** #define ELEMENT_SET 36

- 18.30.1.67 ELEMENT_SOLID_COLOR** #define ELEMENT_SOLID_COLOR 37

- 18.30.1.68 ELEMENT_STOP** #define ELEMENT_STOP 38

- 18.30.1.69 ELEMENT_SVG** #define ELEMENT_SVG 39

- 18.30.1.70 ELEMENT_SWITCH** #define ELEMENT_SWITCH 40

- 18.30.1.71 ELEMENT_TBREAK** #define ELEMENT_TBREAK 41

18.30.1.72 ELEMENT_TEXT #define ELEMENT_TEXT 42

18.30.1.73 ELEMENT_TEXT_AREA #define ELEMENT_TEXT_AREA 43

18.30.1.74 ELEMENT_TITLE #define ELEMENT_TITLE 44

18.30.1.75 ELEMENT_TSPAN #define ELEMENT_TSPAN 45

18.30.1.76 ELEMENT_USE #define ELEMENT_USE 46

18.30.1.77 ELEMENT_VIDEO #define ELEMENT_VIDEO 47

18.30.1.78 ELEMENT_XML #define ELEMENT_XML 0

18.30.1.79 ELLIPSETOEND #define ELLIPSETOEND 16

18.30.1.80 ELLIPSETORAD #define ELLIPSETORAD 8

18.30.1.81 EMB_BIG_ENDIAN #define EMB_BIG_ENDIAN 0

18.30.1.82 EMB_INT16_BIG #define EMB_INT16_BIG 2

18.30.1.83 EMB_INT16_LITTLE #define EMB_INT16_LITTLE 3

18.30.1.84 EMB_INT32_BIG #define EMB_INT32_BIG 4

18.30.1.85 EMB_INT32_LITTLE #define EMB_INT32_LITTLE 5

18.30.1.86 EMB_LITTLE_ENDIAN #define EMB_LITTLE_ENDIAN 1

18.30.1.87 EMB_MAX #define EMB_MAX(
 A,
 B) ((A) > (B)) ? (A) : (B))

18.30.1.88 EMB_MIN #define EMB_MIN(
 A,
 B) ((A) < (B)) ? (A) : (B))

18.30.1.89 ENDIAN_HOST `#define ENDIAN_HOST EMB_LITTLE_ENDIAN`

18.30.1.90 GREEN_TERM_COLOR `#define GREEN_TERM_COLOR "\x1B[0;32m"`

18.30.1.91 HOOP_110X110 `#define HOOP_110X110 1`

18.30.1.92 HOOP_126X110 `#define HOOP_126X110 0`

18.30.1.93 HOOP_140X200 `#define HOOP_140X200 3`

18.30.1.94 HOOP_230X200 `#define HOOP_230X200 4`

18.30.1.95 HOOP_50X50 `#define HOOP_50X50 2`

18.30.1.96 LINETO `#define LINETO 0`

18.30.1.97 MOVETO `#define MOVETO 1`

18.30.1.98 N_PES_VERSIONS `#define N_PES_VERSIONS 13`

18.30.1.99 ObjectTypeRootEntry `#define ObjectTypeRootEntry 0x05`
the root entry

18.30.1.100 ObjectTypeStorage `#define ObjectTypeStorage 0x01`
a directory type object

18.30.1.101 ObjectTypeStream `#define ObjectTypeStream 0x02`
a file type object

18.30.1.102 ObjectTypeUnknown `#define ObjectTypeUnknown 0x00`
Type of directory object Probably unallocated

18.30.1.103 PES0001 `#define PES0001 0`

18.30.1.104 PES0020 `#define PES0020 1`

18.30.1.105 PES0022 `#define PES0022 2`

18.30.1.106 PES0030 `#define PES0030 3`

18.30.1.107 PES0040 `#define PES0040 4`

18.30.1.108 PES0050 `#define PES0050 5`

18.30.1.109 PES0055 `#define PES0055 6`

18.30.1.110 PES0056 `#define PES0056 7`

18.30.1.111 PES0060 `#define PES0060 8`

18.30.1.112 PES0070 `#define PES0070 9`

18.30.1.113 PES0080 `#define PES0080 10`

18.30.1.114 PES0090 `#define PES0090 11`

18.30.1.115 PES0100 `#define PES0100 12`

18.30.1.116 QUADTOCONTROL `#define QUADTOCONTROL 256`

18.30.1.117 QUADTOEND `#define QUADTOEND 512`

18.30.1.118 RED_TERM_COLOR `#define RED_TERM_COLOR "\x1B[0;31m"`

18.30.1.119 RESET_TERM_COLOR `#define RESET_TERM_COLOR "\033[0m"`

18.30.1.120 SVG_ATTRIBUTE `#define SVG_ATTRIBUTE 4`

18.30.1.121 SVG_CATCH_ALL `#define SVG_CATCH_ALL 5`

18.30.1.122 SVG_CREATOR_EMBROIDERMODDER `#define SVG_CREATOR_EMBROIDERMODDER 1`

18.30.1.123 SVG_CREATOR_ILLUSTRATOR `#define SVG_CREATOR_ILLUSTRATOR 2`

18.30.1.124 SVG_CREATOR_INKSCAPE `#define SVG_CREATOR_INKSCAPE 3`

18.30.1.125 SVG_CREATOR_NULL `#define SVG_CREATOR_NULL 0`

18.30.1.126 SVG_ELEMENT `#define SVG_ELEMENT 1`

18.30.1.127 SVG_EXPECT_ATTRIBUTE `#define SVG_EXPECT_ATTRIBUTE 2`

18.30.1.128 SVG_EXPECT_ELEMENT `#define SVG_EXPECT_ELEMENT 1`

18.30.1.129 SVG_EXPECT_NULL `#define SVG_EXPECT_NULL 0`

18.30.1.130 SVG_EXPECT_VALUE `#define SVG_EXPECT_VALUE 3`

18.30.1.131 SVG_MEDIA_PROPERTY `#define SVG_MEDIA_PROPERTY 3`

18.30.1.132 SVG_NULL `#define SVG_NULL 0`

18.30.1.133 SVG_PROPERTY `#define SVG_PROPERTY 2`

18.30.1.134 YELLOW_TERM_COLOR `#define YELLOW_TERM_COLOR "\x1B[1;33m"`

18.30.2 Typedef Documentation

18.30.2.1 bcf_directory `typedef struct _bcf_directory bcf_directory`

Todo possibly add a directory tree in the future.

18.30.2.2 bcf_directory_entry `typedef struct _bcf_directory_entry bcf_directory_entry`

18.30.2.3 bcf_file `typedef struct _bcf_file bcf_file`

18.30.2.4 bcf_file_difat `typedef struct _bcf_file_difat bcf_file_difat`

18.30.2.5 bcf_file_fat `typedef struct _bcf_file_fat bcf_file_fat`

18.30.2.6 bcf_file_header typedef struct `_bcf_file_header` `bcf_file_header`

Todo CLSID should be a separate type.

18.30.2.7 compress typedef struct `Compress` `compress`

18.30.2.8 huffman typedef struct `Huffman` `huffman`

18.30.2.9 StxThread typedef struct `StxThread_` `StxThread`

18.30.2.10 SubDescriptor typedef struct `SubDescriptor_` `SubDescriptor`

18.30.2.11 SvgAttribute typedef struct `SvgAttribute_` `SvgAttribute`

18.30.2.12 ThredExtension typedef struct `ThredExtension_` `ThredExtension`

18.30.2.13 ThredHeader typedef struct `ThredHeader_` `ThredHeader`

18.30.2.14 VipHeader typedef struct `VipHeader_` `VipHeader`

18.30.2.15 vp3Hoop typedef struct `_vp3Hoop` `vp3Hoop`

18.30.3 Enumeration Type Documentation

18.30.3.1 CSV_EXPECT enum `CSV_EXPECT`

Enumerator

CSV_EXPECT_NULL	
CSV_EXPECT_QUOTE1	
CSV_EXPECT_QUOTE2	
CSV_EXPECT_COMMA	

18.30.3.2 CSV_MODE enum `CSV_MODE`

Enumerator

CSV_MODE_NULL	
CSV_MODE_COMMENT	
CSV_MODE_VARIABLE	
CSV_MODE_THREAD	

Enumerator

CSV_MODE_STITCH	
-----------------	--

18.30.4 Function Documentation

18.30.4.1 bcf_difat_create() `bcf_file_difat * bcf_difat_create (`
 `FILE * file,`
 `unsigned int fatSectors,`
 `const unsigned int sectorSize)`

file fatSectors sectorSize

Returns

`bcf_file_difat*`

18.30.4.2 bcf_directory_free() `void bcf_directory_free (`
 `bcf_directory ** dir)`

dir

18.30.4.3 bcf_file_difat_free() `void bcf_file_difat_free (`
 `bcf_file_difat * difat)`

18.30.4.4 bcf_file_fat_free() `void bcf_file_fat_free (`
 `bcf_file_fat ** fat)`

18.30.4.5 bcf_file_free() `void bcf_file_free (`
 `bcf_file * bcfFile)`

bcfFile

18.30.4.6 bcfFile_read() `int bcfFile_read (`
 `FILE * file,`
 `bcf_file * bcfFile)`

file bcfFile

Returns

`int`

18.30.4.7 bcfFileFat_create() `bcf_file_fat * bcfFileFat_create (`
 `const unsigned int sectorSize)`

sectorSize

Returns

`bcf_file_fat*`

18.30.4.8 bcfFileHeader_isValid() int bcfFileHeader_isValid (
 bcf_file_header header)

18.30.4.9 bcfFileHeader_read() bcf_file_header bcfFileHeader_read (
 FILE * file)

file

Returns

bcf_file_header

18.30.4.10 binaryReadString() void binaryReadString (
 FILE * file,
 char * buffer,
 int maxLength)

file buffer maxLength

18.30.4.11 binaryReadUnicodeString() void binaryReadUnicodeString (
 FILE * file,
 char * buffer,
 const int stringLength)

file buffer stringLength

18.30.4.12 binaryWriteInt() void binaryWriteInt (
 FILE * f,
 int data)

f data

Todo replace with embInt_read

18.30.4.13 binaryWriteIntBE() void binaryWriteIntBE (
 FILE * f,
 int data)

f data

Todo replace with embInt_read

18.30.4.14 binaryWriteShort() void binaryWriteShort (
 FILE * f,
 short data)

f data

Todo replace with embInt_read

18.30.4.15 binaryWriteUInt() void binaryWriteUInt (
 FILE * f,
 unsigned int data)

f data

Todo replace with embInt_read

18.30.4.16 binaryWriteUIntBE() void binaryWriteUIntBE (
FILE * *f*,
unsigned int *data*)
f data

Todo replace with embInt_read

18.30.4.17 binaryWriteUShort() void binaryWriteUShort (
FILE * *f*,
unsigned short *data*)
f data

Todo replace with embInt_read

18.30.4.18 binaryWriteUShortBE() void binaryWriteUShortBE (
FILE * *f*,
unsigned short *data*)
f data

Todo replace with embInt_read

18.30.4.19 check_header_present() int check_header_present (
FILE * *file*,
int *minimum_header_length*)
file minimum_header_length

Returns

int

Checks that there are enough bytes to interpret the header, stops possible segfaults when reading in the header bytes.

Returns 0 if there aren't enough, or the length of the file if there are.

18.30.4.20 CompoundFileDirectory() [bcf_directory](#) * CompoundFileDirectory (
const unsigned int *maxNumberOfDirectoryEntries*)
maxNumberOfDirectoryEntries

Returns

[bcf_directory](#)*

18.30.4.21 CompoundFileDirectoryEntry() [bcf_directory_entry](#) * CompoundFileDirectoryEntry (
FILE * *file*)

file

Returns

[bcf_directory_entry](#)*

18.30.4.22 compress_get_bits() int compress_get_bits (
[compress](#) * *c*,
int *length*)

c length Returns .

18.30.4.23 compress_get_position() int compress_get_position (
 compress * c)

c . Returns the position as an int.

18.30.4.24 compress_get_token() int compress_get_token (
 compress * c)

c . Returns the token as an int.

18.30.4.25 compress_load_block() void compress_load_block (
 compress * c)

c . Returns nothing.

18.30.4.26 compress_load_character_huffman() void compress_load_character_huffman (
 compress * c)

Load character table to compress struct c. Returns nothing.

18.30.4.27 compress_load_character_length_huffman() void compress_load_character_length_huffman (
 compress * c)

c . Returns.

18.30.4.28 compress_load_distance_huffman() void compress_load_distance_huffman (
 compress * c)

c . Returns nothing.

18.30.4.29 compress_pop() int compress_pop (
 compress * c,
 int bit_count)

c bit_count . Returns.

18.30.4.30 compress_read_variable_length() int compress_read_variable_length (
 compress * c)

c. Returns.

18.30.4.31 copy_trim() char * copy_trim (
 char const * s)

s

Returns

char*

Todo decription

18.30.4.32 create_test_file_1() int create_test_file_1 (
 const char * outf)

18.30.4.33 create_test_file_2() int create_test_file_2 (
 const char * outf)

18.30.4.34 create_test_file_3() int create_test_file_3 (
 const char * outf)

18.30.4.35 decode_t01_record() `int decode_t01_record (`
 `unsigned char b[3],`
 `int * x,`
 `int * y,`
 `int * flags)`

b x y flags .

Todo remove the unused return argument.

18.30.4.36 decode_tajima_ternary() `void decode_tajima_ternary (`
 `unsigned char b[3],`
 `int * x,`
 `int * y)`

Decode the signed ternary of the tajima format from *b* to the position values *x* and *y*.
There is no return argument.

18.30.4.37 decodeNewStitch() `int decodeNewStitch (`
 `unsigned char value)`

value

Returns

`int`

18.30.4.38 emb_optOut() `char * emb_optOut (`
 `EmbReal num,`
 `char * str)`

Optimizes the number (*num*) for output to a text file and returns it as a string (*str*).
num str

Returns

`char*`

18.30.4.39 emb_readline() `int emb_readline (`
 `FILE * file,`
 `char * line,`
 `int maxLength)`

file line maxLength

Returns

`int`

18.30.4.40 embColor_read() `void embColor_read (`
 `FILE * f,`
 `EmbColor * c,`
 `int toRead)`

f c toRead

18.30.4.41 embColor_write() void embColor_write (

```
FILE * f,
EmbColor c,
int toWrite )
```

f c toWrite

18.30.4.42 embInt_read() void embInt_read (

```
FILE * f,
char * label,
void * b,
int mode )
```

f label b mode

Read and write system for multiple byte types.

The caller passes the function to read/write from, the memory location as a void pointer and a mode identifier that describes the type. This way we can abstract out the endianness of the system running the library and don't have to maintain many functions, just two.

18.30.4.43 embInt_write() void embInt_write (

```
FILE * f,
char * label,
void * b,
int mode )
```

f label b mode

18.30.4.44 encode_t01_record() void encode_t01_record (

```
unsigned char b[3],
int x,
int y,
int flags )
```

Encode into bytes *b* the values of the x-position *x*, y-position *y* and the *flags*.

18.30.4.45 encode_tajima_ternary() int encode_tajima_ternary (

```
unsigned char b[3],
int x,
int y )
```

Encode the signed ternary of the tajima format into *b* the position values *x* and *y*.

If the values of *x* or *y* fall outside of the valid range of -121 and +121 then it returns 0 and 1.

18.30.4.46 entriesInDifatSector() unsigned int entriesInDifatSector (

```
bcf_file_difat * fat )
```

fat

Returns

unsigned int

18.30.4.47 fpad() void fpad (

```
FILE * file,
char c,
int n )
```

f

Returns

int

18.30.4.48 fread_int16() short fread_int16 (
FILE * *f*)

f

Returns

short

18.30.4.49 fread_int32_be() int fread_int32_be (
FILE * *f*)

f

Returns

int

Todo replace with embInt_read

18.30.4.50 fread_uint16() unsigned short fread_uint16 (
FILE * *f*)

f

Returns

unsigned short

Todo replace with embInt_read

18.30.4.51 GetFile() FILE * GetFile (
bcf_file * *bcfFile*,
FILE * *file*,
char * *fileToFind*)

Get the File object.

bcfFile file fileToFind

Returns

FILE*

18.30.4.52 huffman_build_table() void huffman_build_table (
huffman * *h*)

These next 2 functions represent the **Huffman** class in tartarize's code. *h*

18.30.4.53 huffman_table_lookup() int * huffman_table_lookup (
huffman * *h*,
int *byte_lookup*,
int * *lengths*)

18.30.4.54 hus_compress() int hus_compress (
char * *data*,
int *length*,
char * *output*,
int * *output_length*)

data length output output_length . Returns whether it was successful as an int.

This avoids the now unnecessary compression by placing a minimal header of 6 bytes and using only literals in the huffman compressed part (see the sources above).

18.30.4.55 hus_decompress() int hus_decompress (

```

    char * data,
    int length,
    char * output,
    int * output_length )

```

data length *output* *output_length* . Returns whether the decompression was successful.

18.30.4.56 loadFatFromSector() void loadFatFromSector (

```

    bcf_file_fat * fat,
    FILE * file )

```

fat *file*

18.30.4.57 mitDecodeStitch() int mitDecodeStitch (

```

    unsigned char value )

```

value

Returns

int

18.30.4.58 mitEncodeStitch() unsigned char mitEncodeStitch (

```

    EmbReal value )

```

value

Returns

unsigned char

18.30.4.59 numberOfEntriesInDifatSector() unsigned int numberOfEntriesInDifatSector (

```

    bcf_file_difat * fat )

```

18.30.4.60 pfaffDecode() EmbReal pfaffDecode (

```

    unsigned char a1,
    unsigned char a2,
    unsigned char a3 )

```

Decode the bytes *a1*, *a2* and *a3* . Returns the EmbReal floating-point value.

18.30.4.61 pfaffEncode() void pfaffEncode (

```

    FILE * file,
    int dx,
    int dy,
    int flags )

```

file *dx* *dy* *flags*

18.30.4.62 printArcResults() void printArcResults (

```

    EmbReal bulge,
    EmbArc arc,
    EmbReal centerX,
    EmbReal centerY,
    EmbReal radius,
    EmbReal diameter,
    EmbReal chord,
    EmbReal chordMidX,

```

```
EmbReal chordMidY,  
EmbReal sagitta,  
EmbReal apothem,  
EmbReal incAngle,  
char clockwise )
```

18.30.4.63 read100() char read100 (
EmbPattern * pattern,
FILE * file)

18.30.4.64 read10o() char read10o (
EmbPattern * pattern,
FILE * file)

18.30.4.65 readArt() char readArt (
EmbPattern * pattern,
FILE * file)

18.30.4.66 readBmc() char readBmc (
EmbPattern * pattern,
FILE * file)

18.30.4.67 readBro() char readBro (
EmbPattern * pattern,
FILE * file)

18.30.4.68 readCnd() char readCnd (
EmbPattern * pattern,
FILE * file)

18.30.4.69 readCol() char readCol (
EmbPattern * pattern,
FILE * file)

18.30.4.70 readCsd() char readCsd (
EmbPattern * pattern,
FILE * file)

18.30.4.71 readCsv() char readCsv (
EmbPattern * pattern,
FILE * file)

18.30.4.72 readDat() char readDat (
EmbPattern * pattern,
FILE * file)

18.30.4.73 readDem() char readDem (
 EmbPattern * pattern,
 FILE * file)

18.30.4.74 readDescriptions() void readDescriptions (
 FILE * file,
 EmbPattern * pattern)

18.30.4.75 readDsb() char readDsb (
 EmbPattern * pattern,
 FILE * file)

18.30.4.76 readDst() char readDst (
 EmbPattern * pattern,
 FILE * file)

18.30.4.77 readDsz() char readDsz (
 EmbPattern * pattern,
 FILE * file)

18.30.4.78 ZSK USA Embroidery Format (.dsz) The ZSK USA dsz format is stitch-only.

18.30.4.79 readDxf() char readDxf (
 EmbPattern * pattern,
 FILE * file)

18.30.4.80 readEdr() char readEdr (
 EmbPattern * pattern,
 FILE * file)

18.30.4.81 Embird Embroidery Format (.edr) Stitch Only Format

18.30.4.82 readEmd() char readEmd (
 EmbPattern * pattern,
 FILE * file)

18.30.4.83 readExp() char readExp (
 EmbPattern * pattern,
 FILE * file)

18.30.4.84 readExy() char readExy (
 EmbPattern * pattern,
 FILE * file)

18.30.4.85 readEys() char readEys (
 EmbPattern * pattern,
 FILE * file)

18.30.4.86 Sierra Expanded Embroidery Format (.eys) Stitch Only Format.
Smoothie G-Code Embroidery Format (.fxy)?

18.30.4.87 readFeatherPatterns() void readFeatherPatterns (
FILE * *file*,
EmbPattern * *pattern*)

18.30.4.88 readFullSector() unsigned int readFullSector (
FILE * *file*,
bcf_file_difat * *bcfFile*,
unsigned int * *difatEntriesToRead*)
file bcfFile difatEntriesToRead

Returns

unsigned int

18.30.4.89 readFxy() char readFxy (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.90 Embroidery Format (.fxy) Stitch Only Format.

18.30.4.91 readGc() char readGc (
EmbPattern * *pattern*,
FILE * *file*)

Smoothie G-Code

Main Reference: Machinery's Handbook Guide A Guide to Tables, Formulas, & More in the 31st Edition by John Milton Amiss, Franklin D. Jones and Henry Ryffel

18.30.4.92 readGnc() char readGnc (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.93 Great Notions Embroidery Format (.gnc) Stitch Only Format.

18.30.4.94 readGt() char readGt (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.95 Gold Thread Embroidery Format (.gt) Stitch Only Format.

18.30.4.96 readHoopName() void readHoopName (
FILE * *file*,
EmbPattern * *pattern*)

18.30.4.97 readHus() char readHus (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.98 readImageString() void readImageString (
FILE * *file*,
EmbPattern * *pattern*)

18.30.4.99 readInb() char readInb (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.100 Inbro Embroidery Format (.inb) Stitch Only Format.

18.30.4.101 readInf() char readInf (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.102 Embroidery Color Format (.inf) Stitch Only Format.

18.30.4.103 readJef() char readJef (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.104 readKsm() char readKsm (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.105 readMax() char readMax (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.106 readMit() char readMit (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.107 Mitsubishi Embroidery Format (.mit) Stitch Only Format.

18.30.4.108 readMotifPatterns() void readMotifPatterns (
FILE * *file*,
EmbPattern * *pattern*)

18.30.4.109 readNew() char readNew (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.110 Ameco Embroidery Format (.new) Stitch Only Format.

18.30.4.111 readNextSector() void readNextSector (
FILE * *file*,
bcf_directory * *dir*)

file dir

18.30.4.112 readOfm() `char readOfm (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.113 readPcd() `char readPcd (`
 `EmbPattern * pattern,`
 `const char * fileName,`
 `FILE * file)`

18.30.4.114 Pfaff PCD File Format (.pcd) Stitch Only Format.

The format uses a signed 3 byte-length number type.

See the description here (5) for the overview of the format.

For an example of the format see (11).

18.30.4.115 readPcm() `char readPcm (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.116 Pfaff Embroidery Format (.pcm) The Pfaff pcm format is stitch-only.

18.30.4.117 readPcq() `char readPcq (`
 `EmbPattern * pattern,`
 `const char * fileName,`
 `FILE * file)`

18.30.4.118 Embroidery Format (.pcq) The Pfaff pcq format is stitch-only.

18.30.4.119 readPcs() `char readPcs (`
 `EmbPattern * pattern,`
 `const char * fileName,`
 `FILE * file)`

18.30.4.120 Embroidery Format (.pcs) The Pfaff pcs format is stitch-only.

18.30.4.121 readPec() `char readPec (`
 `EmbPattern * pattern,`
 `const char * fileName,`
 `FILE * file)`

18.30.4.122 readPecStitches() `void readPecStitches (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.123 Embroidery Format (.pec) The Brother pec format is stitch-only.

18.30.4.124 readPel() `char readPel (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.125 Embroidery Format (.pel) The Brother pel format is stitch-only.

18.30.4.126 readPem() `char readPem (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.127 Embroidery Format (.pec) The Brother pem format is stitch-only.

18.30.4.128 readPes() `char readPes (`
 `EmbPattern * pattern,`
 `const char * fileName,`
 `FILE * file)`

18.30.4.129 readPESHeaderV10() `void readPESHeaderV10 (`
 `FILE * file,`
 `EmbPattern * pattern)`

18.30.4.130 readPESHeaderV5() `void readPESHeaderV5 (`
 `FILE * file,`
 `EmbPattern * pattern)`

18.30.4.131 readPESHeaderV6() `void readPESHeaderV6 (`
 `FILE * file,`
 `EmbPattern * pattern)`

18.30.4.132 readPESHeaderV7() `void readPESHeaderV7 (`
 `FILE * file,`
 `EmbPattern * pattern)`

18.30.4.133 readPESHeaderV8() `void readPESHeaderV8 (`
 `FILE * file,`
 `EmbPattern * pattern)`

18.30.4.134 readPESHeaderV9() `void readPESHeaderV9 (`
 `FILE * file,`
 `EmbPattern * pattern)`

18.30.4.135 readPhb() `char readPhb (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.136 Embroidery Format (.pec) The Brother phb format is stitch-only.

18.30.4.137 readPhc() `char readPhc (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.138 Embroidery Format (.pec) The Brother phc format is stitch-only.

18.30.4.139 readPlt() `char readPlt (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.140 Embroidery Format (.plt) The AutoCAD plt format is stitch-only.

18.30.4.141 readProgrammableFills() `void readProgrammableFills (`
 `FILE * file,`
 `EmbPattern * pattern)`

18.30.4.142 readRgb() `char readRgb (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.143 Color File (.rgb) The RGB format is a color-only format to act as an external color file for other formats.

18.30.4.144 readSew() `char readSew (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.145 readShv() `char readShv (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.146 readSst() `char readSst (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.147 Embroidery Format (.sst) The Sunstar sst format is stitch-only.

18.30.4.148 readStx() `char readStx (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.149 readSvg() `char readSvg (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.150 readT01() `char readT01 (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.151 Embroidery Format (.pcq) The Pfaff t01 format is stitch-only.

18.30.4.152 readT09() `char readT09 (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.152.1 Embroidery Format (.pcq) The Pfaff t09 format is stitch-only.

18.30.4.153 readTap() `char readTap (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.154 readThr() `char readThr (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.155 Embroidery Format (.thr) The ThreadWorks thr format is stitch-only.

18.30.4.156 readThreads() `void readThreads (`
 `FILE * file,`
 `EmbPattern * pattern)`

18.30.4.157 readTxt() `char readTxt (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.158 File (.txt) The txt format is stitch-only and isn't associated with a specific company.

18.30.4.159 readU00() `char readU00 (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.160 Embroidery Format (.u00) The Barudan u00 format is stitch-only.

18.30.4.161 readU01() `char readU01 (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.162 Embroidery Format (.u00) The Barudan u01 format is stitch-only.

18.30.4.163 readVip() `char readVip (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.164 readVp3() `char readVp3 (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.165 readXxx() `char readXxx (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.166 readZsk() `char readZsk (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.30.4.167 **safe_free()** void safe_free (
void * data)
data

18.30.4.168 **stringInArray()** int stringInArray (
const char * s,
const char ** array)

Tests for the presence of a string *s* in the supplied *array*.
The end of the array is marked by an empty string.

Returns

0 if not present 1 if present.

18.30.4.169 **testEmbCircle()** int testEmbCircle (
void)

18.30.4.170 **testEmbCircle_2()** int testEmbCircle_2 (
void)

18.30.4.171 **testEmbFormat()** int testEmbFormat (
void)

18.30.4.172 **testGeomArc()** int testGeomArc (
void)

18.30.4.173 **testTangentPoints()** void testTangentPoints (
EmbCircle c,
EmbVector p,
EmbVector * t0,
EmbVector * t1)

18.30.4.174 **testThreadColor()** int testThreadColor (
void)

18.30.4.175 **write100()** char write100 (
EmbPattern * pattern,
FILE * file)

18.30.4.176 **write10o()** char write10o (
EmbPattern * pattern,
FILE * file)

18.30.4.177 write_24bit() void write_24bit (
FILE * *file*,
int *x*)
file x

18.30.4.178 writeArt() char writeArt (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.179 writeBmc() char writeBmc (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.180 writeBro() char writeBro (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.181 writeCnd() char writeCnd (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.182 writeCol() char writeCol (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.183 writeCsd() char writeCsd (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.184 writeCsv() char writeCsv (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.185 writeDat() char writeDat (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.186 writeDem() char writeDem (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.187 writeDsb() char writeDsb (
EmbPattern * *pattern*,
FILE * *file*)

18.30.4.188 writeDst() char writeDst (
 EmbPattern * pattern,
 FILE * file)

18.30.4.189 writeDsz() char writeDsz (
 EmbPattern * pattern,
 FILE * file)

18.30.4.190 writeDxf() char writeDxf (
 EmbPattern * pattern,
 FILE * file)

18.30.4.191 writeEdr() char writeEdr (
 EmbPattern * pattern,
 FILE * file)

18.30.4.192 writeEmd() char writeEmd (
 EmbPattern * pattern,
 FILE * file)

18.30.4.193 writeExp() char writeExp (
 EmbPattern * pattern,
 FILE * file)

18.30.4.194 writeExy() char writeExy (
 EmbPattern * pattern,
 FILE * file)

18.30.4.195 writeEys() char writeEys (
 EmbPattern * pattern,
 FILE * file)

18.30.4.196 writeFxy() char writeFxy (
 EmbPattern * pattern,
 FILE * file)

18.30.4.197 writeGc() char writeGc (
 EmbPattern * pattern,
 FILE * file)

18.30.4.198 writeGnc() char writeGnc (
 EmbPattern * pattern,
 FILE * file)

18.30.4.199 writeGt() char writeGt (
 EmbPattern * pattern,
 FILE * file)

18.30.4.200 writeHus() char writeHus (
 EmbPattern * pattern,
 FILE * file)

18.30.4.201 writeInb() char writeInb (
 EmbPattern * pattern,
 FILE * file)

18.30.4.202 writeInf() char writeInf (
 EmbPattern * pattern,
 FILE * file)

18.30.4.203 writeJef() char writeJef (
 EmbPattern * pattern,
 FILE * file)

18.30.4.204 writeKsm() char writeKsm (
 EmbPattern * pattern,
 FILE * file)

18.30.4.205 writeMax() char writeMax (
 EmbPattern * pattern,
 FILE * file)

18.30.4.206 writeMit() char writeMit (
 EmbPattern * pattern,
 FILE * file)

18.30.4.207 writeNew() char writeNew (
 EmbPattern * pattern,
 FILE * file)

18.30.4.208 writeOfm() char writeOfm (
 EmbPattern * pattern,
 FILE * file)

18.30.4.209 writePcd() char writePcd (
 EmbPattern * pattern,
 FILE * file)

18.30.4.210 writePcm() char writePcm (
 EmbPattern * pattern,
 FILE * file)

18.30.4.211 writePcq() char writePcq (
 EmbPattern * pattern,
 FILE * file)

18.30.4.212 writePcs() char writePcs (
 EmbPattern * pattern,
 FILE * file)

18.30.4.213 writePec() char writePec (
 EmbPattern * pattern,
 const char * fileName,
 FILE * file)

18.30.4.214 writePecStitches() void writePecStitches (
 EmbPattern * pattern,
 FILE * file,
 const char * filename)

18.30.4.215 writePel() char writePel (
 EmbPattern * pattern,
 FILE * file)

18.30.4.216 writePem() char writePem (
 EmbPattern * pattern,
 FILE * file)

18.30.4.217 writePes() char writePes (
 EmbPattern * pattern,
 const char * fileName,
 FILE * file)

18.30.4.218 writePhb() char writePhb (
 EmbPattern * pattern,
 FILE * file)

18.30.4.219 writePhc() char writePhc (
 EmbPattern * pattern,
 FILE * file)

18.30.4.220 writePlt() char writePlt (
 EmbPattern * pattern,
 FILE * file)

18.30.4.221 writeRgb() char writeRgb (
 EmbPattern * pattern,
 FILE * file)

18.30.4.222 writeSew() char writeSew (
 EmbPattern * pattern,
 FILE * file)

18.30.4.223 writeShv() char writeShv (
 EmbPattern * pattern,
 FILE * file)

18.30.4.224 writeSst() char writeSst (
 EmbPattern * pattern,
 FILE * file)

18.30.4.225 writeStx() char writeStx (
 EmbPattern * pattern,
 FILE * file)

18.30.4.226 writeSvg() char writeSvg (
 EmbPattern * pattern,
 FILE * file)

Writes the data from *pattern* to a file with the given *fileName*. Returns `true` if successful, otherwise returns `false`.

18.30.4.227 writeT01() char writeT01 (
 EmbPattern * pattern,
 FILE * file)

18.30.4.228 writeT09() char writeT09 (
 EmbPattern * pattern,
 FILE * file)

18.30.4.229 writeTap() char writeTap (
 EmbPattern * pattern,
 FILE * file)

18.30.4.230 writeThr() char writeThr (
 EmbPattern * pattern,
 FILE * file)

18.30.4.231 writeTxt() char writeTxt (
 EmbPattern * pattern,
 FILE * file)

18.30.4.232 writeU00() char writeU00 (
 EmbPattern * pattern,
 FILE * file)

18.30.4.233 writeU01() char writeU01 (
 EmbPattern * pattern,
 FILE * file)

18.30.4.234 writeVip() char writeVip (
 EmbPattern * pattern,
 FILE * file)

18.30.4.235 writeVp3() char writeVp3 (
 EmbPattern * pattern,
 FILE * file)

18.30.4.236 writeXxx() char writeXxx (
 EmbPattern * pattern,
 FILE * file)

18.30.4.237 writeZsk() char writeZsk (
 EmbPattern * pattern,
 FILE * file)

18.30.5 Variable Documentation

18.30.5.1 imageWithFrame const char imageWithFrame[38][48] [extern]

18.31 embroidery_internal.h

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

```
1 #ifndef LIBEMBROIDERY_INTERNAL_HEADER__
2 #define LIBEMBROIDERY_INTERNAL_HEADER__
3
4 #include "embroidery.h"
5
10 /* For FILE * */
11 #include <stdio.h>
12
16 #define CompoundFileSector_MaxRegSector 0xFFFFFFFFFA
17 #define CompoundFileSector_DIFAT_Sector 0xFFFFFFFFFC
18 #define CompoundFileSector_FAT_Sector 0xFFFFFFFFFD
19 #define CompoundFileSector_EndOfChain 0xFFFFFFFFFE
20 #define CompoundFileSector_FreeSector 0xFFFFFFFFFF
21
25 #define ObjectTypeUnknown 0x00
26 #define ObjectTypeStorage 0x01
27 #define ObjectTypeStream 0x02
28 #define ObjectTypeRootEntry 0x05
33 #define CompoundFileStreamId_MaxRegularStreamId 0xFFFFFFFFFA
34 #define CompoundFileStreamId_NoStream 0xFFFFFFFFFF
36 #define ELEMENT_XML 0
```

```

37 #define ELEMENT_A 1
38 #define ELEMENT_ANIMATE 2
39 #define ELEMENT_ANIMATECOLOR 3
40 #define ELEMENT_ANIMATEMOTION 4
41 #define ELEMENT_ANIMATETRANSFORM 5
42 #define ELEMENT_ANIMATION 6
43 #define ELEMENT_AUDIO 7
44 #define ELEMENT_CIRCLE 8
45 #define ELEMENT_DEFS 9
46 #define ELEMENT_DESC 10
47 #define ELEMENT_DISCARD 11
48 #define ELEMENT_ELLIPSE 12
49 #define ELEMENT_FONT 13
50 #define ELEMENT_FONT_FACE 14
51 #define ELEMENT_FONT_FACE_SRC 15
52 #define ELEMENT_FONT_FACE_URI 16
53 #define ELEMENT_FOREIGN_OBJECT 17
54 #define ELEMENT_G 18
55 #define ELEMENT_GLYPH 19
56 #define ELEMENT_HANDLER 20
57 #define ELEMENT_HKERN 21
58 #define ELEMENT_IMAGE 22
59 #define ELEMENT_LINE 23
60 #define ELEMENT_LINEAR_GRADIENT 24
61 #define ELEMENT_LISTENER 25
62 #define ELEMENT_METADATA 26
63 #define ELEMENT_MISSING_GLYPH 27
64 #define ELEMENT_MPATH 28
65 #define ELEMENT_PATH 29
66 #define ELEMENT_POLYGON 30
67 #define ELEMENT_POLYLINE 31
68 #define ELEMENT_PREFETCH 32
69 #define ELEMENT_RADIAL_GRADIENT 33
70 #define ELEMENT_RECT 34
71 #define ELEMENT_SCRIPT 35
72 #define ELEMENT_SET 36
73 #define ELEMENT_SOLID_COLOR 37
74 #define ELEMENT_STOP 38
75 #define ELEMENT_SVG 39
76 #define ELEMENT_SWITCH 40
77 #define ELEMENT_TBREAK 41
78 #define ELEMENT_TEXT 42
79 #define ELEMENT_TEXT_AREA 43
80 #define ELEMENT_TITLE 44
81 #define ELEMENT_TSPAN 45
82 #define ELEMENT_USE 46
83 #define ELEMENT_VIDEO 47
84
85 /* INTERNAL DEFINES */
86 #define RED_TERM_COLOR "\x1B[0;31m"
87 #define GREEN_TERM_COLOR "\x1B[0;32m"
88 #define YELLOW_TERM_COLOR "\x1B[1;33m"
89 #define RESET_TERM_COLOR "\033[0m"
90
91 #define HOOP_126X110 0
92 #define HOOP_110X110 1
93 #define HOOP_50X50 2
94 #define HOOP_140X200 3
95 #define HOOP_230X200 4
96
97 #define EMB_MIN(A, B) ((A) < (B)) ? (A) : (B)
98 #define EMB_MAX(A, B) ((A) > (B)) ? (A) : (B)
99
100 /* Libembroidery's handling of integer types.
101 */
102 #define EMB_BIG_ENDIAN 0
103 #define EMB_LITTLE_ENDIAN 1
104
105 #define ENDIAN_HOST EMB_LITTLE_ENDIAN
106
107 #define EMB_INT16_BIG 2
108 #define EMB_INT16_LITTLE 3
109 #define EMB_INT32_BIG 4
110 #define EMB_INT32_LITTLE 5
111
112 #define PES0001 0
113 #define PES0020 1
114 #define PES0022 2
115 #define PES0030 3
116 #define PES0040 4
117 #define PES0050 5
118 #define PES0055 6
119 #define PES0056 7
120 #define PES0060 8
121 #define PES0070 9
122 #define PES0080 10
123 #define PES0090 11

```

```

124 #define PES0100          12
125 #define N_PES_VERSIONS 13
126
127 /* DXF Version Identifiers */
128 #define DXF_VERSION_R10 "AC1006"
129 #define DXF_VERSION_R11 "AC1009"
130 #define DXF_VERSION_R12 "AC1009"
131 #define DXF_VERSION_R13 "AC1012"
132 #define DXF_VERSION_R14 "AC1014"
133 #define DXF_VERSION_R15 "AC1015"
134 #define DXF_VERSION_R18 "AC1018"
135 #define DXF_VERSION_R21 "AC1021"
136 #define DXF_VERSION_R24 "AC1024"
137 #define DXF_VERSION_R27 "AC1027"
138
139 #define DXF_VERSION_2000 "AC1015"
140 #define DXF_VERSION_2002 "AC1015"
141 #define DXF_VERSION_2004 "AC1018"
142 #define DXF_VERSION_2006 "AC1018"
143 #define DXF_VERSION_2007 "AC1021"
144 #define DXF_VERSION_2009 "AC1021"
145 #define DXF_VERSION_2010 "AC1024"
146 #define DXF_VERSION_2013 "AC1027"
147
148 #define SVG_CREATOR_NULL          0
149 #define SVG_CREATOR_EMBROIDERMODDER 1
150 #define SVG_CREATOR_ILLUSTRATOR   2
151 #define SVG_CREATOR_INKSCAPE     3
152
153 #define SVG_EXPECT_NULL          0
154 #define SVG_EXPECT_ELEMENT      1
155 #define SVG_EXPECT_ATTRIBUTE    2
156 #define SVG_EXPECT_VALUE        3
157
158 /* SVG_TYPES
159 * -----
160 */
161 #define SVG_NULL          0
162 #define SVG_ELEMENT      1
163 #define SVG_PROPERTY     2
164 #define SVG_MEDIA_PROPERTY 3
165 #define SVG_ATTRIBUTE     4
166 #define SVG_CATCH_ALL    5
167
168 /* path flag codes */
169 #define LINETO          0
170 #define MOVETO          1
171 #define BULGETOCONTROL  2
172 #define BULGETOEND      4
173 #define ELLIPSETORAD    8
174 #define ELLIPSETOEND    16
175 #define CUBICTOCONTROL1  32
176 #define CUBICTOCONTROL2  64
177 #define CUBICTOEND      128
178 #define QUADTOCONTROL    256
179 #define QUADTOEND       512
180
181 /* STRUCTS
182 *****/
183
184 /* double-indirection file allocation table references */
185
186 typedef struct _bcf_file_difat
187 {
188     unsigned int fatSectorCount;
189     unsigned int fatSectorEntries[109];
190     unsigned int sectorSize;
191 } bcf_file_difat;
192
193 typedef struct _bcf_file_fat
194 {
195     int fatEntryCount;
196     unsigned int fatEntries[255]; /* maybe make this dynamic */
197     unsigned int numberOfEntriesInFatSector;
198 } bcf_file_fat;
199
200 typedef struct _bcf_directory_entry
201 {
202     char directoryEntryName[32];
203     unsigned short directoryEntryNameLength;
204     unsigned char objectType;
205     unsigned char colorFlag;
206     unsigned int leftSiblingId;
207     unsigned int rightSiblingId;
208     unsigned int childId;
209     unsigned char CLSID[16];
210     unsigned int stateBits;

```



```

223     EmbTime                creationTime;
224     EmbTime                modifiedTime;
225     unsigned int           startingSectorLocation;
226     unsigned long          streamSize; /* should be long long but in our case we shouldn't need
    it, and hard to support on c89 cross platform */
227     unsigned int           streamSizeHigh; /* store the high int of streamsize */
228     struct _bcf_directory_entry* next;
229 } bcf_directory_entry;
230
231 typedef struct _bcf_directory
232 {
233     bcf_directory_entry* dirEntries;
234     unsigned int         maxNumberOfDirectoryEntries;
235 } bcf_directory;
236
237 typedef struct _bcf_file_header
238 {
239     unsigned char    signature[8];
240     unsigned char    CLSID[16];
241     unsigned short   minorVersion;
242     unsigned short   majorVersion;
243     unsigned short   byteOrder;
244     unsigned short   sectorShift;
245     unsigned short   miniSectorShift;
246     unsigned short   reserved1;
247     unsigned int      reserved2;
248     unsigned int      numberOfDirectorySectors;
249     unsigned int      numberOfFATSectors;
250     unsigned int      firstDirectorySectorLocation;
251     unsigned int      transactionSignatureNumber;
252     unsigned int      miniStreamCutoffSize;
253     unsigned int      firstMiniFATSectorLocation;
254     unsigned int      numberOfMiniFatSectors;
255     unsigned int      firstDifatSectorLocation;
256     unsigned int      numberOfDifatSectors;
257 } bcf_file_header;
258
259 typedef struct _bcf_file
260 {
261     bcf_file_header header;
262     bcf_file_difat* difat;
263     bcf_file_fat* fat;
264     bcf_directory* directory;
265 } bcf_file;
266
267 typedef struct _vp3Hoop
268 {
269     int    right;
270     int    bottom;
271     int    left;
272     int    top;
273     int    threadLength;
274     char    unknown2;
275     unsigned char    numberOfColors;
276     unsigned short    unknown3;
277     int    unknown4;
278     int    numberOfBytesRemaining;
279
280     int    xOffset;
281     int    yOffset;
282
283     unsigned char    byte1;
284     unsigned char    byte2;
285     unsigned char    byte3;
286
287     /* Centered hoop dimensions */
288     int    right2;
289     int    left2;
290     int    bottom2;
291     int    top2;
292
293     int    width;
294     int    height;
295 } vp3Hoop;
296
297 typedef struct ThredHeader_ /* thred file header */
298 {
299     unsigned int    sigVersion; /* signature and version */
300     unsigned int    length; /* length of ThredHeader + length of stitch data */
301     unsigned short    numStiches; /* number of stitches */
302     unsigned short    hoopSize; /* size of hoop */
303     unsigned short    reserved[7]; /* reserved for expansion */
304 } ThredHeader;
305
306 typedef struct ThredExtension_ /* thred v1.0 file header extension */
307 {
308     float    hoopX; /* hoop size x dimension in 1/6 mm units */

```

```

334     float hoopY;                /* hoop size y dimension in 1/6 mm units */
335     float stitchGranularity;    /* stitches per millimeter--not implemented */
336     char creatorName[50];       /* name of the file creator */
337     char modifierName[50];      /* name of last file modifier */
338     char auxFormat;             /* auxiliary file format, 0=PCS,1=DST,2=PES */
339     char reserved[31];          /* reserved for expansion */
340 } ThredExtension;
341
342 typedef struct SubDescriptor_
343 {
344     int someNum;
345     int someInt;
346     int someOtherInt;
347     char* colorCode;
348     char* colorName;
349 } SubDescriptor;
350
351 typedef struct StxThread_
352 {
353     char* colorCode;
354     char* colorName;
355     char* sectionName;
356     SubDescriptor* subDescriptors;
357     EmbColor stxColor;
358 } StxThread;
359
360 typedef struct VipHeader_ {
361     int magicCode;
362     int numberOfStitches;
363     int numberOfColors;
364     short positiveXHoopSize;
365     short positiveYHoopSize;
366     short negativeXHoopSize;
367     short negativeYHoopSize;
368     int attributeOffset;
369     int xOffset;
370     int yOffset;
371     unsigned char stringVal[8];
372     short unknown;
373     int colorLength;
374 } VipHeader;
375
376 typedef enum
377 {
378     CSV_EXPECT_NULL,
379     CSV_EXPECT_QUOTE1,
380     CSV_EXPECT_QUOTE2,
381     CSV_EXPECT_COMMA
382 } CSV_EXPECT;
383
384 typedef enum
385 {
386     CSV_MODE_NULL,
387     CSV_MODE_COMMENT,
388     CSV_MODE_VARIABLE,
389     CSV_MODE_THREAD,
390     CSV_MODE_STITCH
391 } CSV_MODE;
392
393 typedef struct SvgAttribute_
394 {
395     char* name;
396     char* value;
397 } SvgAttribute;
398
399 typedef struct Huffman {
400     int default_value;
401     int lengths[1000];
402     int nlengths;
403     int table[1000];
404     int table_width;
405     int ntable;
406 } Huffman;
407
408 typedef struct Compress {
409     int bit_position;
410     char *input_data;
411     int input_length;
412     int bits_total;
413     int block_elements;
414     Huffman character_length_huffman;
415     Huffman character_huffman;
416     Huffman distance_huffman;
417 } Compress;
418
419 /* Function Declarations
420 *****/

```

```

453 void huffman_build_table(huffman *h);
454 int *huffman_table_lookup(huffman *h, int byte_lookup, int *lengths);
455
456 int compress_get_bits(compress *c, int length);
457 int compress_pop(compress *c, int bit_count);
458 int compress_read_variable_length(compress *c);
459 void compress_load_character_length_huffman(compress *c);
460 void compress_load_character_huffman(compress *c);
461 void compress_load_distance_huffman(compress *c);
462 void compress_load_block(compress *c);
463 int compress_get_token(compress *c);
464 int compress_get_position(compress *c);
465
466 void readPecStitches(EmbPattern* pattern, FILE* file);
467 void writePecStitches(EmbPattern* pattern, FILE* file, const char* filename);
468
469 int decodeNewStitch(unsigned char value);
470
471 void pfaffEncode(FILE* file, int x, int y, int flags);
472 EmbReal pfaffDecode(unsigned char a1, unsigned char a2, unsigned char a3);
473
474 unsigned char mitEncodeStitch(EmbReal value);
475 int mitDecodeStitch(unsigned char value);
476
477 int encode_tajima_ternary(unsigned char b[3], int x, int y);
478 void decode_tajima_ternary(unsigned char b[3], int *x, int *y);
479
480 void encode_t01_record(unsigned char b[3], int x, int y, int flags);
481 int decode_t01_record(unsigned char b[3], int *x, int *y, int *flags);
482 void readPESHeaderV5(FILE* file, EmbPattern* pattern);
483 void readPESHeaderV6(FILE* file, EmbPattern* pattern);
484 void readPESHeaderV7(FILE* file, EmbPattern* pattern);
485 void readPESHeaderV8(FILE* file, EmbPattern* pattern);
486 void readPESHeaderV9(FILE* file, EmbPattern* pattern);
487 void readPESHeaderV10(FILE* file, EmbPattern* pattern);
488
489 void readDescriptions(FILE* file, EmbPattern* pattern);
490 void readHoopName(FILE* file, EmbPattern* pattern);
491 void readImageString(FILE* file, EmbPattern* pattern);
492 void readProgrammableFills(FILE* file, EmbPattern* pattern);
493 void readMotifPatterns(FILE* file, EmbPattern* pattern);
494 void readFeatherPatterns(FILE* file, EmbPattern* pattern);
495 void readThreads(FILE* file, EmbPattern* pattern);
496
497 void embInt_read(FILE* f, char *label, void *b, int mode);
498 void embInt_write(FILE* f, char *label, void *b, int mode);
499 int emb_readline(FILE* file, char *line, int maxLength);
500
501 int bcfFile_read(FILE* file, bcf_file* bcfFile);
502 FILE* GetFile(bcf_file* bcfFile, FILE* file, char* fileToFind);
503 void bcf_file_free(bcf_file* bcfFile);
504
505 void binaryReadString(FILE* file, char *buffer, int maxLength);
506 void binaryReadUnicodeString(FILE* file, char *buffer, const int stringLength);
507
508 int stringInArray(const char *s, const char **array);
509 void fpad(FILE *f, char c, int n);
510 char *copy_trim(char const *s);
511 char* emb_optOut(EmbReal num, char* str);
512
513 void write_24bit(FILE* file, int);
514 int check_header_present(FILE* file, int minimum_header_length);
515
516 unsigned short fread_uint16(FILE *file);
517 short fread_int16(FILE* f);
518 int fread_int32_be(FILE* f);
519 void safe_free(void *data);
520 void embInt_read(FILE* f, char *label, void *b, int mode);
521
522 void binaryWriteUIntBE(FILE* f, unsigned int data);
523 void binaryWriteUInt(FILE* f, unsigned int data);
524 void binaryWriteIntBE(FILE* f, int data);
525 void binaryWriteInt(FILE* f, int data);
526 void binaryWriteUShort(FILE* f, unsigned short data);
527 void binaryWriteUShortBE(FILE* f, unsigned short data);
528 void binaryWriteShort(FILE* f, short data);
529
530 bcf_file_difat* bcf_difat_create(FILE* file, unsigned int fatSectors, const unsigned int sectorSize);
531 unsigned int readFullSector(FILE* file, bcf_file_difat* bcfFile, unsigned int*
    numberOfDifatEntriesStillToRead);
532 unsigned int numberOfEntriesInDifatSector(bcf_file_difat* fat);
533 void bcf_file_difat_free(bcf_file_difat* difat);
534
535 unsigned int entriesInDifatSector(bcf_file_difat* fat);
536 bcf_file_fat* bcfFileFat_create(const unsigned int sectorSize);
537 void loadFatFromSector(bcf_file_fat* fat, FILE* file);
538 void bcf_file_fat_free(bcf_file_fat** fat);

```

```

539
540 bcf_directory_entry* CompoundFileDirectoryEntry(FILE* file);
541 bcf_directory* CompoundFileDirectory(const unsigned int maxNumberOfDirectoryEntries);
542 void readNextSector(FILE* file, bcf_directory* dir);
543 void bcf_directory_free(bcf_directory** dir);
544
545 bcf_file_header bcfFileHeader_read(FILE* file);
546 int bcfFileHeader_isValid(bcf_file_header header);
547
548 int hus_compress(char* input, int size, char* output, int *out_size);
549 int hus_decompress(char* input, int size, char* output, int *out_size);
550
551 int encode_tajima_ternary(unsigned char b[3], int x, int y);
552 void decode_tajima_ternary(unsigned char b[3], int *x, int *y);
553 void testTangentPoints(EmbCircle c, EmbVector p, EmbVector *t0, EmbVector *t1);
554 void printArcResults(EmbReal bulge, EmbArc arc,
555                     EmbReal centerX, EmbReal centerY,
556                     EmbReal radius, EmbReal diameter,
557                     EmbReal chord,
558                     EmbReal chordMidX, EmbReal chordMidY,
559                     EmbReal sagitta, EmbReal apothem,
560                     EmbReal incAngle, char clockwise);
561 int create_test_file_1(const char* outf);
562 int create_test_file_2(const char* outf);
563 int create_test_file_3(const char* outf);
564 int testEmbCircle(void);
565 int testEmbCircle_2(void);
566 int testGeomArc(void);
567 int testThreadColor(void);
568 int testEmbFormat(void);
569
570 void embColor_read(FILE *f, EmbColor *c, int toRead);
571 void embColor_write(FILE *f, EmbColor c, int toWrite);
572
573 char read100(EmbPattern *pattern, FILE* file);
574 char write100(EmbPattern *pattern, FILE* file);
575 char read10o(EmbPattern *pattern, FILE* file);
576 char write10o(EmbPattern *pattern, FILE* file);
577 char readArt(EmbPattern *pattern, FILE* file);
578 char writeArt(EmbPattern *pattern, FILE* file);
579 char readBmc(EmbPattern *pattern, FILE* file);
580 char writeBmc(EmbPattern *pattern, FILE* file);
581 char readBro(EmbPattern *pattern, FILE* file);
582 char writeBro(EmbPattern *pattern, FILE* file);
583 char readCnd(EmbPattern *pattern, FILE* file);
584 char writeCnd(EmbPattern *pattern, FILE* file);
585 char readCol(EmbPattern *pattern, FILE* file);
586 char writeCol(EmbPattern *pattern, FILE* file);
587 char readCsd(EmbPattern *pattern, FILE* file);
588 char writeCsd(EmbPattern *pattern, FILE* file);
589 char readCsv(EmbPattern *pattern, FILE* file);
590 char writeCsv(EmbPattern *pattern, FILE* file);
591 char readDat(EmbPattern *pattern, FILE* file);
592 char writeDat(EmbPattern *pattern, FILE* file);
593 char readDem(EmbPattern *pattern, FILE* file);
594 char writeDem(EmbPattern *pattern, FILE* file);
595 char readDsb(EmbPattern *pattern, FILE* file);
596 char writeDsb(EmbPattern *pattern, FILE* file);
597 char readDst(EmbPattern *pattern, FILE* file);
598 char writeDst(EmbPattern *pattern, FILE* file);
599 char readDsz(EmbPattern *pattern, FILE* file);
600 char writeDsz(EmbPattern *pattern, FILE* file);
601 char readDxf(EmbPattern *pattern, FILE* file);
602 char writeDxf(EmbPattern *pattern, FILE* file);
603 char readEdr(EmbPattern *pattern, FILE* file);
604 char writeEdr(EmbPattern *pattern, FILE* file);
605 char readEmd(EmbPattern *pattern, FILE* file);
606 char writeEmd(EmbPattern *pattern, FILE* file);
607 char readExp(EmbPattern *pattern, FILE* file);
608 char writeExp(EmbPattern *pattern, FILE* file);
609 char readExy(EmbPattern *pattern, FILE* file);
610 char writeExy(EmbPattern *pattern, FILE* file);
611 char readEys(EmbPattern *pattern, FILE* file);
612 char writeEys(EmbPattern *pattern, FILE* file);
613 char readFxy(EmbPattern *pattern, FILE* file);
614 char writeFxy(EmbPattern *pattern, FILE* file);
615 char readGc(EmbPattern *pattern, FILE* file);
616 char writeGc(EmbPattern *pattern, FILE* file);
617 char readGnc(EmbPattern *pattern, FILE* file);
618 char writeGnc(EmbPattern *pattern, FILE* file);
619 char readGt(EmbPattern *pattern, FILE* file);
620 char writeGt(EmbPattern *pattern, FILE* file);
621 char readHus(EmbPattern *pattern, FILE* file);
622 char writeHus(EmbPattern *pattern, FILE* file);
623 char readInb(EmbPattern *pattern, FILE* file);
624 char writeInb(EmbPattern *pattern, FILE* file);
625 char readInf(EmbPattern *pattern, FILE* file);

```

```

626 char writeInf(EmbPattern *pattern, FILE* file);
627 char readJef(EmbPattern *pattern, FILE* file);
628 char writeJef(EmbPattern *pattern, FILE* file);
629 char readKsm(EmbPattern *pattern, FILE* file);
630 char writeKsm(EmbPattern *pattern, FILE* file);
631 char readMax(EmbPattern *pattern, FILE* file);
632 char writeMax(EmbPattern *pattern, FILE* file);
633 char readMit(EmbPattern *pattern, FILE* file);
634 char writeMit(EmbPattern *pattern, FILE* file);
635 char readNew(EmbPattern *pattern, FILE* file);
636 char writeNew(EmbPattern *pattern, FILE* file);
637 char readOfm(EmbPattern *pattern, FILE* file);
638 char writeOfm(EmbPattern *pattern, FILE* file);
639 char readPcd(EmbPattern *pattern, const char *fileName, FILE* file);
640 char writePcd(EmbPattern *pattern, FILE* file);
641 char readPcm(EmbPattern *pattern, FILE* file);
642 char writePcm(EmbPattern *pattern, FILE* file);
643 char readPcq(EmbPattern *pattern, const char *fileName, FILE* file);
644 char writePcq(EmbPattern *pattern, FILE* file);
645 char readPcs(EmbPattern *pattern, const char *fileName, FILE* file);
646 char writePcs(EmbPattern *pattern, FILE* file);
647 char readPec(EmbPattern *pattern, const char *fileName, FILE* file);
648 char writePec(EmbPattern *pattern, const char *fileName, FILE* file);
649 char readPel(EmbPattern *pattern, FILE *file);
650 char writePel(EmbPattern *pattern, FILE *file);
651 char readPem(EmbPattern *pattern, FILE *file);
652 char writePem(EmbPattern *pattern, FILE *file);
653 char readPes(EmbPattern *pattern, const char *fileName, FILE* file);
654 char writePes(EmbPattern *pattern, const char *fileName, FILE* file);
655 char readPhb(EmbPattern *pattern, FILE* file);
656 char writePhb(EmbPattern *pattern, FILE *file);
657 char readPhc(EmbPattern *pattern, FILE* file);
658 char writePhc(EmbPattern *pattern, FILE *file);
659 char readPlt(EmbPattern *pattern, FILE* file);
660 char writePlt(EmbPattern *pattern, FILE* file);
661 char readRgb(EmbPattern *pattern, FILE* file);
662 char writeRgb(EmbPattern *pattern, FILE* file);
663 char readSew(EmbPattern *pattern, FILE* file);
664 char writeSew(EmbPattern *pattern, FILE* file);
665 char readShv(EmbPattern *pattern, FILE* file);
666 char writeShv(EmbPattern *pattern, FILE *file);
667 char readSst(EmbPattern *pattern, FILE* file);
668 char writeSst(EmbPattern *pattern, FILE *file);
669 char readStx(EmbPattern *pattern, FILE* file);
670 char writeStx(EmbPattern *pattern, FILE *file);
671 char readSvg(EmbPattern *pattern, FILE* file);
672 char writeSvg(EmbPattern *pattern, FILE* file);
673 char readT01(EmbPattern *pattern, FILE* file);
674 char writeT01(EmbPattern *pattern, FILE* file);
675 char readT09(EmbPattern *pattern, FILE* file);
676 char writeT09(EmbPattern *pattern, FILE* file);
677 char readTap(EmbPattern *pattern, FILE* file);
678 char writeTap(EmbPattern *pattern, FILE* file);
679 char readThr(EmbPattern *pattern, FILE* file);
680 char writeThr(EmbPattern *pattern, FILE* file);
681 char readTxt(EmbPattern *pattern, FILE* file);
682 char writeTxt(EmbPattern *pattern, FILE* file);
683 char readU00(EmbPattern *pattern, FILE* file);
684 char writeU00(EmbPattern *pattern, FILE *file);
685 char readU01(EmbPattern *pattern, FILE* file);
686 char writeU01(EmbPattern *pattern, FILE *file);
687 char readVip(EmbPattern *pattern, FILE* file);
688 char writeVip(EmbPattern *pattern, FILE* file);
689 char readVp3(EmbPattern *pattern, FILE* file);
690 char writeVp3(EmbPattern *pattern, FILE* file);
691 char readXxx(EmbPattern *pattern, FILE* file);
692 char writeXxx(EmbPattern *pattern, FILE* file);
693 char readZsk(EmbPattern *pattern, FILE* file);
694 char writeZsk(EmbPattern *pattern, FILE* file);
695
696 extern const char imageWithFrame[38][48];
697
698 #endif

```

18.32 extern/libembroidery/src/encoding.c File Reference

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "embroidery_internal.h"

```

Functions

- void `write_24bit` (FILE *file, int)
file x
- `EmbColor embColor_fromHexStr` (char *val)
Converts a 6 digit hex string (I.E. "00FF00") into an EmbColor and returns it.
- void `reverse_byte_order` (void *b, int bytes)
- int `decode_t01_record` (unsigned char b[3], int *x, int *y, int *flags)
- void `encode_t01_record` (unsigned char b[3], int x, int y, int flags)
- int `encode_tajima_ternary` (unsigned char b[3], int x, int y)
- void `decode_tajima_ternary` (unsigned char b[3], int *x, int *y)
- void `pfaffEncode` (FILE *file, int dx, int dy, int flags)
- `EmbReal pfaffDecode` (unsigned char a1, unsigned char a2, unsigned char a3)
- unsigned char `mitEncodeStitch` (`EmbReal` value)
value
- int `mitDecodeStitch` (unsigned char value)
value
- int `decodeNewStitch` (unsigned char value)
value
- void `emblnt_read` (FILE *f, char *label, void *b, int mode)
- void `emblnt_write` (FILE *f, char *label, void *b, int mode)

18.32.1 Detailed Description

The functions in this file are grouped together to aid the developer's understanding of the similarities between the file formats. This also helps reduce errors between reimplementation of the same idea.

For example: the Tajima ternary encoding of positions is used by at least 4 formats and the only part that changes is the flag encoding.

18.32.2 Function Documentation

18.32.2.1 `decode_t01_record()` int decode_t01_record (
 unsigned char *b*[3],
 int * *x*,
 int * *y*,
 int * *flags*)

b x y flags .

Todo remove the unused return argument.

18.32.2.2 `decode_tajima_ternary()` void decode_tajima_ternary (
 unsigned char *b*[3],
 int * *x*,
 int * *y*)

Decode the signed ternary of the tajima format from *b* to the position values *x* and *y*.
 There is no return argument.

18.32.2.3 `decodeNewStitch()` int decodeNewStitch (
 unsigned char *value*)

value

Returns

int

18.32.2.4 embColor_fromHexStr() `EmbColor embColor_fromHexStr (`
`char * val)`

Converts a 6 digit hex string (I.E. "00FF00") into an EmbColor and returns it.
val 6 byte code describing the color as a hex string, doesn't require null termination.

Returns

EmbColor the same color as our internal type.

18.32.2.5 embInt_read() `void embInt_read (`
`FILE * f,`
`char * label,`
`void * b,`
`int mode)`

f label b mode

Read and write system for multiple byte types.

The caller passes the function to read/write from, the memory location as a void pointer and a mode identifier that describes the type. This way we can abstract out the endianness of the system running the library and don't have to maintain many functions, just two.

18.32.2.6 embInt_write() `void embInt_write (`
`FILE * f,`
`char * label,`
`void * b,`
`int mode)`

f label b mode

18.32.2.7 encode_t01_record() `void encode_t01_record (`
`unsigned char b[3],`
`int x,`
`int y,`
`int flags)`

Encode into bytes *b* the values of the x-position *x*, y-position *y* and the *flags*.

18.32.2.8 encode_tajima_ternary() `int encode_tajima_ternary (`
`unsigned char b[3],`
`int x,`
`int y)`

Encode the signed ternary of the tajima format into *b* the position values *x* and *y*.

If the values of *x* or *y* fall outside of the valid range of -121 and +121 then it returns 0 and 1.

18.32.2.9 mitDecodeStitch() `int mitDecodeStitch (`
`unsigned char value)`

value

Returns

int

18.32.2.10 mitEncodeStitch() `unsigned char mitEncodeStitch (`
`EmbReal value)`

value

Returns

unsigned char

18.32.2.11 pfaffDecode() `EmbReal pfaffDecode (`
 unsigned char *a1*,
 unsigned char *a2*,
 unsigned char *a3*)

Decode the bytes *a1*, *a2* and *a3*. Returns the EmbReal floating-point value.

18.32.2.12 pfaffEncode() `void pfaffEncode (`
 FILE * *file*,
 int *dx*,
 int *dy*,
 int *flags*)

file dx dy flags

18.32.2.13 reverse_byte_order() `void reverse_byte_order (`
 void * *b*,
 int *bytes*)

Reverses the byte order of *bytes* number of bytes at memory location *b*. Only works for 2 or 4 byte arrays.

18.32.2.14 write_24bit() `void write_24bit (`
 FILE * *file*,
 int *x*)

file x

18.33 extern/libembroidery/src/fill.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "embroidery_internal.h"
```

Functions

- int [lindenmayer_system](#) (L_system *L*, char **state*, int *iterations*, int *complete*)
- static void [join_short_stitches](#) (int **points*, int *n_points*, int *width*, int *tolerance*)
- static int * [threshold_method](#) (EmbImage **image*, int *n_points*, int *subsample_width*, int *subsample_height*, int *threshold*)
- static void [greedy_algorithm](#) (int **points*, int *n_points*, int *width*, EmbReal *bias*)
- static void [save_points_to_pattern](#) (EmbPattern **pattern*, int **points*, int *n_points*, EmbReal *scale*, int *width*, int *height*)
- void [embPattern_horizontal_fill](#) (EmbPattern **pattern*, EmbImage **image*, int *threshold*)
- void [embPattern_crossstitch](#) (EmbPattern **pattern*, EmbImage **image*, int *threshold*)
- int [hilbert_curve](#) (EmbPattern **pattern*, int *iterations*)
- void [generate_dragon_curve](#) (char **state*, int *iterations*)
- int [dragon_curve](#) (int *iterations*)
- void [embPolygon_reduceByDistance](#) (EmbArray **vertices*, EmbArray **simplified*, float *distance*)
- void [embPolygon_reduceByNth](#) (EmbArray **vertices*, EmbArray **out*, int *nth*)
- EmbPattern * [embPattern_combine](#) (EmbPattern **p1*, EmbPattern **p2*)
- void [embPattern_stitchArc](#) (EmbPattern **p*, EmbArc *arc*, int *thread_index*, int *style*)
- void [embPattern_stitchCircle](#) (EmbPattern **p*, EmbCircle *circle*, int *thread_index*, int *style*)
- void [embPattern_stitchEllipse](#) (EmbPattern **p*, EmbEllipse *ellipse*, int *thread_index*, int *style*)
- void [embPattern_stitchPath](#) (EmbPattern **p*, EmbPath *path*, int *thread_index*, int *style*)
- void [embPattern_stitchPolygon](#) (EmbPattern **p*, EmbPolygon *polygon*, int *thread_index*, int *style*)
- void [embPattern_stitchPolyline](#) (EmbPattern **p*, EmbPolyline *polyline*, int *thread_index*, int *style*)
- void [embPattern_stitchRect](#) (EmbPattern **p*, EmbRect *rect*, int *thread_index*, int *style*)
- void [embPattern_stitchText](#) (EmbPattern **p*, EmbRect *rect*, int *thread_index*, int *style*)
- void [embPattern_convertGeometry](#) (EmbPattern **p*)

Variables

- `const char * rules [] = {"+BF-AFA-FB+", "-AF+BFB+FA-"}`
- `L_system hilbert_curve_l_system`

18.33.1 Function Documentation

18.33.1.1 dragon_curve() `int dragon_curve (`
 `int iterations)`

Create the dragon curve for *iterations*.

Returns 0 if the number of iterations is greater than 10 and 1 otherwise.

18.33.1.2 embPattern_combine() `EmbPattern * embPattern_combine (`
 `EmbPattern * p1,`
 `EmbPattern * p2)`

p1 p2

Returns

`EmbPattern*`

18.33.1.3 embPattern_convertGeometry() `void embPattern_convertGeometry (`
 `EmbPattern * p)`

p

18.33.1.4 embPattern_crossstitch() `void embPattern_crossstitch (`
 `EmbPattern * pattern,`
 `EmbImage * image,`
 `int threshold)`

pattern image threshold

Uses a threshold method to determine where to put crosses in the fill.

To improve this, we can remove the vertical stitches when two crosses neighbour. Currently the simple way to do this is to chain crosses that are neighbours exactly one ahead.

18.33.1.5 embPattern_horizontal_fill() `void embPattern_horizontal_fill (`
 `EmbPattern * pattern,`
 `EmbImage * image,`
 `int threshold)`

pattern image threshold

Uses a threshold method to determine where to put lines in the fill.

Needs to pass a "donut test", i.e. an image with black pixels where: $10 < x*x + y*y < 20$ over the area $(-30, 30) \times (-30, 30)$.

Use render then image difference to see how well it passes.

18.33.1.6 embPattern_stitchArc() `void embPattern_stitchArc (`
 `EmbPattern * p,`
 `EmbArc arc,`
 `int thread_index,`
 `int style)`

p arc thread_index style

18.33.1.7 embPattern_stitchCircle() `void embPattern_stitchCircle (`
`EmbPattern * p,`
`EmbCircle circle,`
`int thread_index,`
`int style)`

p circle thread_index style

style determines: stitch density fill pattern outline or fill

For now it's a straight fill of 1000 stitches of the whole object by default.

Consider the intersection of a line in direction "d" that passes through the disc with center "c", radius "r". The start and end points are:

$$(c - r(d/|d|), c + r(d/|d|))$$

Lines that are above and below this with an even separation s can be found by taking the point on the line to be $c + sn$ where the n is the unit normal vector to d and the vector to be d again. The intersection points are therefore a right angled triangle, with one side r , another s and the third the length to be solved, by Pythagoras we have:

$$(c + sn - \sqrt{r^2 - s^2}(d/|d|), c + sn + \sqrt{r^2 - s^2}(d/|d|))$$

repeating this process gives us all the end points and the fill only alters these lines by splitting the ones longer than some tolerance.

18.33.1.8 embPattern_stitchEllipse() `void embPattern_stitchEllipse (`
`EmbPattern * p,`
`EmbEllipse ellipse,`
`int thread_index,`
`int style)`

p ellipse thread_index style

Todo finish stitchEllipse

18.33.1.9 embPattern_stitchPath() `void embPattern_stitchPath (`
`EmbPattern * p,`
`EmbPath path,`
`int thread_index,`
`int style)`

p rect thread_index style

Todo finish stitch path

18.33.1.10 embPattern_stitchPolygon() `void embPattern_stitchPolygon (`
`EmbPattern * p,`
`EmbPolygon polygon,`
`int thread_index,`
`int style)`

p rect thread_index style

Todo finish stitch polygon

18.33.1.11 embPattern_stitchPolyline() `void embPattern_stitchPolyline (`
`EmbPattern * p,`
`EmbPolyline polyline,`
`int thread_index,`
`int style)`

p rect thread_index style

Todo finish stitch polyline

18.33.1.12 embPattern_stitchRect() void embPattern_stitchRect (
 EmbPattern * p,
 EmbRect rect,
 int thread_index,
 int style)

p rect thread_index style

Here we just stitch the rectangle in the direction of it's longer side.

18.33.1.13 embPattern_stitchText() void embPattern_stitchText (
 EmbPattern * p,
 EmbRect rect,
 int thread_index,
 int style)

p rect thread_index style

18.33.1.14 embPolygon_reduceByDistance() void embPolygon_reduceByDistance (
 EmbArray * vertices,
 EmbArray * simplified,
 float distance)

vertices simplified distance

Reduces the polygon by distance.

This is a non-destructive function, so the caller is responsible for freeing "vertices" if they choose to keep "simplified".

18.33.1.15 embPolygon_reduceByNth() void embPolygon_reduceByNth (
 EmbArray * vertices,
 EmbArray * out,
 int nth)

vertices out nth

Reduces the polygon by removing the Nth vertex in the vertices list. This is a non-destructive function, so the caller is responsible for freeing vertices if they choose to keep out.

18.33.1.16 generate_dragon_curve() void generate_dragon_curve (
 char * state,
 int iterations)

state iterations

using the "paper folding" method

Todo find citation for paper folding method

18.33.1.17 greedy_algorithm() static void greedy_algorithm (
 int * points,
 int n_points,
 int width,
 EmbReal bias) [static]

points n_points width bias

18.33.1.18 Greedy Algorithm For each point in the list find the shortest distance to any possible neighbour, then perform a swap to make that neighbour the next item in the list.

To make the stitches lie more on one axis than the other bias the distance operator to prefer horizontal direction.

18.33.1.19 hilbert_curve() `int hilbert_curve (`
`EmbPattern * pattern,`
`int iterations)`

pattern iterations

https://en.wikipedia.org/wiki/Hilbert_curve

Using the Lindenmayer System, so we can save work across different functions.

18.33.1.20 join_short_stitches() `static void join_short_stitches (`
`int * points,`
`int * n_points,`
`int width,`
`int tolerance) [static]`

points n_points width tolerance

Remove points that lie in the middle of two short stitches that could be one longer stitch. Repeat until none are found.

18.33.1.21 lindenmayer_system() `int lindenmayer_system (`
`L_system L,`
`char * state,`
`int iterations,`
`int complete)`

L state iterations complete

Returns

int

This is a slow generation algorithm.

18.33.1.22 save_points_to_pattern() `static void save_points_to_pattern (`
`EmbPattern * pattern,`
`int * points,`
`int n_points,`
`EmbReal scale,`
`int width,`
`int height) [static]`

pattern points n_points scale width height

18.33.1.23 threshold_method() `static int * threshold_method (`
`EmbImage * image,`
`int * n_points,`
`int subsample_width,`
`int subsample_height,`
`int threshold) [static]`

image n_points subsample_width subsample_height threshold

Returns

int*

Identify darker pixels to put stitches in.

18.33.2 Variable Documentation

18.33.2.1 hilbert_curve_l_system `L_system hilbert_curve_l_system`

Initial value:

```
= {
    'A', "AB", "F+-", (char**)rules
}
```

18.33.2.2 rules `const char* rules[] = {"+BF-AFA-FB+", "-AF+BFB+FA+"}`

18.34 extern/libembroidery/src/formats.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <ctype.h>
#include "embroidery_internal.h"
```

Functions

- void [safe_free](#) (void *data)
data
- int [embFormat_getExtension](#) (const char *fileName, char *ending)
fileName ending
- int [emb_identify_format](#) (const char *fileName)
fileName
- short [fread_int16](#) (FILE *f)
f
- unsigned short [fread_uint16](#) (FILE *f)
f
- int [fread_int32_be](#) (FILE *f)
f
- void [fpad](#) (FILE *file, char c, int n)
f
- void [binaryWriteShort](#) (FILE *f, short data)
f data
- void [binaryWriteUShort](#) (FILE *f, unsigned short data)
f data
- void [binaryWriteUShortBE](#) (FILE *f, unsigned short data)
f data
- void [binaryWriteInt](#) (FILE *f, int data)
f data
- void [binaryWriteIntBE](#) (FILE *f, int data)
f data
- void [binaryWriteUInt](#) (FILE *f, unsigned int data)
f data
- void [binaryWriteUIntBE](#) (FILE *f, unsigned int data)
f data
- char [embPattern_read](#) ([EmbPattern](#) *pattern, const char *fileName, int format)
pattern fileName format
- char [embPattern_write](#) ([EmbPattern](#) *pattern, const char *fileName, int format)
pattern fileName format
- char [embPattern_readAuto](#) ([EmbPattern](#) *pattern, const char *fileName)
pattern fileName
- char [embPattern_writeAuto](#) ([EmbPattern](#) *pattern, const char *fileName)
pattern fileName

Variables

- [EmbFormatList](#) [formatTable](#) [[numberOfFormats](#)]
- const char [imageWithFrame](#) [38][48]

18.34.1 Function Documentation

18.34.1.1 [binaryWriteInt\(\)](#) void [binaryWriteInt](#) (
FILE * *f*,
int *data*)

f data

Todo replace with [embInt_read](#)

18.34.1.2 [binaryWriteIntBE\(\)](#) void [binaryWriteIntBE](#) (
FILE * *f*,
int *data*)

f data

Todo replace with [embInt_read](#)

18.34.1.3 [binaryWriteShort\(\)](#) void [binaryWriteShort](#) (
FILE * *f*,
short *data*)

f data

Todo replace with [embInt_read](#)

18.34.1.4 [binaryWriteUInt\(\)](#) void [binaryWriteUInt](#) (
FILE * *f*,
unsigned int *data*)

f data

Todo replace with [embInt_read](#)

18.34.1.5 [binaryWriteUIntBE\(\)](#) void [binaryWriteUIntBE](#) (
FILE * *f*,
unsigned int *data*)

f data

Todo replace with [embInt_read](#)

18.34.1.6 [binaryWriteUShort\(\)](#) void [binaryWriteUShort](#) (
FILE * *f*,
unsigned short *data*)

f data

Todo replace with [embInt_read](#)

18.34.1.7 binaryWriteUShortBE() void binaryWriteUShortBE (
FILE * *f*,
unsigned short *data*)
f data

Todo replace with embInt_read

18.34.1.8 emb_identify_format() int emb_identify_format (
const char * *fileName*)
fileName

Returns

int

18.34.1.9 embFormat_getExtension() int embFormat_getExtension (
const char * *fileName*,
char * *ending*)
fileName ending

Returns

int

18.34.1.10 embPattern_read() char embPattern_read (
EmbPattern * *pattern*,
const char * *fileName*,
int *format*)
pattern fileName format

Returns

char

18.34.1.11 embPattern_readAuto() char embPattern_readAuto (
EmbPattern * *pattern*,
const char * *fileName*)
pattern fileName

Returns

char

18.34.1.12 embPattern_write() char embPattern_write (
EmbPattern * *pattern*,
const char * *fileName*,
int *format*)
pattern fileName format

Returns

char

18.34.1.13 embPattern_writeAuto() `char embPattern_writeAuto (`
 `EmbPattern * pattern,`
 `const char * fileName)`
pattern fileName

Returns

char

18.34.1.14 fpad() `void fpad (`
 `FILE * file,`
 `char c,`
 `int n)`

f

Returns

int

18.34.1.15 fread_int16() `short fread_int16 (`
 `FILE * f)`

f

Returns

short

18.34.1.16 fread_int32_be() `int fread_int32_be (`
 `FILE * f)`

f

Returns

int

Todo replace with `emblnt_read`

18.34.1.17 fread_uint16() `unsigned short fread_uint16 (`
 `FILE * f)`

f

Returns

unsigned short

Todo replace with `emblnt_read`

18.34.1.18 safe_free() `void safe_free (`
 `void * data)`

data

18.34.2 Variable Documentation

18.34.2.1 formatTable `EmbFormatList formatTable[numberOfFormats]`

This file is part of libembroidery.

Copyright 2018-2022 The Embroidermodder Team Licensed under the terms of the zlib license.

This file contains all the read and write functions for the library.

Todo This list needs reviewed in case some stitch formats also can contain object data (EMBFORMAT_↵ STCHANDOBJ). *

18.34.2.2 imageWithFrame `const char imageWithFrame[38][48]`**18.35 extern/libembroidery/src/formats/format_100.c File Reference**

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char `read100` (`EmbPattern` *pattern, FILE *file)
- char `write100` (`EmbPattern` *pattern, FILE *file)

18.35.1 Detailed Description

The Toyota Embroidery Format (.10o)

The Toyota 10o format is a stitch-only format that uses an external color file.

The stitch encoding is in 3 byte chunks.

18.35.2 Function Documentation

18.35.2.1 read100() `char read100 (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.35.2.2 write100() `char write100 (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.36 extern/libembroidery/src/formats/format_10o.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char `read10o` (`EmbPattern` *pattern, FILE *file)
- char `write10o` (`EmbPattern` *pattern, FILE *file)

18.36.1 Detailed Description

The Toyota Embroidery Format (.100)

The Toyota 100 format is a stitch-only format that uses an external color file.

The stitch encoding is in 4 byte chunks.

18.36.2 Function Documentation

18.36.2.1 read10o() `char read10o (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.36.2.2 write10o() `char write10o (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.37 extern/libembroidery/src/formats/format_art.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readArt](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeArt](#) ([EmbPattern](#) *pattern, FILE *file)

18.37.1 Detailed Description

The Bernina Embroidery Format (.art)
We don't know much about this format.

Todo Find a source.

18.37.2 Function Documentation

18.37.2.1 readArt() `char readArt (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.37.2.2 writeArt() `char writeArt (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.38 extern/libembroidery/src/formats/format_bmc.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readBmc](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeBmc](#) ([EmbPattern](#) *pattern, FILE *file)

18.38.1 Detailed Description

The Bitmap Cache Embroidery Format (.bmc)
We don't know much about this format.

Todo Find a source.

18.38.2 Function Documentation

18.38.2.1 readBmc() `char readBmc (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.38.2.2 writeBmc() `char writeBmc (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.39 extern/libembroidery/src/formats/format_bro.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char `readBro` (`EmbPattern *pattern`, `FILE *file`)
- char `writeBro` (`EmbPattern *pattern`, `FILE *file`)

18.39.1 Detailed Description

The Bits and Volts Embroidery Format (.bro)
The Bits and Volts bro format is a stitch-only format that uses an external color file.
The header is 256 bytes. There's a series of unknown variables in the header.
The stitch list uses a variable length encoding which is 2 bytes for any stitch.

18.39.2 Function Documentation

18.39.2.1 readBro() `char readBro (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.39.2.2 writeBro() `char writeBro (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.40 extern/libembroidery/src/formats/format_cnd.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readCnd](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeCnd](#) ([EmbPattern](#) *pattern, FILE *file)

18.40.1 Detailed Description

The Melco Embroidery Format (.cnd)

The Melco cnd format is a stitch-only format.

We don't know much about this format.

Todo Find a source.

18.40.2 Function Documentation

18.40.2.1 readCnd() `char readCnd (`
 [EmbPattern](#) * *pattern*,
 FILE * *file*)

18.40.2.2 writeCnd() `char writeCnd (`
 [EmbPattern](#) * *pattern*,
 FILE * *file*)

18.41 extern/libembroidery/src/formats/format_col.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readCol](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeCol](#) ([EmbPattern](#) *pattern, FILE *file)

18.41.1 Detailed Description

The Embroidery Thread Color Format (.col)

An external color file format for formats that do not record their own colors.

It is a human-readable format that has a header that is a single line containing only the number of threads in decimal followed by the windows line break `\r\n`.

Then the rest of the file is a comma separated value list of all threads with 4 values per line: the index of the thread then the red, green and blue channels of the color in that order.

18.41.1.0.1 Example If we had a pattern called "example" with four colors: black, red, magenta and cyan in that order then the file is (with the white space written out):

```
example.col
4\r
0,0,0,0\r
1,255,0,0\r
2,0,255,0\r
3,0,0,255\r
```

18.41.2 Function Documentation

18.41.2.1 readCol() char readCol (
 EmbPattern * pattern,
 FILE * file)

18.41.2.2 writeCol() char writeCol (
 EmbPattern * pattern,
 FILE * file)

18.42 extern/libembroidery/src/formats/format_csd.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Macros

- #define CsdSubMaskSize 479
- #define CsdXorMaskSize 501

Functions

- void BuildDecryptionTable (int seed)
- unsigned char DecodeCsdByte (long fileOffset, unsigned char val, int type)
- char readCsd (EmbPattern *pattern, FILE *file)
- char writeCsd (EmbPattern *pattern, FILE *file)

Variables

- char _subMask [CsdSubMaskSize]
- char _xorMask [CsdXorMaskSize]
- const unsigned char csd_decryptArray []

18.42.1 Detailed Description

The Singer Embroidery Format (.csd)
Stitch Only Format.

18.42.2 Macro Definition Documentation

18.42.2.1 CsdSubMaskSize #define CsdSubMaskSize 479

18.42.2.2 CsdXorMaskSize #define CsdXorMaskSize 501

18.42.3 Function Documentation

18.42.3.1 BuildDecryptionTable() void BuildDecryptionTable (
 int seed)

18.42.3.2 DecodeCsdByte() unsigned char DecodeCsdByte (
 long fileOffset,
 unsigned char val,
 int type)

18.42.3.3 readCsd() char readCsd (
 EmbPattern * pattern,
 FILE * file)

18.42.3.4 writeCsd() char writeCsd (
 EmbPattern * pattern,
 FILE * file)

18.42.4 Variable Documentation

18.42.4.1 _subMask char _subMask[CsdSubMaskSize]

18.42.4.2 _xorMask char _xorMask[CsdXorMaskSize]

18.42.4.3 csd_decryptArray const unsigned char csd_decryptArray[]

Initial value:

```
= {
    0x43, 0x6E, 0x72, 0x7A, 0x76, 0x6C, 0x61, 0x6F, 0x7C, 0x29, 0x5D, 0x62, 0x60, 0x6E, 0x61, 0x62,
    0x20, 0x41, 0x66, 0x6A, 0x3A, 0x35, 0x5A, 0x63, 0x7C, 0x37, 0x3A, 0x2A, 0x25, 0x24, 0x2A, 0x33,
    0x00, 0x10, 0x14, 0x03, 0x72, 0x4C, 0x48, 0x42, 0x08, 0x7A, 0x5E, 0x0B, 0x6F, 0x45, 0x47, 0x5F,
    0x40, 0x54, 0x5C, 0x57, 0x55, 0x59, 0x53, 0x3A, 0x32, 0x6F, 0x53, 0x54, 0x50, 0x5C, 0x4A, 0x56,
    0x2F, 0x2F, 0x62, 0x2C, 0x22, 0x65, 0x25, 0x28, 0x38, 0x30, 0x38, 0x22, 0x2B, 0x25, 0x3A, 0x6F,
    0x27, 0x38, 0x3E, 0x3F, 0x74, 0x37, 0x33, 0x77, 0x2E, 0x30, 0x3D, 0x34, 0x2E, 0x32, 0x2B, 0x2C,
    0x0C, 0x18, 0x42, 0x13, 0x16, 0x0A, 0x15, 0x02, 0x0B, 0x1C, 0x1E, 0x0E, 0x08, 0x60, 0x64, 0x0D,
    0x09, 0x51, 0x25, 0x1A, 0x18, 0x16, 0x19, 0x1A, 0x58, 0x10, 0x14, 0x5B, 0x08, 0x15, 0x1B, 0x5F,
    0xD5, 0xD2, 0xAE, 0xA3, 0xC1, 0xF0, 0xF4, 0xE8, 0xF8, 0xEC, 0xA6, 0xAB, 0xCD, 0xF8, 0xFD, 0xFB,
    0xE2, 0xF0, 0xFE, 0xFA, 0xF5, 0xB5, 0xF7, 0xF9, 0xFC, 0xB9, 0xF5, 0xEF, 0xF4, 0xF8, 0xEC, 0xBF,
    0xC3, 0xCE, 0xD7, 0xCD, 0xD0, 0xD7, 0xCF, 0xC2, 0xDB, 0xA4, 0xA0, 0xB0, 0xAF, 0xBE, 0x98, 0xE2,
    0xC2, 0x91, 0xE5, 0xDC, 0xDA, 0xD2, 0x96, 0xC4, 0x98, 0xF8, 0xC9, 0xD2, 0xDD, 0xD3, 0x9E, 0xDE,
    0xAE, 0xA5, 0xE2, 0x8C, 0xB6, 0xAC, 0xA3, 0xA9, 0xBC, 0xA8, 0xA6, 0xEB, 0x8B, 0xBF, 0xA1, 0xAC,
    0xB5, 0xA3, 0xBB, 0xB6, 0xA7, 0xD8, 0xDC, 0x9A, 0xAA, 0xF9, 0x82, 0xFB, 0x9D, 0xB9, 0xAB, 0xB3,
    0x94, 0xC1, 0xA0, 0x8C, 0x8B, 0x8E, 0x95, 0x8F, 0x87, 0x99, 0xE7, 0xE1, 0xA3, 0x83, 0x8B, 0xCF,
    0xA3, 0x85, 0x9D, 0x83, 0xD4, 0xB7, 0x83, 0x84, 0x91, 0x97, 0x9F, 0x88, 0x8F, 0xDD, 0xAD, 0x90
}
```

18.43 extern/libembroidery/src/formats/format_csv.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char * [csvStitchFlagToStr](#) (int flags)
- int [csvStrToStitchFlag](#) (const char *str)

- char `readCsv` (`EmbPattern` *pattern, FILE *file)
- char `writeCsv` (`EmbPattern` *pattern, FILE *file)

18.43.1 Detailed Description

Comma Separated Values (.csv)

Comma Separated Values files aren't a universal system, here we aim to offer a broad support. The dialect is detected based on the opening lines, as each manufacturer should label their CSV files there.

18.43.1.0.1 Embroidermodder 2.0 CSV Dialect Our own version has the identifier comment line:

| Control Symbol | Type | Description | |—| | # | COMMENT | | > | VARIABLE | To store records of a pattern's width, height etc. This means that data stored in the header of say a .dst file is preserved. | | \$ | THREAD | | * | STITCH | | * | JUMP | | * | COLOR | To change a color: used for trim as well | | * | END | To end a pattern. | | * | UNKNOWN | For any feature that we can't identify.

18.43.1.0.2 EmBird CSV Dialect

18.43.2 Function Documentation

18.43.2.1 csvStitchFlagToStr() char * csvStitchFlagToStr (
int flags)

18.43.2.2 csvStrToStitchFlag() int csvStrToStitchFlag (
const char * str)

18.43.2.3 readCsv() char readCsv (
EmbPattern * pattern,
FILE * file)

18.43.2.4 writeCsv() char writeCsv (
EmbPattern * pattern,
FILE * file)

18.44 extern/libembroidery/src/formats/format_dat.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char `readDat` (`EmbPattern` *pattern, FILE *file)
- char `writeDat` (`EmbPattern` *pattern, FILE *file)

18.44.1 Function Documentation

18.44.1.1 readDat() char readDat (
EmbPattern * pattern,
FILE * file)

```
18.44.1.2 writeDat()  char writeDat (
                        EmbPattern * pattern,
                        FILE * file )
```

18.45 extern/libembroidery/src/formats/format_dem.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readDem](#) (EmbPattern *pattern, FILE *file)
- char [writeDem](#) (EmbPattern *pattern, FILE *file)

18.45.1 Detailed Description

The Melco Embroidery Format (.dem)
Stitch Only Format

18.45.2 Function Documentation

```
18.45.2.1 readDem()  char readDem (
                        EmbPattern * pattern,
                        FILE * file )
```

```
18.45.2.2 writeDem() char writeDem (
                        EmbPattern * pattern,
                        FILE * file )
```

18.46 extern/libembroidery/src/formats/format_dsb.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readDsb](#) (EmbPattern *pattern, FILE *file)
- char [writeDsb](#) (EmbPattern *pattern, FILE *file)

18.46.1 Detailed Description

The Barudan Embroidery Format (.dsb)

- Stitch Only Format.
- [X] Basic Read Support
- [o] Basic Write Support
- [o] Well Tested Read
- [o] Well Tested Write

18.46.2 Function Documentation

18.46.2.1 readDsb() `char readDsb (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.46.2.2 writeDsb() `char writeDsb (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.47 extern/libembroidery/src/formats/format_dst.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Macros

- `#define cci(c1, c2) (c1*256+c2)`

Functions

- `int decode_record_flags` (unsigned char b2)
- `void encode_record` (FILE *file, int x, int y, int flags)
- `void set_dst_variable` (EmbPattern *pattern, char *var, char *val)
- `char readDst` (EmbPattern *pattern, FILE *file)
- `char writeDst` (EmbPattern *pattern, FILE *file)

18.47.1 Detailed Description

.DST (Tajima) embroidery file read/write routines Format comments are thanks to tspilman@dalcoathletic.com who's notes appeared at <http://www.wotsit.org> under Tajima Format.

18.47.1.1 Tajima Embroidery Format (.dst) Stitch Only Format. [X] Basic Read Support [X] Basic Write Support [] Well Tested Read [] Well Tested Write

.DST (Tajima) embroidery file read/write routines Format comments are thanks to tspilman@dalcoathletic.com who's notes appeared at <http://www.wotsit.org> under Tajima Format.

Other references: [2], [1].

18.47.1.1.1 Header The header contains general information about the design. It is in lines of ASCII, so if you open a DST file as a text file, it's the only part that's easy to read. The line ending symbol is ``0x0D`. The header is necessary for the file to be read by most softwares and hardwares.

The header is 125 bytes of data followed by padding spaces to make it 512 bytes in total.

The lines are as follows.

Label	Size	Description	Example
LA:	17	The design name with no path or extension. The space reserved is 16 characters, but the name must not be longer than 8 and be padded to 16 with spaces (0x20).	"LA:Star "
ST:	8	The stitch count. An integer in the format <code>%07d</code> , that is: a 7 digit number padded by leading zeros. This is the total accross all possible stitch flags.	

Label	Size	Description	Example
CO:	4	The number of color changes (not to be confused with thread count, an all black design we would have the record \textbf{000}). An integer in the format %03d, that is: a 3 digit number padded by leading zeros.	
+X:	6	The extent of the pattern in the positive x direction in millimeters. An integer in the format %05d, that is: a 5 digit number padded by leading zeros.	
-X:	6	The extent of the pattern in the negative x direction in millimeters. An integer in the format %05d, that is: a 5 digit integer padded by leading zeros.	
+Y:	6	The extent of the pattern in the positive y direction in millimeters. An integer in the format %05d, that is: a 5 digit integer padded by leading zeros.	
-Y:	6	The extent of the pattern in the negative y direction in millimeters. An integer in the format %05d, that is: a 5 digit integer padded by leading zeros.	
AX:	7	The difference of the end from the start in the x direction in 0.1mm, the first char should be the sign, followed by an integer in the format %05d, that is: a 5 digit integer padded by leading zeros.	
AY:	7	The difference of the end from the start in the y direction in 0.1mm, the first char should be the sign, followed by an integer in the format %05d, that is: a 5 digit integer padded by leading zeros.	
MX:	7	The x co-ordinate of the last point in the previous file should the design span multiple files. Like AX, it is the sign, followed by a 5 digit integer. If we have a one file design set it to zero.	
MY:	7	The y co-ordinate of the last point in the previous file should the design span multiple files. Like AY, it is the sign, followed by a 5 digit integer. If we have a one file design set it to zero.	
PD:	10	Information about multivolume designs.	

18.47.1.1.2 Stitch Data Uses 3 byte per stitch encoding with the format as follows:

Bit	7	6	5	4	3	2	1	0
Byte 0	y+1	y-1	y+9	y-9	x-9	x+9	x-1	x+1
Byte 1	y+3	y-3	y+27	y-27	x-27	x+27	x-3	x+3
Byte 2	jump	color change	y+81	y-81	x-81	x+81	set	set

T01 and Tap appear to use Tajima Ternary.

Where the stitch type is determined as:

Normal Stitch 0b00000011 0x03 Jump Stitch 0b10000011 0x83 Stop/Change Color 0b11000011 0x↵
C3 End Design 0b11110011 0xF3

Inclusive or'ed with the last byte.

Note that the max stitch length is the largest sum of $1+3+9+27+81=121$ where the unit length is 0.1mm so 12.↵
1mm. The coordinate system is right handed.

18.47.2 Macro Definition Documentation

18.47.2.1 cci #define cci(
 c1,
 c2) (c1*256+c2)

18.47.3 Function Documentation

18.47.3.1 decode_record_flags() int decode_record_flags (
 unsigned char b2)

18.47.3.2 encode_record() void encode_record (

```
FILE * file,
int x,
int y,
int flags )
```

18.47.3.3 readDst() char readDst (

```
EmbPattern * pattern,
FILE * file )
```

18.47.3.4 set_dst_variable() void set_dst_variable (

```
EmbPattern * pattern,
char * var,
char * val )
```

18.47.3.5 writeDst() char writeDst (

```
EmbPattern * pattern,
FILE * file )
```

18.48 extern/libembroidery/src/formats/format_dsz.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readDsz](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeDsz](#) ([EmbPattern](#) *pattern, FILE *file)

18.48.1 Function Documentation

18.48.1.1 readDsz() char readDsz (

```
EmbPattern * pattern,
FILE * file )
```

18.48.1.2 ZSK USA Embroidery Format (.dsz) The ZSK USA dsz format is stitch-only.

18.48.1.3 writeDsz() char writeDsz (

```
EmbPattern * pattern,
FILE * file )
```

18.49 extern/libembroidery/src/formats/format_dxf.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
```

```
#include "../embroidery_internal.h"
```

Functions

- void [readLine](#) (FILE *file, char *str)
- char [readDxf](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeDxf](#) ([EmbPattern](#) *pattern, FILE *file)

18.49.1 Function Documentation

18.49.1.1 readDxf() char readDxf (
 [EmbPattern](#) * pattern,
 FILE * file)

18.49.1.2 readLine() void readLine (
 FILE * file,
 char * str)

18.49.1.3 Drawing Exchange Format (.dxf) Graphics format for drawing files designed and used by AutoDesk for their AutoCAD program. [[dxf_reference](#)]

18.49.1.4 writeDxf() char writeDxf (
 [EmbPattern](#) * pattern,
 FILE * file)

18.50 extern/libembroidery/src/formats/format_edr.c File Reference

```
#include <stdio.h>  
#include <math.h>  
#include "../embroidery_internal.h"
```

Functions

- char [readEdr](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeEdr](#) ([EmbPattern](#) *pattern, FILE *file)

18.50.1 Function Documentation

18.50.1.1 readEdr() char readEdr (
 [EmbPattern](#) * pattern,
 FILE * file)

18.50.1.2 Embird Embroidery Format (.edr) Stitch Only Format

18.50.1.3 writeEdr() char writeEdr (
 [EmbPattern](#) * pattern,
 FILE * file)

18.51 extern/libembroidery/src/formats/format_emd.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [emdDecode](#) (unsigned char inputByte)
- char [readEmd](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeEmd](#) ([EmbPattern](#) *pattern, FILE *file)

18.51.1 Detailed Description

The Elna Embroidery Format (.emd)
Stitch Only Format.

18.51.2 Function Documentation

18.51.2.1 [emdDecode\(\)](#) char emdDecode (
 unsigned char *inputByte*)

18.51.2.2 [readEmd\(\)](#) char readEmd (
 [EmbPattern](#) * *pattern*,
 FILE * *file*)

18.51.2.3 [writeEmd\(\)](#) char writeEmd (
 [EmbPattern](#) * *pattern*,
 FILE * *file*)

18.52 extern/libembroidery/src/formats/format_exp.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [expDecode](#) (unsigned char a1)
- char [readExp](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeExp](#) ([EmbPattern](#) *pattern, FILE *file)

18.52.1 Function Documentation

18.52.1.1 [expDecode\(\)](#) char expDecode (
 unsigned char *a1*)

18.52.1.2 Melco Embroidery Format (.exp) Stitch Only Format.

18.52.1.3 readExp() `char readExp (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.52.1.4 writeExp() `char writeExp (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.53 extern/libembroidery/src/formats/format_exy.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- int `decode_exy_flags` (unsigned char b2)
- char `readExy` (`EmbPattern *pattern`, FILE *file)
- char `writeExy` (`EmbPattern *pattern`, FILE *file)

18.53.1 Function Documentation

18.53.1.1 decode_exy_flags() `int decode_exy_flags (`
 `unsigned char b2)`

18.53.1.2 Eltac Embroidery Format (.exy) Stitch Only Format.

18.53.1.3 readExy() `char readExy (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.53.1.4 writeExy() `char writeExy (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.54 extern/libembroidery/src/formats/format_eyc.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char `readEys` (`EmbPattern *pattern`, FILE *file)
- char `writeEys` (`EmbPattern *pattern`, FILE *file)

18.54.1 Function Documentation

18.54.1.1 readEys() `char readEys (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.54.1.2 Sierra Expanded Embroidery Format (.eys) Stitch Only Format.
Smoothie G-Code Embroidery Format (.fxy)?

18.54.1.3 writeEys() `char writeEys (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.55 extern/libembroidery/src/formats/format_fxy.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readFxy](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeFxy](#) ([EmbPattern](#) *pattern, FILE *file)

18.55.1 Function Documentation

18.55.1.1 readFxy() `char readFxy (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.55.1.2 Embroidery Format (.fxy) Stitch Only Format.

18.55.1.3 writeFxy() `char writeFxy (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.56 extern/libembroidery/src/formats/format_gc.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readGc](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeGc](#) ([EmbPattern](#) *pattern, FILE *file)

18.56.1 Function Documentation

18.56.1.1 readGc() `char readGc (`
 `EmbPattern * pattern,`
 `FILE * file)`

Smoothie G-Code

Main Reference: Machinery's Handbook Guide A Guide to Tables, Formulas, & More in the 31st Edition by John Milton Amiss, Franklin D. Jones and Henry Ryffel

18.56.1.2 writeGc() `char writeGc (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.57 extern/libembroidery/src/formats/format_gnc.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char `readGnc` (`EmbPattern *pattern`, `FILE *file`)
- char `writeGnc` (`EmbPattern *pattern`, `FILE *file`)

18.57.1 Function Documentation

18.57.1.1 readGnc() `char readGnc (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.57.1.2 Great Notions Embroidery Format (.gnc) Stitch Only Format.

18.57.1.3 writeGnc() `char writeGnc (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.58 extern/libembroidery/src/formats/format_gt.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char `readGt` (`EmbPattern *pattern`, `FILE *file`)
- char `writeGt` (`EmbPattern *pattern`, `FILE *file`)

18.58.1 Function Documentation

18.58.1.1 readGt() `char readGt (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.58.1.2 Gold Thread Embroidery Format (.gt) Stitch Only Format.

18.58.1.3 writeGt() char writeGt (
 EmbPattern * pattern,
 FILE * file)

18.59 extern/libembroidery/src/formats/format_hus.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- int [husDecodeStitchType](#) (unsigned char b)
- unsigned char * [husDecompressData](#) (unsigned char *input, int compressedInputLength, int decompressedContentLength)
- unsigned char * [husCompressData](#) (unsigned char *input, int decompressedInputSize, int *compressedSize)
- int [husDecodeByte](#) (unsigned char b)
- unsigned char [husEncodeByte](#) (EmbReal f)
- unsigned char [husEncodeStitchType](#) (int st)
- char [readHus](#) (EmbPattern *pattern, FILE *file)
- char [writeHus](#) (EmbPattern *pattern, FILE *file)

18.59.1 Function Documentation

18.59.1.1 husCompressData() unsigned char * husCompressData (
 unsigned char * input,
 int decompressedInputSize,
 int * compressedSize)

18.59.1.2 husDecodeByte() int husDecodeByte (
 unsigned char b)

18.59.1.3 husDecodeStitchType() int husDecodeStitchType (
 unsigned char b)

18.59.1.4 Husqvarna Viking Embroidery Format (.hus) Stitch Only Format.

18.59.1.5 husDecompressData() unsigned char * husDecompressData (
 unsigned char * input,
 int compressedInputLength,
 int decompressedContentLength)

18.59.1.6 husEncodeByte() unsigned char husEncodeByte (
 EmbReal f)

18.59.1.7 husEncodeStitchType() unsigned char husEncodeStitchType (
int st)

18.59.1.8 readHus() char readHus (
EmbPattern * pattern,
FILE * file)

18.59.1.9 writeHus() char writeHus (
EmbPattern * pattern,
FILE * file)

18.60 extern/libembroidery/src/formats/format_inb.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char readInb (EmbPattern *pattern, FILE *file)
- char writeInb (EmbPattern *pattern, FILE *file)

18.60.1 Function Documentation

18.60.1.1 readInb() char readInb (
EmbPattern * pattern,
FILE * file)

18.60.1.2 Inbro Embroidery Format (.inb) Stitch Only Format.

18.60.1.3 writeInb() char writeInb (
EmbPattern * pattern,
FILE * file)

18.61 extern/libembroidery/src/formats/format_inf.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char readInf (EmbPattern *pattern, FILE *file)
- char writeInf (EmbPattern *pattern, FILE *file)

18.61.1 Function Documentation

18.61.1.1 readInf() `char readInf (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.61.1.2 Embroidery Color Format (.inf) Stitch Only Format.

18.61.1.3 writeInf() `char writeInf (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.62 extern/libembroidery/src/formats/format_jef.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Classes

- struct [hoop_padding](#)

Functions

- int [jefGetHoopSize](#) (int width, int height)
- char [jefDecode](#) (unsigned char inputByte)
- void [jefSetHoopFromId](#) (EmbPattern *pattern, int hoopCode)
- void [read_hoop](#) (FILE *file, struct [hoop_padding](#) *hoop, char *label)
- char [readJef](#) (EmbPattern *pattern, FILE *file)
- void [jefEncode](#) (unsigned char *b, char dx, char dy, int flags)
- char [writeJef](#) (EmbPattern *pattern, FILE *file)

18.62.1 Function Documentation

18.62.1.1 jefDecode() `char jefDecode (`
 unsigned char *inputByte*)

18.62.1.2 jefEncode() `void jefEncode (`
 unsigned char * *b*,
 char *dx*,
 char *dy*,
 int *flags*)

18.62.1.3 jefGetHoopSize() `int jefGetHoopSize (`
 int *width*,
 int *height*)

18.62.1.4 Janome Embroidery Format (.jef) Stitch Only Format.

18.62.1.5 jefSetHoopFromId() `void jefSetHoopFromId (`
 EmbPattern * *pattern*,
 int *hoopCode*)

18.62.1.6 read_hoop() void read_hoop (
FILE * *file*,
struct hoop_padding * *hoop*,
char * *label*)

18.62.1.7 readJef() char readJef (
EmbPattern * *pattern*,
FILE * *file*)

18.62.1.8 writeJef() char writeJef (
EmbPattern * *pattern*,
FILE * *file*)

18.63 extern/libembroidery/src/formats/format_ksm.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- void [ksmEncode](#) (unsigned char **b*, char *dx*, char *dy*, int *flags*)
- char [readKsm](#) (EmbPattern **pattern*, FILE **file*)
- char [writeKsm](#) (EmbPattern **pattern*, FILE **file*)

18.63.1 Function Documentation

18.63.1.1 ksmEncode() void ksmEncode (
unsigned char * *b*,
char *dx*,
char *dy*,
int *flags*)

18.63.1.2 Pfaff professional Design format (.ksm) Stitch Only Format.

18.63.1.3 readKsm() char readKsm (
EmbPattern * *pattern*,
FILE * *file*)

18.63.1.4 writeKsm() char writeKsm (
EmbPattern * *pattern*,
FILE * *file*)

18.64 extern/libembroidery/src/formats/format_max.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readMax](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeMax](#) ([EmbPattern](#) *pattern, FILE *file)

Variables

- const unsigned char [max_header](#) []

18.64.1 Function Documentation

18.64.1.1 readMax() char readMax (
[EmbPattern](#) * pattern,
 FILE * file)

18.64.1.2 writeMax() char writeMax (
[EmbPattern](#) * pattern,
 FILE * file)

18.64.2 Variable Documentation

18.64.2.1 max_header const unsigned char max_header[]

Initial value:

```
= {
    0x56, 0x43, 0x53, 0x4D, 0xFC, 0x03, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00,
    0xF6, 0x25, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00, 0x00, 0x05, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x31, 0x33, 0x37, 0x38,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x4D, 0x61, 0x64, 0x65, 0x69, 0x72, 0x61, 0x20,
    0x52, 0x61, 0x79, 0x6F, 0x6E, 0x20, 0x34, 0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x01, 0x38, 0x09, 0x31, 0x33, 0x30, 0x2F, 0x37, 0x30, 0x35, 0x20, 0x48, 0xFA, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00
}
```

18.64.2.2 Pfaff Embroidery Format (.max) Stitch Only Format.

18.65 extern/libembroidery/src/formats/format_mit.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readMit](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeMit](#) ([EmbPattern](#) *pattern, FILE *file)

18.65.1 Function Documentation

18.65.1.1 readMit() `char readMit (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.65.1.2 Mitsubishi Embroidery Format (.mit) Stitch Only Format.

18.65.1.3 writeMit() `char writeMit (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.66 extern/libembroidery/src/formats/format_new.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readNew](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeNew](#) ([EmbPattern](#) *pattern, FILE *file)

18.66.1 Function Documentation

18.66.1.1 readNew() `char readNew (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.66.1.2 Ameco Embroidery Format (.new) Stitch Only Format.

18.66.1.3 writeNew() `char writeNew (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.67 extern/libembroidery/src/formats/format_ofm.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char * [ofmReadLibrary](#) (FILE *file)
- static int [ofmReadClass](#) (FILE *file)
- void [ofmReadBlockHeader](#) (FILE *file)
- void [ofmReadColorChange](#) (FILE *file, [EmbPattern](#) *pattern)
- void [ofmReadThreads](#) (FILE *file, [EmbPattern](#) *p)
- [EmbReal](#) [ofmDecode](#) (unsigned char b1, unsigned char b2)
- void [ofmReadExpanded](#) (FILE *file, [EmbPattern](#) *p)
- char [readOfm](#) ([EmbPattern](#) *pattern, FILE *fileCompound)
- char [writeOfm](#) ([EmbPattern](#) *pattern, FILE *file)

18.67.1 Function Documentation

18.67.1.1 ofmDecode() `EmbReal ofmDecode (`
 unsigned char *b1*,
 unsigned char *b2*)

18.67.1.2 ofmReadBlockHeader() `void ofmReadBlockHeader (`
 FILE * *file*)

18.67.1.3 ofmReadClass() `static int ofmReadClass (`
 FILE * *file*) [static]

18.67.1.4 ofmReadColorChange() `void ofmReadColorChange (`
 FILE * *file*,
 EmbPattern * *pattern*)

18.67.1.5 ofmReadExpanded() `void ofmReadExpanded (`
 FILE * *file*,
 EmbPattern * *p*)

18.67.1.6 ofmReadLibrary() `char * ofmReadLibrary (`
 FILE * *file*)

18.67.1.7 Melco Embroidery Format (.ofm) Stitch Only Format.

18.67.1.8 ofmReadThreads() `void ofmReadThreads (`
 FILE * *file*,
 EmbPattern * *p*)

18.67.1.9 readOfm() `char readOfm (`
 EmbPattern * *pattern*,
 FILE * *fileCompound*)

18.67.1.10 writeOfm() `char writeOfm (`
 EmbPattern * *pattern*,
 FILE * *file*)

18.68 extern/libembroidery/src/formats/format_pcd.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readPcd](#) ([EmbPattern](#) *pattern, const char *fileName, FILE *file)
- char [writePcd](#) ([EmbPattern](#) *pattern, FILE *file)

18.68.1 Function Documentation

18.68.1.1 readPcd() `char readPcd (`
 [EmbPattern](#) * *pattern*,
 const char * *fileName*,
 FILE * *file*)

18.68.1.2 Pfaff PCD File Format (.pcd) Stitch Only Format.

The format uses a signed 3 byte-length number type.

See the description here (5) for the overview of the format.

For an example of the format see (11).

18.68.1.3 writePcd() `char writePcd (`
 [EmbPattern](#) * *pattern*,
 FILE * *file*)

18.69 extern/libembroidery/src/formats/format_pcm.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readPcm](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writePcm](#) ([EmbPattern](#) *pattern, FILE *file)

18.69.1 Function Documentation

18.69.1.1 readPcm() `char readPcm (`
 [EmbPattern](#) * *pattern*,
 FILE * *file*)

18.69.1.2 Pfaff Embroidery Format (.pcm) The Pfaff pcm format is stitch-only.

18.69.1.3 writePcm() `char writePcm (`
 [EmbPattern](#) * *pattern*,
 FILE * *file*)

18.70 extern/libembroidery/src/formats/format_pcq.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```


Functions

- char [readPcq](#) ([EmbPattern](#) *pattern, const char *fileName, FILE *file)
- char [writePcq](#) ([EmbPattern](#) *pattern, FILE *file)

18.70.1 Function Documentation

18.70.1.1 readPcq() char readPcq (
 [EmbPattern](#) * pattern,
 const char * fileName,
 FILE * file)

18.70.1.2 Embroidery Format (.pcq) The Pfaff pcq format is stitch-only.

18.70.1.3 writePcq() char writePcq (
 [EmbPattern](#) * pattern,
 FILE * file)

18.71 extern/libembroidery/src/formats/format_pcs.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readPcs](#) ([EmbPattern](#) *pattern, const char *fileName, FILE *file)
- char [writePcs](#) ([EmbPattern](#) *pattern, FILE *file)

18.71.1 Function Documentation

18.71.1.1 readPcs() char readPcs (
 [EmbPattern](#) * pattern,
 const char * fileName,
 FILE * file)

18.71.1.2 Embroidery Format (.pcq) The Pfaff pcs format is stitch-only.

18.71.1.3 writePcs() char writePcs (
 [EmbPattern](#) * pattern,
 FILE * file)

18.72 extern/libembroidery/src/formats/format_pec.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- void `readPecStitches` (`EmbPattern` *pattern, FILE *file)
- void `pecEncodeJump` (FILE *file, int x, int types)
- void `pecEncodeStop` (FILE *file, unsigned char val)
- char `readPec` (`EmbPattern` *pattern, const char *fileName, FILE *file)
- void `pecEncode` (FILE *file, `EmbPattern` *p)
- void `writelImage` (FILE *file, unsigned char image[][48])
- void `writePecStitches` (`EmbPattern` *pattern, FILE *file, const char *fileName)
- char `writePec` (`EmbPattern` *pattern, const char *fileName, FILE *file)

18.72.1 Function Documentation

18.72.1.1 `pecEncode()` void `pecEncode` (
FILE * *file*,
EmbPattern * *p*)

18.72.1.2 `pecEncodeJump()` void `pecEncodeJump` (
FILE * *file*,
int *x*,
int *types*)

18.72.1.3 `pecEncodeStop()` void `pecEncodeStop` (
FILE * *file*,
unsigned char *val*)

18.72.1.4 `readPec()` char `readPec` (
EmbPattern * *pattern*,
const char * *fileName*,
FILE * *file*)

18.72.1.5 `readPecStitches()` void `readPecStitches` (
EmbPattern * *pattern*,
FILE * *file*)

18.72.1.6 Embroidery Format (.pec) The Brother pec format is stitch-only.

18.72.1.7 `writelImage()` void `writeImage` (
FILE * *file*,
unsigned char *image*[][48])

Write a PES embedded *image* to the given *file* pointer.

18.72.1.8 `writePec()` char `writePec` (
EmbPattern * *pattern*,
const char * *fileName*,
FILE * *file*)

18.72.1.9 writePecStitches() void writePecStitches (
 EmbPattern * pattern,
 FILE * file,
 const char * fileName)

18.73 extern/libembroidery/src/formats/format_pel.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char readPel (EmbPattern *pattern, FILE *file)
- char writePel (EmbPattern *pattern, FILE *file)

18.73.1 Function Documentation

18.73.1.1 readPel() char readPel (
 EmbPattern * pattern,
 FILE * file)

18.73.1.2 Embroidery Format (.pec) The Brother pel format is stitch-only.

18.73.1.3 writePel() char writePel (
 EmbPattern * pattern,
 FILE * file)

18.74 extern/libembroidery/src/formats/format_pem.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char readPem (EmbPattern *pattern, FILE *file)
- char writePem (EmbPattern *pattern, FILE *file)

18.74.1 Function Documentation

18.74.1.1 readPem() char readPem (
 EmbPattern * pattern,
 FILE * file)

18.74.1.2 Embroidery Format (.pec) The Brother pem format is stitch-only.

18.74.1.3 writePem() char writePem (
 EmbPattern * pattern,
 FILE * file)

18.75 extern/libembroidery/src/formats/format_pes.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readPes](#) ([EmbPattern](#) *pattern, const char *fileName, FILE *file)
- void [readDescriptions](#) (FILE *file, [EmbPattern](#) *pattern)
- void [readPESHeaderV5](#) (FILE *file, [EmbPattern](#) *pattern)
- void [readPESHeaderV6](#) (FILE *file, [EmbPattern](#) *pattern)
- void [readPESHeaderV7](#) (FILE *file, [EmbPattern](#) *pattern)
- void [readPESHeaderV8](#) (FILE *file, [EmbPattern](#) *pattern)
- void [readPESHeaderV9](#) (FILE *file, [EmbPattern](#) *pattern)
- void [readPESHeaderV10](#) (FILE *file, [EmbPattern](#) *pattern)
- void [readHoopName](#) (FILE *file, [EmbPattern](#) *pattern)
- void [readImageString](#) (FILE *file, [EmbPattern](#) *pattern)
- void [readProgrammableFills](#) (FILE *file, [EmbPattern](#) *pattern)
- void [readMotifPatterns](#) (FILE *file, [EmbPattern](#) *pattern)
- void [readFeatherPatterns](#) (FILE *file, [EmbPattern](#) *pattern)
- void [readThreads](#) (FILE *file, [EmbPattern](#) *pattern)
- void [pesWriteSewSegSection](#) ([EmbPattern](#) *pattern, FILE *file)
- void [pesWriteEmbOneSection](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writePes](#) ([EmbPattern](#) *pattern, const char *fileName, FILE *file)

Variables

- const char * [pes_version_strings](#) []
- int [pes_version](#) = PES0001

18.75.1 Function Documentation

18.75.1.1 [pesWriteEmbOneSection\(\)](#) void [pesWriteEmbOneSection](#) (
[EmbPattern](#) * *pattern*,
FILE * *file*)

18.75.1.2 [pesWriteSewSegSection\(\)](#) void [pesWriteSewSegSection](#) (
[EmbPattern](#) * *pattern*,
FILE * *file*)

18.75.1.3 [readDescriptions\(\)](#) void [readDescriptions](#) (
FILE * *file*,
[EmbPattern](#) * *pattern*)

18.75.1.4 [readFeatherPatterns\(\)](#) void [readFeatherPatterns](#) (
FILE * *file*,
[EmbPattern](#) * *pattern*)

18.75.1.5 readHoopName() void readHoopName (
FILE * *file*,
EmbPattern * *pattern*)

18.75.1.6 readImageString() void readImageString (
FILE * *file*,
EmbPattern * *pattern*)

18.75.1.7 readMotifPatterns() void readMotifPatterns (
FILE * *file*,
EmbPattern * *pattern*)

18.75.1.8 readPes() char readPes (
EmbPattern * *pattern*,
const char * *fileName*,
FILE * *file*)

18.75.1.9 readPESHeaderV10() void readPESHeaderV10 (
FILE * *file*,
EmbPattern * *pattern*)

18.75.1.10 readPESHeaderV5() void readPESHeaderV5 (
FILE * *file*,
EmbPattern * *pattern*)

18.75.1.11 readPESHeaderV6() void readPESHeaderV6 (
FILE * *file*,
EmbPattern * *pattern*)

18.75.1.12 readPESHeaderV7() void readPESHeaderV7 (
FILE * *file*,
EmbPattern * *pattern*)

18.75.1.13 readPESHeaderV8() void readPESHeaderV8 (
FILE * *file*,
EmbPattern * *pattern*)

18.75.1.14 readPESHeaderV9() void readPESHeaderV9 (
FILE * *file*,
EmbPattern * *pattern*)

18.75.1.15 readProgrammableFills() void readProgrammableFills (
FILE * *file*,
EmbPattern * *pattern*)

18.75.1.16 readThreads() `void readThreads (`
 `FILE * file,`
 `EmbPattern * pattern)`

18.75.1.17 writePes() `char writePes (`
 `EmbPattern * pattern,`
 `const char * fileName,`
 `FILE * file)`

18.75.2 Variable Documentation

18.75.2.1 pes_version `int pes_version = PES0001`

18.75.2.2 pes_version_strings `const char* pes_version_strings[]`
Initial value:

```
= {  
    "#PES0001",  
    "#PES0020",  
    "#PES0022",  
    "#PES0030",  
    "#PES0040",  
    "#PES0050",  
    "#PES0055",  
    "#PES0056",  
    "#PES0060",  
    "#PES0070",  
    "#PES0080",  
    "#PES0090",  
    "#PES0100",  
}
```

18.75.2.3 Embroidery Format (.pec) The Brother pes format is stitch-only.

18.76 extern/libembroidery/src/formats/format_phb.c File Reference

```
#include <stdio.h>  
#include <math.h>  
#include "../embroidery_internal.h"
```

Functions

- char [readPhb](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writePhb](#) ([EmbPattern](#) *pattern, FILE *file)

18.76.1 Function Documentation

18.76.1.1 readPhb() `char readPhb (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.76.1.2 Embroidery Format (.pec) The Brother phb format is stitch-only.

18.76.1.3 writePhb() `char writePhb (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.77 extern/libembroidery/src/formats/format_phc.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readPhc](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writePhc](#) ([EmbPattern](#) *pattern, FILE *file)

18.77.1 Function Documentation

18.77.1.1 readPhc() char readPhc (
 [EmbPattern](#) * pattern,
 FILE * file)

18.77.1.2 Embroidery Format (.pec) The Brother phc format is stitch-only.

18.77.1.3 writePhc() char writePhc (
 [EmbPattern](#) * pattern,
 FILE * file)

18.78 extern/libembroidery/src/formats/format_plt.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readPlt](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writePlt](#) ([EmbPattern](#) *pattern, FILE *file)

18.78.1 Function Documentation

18.78.1.1 readPlt() char readPlt (
 [EmbPattern](#) * pattern,
 FILE * file)

18.78.1.2 Embroidery Format (.plt) The AutoCAD plt format is stitch-only.

18.78.1.3 writePlt() char writePlt (
 [EmbPattern](#) * pattern,
 FILE * file)

18.79 extern/libembroidery/src/formats/format_rgb.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readRgb](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeRgb](#) ([EmbPattern](#) *pattern, FILE *file)

18.79.1 Function Documentation

18.79.1.1 readRgb() char readRgb (
 [EmbPattern](#) * pattern,
 FILE * file)

18.79.1.2 Color File (.rgb) The RGB format is a color-only format to act as an external color file for other formats.

18.79.1.3 writeRgb() char writeRgb (
 [EmbPattern](#) * pattern,
 FILE * file)

18.80 extern/libembroidery/src/formats/format_sew.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [sewDecode](#) (unsigned char inputByte)
- char [readSew](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeSew](#) ([EmbPattern](#) *pattern, FILE *file)

18.80.1 Function Documentation

18.80.1.1 readSew() char readSew (
 [EmbPattern](#) * pattern,
 FILE * file)

18.80.1.2 sewDecode() char sewDecode (
 unsigned char inputByte)

18.80.1.3 Embroidery Format (.sew) The Janome sew format is stitch-only.

18.80.1.4 writeSew() char writeSew (
 [EmbPattern](#) * pattern,
 FILE * file)

18.81 extern/libembroidery/src/formats/format_shv.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [shvDecode](#) (unsigned char inputByte)
- short [shvDecodeShort](#) (unsigned short inputByte)
- char [readShv](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeShv](#) ([EmbPattern](#) *pattern, FILE *file)

18.81.1 Function Documentation

18.81.1.1 [readShv\(\)](#) char readShv (
 [EmbPattern](#) * pattern,
 FILE * file)

18.81.1.2 [shvDecode\(\)](#) char shvDecode (
 unsigned char inputByte)

18.81.1.3 Viking Embroidery Format (.shv) The Husqvarna Viking shv format is stitch-only.

18.81.1.4 [shvDecodeShort\(\)](#) short shvDecodeShort (
 unsigned short inputByte)

18.81.1.5 [writeShv\(\)](#) char writeShv (
 [EmbPattern](#) * pattern,
 FILE * file)

18.82 extern/libembroidery/src/formats/format_sst.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readSst](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeSst](#) ([EmbPattern](#) *pattern, FILE *file)

18.82.1 Function Documentation

18.82.1.1 [readSst\(\)](#) char readSst (
 [EmbPattern](#) * pattern,
 FILE * file)

18.82.1.2 Embroidery Format (.sst) The Sunstar sst format is stitch-only.

18.82.1.3 writeSst() `char writeSst (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.83 extern/libembroidery/src/formats/format_stx.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- int `stxReadThread` (`StxThread *thread`, `FILE *file`)
- char `readStx` (`EmbPattern *pattern`, `FILE *file`)
- char `writeStx` (`EmbPattern *pattern`, `FILE *file`)

18.83.1 Function Documentation

18.83.1.1 readStx() `char readStx (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.83.1.2 stxReadThread() `int stxReadThread (`
 `StxThread * thread,`
 `FILE * file)`

18.83.1.3 Stitch Embroidery Format (.stx) The Data Stitch stx format is stitch-only.

18.83.1.4 writeStx() `char writeStx (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.84 extern/libembroidery/src/formats/format_svg.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char `readSvg` (`EmbPattern *pattern`, `FILE *file`)
- char `writeSvg` (`EmbPattern *pattern`, `FILE *file`)

Variables

- int `svgCreator`
- int `svgExpect`
- int `svgMultiValue`
- int `current_element_id`
- `SvgAttribute` `attributeList` [1000]
- int `n_attributes` = 0
- char `currentAttribute` [1000]
- char `currentValue` [1000]

18.84.1 Function Documentation

18.84.1.1 `readSvg()` `char readSvg (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.84.1.2 `writeSvg()` `char writeSvg (`
 `EmbPattern * pattern,`
 `FILE * file)`

Writes the data from *pattern* to a file with the given *fileName*. Returns `true` if successful, otherwise returns `false`.

18.84.2 Variable Documentation

18.84.2.1 `attributeList` `SvgAttribute attributeList`[1000]

18.84.2.2 `current_element_id` `int current_element_id`

18.84.2.3 `currentAttribute` `char currentAttribute`[1000]

18.84.2.4 `currentValue` `char currentValue`[1000]

18.84.2.5 `n_attributes` `int n_attributes` = 0

18.84.2.6 `svgCreator` `int svgCreator`

18.84.2.7 `Vector Graphics (.svg)` The scalable vector graphics (SVG) format is a graphics format maintained by ...

18.84.2.8 `svgExpect` `int svgExpect`

18.84.2.9 `svgMultiValue` `int svgMultiValue`

18.85 extern/libembroidery/src/formats/format_t01.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readT01](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeT01](#) ([EmbPattern](#) *pattern, FILE *file)

18.85.1 Function Documentation

18.85.1.1 readT01() char readT01 (
 [EmbPattern](#) * pattern,
 FILE * file)

18.85.1.2 Embroidery Format (.pcq) The Pfaff t01 format is stitch-only.

18.85.1.3 writeT01() char writeT01 (
 [EmbPattern](#) * pattern,
 FILE * file)

18.86 extern/libembroidery/src/formats/format_t09.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readT09](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeT09](#) ([EmbPattern](#) *pattern, FILE *file)

18.86.1 Function Documentation

18.86.1.1 readT09() char readT09 (
 [EmbPattern](#) * pattern,
 FILE * file)

18.86.1.1.1 Embroidery Format (.pcq) The Pfaff t09 format is stitch-only.

18.86.1.2 writeT09() char writeT09 (
 [EmbPattern](#) * pattern,
 FILE * file)

18.87 extern/libembroidery/src/formats/format_tap.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../embroidery_internal.h"
```

Functions

- void [encode_tap_record](#) (FILE *file, int x, int y, int flags)
- int [decode_tap_record_flags](#) (unsigned char b2)
- char [readTap](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeTap](#) ([EmbPattern](#) *pattern, FILE *file)

18.87.1 Function Documentation

18.87.1.1 [decode_tap_record_flags\(\)](#) int [decode_tap_record_flags](#) (
 unsigned char *b2*)

18.87.1.2 [encode_tap_record\(\)](#) void [encode_tap_record](#) (
 FILE * *file*,
 int *x*,
 int *y*,
 int *flags*)

18.87.1.3 Embroidery Format (.tap) The Happy tap format is stitch-only.

18.87.1.4 [readTap\(\)](#) char [readTap](#) (
 [EmbPattern](#) * *pattern*,
 FILE * *file*)

18.87.1.5 [writeTap\(\)](#) char [writeTap](#) (
 [EmbPattern](#) * *pattern*,
 FILE * *file*)

18.88 extern/libembroidery/src/formats/format_thr.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readThr](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeThr](#) ([EmbPattern](#) *pattern, FILE *file)

18.88.1 Function Documentation

18.88.1.1 readThr() `char readThr (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.88.1.2 Embroidery Format (.thr) The ThreadWorks thr format is stitch-only.

18.88.1.3 writeThr() `char writeThr (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.89 extern/libembroidery/src/formats/format_txt.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readTxt](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeTxt](#) ([EmbPattern](#) *pattern, FILE *file)

18.89.1 Function Documentation

18.89.1.1 readTxt() `char readTxt (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.89.1.2 File (.txt) The txt format is stitch-only and isn't associated with a specific company.

18.89.1.3 writeTxt() `char writeTxt (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.90 extern/libembroidery/src/formats/format_u00.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readU00](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeU00](#) ([EmbPattern](#) *pattern, FILE *file)

18.90.1 Function Documentation

18.90.1.1 readU00() `char readU00 (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.90.1.2 Embroidery Format (.u00) The Barudan u00 format is stitch-only.

18.90.1.3 writeU00() `char writeU00 (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.91 extern/libembroidery/src/formats/format_u01.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char [readU01](#) ([EmbPattern](#) *pattern, FILE *file)
- char [writeU01](#) ([EmbPattern](#) *pattern, FILE *file)

18.91.1 Function Documentation

18.91.1.1 readU01() `char readU01 (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.91.1.2 Embroidery Format (.u00) The Barudan u01 format is stitch-only.

18.91.1.3 writeU01() `char writeU01 (`
 `EmbPattern * pattern,`
 `FILE * file)`

18.92 extern/libembroidery/src/formats/format_vip.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- int [vipDecodeByte](#) (unsigned char b)
- int [vipDecodeStitchType](#) (unsigned char b)
- unsigned char * [vipDecompressData](#) (unsigned char *input, int compressedInputLength, int decompressedContentLength)
- char [readVip](#) ([EmbPattern](#) *pattern, FILE *file)
- unsigned char * [vipCompressData](#) (unsigned char *input, int decompressedInputSize, int *compressedSize)
- unsigned char [vipEncodeByte](#) ([EmbReal](#) f)
- unsigned char [vipEncodeStitchType](#) (int st)
- char [writeVip](#) ([EmbPattern](#) *pattern, FILE *file)

Variables

- const unsigned char [vipDecodingTable](#) []

18.92.1 Function Documentation

18.92.1.1 readVip() char readVip (
 EmbPattern * pattern,
 FILE * file)

18.92.1.2 vipCompressData() unsigned char * vipCompressData (
 unsigned char * input,
 int decompressedInputSize,
 int * compressedSize)

18.92.1.3 vipDecodeByte() int vipDecodeByte (
 unsigned char b)

18.92.1.4 vipDecodeStitchType() int vipDecodeStitchType (
 unsigned char b)

18.92.1.5 vipDecompressData() unsigned char * vipDecompressData (
 unsigned char * input,
 int compressedInputLength,
 int decompressedContentLength)

18.92.1.6 vipEncodeByte() unsigned char vipEncodeByte (
 EmbReal f)

18.92.1.7 vipEncodeStitchType() unsigned char vipEncodeStitchType (
 int st)

18.92.1.8 writeVip() char writeVip (
 EmbPattern * pattern,
 FILE * file)

18.92.2 Variable Documentation

18.92.2.1 vipDecodingTable const unsigned char vipDecodingTable[]

Initial value:

```
= {
    0x2E, 0x82, 0xE4, 0x6F, 0x38, 0xA9, 0xDC, 0xC6, 0x7B, 0xB6, 0x28, 0xAC, 0xFD, 0xAA, 0x8A, 0x4E,
    0x76, 0x2E, 0xF0, 0xE4, 0x25, 0x1B, 0x8A, 0x68, 0x4E, 0x92, 0xB9, 0xB4, 0x95, 0xF0, 0x3E, 0xEF,
    0xF7, 0x40, 0x24, 0x18, 0x39, 0x31, 0xBB, 0xE1, 0x53, 0xA8, 0x1F, 0xB1, 0x3A, 0x07, 0xFB, 0xCB,
    0xE6, 0x00, 0x81, 0x50, 0x0E, 0x40, 0xE1, 0x2C, 0x73, 0x50, 0x0D, 0x91, 0xD6, 0x0A, 0x5D, 0xD6,
    0x8B, 0xB8, 0x62, 0xAE, 0x47, 0x00, 0x53, 0x5A, 0xB7, 0x80, 0xAA, 0x28, 0xF7, 0x5D, 0x70, 0x5E,
    0x2C, 0x0B, 0x98, 0xE3, 0xA0, 0x98, 0x60, 0x47, 0x89, 0x9B, 0x82, 0xFB, 0x40, 0xC9, 0xB4, 0x00,
    0x0E, 0x68, 0x6A, 0x1E, 0x09, 0x85, 0xC0, 0x53, 0x81, 0xD1, 0x98, 0x89, 0xAF, 0xE8, 0x85, 0x4F,
    0xE3, 0x69, 0x89, 0x03, 0xA1, 0x2E, 0x8F, 0xCF, 0xED, 0x91, 0x9F, 0x58, 0x1E, 0xD6, 0x84, 0x3C,
    0x09, 0x27, 0xBD, 0xF4, 0xC3, 0x90, 0xC0, 0x51, 0x1B, 0x2B, 0x63, 0xBC, 0xB9, 0x3D, 0x40, 0x4D,
    0x62, 0x6F, 0xE0, 0x8C, 0xF5, 0x5D, 0x08, 0xFD, 0x3D, 0x50, 0x36, 0xD7, 0xC9, 0xC9, 0x43, 0xE4,
    0x2D, 0xCB, 0x95, 0xB6, 0xF4, 0x0D, 0xEA, 0xC2, 0xFD, 0x66, 0x3F, 0x5E, 0xBD, 0x69, 0x06, 0x2A,
    0x03, 0x19, 0x47, 0x2B, 0xDF, 0x38, 0xEA, 0x4F, 0x80, 0x49, 0x95, 0xB2, 0xD6, 0xF9, 0x9A, 0x75,
    0xF4, 0xD8, 0x9B, 0x1D, 0xB0, 0xA4, 0x69, 0xDB, 0xA9, 0x21, 0x79, 0x6F, 0xD8, 0xDE, 0x33, 0xFE,
```



```

0x9F, 0x04, 0xE5, 0x9A, 0x6B, 0x9B, 0x73, 0x83, 0x62, 0x7C, 0xB9, 0x66, 0x76, 0xF2, 0x5B, 0xC9,
0x5E, 0xFC, 0x74, 0xAA, 0x6C, 0xF1, 0xCD, 0x93, 0xCE, 0xE9, 0x80, 0x53, 0x03, 0x3B, 0x97, 0x4B,
0x39, 0x76, 0xC2, 0xC1, 0x56, 0xCB, 0x70, 0xFD, 0x3B, 0x3E, 0x52, 0x57, 0x81, 0x5D, 0x56, 0x8D,
0x51, 0x90, 0xD4, 0x76, 0xD7, 0xD5, 0x16, 0x02, 0x6D, 0xF2, 0x4D, 0xE1, 0x0E, 0x96, 0x4F, 0xA1,
0x3A, 0xA0, 0x60, 0x59, 0x64, 0x04, 0x1A, 0xE4, 0x67, 0xB6, 0xED, 0x3F, 0x74, 0x20, 0x55, 0x1F,
0xFB, 0x23, 0x92, 0x91, 0x53, 0xC8, 0x65, 0xAB, 0x9D, 0x51, 0xD6, 0x73, 0xDE, 0x01, 0xB1, 0x80,
0xB7, 0xC0, 0xD6, 0x80, 0x1C, 0x2E, 0x3C, 0x83, 0x63, 0xEE, 0xBC, 0x33, 0x25, 0xE2, 0x0E, 0x7A,
0x67, 0xDE, 0x3F, 0x71, 0x14, 0x49, 0x9C, 0x92, 0x93, 0x0D, 0x26, 0x9A, 0x0E, 0xDA, 0xED, 0x6F,
0xA4, 0x89, 0x0C, 0x1B, 0xF0, 0xA1, 0xDF, 0xE1, 0x9E, 0x3C, 0x04, 0x78, 0xE4, 0xAB, 0x6D, 0xFF,
0x9C, 0xAF, 0xCA, 0xC7, 0x88, 0x17, 0x9C, 0xE5, 0xB7, 0x33, 0x6D, 0xDC, 0xED, 0x8F, 0x6C, 0x18,
0x1B, 0x71, 0x06, 0xB1, 0xC5, 0xE2, 0xCF, 0x13, 0x77, 0x81, 0xC5, 0xB7, 0x0A, 0x14, 0x0A, 0x6B,
0x40, 0x26, 0xA0, 0x88, 0xD1, 0x62, 0x6A, 0xB3, 0x50, 0x12, 0xB9, 0x9B, 0xB5, 0x83, 0x9B, 0x37
}

```

18.92.2.2 Embroidery Format (.pcq) The Pfaff vip format is stitch-only.

18.93 extern/libembroidery/src/formats/format_vp3.c File Reference

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"

```

Functions

- unsigned char * [vp3ReadString](#) (FILE *file)
- int [vp3Decode](#) (unsigned char inputByte)
- short [vp3DecodeInt16](#) (unsigned short inputByte)
- [vp3Hoop vp3ReadHoopSection](#) (FILE *file)
- char [readVp3](#) ([EmbPattern](#) *pattern, FILE *file)
- void [vp3WriteStringLen](#) (FILE *file, const char *str, int len)
- void [vp3WriteString](#) (FILE *file, const char *str)
- void [vp3PatchByteCount](#) (FILE *file, int offset, int adjustment)
- char [writeVp3](#) ([EmbPattern](#) *pattern, FILE *file)

18.93.1 Function Documentation

18.93.1.1 [readVp3\(\)](#) char readVp3 (
[EmbPattern](#) * pattern,
FILE * file)

18.93.1.2 [vp3Decode\(\)](#) int vp3Decode (
unsigned char inputByte)

18.93.1.3 [vp3DecodeInt16\(\)](#) short vp3DecodeInt16 (
unsigned short inputByte)

18.93.1.4 [vp3PatchByteCount\(\)](#) void vp3PatchByteCount (
FILE * file,
int offset,
int adjustment)

18.93.1.5 `vp3ReadHoopSection()` `vp3Hoop` `vp3ReadHoopSection` (
 FILE * *file*)

18.93.1.6 `vp3ReadString()` unsigned char * `vp3ReadString` (
 FILE * *file*)

18.93.1.7 **Embroidery Format (.pcq)** The Pfaff vp3 format is stitch-only.

18.93.1.8 `vp3WriteString()` void `vp3WriteString` (
 FILE * *file*,
 const char * *str*)

18.93.1.9 `vp3WriteStringLen()` void `vp3WriteStringLen` (
 FILE * *file*,
 const char * *str*,
 int *len*)

18.93.1.10 `writeVp3()` char `writeVp3` (
 EmbPattern * *pattern*,
 FILE * *file*)

18.94 extern/libembroidery/src/formats/format_XXX.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char `xxxDecodeByte` (unsigned char *inputByte*)
- char `readXxx` (EmbPattern **pattern*, FILE **file*)
- void `xxxEncodeStop` (FILE **file*, EmbStitch *s*)
- void `xxxEncodeStitch` (FILE **file*, EmbReal *deltaX*, EmbReal *deltaY*, int *flags*)
- void `xxxEncodeDesign` (FILE **file*, EmbPattern **p*)
- char `writeXxx` (EmbPattern **pattern*, FILE **file*)

18.94.1 Function Documentation

18.94.1.1 `readXxx()` char `readXxx` (
 EmbPattern * *pattern*,
 FILE * *file*)

18.94.1.2 `writeXxx()` char `writeXxx` (
 EmbPattern * *pattern*,
 FILE * *file*)

18.94.1.3 `xxxDecodeByte()` char `xxxDecodeByte` (
 unsigned char *inputByte*)

18.94.1.4 Embroidery Format (.xxx) The Singer xxx format is stitch-only.

18.94.1.5 xxxEncodeDesign() void xxxEncodeDesign (
FILE * *file*,
EmbPattern * *p*)

18.94.1.6 xxxEncodeStitch() void xxxEncodeStitch (
FILE * *file*,
EmbReal *deltaX*,
EmbReal *deltaY*,
int *flags*)

18.94.1.7 xxxEncodeStop() void xxxEncodeStop (
FILE * *file*,
EmbStitch *s*)

18.95 extern/libembroidery/src/formats/format_zsk.c File Reference

```
#include <stdio.h>
#include <string.h>
#include "../embroidery_internal.h"
```

Functions

- char readZsk (EmbPattern *pattern, FILE *file)
- char writeZsk (EmbPattern *pattern, FILE *file)

18.95.1 Detailed Description

The ZSK USA Embroidery Format (.zsk)
The ZSK USA zsk format is stitch-only.

18.95.2 Function Documentation

18.95.2.1 readZsk() char readZsk (
EmbPattern * *pattern*,
FILE * *file*)

18.95.2.2 writeZsk() char writeZsk (
EmbPattern * *pattern*,
FILE * *file*)

18.96 extern/libembroidery/src/geometry.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "embroidery.h"
```

Functions

- `EmbGeometry * embGeometry_init (int type_in)`
Our generic object interface backends to each individual type.
- `void embGeometry_free (EmbGeometry *obj)`
Free the memory occupied by a non-stitch geometry object.
- `void embGeometry_move (EmbGeometry *obj, EmbVector delta)`
Translate obj by the vector delta.
- `EmbRect embGeometry_boundingRect (EmbGeometry *obj)`
Calculate the bounding box of geometry obj based on what kind of geometric object it is.
- `void embGeometry_vulcanize (EmbGeometry *obj)`
Toggle the rubber mode of the object.

18.96.1 Function Documentation

18.96.1.1 embGeometry_boundingRect() `EmbRect embGeometry_boundingRect (EmbGeometry * obj)`

Calculate the bounding box of geometry *obj* based on what kind of geometric object it is.
obj A pointer to the geometry memory.

Returns

EmbRect The bounding box in the same scale as the input geometry.

In the case of a failure the bounding box returned is always the unit square with top left corner at (0, 0).

18.96.1.2 embGeometry_free() `void embGeometry_free (EmbGeometry * obj)`

Free the memory occupied by a non-stitch geometry object.
obj Pointer to geometry memory.

18.96.1.3 embGeometry_init() `EmbGeometry * embGeometry_init (int type_in)`

Our generic object interface backends to each individual type.
type_in

Returns

EmbGeometry*

18.96.1.4 embGeometry_move() `void embGeometry_move (EmbGeometry * obj, EmbVector delta)`

Translate *obj* by the vector *delta*.
obj A pointer to the geometry memory. *delta* A vector in the 0.1mm scale to offset the geometry by.

18.96.1.5 embGeometry_vulcanize() `void embGeometry_vulcanize (EmbGeometry * obj)`

Toggle the rubber mode of the object.
obj

Todo Review. This could be controlled by a simple flag.

18.97 extern/libembroidery/src/geometry/arc.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

Functions

- [EmbArc embArc_init](#) (void)
- char [embArc_clockwise](#) (EmbArc arc)
- void [getArcCenter](#) (EmbArc arc, EmbVector *arcCenter)
- char [getArcDataFromBulge](#) (EmbReal bulge, EmbArc *arc, EmbReal *arcCenterX, EmbReal *arcCenterY, EmbReal *radius, EmbReal *diameter, EmbReal *chord, EmbReal *chordMidX, EmbReal *chordMidY, EmbReal *sagitta, EmbReal *apothem, EmbReal *incAngleInDegrees, char *clockwise)
- char [clockwise](#) (EmbGeometry *obj)
- void [embArc_setCenter](#) (EmbArc *arc, EmbVector point)
- void [embArc_setRadius](#) (EmbArc *arc, float radius)
- void [embArc_setStartAngle](#) (EmbArc *arc, float angle)
- void [embArc_setEndAngle](#) (EmbArc *arc, float angle)
- float [embArc_startAngle](#) (EmbArc arc)
- float [embArc_endAngle](#) (EmbArc arc)
- float [embArc_area](#) (EmbArc arc)
- float [embArc_arcLength](#) (EmbArc arc)
- float [embArc_chord](#) (EmbArc arc)
- float [embArc_includedAngle](#) (EmbArc arc)
- char [Arc_clockwise](#) ()
- void [embArc_updatePath](#) (EmbArc arc)
- void [embArc_paint](#) (void)
- void [embArc_updateRubber](#) (EmbArc arc, int pattern, int layer, int index)
- EmbVector [embArc_mouseSnapPoint](#) (EmbArc arc, EmbVector mousePoint)
- void [embArc_gripEdit](#) (EmbArc *arc, EmbVector before, EmbVector after)
- void [set_object_color](#) (EmbGeometry *obj, EmbColor color)
- void [embBase_setColorRGB](#) (EmbGeometry *obj, unsigned int rgb)
- void [Base_setLineType](#) (EmbGeometry *obj, int lineType)
- void [Base_setLineWeight](#) (EmbGeometry *obj, float lineWeight)
- EmbVector [Base_objectRubberPoint](#) (EmbGeometry *obj, const char *key)
- const char * [Base_objectRubberText](#) (EmbGeometry *obj, const char *key)
- void [embCircle_prompt](#) (const char *str)
- void [embCircle_setArea](#) (EmbCircle *circle, float area)
- void [embCircle_setCircumference](#) (EmbCircle *circle, float circumference)
- void [embEllipse_main](#) ()
- void [embEllipse_click](#) (float x, float y)
- EmbVector [embRect_bottomLeft](#) (EmbRect rect)
- EmbVector [embRect_bottomRight](#) (EmbRect rect)

18.97.1 Function Documentation

18.97.1.1 [Arc_clockwise\(\)](#) char Arc_clockwise ()

18.97.1.2 Base_objectRubberPoint() `EmbVector` Base_objectRubberPoint (
 `EmbGeometry` * *obj*,
const char * *key*)

18.97.1.3 Base_objectRubberText() const char * Base_objectRubberText (
 `EmbGeometry` * *obj*,
const char * *key*)

18.97.1.4 Base_setLineType() void Base_setLineType (
 `EmbGeometry` * *obj*,
int *lineType*)

18.97.1.5 Base_setLineWeight() void Base_setLineWeight (
 `EmbGeometry` * *obj*,
float *lineWeight*)

18.97.1.6 clockwise() char clockwise (
 `EmbGeometry` * *obj*)

18.97.1.7 embArc_arcLength() float embArc_arcLength (
 `EmbArc` *arc*)

18.97.1.8 embArc_area() float embArc_area (
 `EmbArc` *arc*)

18.97.1.9 embArc_chord() float embArc_chord (
 `EmbArc` *arc*)

18.97.1.10 embArc_clockwise() char embArc_clockwise (
 `EmbArc` *arc*)

18.97.1.11 embArc_endAngle() float embArc_endAngle (
 `EmbArc` *arc*)

18.97.1.12 embArc_gripEdit() void embArc_gripEdit (
 `EmbArc` * *arc*,
`EmbVector` *before*,
`EmbVector` *after*)

18.97.1.13 embArc_includedAngle() float embArc_includedAngle (
 `EmbArc` *arc*)

- 18.97.1.14** `embArc_init()` `EmbArc embArc_init (`
`void)`
- 18.97.1.15** `embArc_mouseSnapPoint()` `EmbVector embArc_mouseSnapPoint (`
`EmbArc arc,`
`EmbVector mousePoint)`
- 18.97.1.16** `embArc_paint()` `void embArc_paint (`
`void)`
- 18.97.1.17** `embArc_setCenter()` `void embArc_setCenter (`
`EmbArc * arc,`
`EmbVector point)`
- 18.97.1.18** `embArc_setEndAngle()` `void embArc_setEndAngle (`
`EmbArc * arc,`
`float angle)`
- 18.97.1.19** `embArc_setRadius()` `void embArc_setRadius (`
`EmbArc * arc,`
`float radius)`
- 18.97.1.20** `embArc_setStartAngle()` `void embArc_setStartAngle (`
`EmbArc * arc,`
`float angle)`
- 18.97.1.21** `embArc_startAngle()` `float embArc_startAngle (`
`EmbArc arc)`
- 18.97.1.22** `embArc_updatePath()` `void embArc_updatePath (`
`EmbArc arc)`
- 18.97.1.23** `embArc_updateRubber()` `void embArc_updateRubber (`
`EmbArc arc,`
`int pattern,`
`int layer,`
`int index)`
- 18.97.1.24** `embBase_setColorRGB()` `void embBase_setColorRGB (`
`EmbGeometry * obj,`
`unsigned int rgb)`
- 18.97.1.25** `embCircle_prompt()` `void embCircle_prompt (`
`const char * str)`

18.97.1.26 embCircle_setArea() void embCircle_setArea (
 EmbCircle * circle,
 float area)

18.97.1.27 embCircle_setCircumference() void embCircle_setCircumference (
 EmbCircle * circle,
 float circumference)

18.97.1.28 embEllipse_click() void embEllipse_click (
 float x,
 float y)

18.97.1.29 embEllipse_main() void embEllipse_main ()

18.97.1.30 embRect_bottomLeft() EmbVector embRect_bottomLeft (
 EmbRect rect)

18.97.1.31 embRect_bottomRight() EmbVector embRect_bottomRight (
 EmbRect rect)

18.97.1.32 getArcCenter() void getArcCenter (
 EmbArc arc,
 EmbVector * arcCenter)

18.97.1.33 getArcDataFromBulge() char getArcDataFromBulge (
 EmbReal bulge,
 EmbArc * arc,
 EmbReal * arcCenterX,
 EmbReal * arcCenterY,
 EmbReal * radius,
 EmbReal * diameter,
 EmbReal * chord,
 EmbReal * chordMidX,
 EmbReal * chordMidY,
 EmbReal * sagitta,
 EmbReal * apothem,
 EmbReal * incAngleInDegrees,
 char * clockwise)

18.97.1.34 set_object_color() void set_object_color (
 EmbGeometry * obj,
 EmbColor color)

18.98 extern/libembroidery/src/geometry/circle.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```



```
#include "../embroidery.h"
```

Functions

- [EmbCircle embCircle_init](#) (void)
- [EmbReal embCircle_area](#) (EmbCircle circle)
- [EmbReal embCircle_circumference](#) (EmbCircle circle)
- int [getCircleCircleIntersections](#) (EmbCircle c0, EmbCircle c1, EmbVector *p0, EmbVector *p1)
- int [getCircleTangentPoints](#) (EmbCircle c, EmbVector point, EmbVector *t0, EmbVector *t1)

18.98.1 Function Documentation

18.98.1.1 [embCircle_area\(\)](#) [EmbReal](#) embCircle_area (
[EmbCircle](#) circle)

18.98.1.2 [embCircle_circumference\(\)](#) [EmbReal](#) embCircle_circumference (
[EmbCircle](#) circle)

18.98.1.3 [embCircle_init\(\)](#) [EmbCircle](#) embCircle_init (
void)

18.98.1.4 [getCircleCircleIntersections\(\)](#) int getCircleCircleIntersections (
[EmbCircle](#) c0,
[EmbCircle](#) c1,
[EmbVector](#) * p0,
[EmbVector](#) * p1)

18.98.1.5 [getCircleTangentPoints\(\)](#) int getCircleTangentPoints (
[EmbCircle](#) c,
[EmbVector](#) point,
[EmbVector](#) * t0,
[EmbVector](#) * t1)

18.99 extern/libembroidery/src/geometry/ellipse.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

Functions

- [EmbEllipse embEllipse_init](#) (void)
- [EmbReal embEllipse_area](#) (EmbEllipse ellipse)
- [EmbReal embEllipse_perimeter](#) (EmbEllipse ellipse)
- [EmbReal embEllipse_diameterX](#) (EmbEllipse ellipse)
- [EmbReal embEllipse_diameterY](#) (EmbEllipse ellipse)
- [EmbReal embEllipse_width](#) (EmbEllipse ellipse)
- [EmbReal embEllipse_height](#) (EmbEllipse ellipse)

- void `embEllipse_setSize` (float width, float height)
- void `embEllipse_setRadiusMajor` (float radius)
- void `embEllipse_setRadiusMinor` (float radius)
- void `embEllipse_setDiameterMajor` (`EmbEllipse` *ellipse, float diameter)
- void `embEllipse_setDiameterMinor` (`EmbEllipse` *ellipse, float diameter)
- `EmbVector` `ellipse_objectQuadrant0` (`EmbEllipse` *ellipse)
- `EmbVector` `ellipse_objectQuadrant90` (`EmbEllipse` *ellipse)
- `EmbVector` `ellipse_objectQuadrant180` (`EmbEllipse` *ellipse)
- `EmbVector` `ellipse_objectQuadrant270` (`EmbEllipse` *ellipse)
- void `embEllipse_updatePath` ()

18.99.1 Function Documentation

18.99.1.1 `ellipse_objectQuadrant0()` `EmbVector` `ellipse_objectQuadrant0` (
 `EmbEllipse` * *ellipse*)

18.99.1.2 `ellipse_objectQuadrant180()` `EmbVector` `ellipse_objectQuadrant180` (
 `EmbEllipse` * *ellipse*)

18.99.1.3 `ellipse_objectQuadrant270()` `EmbVector` `ellipse_objectQuadrant270` (
 `EmbEllipse` * *ellipse*)

18.99.1.4 `ellipse_objectQuadrant90()` `EmbVector` `ellipse_objectQuadrant90` (
 `EmbEllipse` * *ellipse*)

18.99.1.5 `embEllipse_area()` `EmbReal` `embEllipse_area` (
 `EmbEllipse` *ellipse*)

18.99.1.6 `embEllipse_diameterX()` `EmbReal` `embEllipse_diameterX` (
 `EmbEllipse` *ellipse*)

18.99.1.7 `embEllipse_diameterY()` `EmbReal` `embEllipse_diameterY` (
 `EmbEllipse` *ellipse*)

18.99.1.8 `embEllipse_height()` `EmbReal` `embEllipse_height` (
 `EmbEllipse` *ellipse*)

18.99.1.9 `embEllipse_init()` `EmbEllipse` `embEllipse_init` (
 void)

18.99.1.10 `embEllipse_perimeter()` `EmbReal` `embEllipse_perimeter` (
 `EmbEllipse` *ellipse*)

18.99.1.11 embEllipse_setDiameterMajor() void embEllipse_setDiameterMajor (
 EmbEllipse * ellipse,
 float diameter)

18.99.1.12 embEllipse_setDiameterMinor() void embEllipse_setDiameterMinor (
 EmbEllipse * ellipse,
 float diameter)

18.99.1.13 embEllipse_setRadiusMajor() void embEllipse_setRadiusMajor (
 float radius)

18.99.1.14 embEllipse_setRadiusMinor() void embEllipse_setRadiusMinor (
 float radius)

18.99.1.15 embEllipse_setSize() void embEllipse_setSize (
 float width,
 float height)

18.99.1.16 embEllipse_updatePath() void embEllipse_updatePath ()

18.99.1.17 embEllipse_width() EmbReal embEllipse_width (
 EmbEllipse ellipse)

18.100 extern/libembroidery/src/geometry/functions.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

Functions

- int [emb_round](#) (EmbReal x)
- [EmbReal radians](#) (EmbReal degree)
- [EmbReal degrees](#) (EmbReal radian)

18.100.1 Function Documentation

18.100.1.1 degrees() EmbReal degrees (
 EmbReal radian)

18.100.1.2 emb_round() int emb_round (
 EmbReal x)

18.100.1.3 radians() `EmbReal radians (`
`EmbReal degree)`

18.101 extern/libembroidery/src/geometry/line.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

Functions

- void `embLine_normalVector` (`EmbLine` line, `EmbVector` *result, int clockwise)
- `EmbVector` `embLine_toVector` (`EmbLine` line)
- `EmbVector` `embLine_intersectionPoint` (`EmbLine` line1, `EmbLine` line2)

18.101.1 Function Documentation

18.101.1.1 embLine_intersectionPoint() `EmbVector embLine_intersectionPoint (`
`EmbLine line1,`
`EmbLine line2)`

18.101.1.2 embLine_normalVector() `void embLine_normalVector (`
`EmbLine line,`
`EmbVector * result,`
`int clockwise)`

Finds the normalized vector perpendicular (clockwise) to the line given by v1->v2 (normal to the line)

18.101.1.3 embLine_toVector() `EmbVector embLine_toVector (`
`EmbLine line)`

18.102 extern/libembroidery/src/geometry/path.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

18.103 extern/libembroidery/src/geometry/polygon.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

18.104 extern/libembroidery/src/geometry/polyline.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

18.105 extern/libembroidery/src/geometry/rect.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

Functions

- [EmbRect embRect_init](#) (void)
- [EmbReal embRect_area](#) (EmbRect rect)

18.105.1 Function Documentation

18.105.1.1 [embRect_area\(\)](#) [EmbReal](#) embRect_area (
 [EmbRect](#) rect)

18.105.1.2 [embRect_init\(\)](#) [EmbRect](#) embRect_init (
 void)

18.106 extern/libembroidery/src/geometry/text.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

Functions

- void [textSingle_setTextFont](#) (const char *font)
- void [textSingle_setJustify](#) (const char *justify)
- void [textSingle_setTextSize](#) (float size)
- void [textSingle_setTextStyle](#) (char bold, char italic, char under, char strike, char over)
- void [textSingle_setTextBold](#) (char val)
- void [textSingle_setTextItalic](#) (char val)
- void [textSingle_setTextUnderline](#) (char val)
- void [textSingle_setTextStrikeOut](#) (char val)
- void [textSingle_setTextOverline](#) (char val)
- void [textSingle_setTextBackward](#) (char val)
- void [textSingle_setTextUpsideDown](#) (char val)
- void [textSingle_paint](#) ()
- void [textSingle_updateRubber](#) ()
- [EmbVector](#) [textSingle_mouseSnapPoint](#) ([EmbVector](#) mousePoint)
- void [textSingle_gripEdit](#) ([EmbVector](#) before, [EmbVector](#) after)

18.106.1 Function Documentation

18.106.1.1 [textSingle_gripEdit\(\)](#) void textSingle_gripEdit (
 [EmbVector](#) before,
 [EmbVector](#) after)

18.106.1.2 textSingle_mouseSnapPoint() `EmbVector textSingle_mouseSnapPoint (EmbVector mousePoint)`

18.106.1.3 textSingle_paint() `void textSingle_paint ()`

18.106.1.4 textSingle_setJustify() `void textSingle_setJustify (const char * justify)`

18.106.1.5 textSingle_setTextBackward() `void textSingle_setTextBackward (char val)`

18.106.1.6 textSingle_setTextBold() `void textSingle_setTextBold (char val)`

18.106.1.7 textSingle_setTextFont() `void textSingle_setTextFont (const char * font)`

18.106.1.8 textSingle_setTextItalic() `void textSingle_setTextItalic (char val)`

18.106.1.9 textSingle_setTextOverline() `void textSingle_setTextOverline (char val)`

18.106.1.10 textSingle_setTextSize() `void textSingle_setTextSize (float size)`

18.106.1.11 textSingle_setTextStrikeOut() `void textSingle_setTextStrikeOut (char val)`

18.106.1.12 textSingle_setTextStyle() `void textSingle_setTextStyle (char bold, char italic, char under, char strike, char over)`

18.106.1.13 textSingle_setTextUnderline() `void textSingle_setTextUnderline (char val)`

18.106.1.14 textSingle_setTextUpsideDown() `void textSingle_setTextUpsideDown (char val)`

18.106.1.15 textSingle_updateRubber() void textSingle_updateRubber ()

18.107 extern/libembroidery/src/geometry/vector.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

Functions

- void [embVector_normalize](#) (EmbVector vector, EmbVector *result)
- void [embVector_multiply](#) (EmbVector vector, EmbReal magnitude, EmbVector *result)
- EmbVector [embVector_add](#) (EmbVector a, EmbVector b)
- EmbVector [embVector_average](#) (EmbVector a, EmbVector b)
- EmbVector [embVector_subtract](#) (EmbVector v1, EmbVector v2)
- EmbReal [embVector_dot](#) (EmbVector a, EmbVector b)
- EmbReal [embVector_cross](#) (EmbVector a, EmbVector b)

The "cross product" as vectors a and b returned as a real value.

- void [embVector_transpose_product](#) (EmbVector v1, EmbVector v2, EmbVector *result)
- EmbReal [embVector_length](#) (EmbVector vector)
- EmbReal [embVector_relativeX](#) (EmbVector a1, EmbVector a2, EmbVector a3)
- EmbReal [embVector_relativeY](#) (EmbVector a1, EmbVector a2, EmbVector a3)
- EmbReal [embVector_angle](#) (EmbVector v)
- EmbVector [embVector_unit](#) (EmbReal alpha)
- EmbReal [embVector_distance](#) (EmbVector a, EmbVector b)

18.107.1 Function Documentation

18.107.1.1 embVector_add() EmbVector embVector_add (

EmbVector a,

EmbVector b)

The sum of vectors *a* and *b* returned as a vector.

Equivalent to:

$$\mathbf{c} = \mathbf{a} + \mathbf{b} = \begin{pmatrix} a_x + b_x \\ a_y + b_y \end{pmatrix}$$

18.107.1.2 embVector_angle() EmbReal embVector_angle (

EmbVector v)

The angle, measured anti-clockwise from the x-axis, of a vector *v*.

18.107.1.3 embVector_average() EmbVector embVector_average (

EmbVector a,

EmbVector b)

The average of vectors *v1* and *v2* returned as a vector.

Equivalent to:

$$\mathbf{c} = \frac{\mathbf{a} + \mathbf{b}}{2} = \begin{pmatrix} \frac{a_x + b_x}{2} \\ \frac{a_y + b_y}{2} \end{pmatrix}$$

18.107.1.4 embVector_cross() `EmbReal embVector_cross (`
`EmbVector a,`
`EmbVector b)`

The "cross product" as vectors *a* and *b* returned as a real value.

Technically, this is the magnitude of the cross product when the embroidery is placed in the $z=0$ plane (since the cross product is defined for 3-dimensional vectors). That is:

$$|c| = \left| \begin{pmatrix} a_x \\ a_y \\ 0 \end{pmatrix} \times \begin{pmatrix} b_x \\ b_y \\ 0 \end{pmatrix} \right| = \left| \begin{pmatrix} 0 \\ 0 \\ a_x b_y - a_y b_x \end{pmatrix} \right| = a_x b_y - a_y b_x$$

18.107.1.5 embVector_distance() `EmbReal embVector_distance (`
`EmbVector a,`
`EmbVector b)`

The distance between *a* and *b* returned as a real value.

$$d = |\mathbf{a} - \mathbf{b}| = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}$$

18.107.1.6 embVector_dot() `EmbReal embVector_dot (`
`EmbVector a,`
`EmbVector b)`

The dot product as vectors *v1* and *v2* returned as a `EmbReal`.

Equivalent to:

$$c = \mathbf{a} \cdot \mathbf{b} = a_x b_x + a_y b_y$$

18.107.1.7 embVector_length() `EmbReal embVector_length (`
`EmbVector vector)`

The length or absolute value of the vector *vector*.

Equivalent to:

$$|v| = \sqrt{v_x^2 + v_y^2}$$

18.107.1.8 embVector_multiply() `void embVector_multiply (`
`EmbVector vector,`
`EmbReal magnitude,`
`EmbVector * result)`

The scalar multiple *magnitude* of a vector *vector*. Returned as *result*.

Todo make result return argument.

18.107.1.9 embVector_normalize() `void embVector_normalize (`
`EmbVector vector,`
`EmbVector * result)`

Finds the unit length vector *result* in the same direction as *vector*.

Equivalent to:

$$\mathbf{u} = \frac{v}{|\mathbf{v}|}$$

Todo make result return argument.

18.107.1.10 `embVector_relativeX()` `EmbReal` `embVector_relativeX` (
`EmbVector` *a1*,
`EmbVector` *a2*,
`EmbVector` *a3*)

The x-component of the vector

18.107.1.11 `embVector_relativeY()` `EmbReal` `embVector_relativeY` (
`EmbVector` *a1*,
`EmbVector` *a2*,
`EmbVector` *a3*)

The y-component of the vector

18.107.1.12 `embVector_subtract()` `EmbVector` `embVector_subtract` (
`EmbVector` *v1*,
`EmbVector` *v2*)

The difference between vectors *v1* and *v2* returned as *result*.

Equivalent to:

$$\mathbf{c} = \mathbf{a} - \mathbf{b} = \begin{pmatrix} a_x - b_x \\ a_y - b_y \end{pmatrix}$$

18.107.1.13 `embVector_transpose_product()` `void` `embVector_transpose_product` (
`EmbVector` *v1*,
`EmbVector` *v2*,
`EmbVector` * *result*)

Since we aren't using full vector algebra here, all vectors are "vertical". so this is like the product $\mathbf{v1}^T \mathbf{v2}$ for our vectors *v1* and *v2* so a "component-wise product". The result is stored at the pointer *result*.

That is $\begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} xa \\ yb \end{pmatrix}$

18.107.1.14 `embVector_unit()` `EmbVector` `embVector_unit` (
`EmbReal` *alpha*)

The unit vector in the direction *angle*.

$$\mathbf{a}_\alpha = \begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \end{pmatrix}$$

18.108 extern/libembroidery/src/image.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "embroidery_internal.h"
```

Functions

- void `writelnImage` (FILE *file, unsigned char image[][48])
- float `image_diff` (unsigned char *a, unsigned char *b, int size)

18.108.1 Detailed Description

This backends to the stb libraries and nanosvg library.
 Use Python PEP7 for coding style.

18.108.2 Function Documentation

18.108.2.1 image_diff() float image_diff (
 unsigned char * *a*,
 unsigned char * *b*,
 int *size*)

The distance between the arrays *a* and *b* of length *size*.

18.108.2.2 writelImage() void writeImage (
 FILE * *file*,
 unsigned char *image*[][48])

Write a PES embedded *image* to the given *file* pointer.

18.109 extern/libembroidery/src/main.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <time.h>
#include "embroidery_internal.h"
```

Macros

- #define [FLAG_TO](#) 0
- #define [FLAG_TO_SHORT](#) 1
- #define [FLAG_HELP](#) 2
- #define [FLAG_HELP_SHORT](#) 3
- #define [FLAG_FORMATS](#) 4
- #define [FLAG_FORMATS_SHORT](#) 5
- #define [FLAG_QUIET](#) 6
- #define [FLAG_QUIET_SHORT](#) 7
- #define [FLAG_VERBOSE](#) 8
- #define [FLAG_VERBOSE_SHORT](#) 9
- #define [FLAG_VERSION](#) 10
- #define [FLAG_VERSION_SHORT](#) 11
- #define [FLAG_CIRCLE](#) 12
- #define [FLAG_CIRCLE_SHORT](#) 13
- #define [FLAG_ELLIPSE](#) 14
- #define [FLAG_ELLIPSE_SHORT](#) 15
- #define [FLAG_LINE](#) 16
- #define [FLAG_LINE_SHORT](#) 17
- #define [FLAG_POLYGON](#) 18
- #define [FLAG_POLYGON_SHORT](#) 19
- #define [FLAG_POLYLINE](#) 20
- #define [FLAG_POLYLINE_SHORT](#) 21
- #define [FLAG_RENDER](#) 22
- #define [FLAG_RENDER_SHORT](#) 23
- #define [FLAG_SATIN](#) 24
- #define [FLAG_SATIN_SHORT](#) 25
- #define [FLAG_STITCH](#) 26
- #define [FLAG_STITCH_SHORT](#) 27
- #define [FLAG_TEST](#) 28
- #define [FLAG_FULL_TEST_SUITE](#) 29
- #define [FLAG_HILBERT_CURVE](#) 30
- #define [FLAG_SIERPINSKI_TRIANGLE](#) 31
- #define [FLAG_FILL](#) 32

- #define `FLAG_FILL_SHORT` 33
- #define `FLAG_SIMULATE` 34
- #define `FLAG_COMBINE` 35
- #define `FLAG_CROSS_STITCH` 36
- #define `NUM_FLAGS` 37

Functions

- void `embVector_print` (`EmbVector` v, char *label)
v label
- void `embArc_print` (`EmbArc` arc)
arc
- int `check_header_present` (FILE *file, int minimum_header_length)
file minimum_header_length
- unsigned int `sectorSize` (`bcf_file` *bcfFile)
bcfFile
- int `haveExtraDIFATsectors` (`bcf_file` *file)
file
- int `seekToSector` (`bcf_file` *bcfFile, FILE *file, const unsigned int sector)
bcfFile file sector
- void `parseDIFATsectors` (FILE *file, `bcf_file` *bcfFile)
file bcfFile
- int `bcfFile_read` (FILE *file, `bcf_file` *bcfFile)
file bcfFile
- FILE * `GetFile` (`bcf_file` *bcfFile, FILE *file, char *fileToFind)
Get the File object.
- void `bcf_file_free` (`bcf_file` *bcfFile)
bcfFile
- `bcf_file_difat` * `bcf_difat_create` (FILE *file, unsigned int fatSectors, const unsigned int `sectorSize`)
file fatSectors sectorSize
- unsigned int `entriesInDifatSector` (`bcf_file_difat` *fat)
fat
- unsigned int `readFullSector` (FILE *file, `bcf_file_difat` *bcfFile, unsigned int *difatEntriesToRead)
file bcfFile difatEntriesToRead
- void `parseDirectoryEntryName` (FILE *file, `bcf_directory_entry` *dir)
file dir
- `bcf_directory` * `CompoundFileDirectory` (const unsigned int maxNumberOfDirectoryEntries)
maxNumberOfDirectoryEntries
- `EmbTime` `parseTime` (FILE *file)
file
- `bcf_directory_entry` * `CompoundFileDirectoryEntry` (FILE *file)
file
- void `readNextSector` (FILE *file, `bcf_directory` *dir)
file dir
- void `bcf_directory_free` (`bcf_directory` **dir)
dir
- `bcf_file_fat` * `bcfFileFat_create` (const unsigned int `sectorSize`)
sectorSize
- void `loadFatFromSector` (`bcf_file_fat` *fat, FILE *file)
fat file
- `bcf_file_header` `bcfFileHeader_read` (FILE *file)

- file*
- void `embSatinOutline_generateSatinOutline` (`EmbArray` *lines, `EmbReal` thickness, `EmbSatinOutline` *result)
- lines thickness result*
- `EmbArray` * `embSatinOutline_renderStitches` (`EmbSatinOutline` *result, `EmbReal` density)
- result density*
- void `write_24bit` (`FILE` *file, int x)
- file x*
- int `embColor_distance` (`EmbColor` a, `EmbColor` b)
- a b*
- void `embColor_read` (`FILE` *f, `EmbColor` *c, int toRead)
- f c toRead*
- void `embColor_write` (`FILE` *f, `EmbColor` c, int toWrite)
- f c toWrite*
- int `embThread_findNearestColor` (`EmbColor` color, `EmbColor` *color_list, int n_colors)
- int `embThread_findNearestThread` (`EmbColor` color, `EmbThread` *thread_list, int n_threads)
- color thread_list n_threads*
- `EmbThread` `embThread_getRandom` (void)
- void `binaryReadString` (`FILE` *file, char *buffer, int maxLength)
- file buffer maxLength*
- void `binaryReadUnicodeString` (`FILE` *file, char *buffer, const int stringLength)
- file buffer stringLength*
- int `stringInArray` (const char *s, const char **array)
- int `emb_readline` (`FILE` *file, char *line, int maxLength)
- file line maxLength*
- void `get_trim_bounds` (char const *s, char const **firstWord, char const **trailingSpace)
- Get the trim bounds object.*
- char * `copy_trim` (char const *s)
- s*
- char * `emb_optOut` (`EmbReal` num, char *str)
- Optimizes the number (num) for output to a text file and returns it as a string (str).*
- void `embTime_initNow` (`EmbTime` *t)
- t*
- `EmbTime` `embTime_time` (`EmbTime` *t)
- t*

Variables

- `EmbThread` `black_thread` = { { 0, 0, 0 }, "Black", "Black" }
- int `emb_verbose` = 0
- Verbosity level.*
- int `emb_error` = 0
- Error code storage for optional control flow blocking.*
- const `EmbReal` `embConstantPi` = 3.1415926535
- const unsigned int `difatEntriesInHeader` = 109
- const unsigned int `sizeOfFatEntry` = sizeof(unsigned int)
- const unsigned int `sizeOfDifatEntry` = 4
- const unsigned int `sizeOfChainingEntryAtEndOfDifatSector` = 4
- const unsigned int `sizeOfDirectoryEntry` = 128
- char const `WHITESPACE` [] = "\t\n\r"

18.109.1 Macro Definition Documentation

18.109.1.1 FLAG_CIRCLE `#define FLAG_CIRCLE 12`

18.109.1.2 FLAG_CIRCLE_SHORT `#define FLAG_CIRCLE_SHORT 13`

18.109.1.3 FLAG_COMBINE `#define FLAG_COMBINE 35`

18.109.1.4 FLAG_CROSS_STITCH `#define FLAG_CROSS_STITCH 36`

18.109.1.5 FLAG_ELLIPSE `#define FLAG_ELLIPSE 14`

18.109.1.6 FLAG_ELLIPSE_SHORT `#define FLAG_ELLIPSE_SHORT 15`

18.109.1.7 FLAG_FILL `#define FLAG_FILL 32`

18.109.1.8 FLAG_FILL_SHORT `#define FLAG_FILL_SHORT 33`

18.109.1.9 FLAG_FORMATS `#define FLAG_FORMATS 4`

18.109.1.10 FLAG_FORMATS_SHORT `#define FLAG_FORMATS_SHORT 5`

18.109.1.11 FLAG_FULL_TEST_SUITE `#define FLAG_FULL_TEST_SUITE 29`

18.109.1.12 FLAG_HELP `#define FLAG_HELP 2`

18.109.1.13 FLAG_HELP_SHORT `#define FLAG_HELP_SHORT 3`

18.109.1.14 FLAG_HILBERT_CURVE `#define FLAG_HILBERT_CURVE 30`

18.109.1.15 FLAG_LINE `#define FLAG_LINE 16`

18.109.1.16 FLAG_LINE_SHORT `#define FLAG_LINE_SHORT 17`

18.109.1.17 FLAG_POLYGON `#define FLAG_POLYGON 18`

18.109.1.18 FLAG_POLYGON_SHORT `#define FLAG_POLYGON_SHORT 19`

18.109.1.19 FLAG_POLYLINE `#define FLAG_POLYLINE 20`

18.109.1.20 FLAG_POLYLINE_SHORT `#define FLAG_POLYLINE_SHORT 21`

18.109.1.21 FLAG_QUIET `#define FLAG_QUIET 6`

18.109.1.22 FLAG_QUIET_SHORT `#define FLAG_QUIET_SHORT 7`

18.109.1.23 FLAG_RENDER `#define FLAG_RENDER 22`

18.109.1.24 FLAG_RENDER_SHORT `#define FLAG_RENDER_SHORT 23`

18.109.1.25 FLAG_SATIN `#define FLAG_SATIN 24`

18.109.1.26 FLAG_SATIN_SHORT `#define FLAG_SATIN_SHORT 25`

18.109.1.27 FLAG_SIERPINSKI_TRIANGLE `#define FLAG_SIERPINSKI_TRIANGLE 31`

18.109.1.28 FLAG_SIMULATE `#define FLAG_SIMULATE 34`

18.109.1.29 FLAG_STITCH `#define FLAG_STITCH 26`

18.109.1.30 FLAG_STITCH_SHORT `#define FLAG_STITCH_SHORT 27`

18.109.1.31 FLAG_TEST `#define FLAG_TEST 28`

18.109.1.32 FLAG_TO `#define FLAG_TO 0`

18.109.1.33 FLAG_TO_SHORT `#define FLAG_TO_SHORT 1`

18.109.1.34 FLAG_VERBOSE `#define FLAG_VERBOSE 8`

18.109.1.35 FLAG_VERBOSE_SHORT `#define FLAG_VERBOSE_SHORT 9`

18.109.1.36 FLAG_VERSION `#define FLAG_VERSION 10`

18.109.1.37 FLAG_VERSION_SHORT `#define FLAG_VERSION_SHORT 11`

18.109.1.38 NUM_FLAGS `#define NUM_FLAGS 37`

18.109.2 Function Documentation

18.109.2.1 bcf_difat_create() `bcf_file_difat * bcf_difat_create (`
 `FILE * file,`
 `unsigned int fatSectors,`
 `const unsigned int sectorSize)`
file fatSectors sectorSize

Returns

`bcf_file_difat*`

18.109.2.2 bcf_directory_free() `void bcf_directory_free (`
 `bcf_directory ** dir)`
dir

18.109.2.3 bcf_file_free() `void bcf_file_free (`
 `bcf_file * bcfFile)`
bcfFile

18.109.2.4 bcfFile_read() `int bcfFile_read (`
 `FILE * file,`
 `bcf_file * bcfFile)`
file bcfFile

Returns

`int`

18.109.2.5 bcfFileFat_create() `bcf_file_fat * bcfFileFat_create (`
 `const unsigned int sectorSize)`
sectorSize

Returns

`bcf_file_fat*`

18.109.2.6 bcfFileHeader_read() `bcf_file_header bcfFileHeader_read (`
 `FILE * file)`
file

Returns

`bcf_file_header`

18.109.2.7 binaryReadString() `void binaryReadString (`
 `FILE * file,`
 `char * buffer,`
 `int maxLength)`
file buffer maxLength

18.109.2.8 binaryReadUnicodeString() `void binaryReadUnicodeString (`
 `FILE * file,`
 `char * buffer,`
 `const int stringLength)`
file buffer stringLength

18.109.2.9 check_header_present() `int check_header_present (`
 `FILE * file,`
 `int minimum_header_length)`
file minimum_header_length

Returns

`int`

Checks that there are enough bytes to interpret the header, stops possible segfaults when reading in the header bytes.

Returns 0 if there aren't enough, or the length of the file if there are.

18.109.2.10 CompoundFileDirectory() `bcf_directory * CompoundFileDirectory (`
 `const unsigned int maxNumberOfDirectoryEntries)`
maxNumberOfDirectoryEntries

Returns

`bcf_directory*`

18.109.2.11 CompoundFileDirectoryEntry() `bcf_directory_entry * CompoundFileDirectoryEntry (`
 `FILE * file)`
file

Returns

`bcf_directory_entry*`

18.109.2.12 copy_trim() `char * copy_trim (`
 `char const * s)`

s

Returns

`char*`

Todo decription

18.109.2.13 emb_optOut() char * emb_optOut (
 EmbReal num,
 char * str)

Optimizes the number (*num*) for output to a text file and returns it as a string (*str*).
num str

Returns

char*

18.109.2.14 emb_readline() int emb_readline (
 FILE * file,
 char * line,
 int maxLength)

file line maxLength

Returns

int

18.109.2.15 embArc_print() void embArc_print (
 EmbArc arc)

arc

Todo move to [arc.c](#)

18.109.2.16 embColor_distance() int embColor_distance (
 EmbColor a,
 EmbColor b)

a b

Returns

int

18.109.2.17 embColor_read() void embColor_read (
 FILE * f,
 EmbColor * c,
 int toRead)

f c toRead

18.109.2.18 embColor_write() void embColor_write (
 FILE * f,
 EmbColor c,
 int toWrite)

f c toWrite

18.109.2.19 embSatinOutline_generateSatinOutline() void embSatinOutline_generateSatinOutline (
 EmbArray * lines,
 EmbReal thickness,
 EmbSatinOutline * result)

lines thickness result

18.109.2.20 embSatinOutline_renderStitches() `EmbArray * embSatinOutline_renderStitches (`
`EmbSatinOutline * result,`
`EmbReal density)`

result density

Returns

EmbArray*

18.109.2.21 embThread_findNearestColor() `int embThread_findNearestColor (`
`EmbColor color,`
`EmbColor * color_list,`
`int n_colors)`

Returns the closest color to the required color based on a list of available threads. The algorithm is a simple least squares search against the list. If the (square of) Euclidean 3-dimensional distance between the points in (red, green, blue) space is smaller then the index is saved and the remaining index is returned to the caller.

color The EmbColor color to match. *colors* The EmbThreadList pointer to start the search at. *mode* Is the argument an array of threads (0) or colors (1)?

Returns

closestIndex The entry in the ThreadList that matches.

18.109.2.22 embThread_findNearestThread() `int embThread_findNearestThread (`
`EmbColor color,`
`EmbThread * thread_list,`
`int n_threads)`

color thread_list n_threads

Returns

int

18.109.2.23 embThread_getRandom() `EmbThread embThread_getRandom (`
`void)`

Returns a random thread color, useful in filling in cases where the actual color of the thread doesn't matter but one needs to be declared to test or render a pattern.

Returns

c The resulting color.

18.109.2.24 embTime_initNow() `void embTime_initNow (`
`EmbTime * t)`

t

18.109.2.25 embTime_time() `EmbTime embTime_time (`
`EmbTime * t)`

t

Returns

EmbTime

18.109.2.26 embVector_print() void embVector_print (
 EmbVector v,
 char * label)

v label

move to [vector.c](#)

18.109.2.27 entriesInDifatSector() unsigned int entriesInDifatSector (
 bcf_file_difat * fat)

fat

Returns

unsigned int

18.109.2.28 get_trim_bounds() void get_trim_bounds (
 char const * s,
 char const ** firstWord,
 char const ** trailingSpace)

Get the trim bounds object.

s firstWord trailingSpace

18.109.2.29 GetFile() FILE * GetFile (
 bcf_file * bcfFile,
 FILE * file,
 char * fileToFind)

Get the File object.

bcfFile file fileToFind

Returns

FILE*

18.109.2.30 haveExtraDIFATSectors() int haveExtraDIFATSectors (
 bcf_file * file)

file

Returns

int

18.109.2.31 loadFatFromSector() void loadFatFromSector (
 bcf_file_fat * fat,
 FILE * file)

fat file

18.109.2.32 parseDIFATSectors() void parseDIFATSectors (
 FILE * file,
 bcf_file * bcfFile)

file bcfFile

18.109.2.33 parseDirectoryEntryName() void parseDirectoryEntryName (
 FILE * file,
 bcf_directory_entry * dir)

file dir

18.109.2.34 parseTime() `EmbTime parseTime (`
 `FILE * file)`

file

Returns

EmbTime

18.109.2.35 readFullSector() `unsigned int readFullSector (`
 `FILE * file,`
 `bcf_file_difat * bcfFile,`
 `unsigned int * difatEntriesToRead)`

file bcfFile difatEntriesToRead

Returns

unsigned int

18.109.2.36 readNextSector() `void readNextSector (`
 `FILE * file,`
 `bcf_directory * dir)`

file dir

18.109.2.37 sectorSize() `unsigned int sectorSize (`
 `bcf_file * bcfFile)`

bcfFile

Returns

unsigned int

18.109.2.38 seekToSector() `int seekToSector (`
 `bcf_file * bcfFile,`
 `FILE * file,`
 `const unsigned int sector)`

bcfFile file sector

Returns

int

18.109.2.39 stringInArray() `int stringInArray (`
 `const char * s,`
 `const char ** array)`

Tests for the presence of a string *s* in the supplied *array*.
The end of the array is marked by an empty string.

Returns

0 if not present 1 if present.

18.109.2.40 write_24bit() void write_24bit (
 FILE * *file*,
 int *x*)
file x

18.109.3 Variable Documentation

18.109.3.1 black_thread EmbThread black_thread = { { 0, 0, 0 }, "Black", "Black" }

18.109.3.2 difatEntriesInHeader const unsigned int difatEntriesInHeader = 109

18.109.3.3 emb_error int emb_error = 0
 Error code storage for optional control flow blocking.

18.109.3.4 emb_verbose int emb_verbose = 0
 Verbosity level.

18.109.3.5 embConstantPi const EmbReal embConstantPi = 3.1415926535

18.109.3.6 sizeOfChainingEntryAtEndOfDifatSector const unsigned int sizeOfChainingEntryAtEndOfDifatSector = 4

18.109.3.7 sizeOfDifatEntry const unsigned int sizeOfDifatEntry = 4

18.109.3.8 sizeOfDirectoryEntry const unsigned int sizeOfDirectoryEntry = 128

18.109.3.9 sizeOfFatEntry const unsigned int sizeOfFatEntry = sizeof(unsigned int)

18.109.3.10 WHITESPACE char const WHITESPACE[] = " \t\n\r"

18.110 extern/libembroidery/src/pattern.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "embroidery_internal.h"
```

Functions

- EmbPattern * embPattern_create (void)
- void embPattern_hideStitchesOverLength (EmbPattern *p, int length)
- int embPattern_addThread (EmbPattern *pattern, EmbThread thread)
- void embPattern_fixColorCount (EmbPattern *p)

- void `embPattern_copystitch_listToPolylines` (`EmbPattern *p`)
- void `embPattern_copyPolylinesToStitch_list` (`EmbPattern *p`)
- void `embPattern_movestitch_listToPolylines` (`EmbPattern *p`)
- void `embPattern_movePolylinesToStitch_list` (`EmbPattern *p`)
- void `embPattern_addStitchAbs` (`EmbPattern *p`, `EmbReal x`, `EmbReal y`, `int flags`, `int isAutoColorIndex`)
- void `embPattern_addStitchRel` (`EmbPattern *p`, `EmbReal dx`, `EmbReal dy`, `int flags`, `int isAutoColorIndex`)
- void `embPattern_changeColor` (`EmbPattern *p`, `int index`)
- void `embPattern_scale` (`EmbPattern *p`, `EmbReal scale`)
- `EmbRect` `embPattern_calcBoundingBox` (`EmbPattern *p`)
- void `embPattern_flipHorizontal` (`EmbPattern *p`)
- void `embPattern_flipVertical` (`EmbPattern *p`)
- void `embPattern_flip` (`EmbPattern *p`, `int horz`, `int vert`)
- void `embPattern_combineJumpStitches` (`EmbPattern *p`)
- void `embPattern_correctForMaxStitchLength` (`EmbPattern *p`, `EmbReal maxStitchLength`, `EmbReal maxJumpLength`)
- void `embPattern_center` (`EmbPattern *p`)
- void `embPattern_loadExternalColorFile` (`EmbPattern *p`, `const char *fileName`)
- void `embPattern_free` (`EmbPattern *p`)
- void `embPattern_addCircleAbs` (`EmbPattern *p`, `EmbCircle circle`)
- void `embPattern_addEllipseAbs` (`EmbPattern *p`, `EmbEllipse ellipse`)
- void `embPattern_addLineAbs` (`EmbPattern *p`, `EmbLine line`)
- void `embPattern_addPathAbs` (`EmbPattern *p`, `EmbPath obj`)
- void `embPattern_addPointAbs` (`EmbPattern *p`, `EmbPoint obj`)
- void `embPattern_addPolygonAbs` (`EmbPattern *p`, `EmbPolygon obj`)
- void `embPattern_addPolylineObjectAbs` (`EmbPattern *p`, `EmbPolyline obj`)
- void `embPattern_addRectAbs` (`EmbPattern *p`, `EmbRect rect`)
- void `embPattern_end` (`EmbPattern *p`)
- int `embPattern_color_count` (`EmbPattern *pattern`, `EmbColor startColor`)
- void `embPattern_designDetails` (`EmbPattern *pattern`)
- int `convert` (`const char *inf`, `const char *outf`)
- float `embPattern_totalStitchLength` (`EmbPattern *pattern`)
- float `embPattern_minimumStitchLength` (`EmbPattern *pattern`)
- float `embPattern_maximumStitchLength` (`EmbPattern *pattern`)
- void `embPattern_lengthHistogram` (`EmbPattern *pattern`, `int *bin`, `int NUMBINS`)
- int `embPattern_realStitches` (`EmbPattern *pattern`)
- int `embPattern_jumpStitches` (`EmbPattern *pattern`)
- int `embPattern_trimStitches` (`EmbPattern *pattern`)

18.110.1 Detailed Description

The file is for the management of the main struct: `EmbPattern`.

18.110.2 Function Documentation

18.110.2.1 `convert()` `int convert (`
`const char * inf,`
`const char * outf)`

18.110.2.2 `embPattern_addCircleAbs()` `void embPattern_addCircleAbs (`
`EmbPattern * p,`
`EmbCircle circle)`

Adds a circle object to pattern (*p*) with its center at the absolute position (*cx,cy*) with a radius of (*r*). Positive *y* is up. Units are in millimeters.

18.110.2.3 embPattern_addEllipseAbs() void embPattern_addEllipseAbs (
 EmbPattern * p,
 EmbEllipse ellipse)

Adds an ellipse object to pattern (*p*) with its center at the absolute position (*cx,cy*) with radii of (*rx,ry*). Positive y is up. Units are in millimeters.

18.110.2.4 embPattern_addLineAbs() void embPattern_addLineAbs (
 EmbPattern * p,
 EmbLine line)

Adds a line object to pattern (*p*) starting at the absolute position (*x1,y1*) and ending at the absolute position (*x2,y2*). Positive y is up. Units are in millimeters.

18.110.2.5 embPattern_addPathAbs() void embPattern_addPathAbs (
 EmbPattern * p,
 EmbPath obj)

18.110.2.6 embPattern_addPointAbs() void embPattern_addPointAbs (
 EmbPattern * p,
 EmbPoint obj)

Adds a point object to pattern (*p*) at the absolute position (*x,y*). Positive y is up. Units are in millimeters.

18.110.2.7 embPattern_addPolygonAbs() void embPattern_addPolygonAbs (
 EmbPattern * p,
 EmbPolygon obj)

18.110.2.8 embPattern_addPolylineObjectAbs() void embPattern_addPolylineObjectAbs (
 EmbPattern * p,
 EmbPolyline obj)

18.110.2.9 embPattern_addRectAbs() void embPattern_addRectAbs (
 EmbPattern * p,
 EmbRect rect)

Adds a rectangle object to pattern (*p*) at the absolute position (*x,y*) with a width of (*w*) and a height of (*h*). Positive y is up. Units are in millimeters.

18.110.2.10 embPattern_addStitchAbs() void embPattern_addStitchAbs (
 EmbPattern * p,
 EmbReal x,
 EmbReal y,
 int flags,
 int isAutoColorIndex)

Adds a stitch to the pattern (*p*) at the absolute position (*x,y*). Positive y is up. Units are in millimeters.

18.110.2.11 embPattern_addStitchRel() void embPattern_addStitchRel (
 EmbPattern * p,
 EmbReal dx,
 EmbReal dy,
 int flags,
 int isAutoColorIndex)

Adds a stitch to the pattern (*p*) at the relative position (*dx,dy*) to the previous stitch. Positive y is up. Units are in millimeters.

18.110.2.12 embPattern_addThread() `int embPattern_addThread (`
`EmbPattern * pattern,`
`EmbThread thread)`

pattern thread

Returns

`int`

18.110.2.13 embPattern_calcBoundingBox() `EmbRect embPattern_calcBoundingBox (`
`EmbPattern * p)`

Returns an EmbRect that encapsulates all stitches and objects in the pattern (*p*).

18.110.2.14 embPattern_center() `void embPattern_center (`
`EmbPattern * p)`

Center the pattern *p*.

18.110.2.15 embPattern_changeColor() `void embPattern_changeColor (`
`EmbPattern * p,`
`int index)`

Change the currentColorIndex of pattern *p* to *index*.

18.110.2.16 embPattern_color_count() `int embPattern_color_count (`
`EmbPattern * pattern,`
`EmbColor startColor)`

18.110.2.17 embPattern_combineJumpStitches() `void embPattern_combineJumpStitches (`
`EmbPattern * p)`

p

18.110.2.18 embPattern_copyPolylinesToStitch_list() `void embPattern_copyPolylinesToStitch_list (`
`EmbPattern * p)`

Copies all of the EmbPolylineObjectList data to Embstitch_list data for pattern (*p*).

18.110.2.19 embPattern_copystitch_listToPolylines() `void embPattern_copystitch_listToPolylines (`
`EmbPattern * p)`

Copies all of the Embstitch_list data to EmbPolylineObjectList data for pattern (*p*).

18.110.2.20 embPattern_correctForMaxStitchLength() `void embPattern_correctForMaxStitchLength (`
`EmbPattern * p,`
`EmbReal maxStitchLength,`
`EmbReal maxJumpLength)`

Todo The params determine the max XY movement rather than the length. They need renamed or clarified further.

18.110.2.21 embPattern_create() `EmbPattern * embPattern_create (`
`void)`

Returns a pointer to an EmbPattern. It is created on the heap. The caller is responsible for freeing the allocated memory with `embPattern_free()`.

Returns

`EmbPattern*`

18.110.2.22 embPattern_designDetails() void embPattern_designDetails (
 EmbPattern * pattern)

18.110.2.23 embPattern_end() void embPattern_end (
 EmbPattern * p)

18.110.2.24 embPattern_fixColorCount() void embPattern_fixColorCount (
 EmbPattern * p)

p

18.110.2.25 embPattern_flip() void embPattern_flip (
 EmbPattern * p,
 int horz,
 int vert)

Flips the entire pattern (*p*) horizontally about the x-axis if (*horz*) is true. Flips the entire pattern (*p*) vertically about the y-axis if (*vert*) is true.

18.110.2.26 embPattern_flipHorizontal() void embPattern_flipHorizontal (
 EmbPattern * p)

Flips the entire pattern (*p*) horizontally about the y-axis.

18.110.2.27 embPattern_flipVertical() void embPattern_flipVertical (
 EmbPattern * p)

Flips the entire pattern (*p*) vertically about the x-axis.

18.110.2.28 embPattern_free() void embPattern_free (
 EmbPattern * p)

Frees all memory allocated in the pattern (*p*).

18.110.2.29 embPattern_hideStitchesOverLength() void embPattern_hideStitchesOverLength (
 EmbPattern * p,
 int length)

p length

18.110.2.30 embPattern_jumpStitches() int embPattern_jumpStitches (
 EmbPattern * pattern)

18.110.2.31 embPattern_lengthHistogram() void embPattern_lengthHistogram (
 EmbPattern * pattern,
 int * bin,
 int NUMBINS)

18.110.2.32 embPattern_loadExternalColorFile() void embPattern_loadExternalColorFile (
 EmbPattern * p,
 const char * fileName)

TODO: Description needed.

18.110.2.33 embPattern_maximumStitchLength() float embPattern_maximumStitchLength (
 EmbPattern * pattern)

18.110.2.34 `embPattern_minimumStitchLength()` `float embPattern_minimumStitchLength (EmbPattern * pattern)`

18.110.2.35 `embPattern_movePolylinesTostitch_list()` `void embPattern_movePolylinesTostitch_list (EmbPattern * p)`

Moves all of the EmbPolylineObjectList data to Embstitch_list data for pattern (*p*).

18.110.2.36 `embPattern_movestitch_listToPolylines()` `void embPattern_movestitch_listToPolylines (EmbPattern * p)`

Moves all of the Embstitch_list data to EmbPolylineObjectList data for pattern (*p*).

18.110.2.37 `embPattern_realStitches()` `int embPattern_realStitches (EmbPattern * pattern)`

18.110.2.38 `embPattern_scale()` `void embPattern_scale (EmbPattern * p, EmbReal scale)`

Very simple scaling of the x and y axis for every point. Doesn't insert or delete stitches to preserve density.

18.110.2.39 `embPattern_totalStitchLength()` `float embPattern_totalStitchLength (EmbPattern * pattern)`

pattern

Returns

float

18.110.2.40 `embPattern_trimStitches()` `int embPattern_trimStitches (EmbPattern * pattern)`

18.111 extern/libembroidery/src/thread-color.c File Reference

```
#include <stdio.h>
#include <string.h>
#include "embroidery_internal.h"
```

Functions

- int `threadColor` (const char *name, int brand)
- int `threadColorNum` (unsigned int color, int brand)
- const char * `threadColorName` (unsigned int color, int brand)

Variables

- const unsigned char `_dxfColorTable` [][3] = {{ 0, 0, 0 }}
- const `EmbThread husThreads` [] = {{{ 0, 0, 0 }, "END", "END"}}
- const `EmbThread jefThreads` [] = {{{ 0, 0, 0 }, "END", "END"}}
- const `EmbThread shvThreads` [] = {{{ 0, 0, 0 }, "END", "END"}}
- const `EmbThread pcmThreads` [] = {{{ 0, 0, 0 }, "END", "END"}}
- const `EmbThread pecThreads` [] = {{{ 0, 0, 0 }, "END", "END"}}
- const int `shvThreadCount` = 42
- const int `pecThreadCount` = 65
- `thread_color` * `brand_codes` []
- const char * `brand_codes_files` []

18.111.1 Function Documentation

18.111.1.1 threadColor() `int threadColor (`
 `const char * name,`
 `int brand)`

18.111.1.2 threadColorName() `const char * threadColorName (`
 `unsigned int color,`
 `int brand)`

18.111.1.3 threadColorNum() `int threadColorNum (`
 `unsigned int color,`
 `int brand)`

18.111.2 Variable Documentation

18.111.2.1 _dxfColorTable `const unsigned char _dxfColorTable[][3] = {{ 0, 0, 0 }}`

18.111.2.2 brand_codes `thread_color* brand_codes[]`

18.111.2.3 brand_codes_files `const char* brand_codes_files[]`

Initial value:

```
= {
    "arc_polyester_colors.csv",
    "arc_rayon_colors.csv",
    "coats_and_clark_rayon_colors.csv",
    "exquisite_polyester_colors.csv",
    "fufu_Polyester_colors.csv",
    "fufu_Rayon_colors.csv",
    "Hemingworth_Polyester_colors.csv",
    "Isacord_Polyester_colors.csv",
    "Isafil_Rayon_colors.csv",
    "Marathon_Polyester_colors.csv",
    "Marathon_Rayon_colors.csv",
    "Madeira_Polyester_colors.csv",
    "Madeira_Rayon_colors.csv",
    "Metro_Polyester_colors.csv",
    "Pantone_colors.csv",
    "RobisonAnton_Polyester_colors.csv",
    "RobisonAnton_Rayon_colors.csv",
    "Sigma_Polyester_colors.csv",
    "Sulky_Rayon_colors.csv",
    "ThreadArt_Rayon_colors.csv",
    "ThreadArt_Polyester_colors.csv",
    "ThreaDelight_Polyester_colors.csv",
    "Z102_Isacord_Polyester_colors.csv",
    "svg_color_colors.csv"
}
```

18.111.2.4 husThreads `const EmbThread husThreads[] = {{{ 0, 0, 0 }, "END", "END"}}`

18.111.2.5 jefThreads `const EmbThread jefThreads[] = {{{ 0, 0, 0 }, "END", "END"}}`

18.111.2.6 pcmThreads `const EmbThread pcmThreads[] = {{{ 0, 0, 0 }, "END", "END"}}`

18.111.2.7 pecThreadCount `const int pecThreadCount = 65`

18.111.2.8 pecThreads `const EmbThread pecThreads[] = {{{ 0, 0, 0 }, "END", "END"}}`

18.111.2.9 shvThreadCount `const int shvThreadCount = 42`

18.111.2.10 shvThreads `const EmbThread shvThreads[] = {{{ 0, 0, 0 }, "END", "END"}}`

18.112 privacy_policy.md File Reference

References

- [1] acatina. Technical info. [411](#)
- [2] KDE Community. Projects/liberty/file formats/tajima ternary - kde community wiki. [411](#)
- [3] G. van Rossum and B. Warsaw. Python pep 7. [12](#)

Index

- `__mainWin`
 - Application, [50](#)
- `_appVer_`
 - embroidermodder.cpp, [222](#)
- `_bcf_directory,` [40](#)
 - dirEntries, [40](#)
 - maxNumberOfDirectoryEntries, [40](#)
- `_bcf_directory_entry,` [40](#)
 - childId, [41](#)
 - CLSID, [41](#)
 - colorFlag, [41](#)
 - creationTime, [41](#)
 - directoryEntryName, [41](#)
 - directoryEntryNameLength, [41](#)
 - leftSiblingId, [41](#)
 - modifiedTime, [41](#)
 - next, [41](#)
 - objectType, [41](#)
 - rightSiblingId, [41](#)
 - startingSectorLocation, [42](#)
 - stateBits, [42](#)
 - streamSize, [42](#)
 - streamSizeHigh, [42](#)
- `_bcf_file,` [42](#)
 - difat, [42](#)
 - directory, [42](#)
 - fat, [42](#)
 - header, [43](#)
- `_bcf_file_difat,` [43](#)
 - fatSectorCount, [43](#)
 - fatSectorEntries, [43](#)
 - sectorSize, [43](#)
- `_bcf_file_fat,` [43](#)
 - fatEntries, [44](#)
 - fatEntryCount, [44](#)
 - numberOfEntriesInFatSector, [44](#)
- `_bcf_file_header,` [44](#)
 - byteOrder, [45](#)
 - CLSID, [45](#)
 - firstDifatSectorLocation, [45](#)
 - firstDirectorySectorLocation, [45](#)
 - firstMiniFATSectorLocation, [45](#)
 - majorVersion, [45](#)
 - miniSectorShift, [45](#)
 - miniStreamCutoffSize, [45](#)
 - minorVersion, [45](#)
 - numberOfDifatSectors, [45](#)
 - numberOfDirectorySectors, [46](#)
 - numberOfFAT Sectors, [46](#)
 - numberOfMiniFatSectors, [46](#)
 - reserved1, [46](#)
 - reserved2, [46](#)
 - sectorShift, [46](#)
 - signature, [46](#)
 - transactionSignatureNumber, [46](#)
- `_dxfColorTable`
 - embroidery.h, [336](#)
 - thread-color.c, [477](#)
- `_mainWin`
 - embroidermodder.h, [236](#)
 - mainwindow.cpp, [292](#)
- `_subMask`
 - format_csd.c, [408](#)
- `_vp3Hoop,` [46](#)
 - bottom, [47](#)
 - bottom2, [47](#)
 - byte1, [47](#)
 - byte2, [47](#)
 - byte3, [47](#)
 - height, [47](#)
 - left, [48](#)
 - left2, [48](#)
 - numberOfBytesRemaining, [48](#)
 - numberOfColors, [48](#)
 - right, [48](#)
 - right2, [48](#)
 - threadLength, [48](#)
 - top, [48](#)
 - top2, [48](#)
 - unknown2, [48](#)
 - unknown3, [48](#)
 - unknown4, [49](#)
 - width, [49](#)
 - xOffset, [49](#)
 - yOffset, [49](#)
- `_xorMask`
 - format_csd.c, [408](#)
- `~CmdPrompt`
 - CmdPrompt, [52](#)
- `~CmdPromptHandle`
 - CmdPromptHandle, [60](#)
- `~CmdPromptHistory`
 - CmdPromptHistory, [63](#)
- `~CmdPromptInput`
 - CmdPromptInput, [67](#)
- `~CmdPromptSplitter`
 - CmdPromptSplitter, [74](#)
- `~EmbDetailsDialog`
 - EmbDetailsDialog, [82](#)
- `~Geometry`
 - Geometry, [109](#)
- `~ImageWidget`
 - ImageWidget, [133](#)
- `~LayerManager`
 - LayerManager, [135](#)
- `~MainWindow`
 - MainWindow, [141](#)
- `~MdiArea`
 - MdiArea, [157](#)
- `~MdiWindow`

- MdiWindow, 161
- ~PreviewDialog
 - PreviewDialog, 169
- ~PropertyEditor
 - PropertyEditor, 171
- ~SaveObject
 - SaveObject, 177
- ~Settings_Dialog
 - Settings_Dialog, 190
- ~UndoEditor
 - UndoEditor, 205
- ~View
 - View, 209
- 10o, 9, 403
- 100, 9, 403
- about
 - MainWindow, 141
- about_action
 - mainwindow.cpp, 271
- accept_
 - settings-dialog.cpp, 298
- acceptChanges
 - Settings_Dialog, 191
- actionHash
 - embroidermodder.h, 237
 - mainwindow.cpp, 292
- activeCommand
 - MainWindow, 141
- activeMdiWindow
 - MainWindow, 141
- activeScene
 - embroidermodder.h, 228
 - mainwindow.cpp, 271
- activeUndoStack
 - MainWindow, 141
- activeView
 - embroidermodder.h, 228
 - mainwindow.cpp, 271
- actuator
 - embroidermodder.h, 228
 - mainwindow.cpp, 271
- add_arc_action
 - mainwindow.cpp, 271
- add_circle_action
 - mainwindow.cpp, 272
- add_dim_leader_action
 - mainwindow.cpp, 272
- add_ellipse_action
 - mainwindow.cpp, 272
- add_geometry_action
 - mainwindow.cpp, 272
- add_horizontal_dimension_action
 - mainwindow.cpp, 272
- add_image_action
 - mainwindow.cpp, 272
- add_infinite_line_action
 - mainwindow.cpp, 272
- add_line_action
 - mainwindow.cpp, 272
- add_path_action
 - mainwindow.cpp, 273
- add_point_action
 - mainwindow.cpp, 273
- add_polygon_action
 - mainwindow.cpp, 273
- add_polyline
 - embroidermodder.h, 229
 - objects.cpp, 295
- add_polyline_action
 - mainwindow.cpp, 273
- add_ray_action
 - mainwindow.cpp, 274
- add_rectangle_action
 - mainwindow.cpp, 274
- add_regular_polygon_action
 - mainwindow.cpp, 274
- add_rounded_rectangle_action
 - mainwindow.cpp, 274
- add_rubber_action
 - mainwindow.cpp, 274
- add_slot_action
 - mainwindow.cpp, 274
- add_text_multi_action
 - mainwindow.cpp, 275
- add_text_single_action
 - mainwindow.cpp, 275
- add_to_path
 - interface.cpp, 257
- add_to_selection_action
 - mainwindow.cpp, 275
- add_triangle_action
 - mainwindow.cpp, 275
- add_vertical_dimension_action
 - mainwindow.cpp, 276
- addArc
 - SaveObject, 178
- addBlock
 - SaveObject, 178
- addCircle
 - SaveObject, 178
- addColorsToComboBox
 - Settings_Dialog, 191
- addDimAligned
 - SaveObject, 178
- addDimAngular
 - SaveObject, 179
- addDimArcLength
 - SaveObject, 179
- addDimDiameter
 - SaveObject, 179
- addDimLeader
 - SaveObject, 179
- addDimLinear
 - SaveObject, 180
- addDimOrdinate

- SaveObject, [180](#)
- addDimRadius
 - SaveObject, [180](#)
- addEllipse
 - SaveObject, [181](#)
- addEllipseArc
 - SaveObject, [181](#)
- addGrid
 - SaveObject, [181](#)
- addHatch
 - SaveObject, [181](#)
- addImage
 - SaveObject, [182](#)
- addInfiniteLine
 - SaveObject, [182](#)
- addLayer
 - LayerManager, [135](#)
- addLine
 - SaveObject, [182](#)
- addObject
 - View, [209](#)
- addPath
 - SaveObject, [182](#)
- addPoint
 - SaveObject, [183](#)
- addPolygon
 - SaveObject, [183](#)
- addPolyline
 - SaveObject, [183](#)
- addRay
 - SaveObject, [184](#)
- addRectangle
 - SaveObject, [184](#)
- addSlot
 - SaveObject, [184](#)
- addSpline
 - SaveObject, [184](#)
- addStack
 - UndoEditor, [205](#)
- addTextMulti
 - SaveObject, [185](#)
- addTextSingle
 - SaveObject, [185](#)
- addToRubberRoom
 - View, [209](#)
- after
 - UndoableCommand, [203](#)
- alert
 - CmdPrompt, [53](#)
- alert_action
 - mainwindow.cpp, [276](#)
- alignScenePointWithViewPoint
 - View, [209](#)
- allGripPoints
 - Geometry, [109](#)
- allow_rubber_action
 - mainwindow.cpp, [276](#)
- allowRubber
 - View, [210](#)
- allowZoomIn
 - View, [210](#)
- allowZoomOut
 - View, [210](#)
- alpha
 - SelectBox, [188](#)
- alphabet
 - LSYSTEM, [136](#)
- Ameco, [373](#), [424](#)
- angle
 - UndoableCommand, [203](#)
- append_history_action
 - mainwindow.cpp, [276](#)
- append_prompt_history_action
 - mainwindow.cpp, [276](#)
- appendHistory
 - CmdPrompt, [53](#)
 - CmdPromptHistory, [63](#)
 - CmdPromptInput, [67](#)
- appendTheHistory
 - CmdPrompt, [53](#)
- Application, [49](#)
 - __mainWin, [50](#)
 - Application, [49](#)
 - event, [50](#)
 - setMainWin, [50](#)
- applyFormatting
 - CmdPromptHistory, [64](#)
 - CmdPromptInput, [67](#)
- arc
 - EmbGeometry_, [87](#)
- arc.c
 - Arc_clockwise, [447](#)
 - Base_objectRubberPoint, [447](#)
 - Base_objectRubberText, [448](#)
 - Base_setLineType, [448](#)
 - Base_setLineWeight, [448](#)
 - clockwise, [448](#)
 - embArc_arcLength, [448](#)
 - embArc_area, [448](#)
 - embArc_chord, [448](#)
 - embArc_clockwise, [448](#)
 - embArc_endAngle, [448](#)
 - embArc_gripEdit, [448](#)
 - embArc_includedAngle, [448](#)
 - embArc_init, [448](#)
 - embArc_mouseSnapPoint, [449](#)
 - embArc_paint, [449](#)
 - embArc_setCenter, [449](#)
 - embArc_setEndAngle, [449](#)
 - embArc_setRadius, [449](#)
 - embArc_setStartAngle, [449](#)
 - embArc_startAngle, [449](#)
 - embArc_updatePath, [449](#)
 - embArc_updateRubber, [449](#)
 - embBase_setColorRGB, [449](#)
 - embCircle_prompt, [449](#)

- embCircle_setArea, 449
- embCircle_setCircumference, 450
- embEllipse_click, 450
- embEllipse_main, 450
- embRect_bottomLeft, 450
- embRect_bottomRight, 450
- getArcCenter, 450
- getArcDataFromBulge, 450
- set_object_color, 450
- Arc_clockwise
 - arc.c, 447
- Arc_Polyester
 - embroidery.h, 311
- Arc_Rayon
 - embroidery.h, 312
- arcEndPoint
 - Geometry, 128
- arcMidPoint
 - Geometry, 128
- arcStartPoint
 - Geometry, 128
- array.c
 - embArray_addArc, 301
 - embArray_addCircle, 301
 - embArray_addEllipse, 301
 - embArray_addFlag, 301
 - embArray_addLine, 301
 - embArray_addPath, 301
 - embArray_addPoint, 301
 - embArray_addPolygon, 301
 - embArray_addPolyline, 301
 - embArray_addRect, 301
 - embArray_addStitch, 301
 - embArray_addVector, 302
 - embArray_copy, 302
 - embArray_create, 302
 - embArray_free, 302
 - embArray_resize, 302
- ArrowStyle
 - Geometry, 107
- arrowStyleAngle
 - Geometry, 128
- arrowStyleLength
 - Geometry, 128
- arrowStylePath
 - Geometry, 128
- art, 9, 404
- attributeList
 - format_svg.c, 437
- attributeOffset
 - VipHeader_, 220
- AutoCAD, 376, 414, 433
- AutoDesk, 414
- auxFormat
 - ThredExtension_, 200
- axiom
 - LSYSTEM, 137
- b
 - EmbColor_, 81
 - Node_, 168
 - Barudan, 377, 410, 441
 - Base_objectRubberPoint
 - arc.c, 447
 - Base_objectRubberText
 - arc.c, 448
 - Base_setLineType
 - arc.c, 448
 - Base_setLineWeight
 - arc.c, 448
 - bcf_difat_create
 - embroidery_internal.h, 362
 - main.c, 465
 - bcf_directory
 - embroidery_internal.h, 360
 - bcf_directory_entry
 - embroidery_internal.h, 360
 - bcf_directory_free
 - embroidery_internal.h, 362
 - main.c, 465
 - bcf_file
 - embroidery_internal.h, 360
 - bcf_file_difat
 - embroidery_internal.h, 360
 - bcf_file_difat_free
 - embroidery_internal.h, 362
 - bcf_file_fat
 - embroidery_internal.h, 360
 - bcf_file_fat_free
 - embroidery_internal.h, 362
 - bcf_file_free
 - embroidery_internal.h, 362
 - main.c, 465
 - bcf_file_header
 - embroidery_internal.h, 360
 - bcfFile_read
 - embroidery_internal.h, 362
 - main.c, 465
 - bcfFileFat_create
 - embroidery_internal.h, 362
 - main.c, 465
 - bcfFileHeader_isValid
 - embroidery_internal.h, 362
 - bcfFileHeader_read
 - embroidery_internal.h, 363
 - main.c, 465
 - before
 - UndoableCommand, 203
 - Bernina, 404
 - beziers
 - EmbSpline_, 98
 - bgColor
 - MdiArea, 159
 - bgLogo
 - MdiArea, 159
 - bgTexture
 - MdiArea, 159

- binaryReadString
 - embroidery_internal.h, 363
 - main.c, 465
- binaryReadUnicodeString
 - embroidery_internal.h, 363
 - main.c, 466
- binaryWriteInt
 - embroidery_internal.h, 363
 - formats.c, 400
- binaryWriteIntBE
 - embroidery_internal.h, 363
 - formats.c, 400
- binaryWriteShort
 - embroidery_internal.h, 363
 - formats.c, 400
- binaryWriteUInt
 - embroidery_internal.h, 363
 - formats.c, 400
- binaryWriteUIntBE
 - embroidery_internal.h, 363
 - formats.c, 400
- binaryWriteUShort
 - embroidery_internal.h, 364
 - formats.c, 400
- binaryWriteUShortBE
 - embroidery_internal.h, 364
 - formats.c, 400
- bit_position
 - Compress, 75
- Bitmap Cache, 405
- Bits and Volts, 405
- bits_total
 - Compress, 75
- black_thread
 - embroidery.h, 336
 - main.c, 471
- blink
 - CmdPrompt, 53
- blink_prompt_action
 - mainwindow.cpp, 277
- blinkState
 - CmdPrompt, 59
- blinkTimer
 - CmdPrompt, 59
- block_elements
 - Compress, 75
- bmc, 405
- BOOL_TYPE
 - embroidermodder.h, 226
- bottom
 - _vp3Hoop, 47
 - EmbRect_, 96
 - hoop_padding, 131
- bottom2
 - _vp3Hoop, 47
- boundingRect
 - EmbDetailsDialog, 83
 - Geometry, 109
- Box
 - Geometry, 107
- boxDir
 - SelectBox, 188
- brand_codes
 - thread-color.c, 477
- brand_codes_files
 - thread-color.c, 477
- bro, 9, 405
- Brother, 374, 375, 428, 429, 432, 433
- BuildDecryptionTable
 - format_csd.c, 407
- BULGETOCONTROL
 - embroidery_internal.h, 352
- BULGETOEND
 - embroidery_internal.h, 353
- buttonBox
 - EmbDetailsDialog, 83
 - Settings_Dialog, 196
- buttonCustomFilterClearAll
 - Settings_Dialog, 191
- buttonCustomFilterClearAllClicked
 - Settings_Dialog, 191
- buttonCustomFilterSelectAll
 - Settings_Dialog, 191
- buttonCustomFilterSelectAllClicked
 - Settings_Dialog, 191
- buttonQSnapClearAll
 - Settings_Dialog, 191
- buttonQSnapClearAllClicked
 - Settings_Dialog, 191
- buttonQSnapSelectAll
 - Settings_Dialog, 191
- buttonQSnapSelectAllClicked
 - Settings_Dialog, 191
- buttons
 - StatusBar, 198
- buttonTipOfTheDayClicked
 - MainWindow, 142
- byte1
 - _vp3Hoop, 47
- byte2
 - _vp3Hoop, 47
- byte3
 - _vp3Hoop, 47
- byteOrder
 - _bcf_file_header, 45
- calculate_angle_action
 - mainwindow.cpp, 277
- calculate_distance_action
 - mainwindow.cpp, 277
- calculateArcData
 - Geometry, 109
- canRedo
 - UndoEditor, 205
- canUndo
 - UndoEditor, 205
- cascade

- MdiArea, 157
- catalogNumber
 - EmbThread_, 100
- cci
 - format_dst.c, 412
- center
 - EmbCircle_, 80
 - EmbEllipse_, 85
 - View, 210
- centerAt
 - View, 210
- changeFormatting
 - CmdPromptInput, 67
- changelog_action
 - mainwindow.cpp, 277
- character_huffman
 - Compress, 75
- character_length_huffman
 - Compress, 75
- check_for_color_file
 - EmbFormatList_, 85
- check_header_present
 - embroidery_internal.h, 364
 - main.c, 466
- checkBoxCustomFilterStateChanged
 - Settings_Dialog, 191
- checkBoxes
 - embroidermodder.h, 237
 - mainwindow.cpp, 292
- checkBoxGeneralMdiBGUseColorStateChanged
 - Settings_Dialog, 191
- checkBoxGeneralMdiBGUseLogoStateChanged
 - Settings_Dialog, 191
- checkBoxGeneralMdiBGUseTextureStateChanged
 - Settings_Dialog, 191
- checkBoxGridCenterOnOriginStateChanged
 - Settings_Dialog, 192
- checkBoxGridColorMatchCrossHairStateChanged
 - Settings_Dialog, 192
- checkBoxGridLoadFromFileStateChanged
 - Settings_Dialog, 192
- checkBoxLwtRealRenderStateChanged
 - Settings_Dialog, 192
- checkBoxLwtShowLwtStateChanged
 - Settings_Dialog, 192
- checkBoxPromptSaveHistoryAsHtmlStateChanged
 - Settings_Dialog, 192
- checkBoxRulerShowOnLoadStateChanged
 - Settings_Dialog, 192
- checkBoxShowScrollBarsStateChanged
 - Settings_Dialog, 192
- checkBoxTipOfTheDay
 - mainwindow.cpp, 292
- checkChangedText
 - CmdPromptInput, 68
- checkCursorPosition
 - CmdPromptInput, 68
- checkEditedText
 - CmdPromptInput, 68
- checkForUpdates
 - MainWindow, 142
- checkSelection
 - CmdPromptInput, 68
- childId
 - _bcf_directory_entry, 41
- chooseDisplayBackgroundColor
 - Settings_Dialog, 192
- chooseDisplayCrossHairColor
 - Settings_Dialog, 192
- chooseDisplaySelectBoxLeftColor
 - Settings_Dialog, 192
- chooseDisplaySelectBoxLeftFill
 - Settings_Dialog, 192
- chooseDisplaySelectBoxRightColor
 - Settings_Dialog, 193
- chooseDisplaySelectBoxRightFill
 - Settings_Dialog, 193
- chooseGeneralMdiBackgroundColor
 - Settings_Dialog, 193
- chooseGeneralMdiBackgroundLogo
 - Settings_Dialog, 193
- chooseGeneralMdiBackgroundTexture
 - Settings_Dialog, 193
- chooseGridColor
 - Settings_Dialog, 193
- choosePromptBackgroundColor
 - Settings_Dialog, 193
- choosePromptTextColor
 - Settings_Dialog, 193
- chooseRulerColor
 - Settings_Dialog, 193
- CHUNK_SIZE
 - embroidery.h, 312
- circle
 - EmbGeometry_, 87
- circle.c
 - embCircle_area, 451
 - embCircle_circumference, 451
 - embCircle_init, 451
 - getCircleCircleIntersections, 451
 - getCircleTangentPoints, 451
- circle_click
 - Geometry, 110
- clear_rubber_action
 - mainwindow.cpp, 277
- clear_selection_action
 - mainwindow.cpp, 277
- clearAllFields
 - PropertyEditor, 171
- clearFormatting
 - CmdPromptInput, 68
- clearRubberRoom
 - View, 210
- clearSelection
 - View, 210
- clockwise

- arc.c, 448
- Closed
 - Geometry, 107
- closeEvent
 - MainWindow, 142
 - MdiWindow, 162
- closest_point
 - objects.cpp, 295
- closeToolBar
 - MainWindow, 142
- CLSID
 - _bcf_directory_entry, 41
 - _bcf_file_header, 45
- cmdActive
 - CmdPromptInput, 72
- CmdPrompt, 50
 - ~CmdPrompt, 52
 - alert, 53
 - appendHistory, 53
 - appendTheHistory, 53
 - blink, 53
 - blinkState, 59
 - blinkTimer, 59
 - CmdPrompt, 52
 - copyPressed, 53
 - cutPressed, 53
 - deletePressed, 54
 - downPressed, 54
 - escapePressed, 54
 - F10Pressed, 54
 - F11Pressed, 54
 - F12Pressed, 54
 - F1Pressed, 54
 - F2Pressed, 54
 - F3Pressed, 54
 - F4Pressed, 54
 - F5Pressed, 54
 - F6Pressed, 55
 - F7Pressed, 55
 - F8Pressed, 55
 - F9Pressed, 55
 - floatingChanged, 55
 - historyAppended, 55
 - pastePressed, 55
 - promptDivider, 59
 - promptHistory, 59
 - promptInput, 59
 - promptSplitter, 59
 - promptVBoxLayout, 59
 - redoPressed, 55
 - runCommand, 55
 - saveHistory, 56
 - selectAllPressed, 56
 - setCurrentText, 56
 - setHistory, 56
 - setPrefix, 56
 - setPromptBackgroundColor, 56
 - setPromptFontFamily, 57
 - setPromptFontSize, 57
 - setPromptFontStyle, 57
 - setPromptTextColor, 57
 - shiftPressed, 58
 - shiftReleased, 58
 - showSettings, 58
 - startBlinking, 58
 - startCommand, 58
 - stopBlinking, 58
 - styleHash, 59
 - tabPressed, 58
 - undoPressed, 58
 - updateStyle, 58
 - upPressed, 58
- CmdPromptHandle, 59
 - ~CmdPromptHandle, 60
 - CmdPromptHandle, 60
 - handleMoved, 61
 - handlePressed, 61
 - handleReleased, 61
 - mouseMoveEvent, 61
 - mousePressEvent, 61
 - mouseReleaseEvent, 61
 - moveY, 62
 - pressY, 62
 - releaseY, 62
- CmdPromptHistory, 62
 - ~CmdPromptHistory, 63
 - appendHistory, 63
 - applyFormatting, 64
 - CmdPromptHistory, 63
 - contextMenuEvent, 64
 - historyAppended, 64
 - resizeHistory, 64
 - startResizeHistory, 65
 - stopResizeHistory, 65
 - tmpHeight, 65
- CmdPromptInput, 65
 - ~CmdPromptInput, 67
 - appendHistory, 67
 - applyFormatting, 67
 - changeFormatting, 67
 - checkChangedText, 68
 - checkCursorPosition, 68
 - checkEditedText, 68
 - checkSelection, 68
 - clearFormatting, 68
 - cmdActive, 72
 - CmdPromptInput, 67
 - contextMenuEvent, 69
 - copyClip, 69
 - copyPressed, 69
 - curCmd, 72
 - curText, 73
 - cutPressed, 69
 - defaultPrefix, 73
 - deletePressed, 69
 - downPressed, 69

- endCommand, 69
- escapePressed, 69
- eventFilter, 69
- F10Pressed, 70
- F11Pressed, 70
- F12Pressed, 70
- F1Pressed, 70
- F2Pressed, 70
- F3Pressed, 70
- F4Pressed, 70
- F5Pressed, 70
- F6Pressed, 70
- F7Pressed, 70
- F8Pressed, 71
- F9Pressed, 71
- isBlinking, 73
- lastCmd, 73
- pasteClip, 71
- pastePressed, 71
- prefix, 73
- processInput, 71
- rapidFireEnabled, 73
- redoPressed, 71
- runCommand, 71
- selectAllPressed, 71
- shiftPressed, 71
- shiftReleased, 71
- showSettings, 72
- startCommand, 72
- stopBlinking, 72
- tabPressed, 72
- undoPressed, 72
- updateCurrentText, 72
- upPressed, 72
- CmdPromptSplitter, 73
 - ~CmdPromptSplitter, 74
 - CmdPromptSplitter, 74
 - createHandle, 74
 - moveResizeHistory, 74
 - pressResizeHistory, 74
 - releaseResizeHistory, 75
- cnd, 9, 406
- CoatsAndClark_Rayon
 - embroidery.h, 312
- CODE_OF_CONDUCT.md, 221
- col, 9, 406
- color
 - EmbGeometry_, 87
 - EmbLine_, 91
 - EmbPath_, 93
 - EmbPoint_, 95
 - EmbStitch_, 98
 - EmbThread_, 100
- color_only
 - EmbFormatList_, 85
- colorChanges
 - EmbDetailsDialog, 83
- colorCode
 - StxThread_, 198
 - SubDescriptor_, 199
- colorFlag
 - _bcf_directory_entry, 41
- colorLength
 - VipHeader_, 220
- colorName
 - StxThread_, 198
 - SubDescriptor_, 199
- colorSelector
 - MainWindow, 154
- colorSelectorIndexChanged
 - MainWindow, 142
- colorTotal
 - EmbDetailsDialog, 83
- comboBoxes
 - embroidermodder.h, 237
 - mainwindow.cpp, 292
- comboBoxGridTypeCurrentIndexChanged
 - Settings_Dialog, 193
- comboBoxIconSizeCurrentIndexChanged
 - Settings_Dialog, 193
- comboBoxPromptFontFamilyCurrentIndexChanged
 - Settings_Dialog, 194
- comboBoxPromptFontStyleCurrentIndexChanged
 - Settings_Dialog, 194
- comboBoxQSnapLocatorColorCurrentIndexChanged
 - Settings_Dialog, 194
- comboBoxRulerMetricCurrentIndexChanged
 - Settings_Dialog, 194
- comboBoxScrollBarWidgetCurrentIndexChanged
 - Settings_Dialog, 194
- comboBoxSelected
 - PropertyEditor, 174
- comboBoxSelectionCoolGripColorCurrentIndexChanged
 - Settings_Dialog, 194
- comboBoxSelectionHotGripColorCurrentIndexChanged
 - Settings_Dialog, 194
- comboBoxTextSingleFont
 - property-editor.cpp, 296
- Command
 - embroidermodder.h, 226
- command
 - UndoableCommand, 203
- command_map
 - mainwindow.cpp, 292
- CompoundFileDirectory
 - embroidery_internal.h, 364
 - main.c, 466
- CompoundFileDirectoryEntry
 - embroidery_internal.h, 364
 - main.c, 466
- CompoundFileSector_DIFAT_Sector
 - embroidery_internal.h, 353
- CompoundFileSector_EndOfChain
 - embroidery_internal.h, 353
- CompoundFileSector_FAT_Sector
 - embroidery_internal.h, 353

- CompoundFileSector_FreeSector
 - embroidery_internal.h, [353](#)
- CompoundFileSector_MaxRegSector
 - embroidery_internal.h, [353](#)
- CompoundFileStreamId_MaxRegularStreamId
 - embroidery_internal.h, [353](#)
- CompoundFileStreamId_NoStream
 - embroidery_internal.h, [353](#)
- Compress, [75](#)
 - bit_position, [75](#)
 - bits_total, [75](#)
 - block_elements, [75](#)
 - character_huffman, [75](#)
 - character_length_huffman, [75](#)
 - distance_huffman, [76](#)
 - input_data, [76](#)
 - input_length, [76](#)
- compress
 - embroidery_internal.h, [361](#)
- compress.c
 - compress_get_bits, [303](#)
 - compress_get_position, [303](#)
 - compress_get_token, [303](#)
 - compress_init, [303](#)
 - compress_load_block, [303](#)
 - compress_load_character_huffman, [303](#)
 - compress_load_character_length_huffman, [303](#)
 - compress_load_distance_huffman, [303](#)
 - compress_peek, [303](#)
 - compress_pop, [303](#)
 - compress_read_variable_length, [303](#)
 - huffman_build_table, [304](#)
 - huffman_lookup, [304](#)
 - huffman_lookup_data, [304](#)
 - hus_compress, [304](#)
 - hus_decompress, [304](#)
- compress_get_bits
 - compress.c, [303](#)
 - embroidery_internal.h, [364](#)
- compress_get_position
 - compress.c, [303](#)
 - embroidery_internal.h, [364](#)
- compress_get_token
 - compress.c, [303](#)
 - embroidery_internal.h, [365](#)
- compress_init
 - compress.c, [303](#)
- compress_load_block
 - compress.c, [303](#)
 - embroidery_internal.h, [365](#)
- compress_load_character_huffman
 - compress.c, [303](#)
 - embroidery_internal.h, [365](#)
- compress_load_character_length_huffman
 - compress.c, [303](#)
 - embroidery_internal.h, [365](#)
- compress_load_distance_huffman
 - compress.c, [303](#)
 - embroidery_internal.h, [365](#)
- compress_peek
 - compress.c, [303](#)
 - embroidery_internal.h, [365](#)
- compress_pop
 - compress.c, [303](#)
 - embroidery_internal.h, [365](#)
- compress_read_variable_length
 - compress.c, [303](#)
 - embroidery_internal.h, [365](#)
- config
 - embroidermodder.h, [237](#)
 - mainwindow.cpp, [292](#)
- config_tables
 - embroidermodder.h, [237](#)
 - mainwindow.cpp, [292](#)
- constants
 - LSYSTEM, [137](#)
- construct_command
 - embroidermodder.h, [229](#)
 - mainwindow.cpp, [278](#)
- contains
 - embroidermodder.h, [229](#)
 - view.cpp, [300](#)
- context_menu_action
 - StatusBar, [197](#)
- context_menu_event
 - StatusBar, [197](#)
- contextMenuEvent
 - CmdPromptHistory, [64](#)
 - CmdPromptInput, [69](#)
 - View, [210](#)
- control1
 - EmbBezier_, [79](#)
- control2
 - EmbBezier_, [79](#)
- convert
 - embroidery.h, [321](#)
 - pattern.c, [472](#)
- convert_args_to_type
 - embroidermodder.h, [229](#)
 - mainwindow.cpp, [278](#)
- copy
 - View, [210](#)
- copy_action
 - mainwindow.cpp, [278](#)
- copy_selected_action
 - mainwindow.cpp, [278](#)
- copy_trim
 - embroidery_internal.h, [365](#)
 - main.c, [466](#)
- copyClip
 - CmdPromptInput, [69](#)
- copyPressed
 - CmdPrompt, [53](#)
 - CmdPromptInput, [69](#)
- copySelected
 - View, [210](#)
- cornerButtonClicked

- View, 210
- count
 - EmbArray_, 78
- create_checkbox
 - Settings_Dialog, 194
- create_float_spinbox
 - Settings_Dialog, 194
- create_icon
 - MainWindow, 143
- create_menu
 - embroidermodder.h, 229
 - mainwindow-menus.cpp, 263
- create_test_file_1
 - embroidery_internal.h, 365
- create_test_file_2
 - embroidery_internal.h, 365
- create_test_file_3
 - embroidery_internal.h, 365
- create_toolbar
 - MainWindow, 143
- createAllActions
 - MainWindow, 143
- createAllMenus
 - MainWindow, 143
- createAllToolbars
 - MainWindow, 144
- createComboBoxSelected
 - PropertyEditor, 171
- createGrid
 - View, 210
- createGridIso
 - View, 210
- createGridPolar
 - View, 210
- createGridRect
 - View, 210
- createGroupBox
 - PropertyEditor, 172
- createHandle
 - CmdPromptSplitter, 74
- createHistogram
 - EmbDetailsDialog, 83
- createLineEdit
 - PropertyEditor, 172
- createMainWidget
 - EmbDetailsDialog, 83
- createObjectList
 - View, 210
- createOrigin
 - View, 211
- createRulerTextPath
 - View, 211
- createTabDisplay
 - Settings_Dialog, 194
- createTabFilePaths
 - Settings_Dialog, 194
- createTabGeneral
 - Settings_Dialog, 195
- createTabGridRuler
 - Settings_Dialog, 195
- createTabLineWeight
 - Settings_Dialog, 195
- createTabOpenSave
 - Settings_Dialog, 195
- createTabOrthoPolar
 - Settings_Dialog, 195
- createTabPrinting
 - Settings_Dialog, 195
- createTabPrompt
 - Settings_Dialog, 195
- createTabQuickSnap
 - Settings_Dialog, 195
- createTabQuickTrack
 - Settings_Dialog, 195
- createTabSelection
 - Settings_Dialog, 195
- createTabSnap
 - Settings_Dialog, 195
- createToolButton
 - PropertyEditor, 172
- createToolButtonPickAdd
 - PropertyEditor, 172
- createToolButtonQSelect
 - PropertyEditor, 172
- creationTime
 - _bcf_directory_entry, 41
- creatorName
 - ThredExtension_, 200
- crosshairColor
 - View, 216
- crosshairSize
 - View, 216
- csd, 9, 407
- csd_decryptArray
 - format_csd.c, 408
- CsdSubMaskSize
 - format_csd.c, 407
- CsdXorMaskSize
 - format_csd.c, 407
- csv, 409
- CSV_EXPECT
 - embroidery_internal.h, 361
- CSV_EXPECT_COMMA
 - embroidery_internal.h, 361
- CSV_EXPECT_NULL
 - embroidery_internal.h, 361
- CSV_EXPECT_QUOTE1
 - embroidery_internal.h, 361
- CSV_EXPECT_QUOTE2
 - embroidery_internal.h, 361
- CSV_MODE
 - embroidery_internal.h, 361
- CSV_MODE_COMMENT
 - embroidery_internal.h, 361
- CSV_MODE_NULL
 - embroidery_internal.h, 361

- CSV_MODE_STITCH
 - embroidery_internal.h, [362](#)
- CSV_MODE_THREAD
 - embroidery_internal.h, [361](#)
- CSV_MODE_VARIABLE
 - embroidery_internal.h, [361](#)
- csvStitchFlagToStr
 - format_csv.c, [409](#)
- csvStrToStitchFlag
 - format_csv.c, [409](#)
- CUBICTOCONTROL1
 - embroidery_internal.h, [353](#)
- CUBICTOCONTROL2
 - embroidery_internal.h, [353](#)
- CUBICTOEND
 - embroidery_internal.h, [353](#)
- curCmd
 - CmdPromptInput, [72](#)
- curColor
 - MdiWindow, [166](#)
- curFile
 - MdiWindow, [166](#)
- curLayer
 - MdiWindow, [166](#)
- curLineType
 - MdiWindow, [167](#)
- curLineWeight
 - MdiWindow, [167](#)
- current_element_id
 - format_svg.c, [437](#)
- currentAttribute
 - format_svg.c, [437](#)
- currentColorChanged
 - MdiWindow, [162](#)
- currentColorIndex
 - EmbPattern_, [94](#)
- currentDisplayBackgroundColorChanged
 - Settings_Dialog, [195](#)
- currentDisplayCrossHairColorChanged
 - Settings_Dialog, [195](#)
- currentDisplaySelectBoxLeftColorChanged
 - Settings_Dialog, [195](#)
- currentDisplaySelectBoxLeftFillChanged
 - Settings_Dialog, [195](#)
- currentDisplaySelectBoxRightColorChanged
 - Settings_Dialog, [195](#)
- currentDisplaySelectBoxRightFillChanged
 - Settings_Dialog, [196](#)
- currentGeneralMdiBackgroundColorChanged
 - Settings_Dialog, [196](#)
- currentGridColorChanged
 - Settings_Dialog, [196](#)
- currentLayerChanged
 - MdiWindow, [162](#)
- currentLinetypeChanged
 - MdiWindow, [162](#)
- currentLineweightChanged
 - MdiWindow, [162](#)
- currentPromptBackgroundColorChanged
 - Settings_Dialog, [196](#)
- currentPromptTextColorChanged
 - Settings_Dialog, [196](#)
- currentRulerColorChanged
 - Settings_Dialog, [196](#)
- currentValue
 - format_svg.c, [437](#)
- curText
 - CmdPromptInput, [73](#)
- curved
 - Geometry, [128](#)
- cut
 - View, [211](#)
- cut_action
 - mainwindow.cpp, [278](#)
- cut_selected_action
 - mainwindow.cpp, [279](#)
- cutCopyMousePoint
 - View, [216](#)
- cutCopyObjectList
 - MainWindow, [154](#)
- cutPressed
 - CmdPrompt, [53](#)
 - CmdPromptInput, [69](#)
- d
 - em2_dev_script, [39](#)
- dat, [9](#)
- data
 - EmblImage_, [89](#)
- day
 - EmbTime_, [101](#)
- day_vision_action
 - mainwindow.cpp, [279](#)
- debug_action
 - mainwindow.cpp, [279](#)
- debug_message
 - embroidermodder.h, [229](#)
 - interface.cpp, [257](#)
- decode_exy_flags
 - format_exy.c, [416](#)
- decode_record_flags
 - format_dst.c, [412](#)
- decode_t01_record
 - embroidery_internal.h, [365](#)
 - encoding.c, [392](#)
- decode_tajima_ternary
 - embroidery_internal.h, [366](#)
 - encoding.c, [392](#)
- decode_tap_record_flags
 - format_tap.c, [439](#)
- DecodeCsdByte
 - format_csd.c, [408](#)
- decodeNewStitch
 - embroidery_internal.h, [366](#)
 - encoding.c, [392](#)
- default_value
 - Huffman, [132](#)

- defaultPrefix
 - CmdPromptInput, 73
- degrees
 - embroidery.h, 321
 - functions.c, 453
- degrees__
 - embroidermodder.h, 230
 - interface.cpp, 257
- delete_selected_action
 - mainwindow.cpp, 279
- deleteObject
 - View, 211
- deletePressed
 - CmdPrompt, 54
 - CmdPromptInput, 69
 - MainWindow, 144
 - MdiWindow, 163
 - View, 211
- deleteSelected
 - View, 211
- delta
 - UndoableCommand, 204
- dem, 9, 410
- description
 - EmbFormatList_, 85
 - EmbThread_, 100
- design_details_action
 - mainwindow.cpp, 279
- designDetails
 - MdiWindow, 163
- dialog
 - embroidermodder.h, 237
 - mainwindow.cpp, 292
- Dictionary
 - embroidermodder.h, 226
- difat
 - _bcf_file, 42
- difatEntriesInHeader
 - main.c, 471
- dimensions
 - EmblImage_, 89
- dirBrush
 - SelectBox, 188
- directory
 - _bcf_file, 42
- directoryEntryName
 - _bcf_directory_entry, 41
- directoryEntryNameLength
 - _bcf_directory_entry, 41
- dirEntries
 - _bcf_directory, 40
- dirPen
 - SelectBox, 188
- disable_action
 - mainwindow.cpp, 279
- display_props
 - settings-dialog.cpp, 298
- distance_huffman
 - Compress, 76
- do_nothing_action
 - mainwindow.cpp, 279
- docIndex
 - MainWindow, 154
- dockPropEdit
 - embroidermodder.h, 237
 - mainwindow.cpp, 292
- dockUndoEdit
 - embroidermodder.h, 237
 - mainwindow.cpp, 293
- done
 - UndoableCommand, 204
- Dot
 - Geometry, 107
- doubleSpinBoxes
 - embroidermodder.h, 237
 - mainwindow.cpp, 293
- downPressed
 - CmdPrompt, 54
 - CmdPromptInput, 69
- dragon_curve
 - fill.c, 395
- drawBackground
 - View, 211
- drawForeground
 - View, 211
- drawRubberLine
 - Geometry, 110
- dsb, 9, 410
- dst, 9, 411
- dstJumpsPerTrim
 - EmbPattern_, 94
- dsz, 9, 371, 413
- dx, 9, 414
- dx_color
 - embroidery.h, 312
- DXF_VERSION_2000
 - embroidery_internal.h, 353
- DXF_VERSION_2002
 - embroidery_internal.h, 353
- DXF_VERSION_2004
 - embroidery_internal.h, 353
- DXF_VERSION_2006
 - embroidery_internal.h, 353
- DXF_VERSION_2007
 - embroidery_internal.h, 353
- DXF_VERSION_2009
 - embroidery_internal.h, 353
- DXF_VERSION_2010
 - embroidery_internal.h, 354
- DXF_VERSION_2013
 - embroidery_internal.h, 354
- DXF_VERSION_R10
 - embroidery_internal.h, 354
- DXF_VERSION_R11
 - embroidery_internal.h, 354
- DXF_VERSION_R12

- embroidery_internal.h, [354](#)
- DXF_VERSION_R13
 - embroidery_internal.h, [354](#)
- DXF_VERSION_R14
 - embroidery_internal.h, [354](#)
- DXF_VERSION_R15
 - embroidery_internal.h, [354](#)
- DXF_VERSION_R18
 - embroidery_internal.h, [354](#)
- DXF_VERSION_R21
 - embroidery_internal.h, [354](#)
- DXF_VERSION_R24
 - embroidery_internal.h, [354](#)
- DXF_VERSION_R27
 - embroidery_internal.h, [354](#)
- edr, [9](#), [371](#), [414](#)
- ELEMENT_A
 - embroidery_internal.h, [354](#)
- ELEMENT_ANIMATE
 - embroidery_internal.h, [354](#)
- ELEMENT_ANIMATECOLOR
 - embroidery_internal.h, [354](#)
- ELEMENT_ANIMATEMOTION
 - embroidery_internal.h, [354](#)
- ELEMENT_ANIMATETRANSFORM
 - embroidery_internal.h, [354](#)
- ELEMENT_ANIMATION
 - embroidery_internal.h, [354](#)
- ELEMENT_AUDIO
 - embroidery_internal.h, [355](#)
- ELEMENT_CIRCLE
 - embroidery_internal.h, [355](#)
- ELEMENT_DEFS
 - embroidery_internal.h, [355](#)
- ELEMENT_DESC
 - embroidery_internal.h, [355](#)
- ELEMENT_DISCARD
 - embroidery_internal.h, [355](#)
- ELEMENT_ELLIPSE
 - embroidery_internal.h, [355](#)
- ELEMENT_FONT
 - embroidery_internal.h, [355](#)
- ELEMENT_FONT_FACE
 - embroidery_internal.h, [355](#)
- ELEMENT_FONT_FACE_SRC
 - embroidery_internal.h, [355](#)
- ELEMENT_FONT_FACE_URI
 - embroidery_internal.h, [355](#)
- ELEMENT_FOREIGN_OBJECT
 - embroidery_internal.h, [355](#)
- ELEMENT_G
 - embroidery_internal.h, [355](#)
- ELEMENT_GLYPH
 - embroidery_internal.h, [355](#)
- ELEMENT_HANDLER
 - embroidery_internal.h, [355](#)
- ELEMENT_HKERN
 - embroidery_internal.h, [355](#)
- ELEMENT_IMAGE
 - embroidery_internal.h, [355](#)
- ELEMENT_LINE
 - embroidery_internal.h, [355](#)
- ELEMENT_LINEAR_GRADIENT
 - embroidery_internal.h, [355](#)
- ELEMENT_LISTENER
 - embroidery_internal.h, [356](#)
- ELEMENT_METADATA
 - embroidery_internal.h, [356](#)
- ELEMENT_MISSING_GLYPH
 - embroidery_internal.h, [356](#)
- ELEMENT_MPATH
 - embroidery_internal.h, [356](#)
- ELEMENT_PATH
 - embroidery_internal.h, [356](#)
- ELEMENT_POLYGON
 - embroidery_internal.h, [356](#)
- ELEMENT_POLYLINE
 - embroidery_internal.h, [356](#)
- ELEMENT_PREFETCH
 - embroidery_internal.h, [356](#)
- ELEMENT_RADIAL_GRADIENT
 - embroidery_internal.h, [356](#)
- ELEMENT_RECT
 - embroidery_internal.h, [356](#)
- ELEMENT_SCRIPT
 - embroidery_internal.h, [356](#)
- ELEMENT_SET
 - embroidery_internal.h, [356](#)
- ELEMENT_SOLID_COLOR
 - embroidery_internal.h, [356](#)
- ELEMENT_STOP
 - embroidery_internal.h, [356](#)
- ELEMENT_SVG
 - embroidery_internal.h, [356](#)
- ELEMENT_SWITCH
 - embroidery_internal.h, [356](#)
- ELEMENT_TBREAK
 - embroidery_internal.h, [356](#)
- ELEMENT_TEXT
 - embroidery_internal.h, [356](#)
- ELEMENT_TEXT_AREA
 - embroidery_internal.h, [357](#)
- ELEMENT_TITLE
 - embroidery_internal.h, [357](#)
- ELEMENT_TSPAN
 - embroidery_internal.h, [357](#)
- ELEMENT_USE
 - embroidery_internal.h, [357](#)
- ELEMENT_VIDEO
 - embroidery_internal.h, [357](#)
- ELEMENT_XML
 - embroidery_internal.h, [357](#)
- ellipse
 - EmbGeometry_, [87](#)
- ellipse.c
 - ellipse_objectQuadrant0, [452](#)

- ellipse_objectQuadrant180, [452](#)
- ellipse_objectQuadrant270, [452](#)
- ellipse_objectQuadrant90, [452](#)
- embEllipse_area, [452](#)
- embEllipse_diameterX, [452](#)
- embEllipse_diameterY, [452](#)
- embEllipse_height, [452](#)
- embEllipse_init, [452](#)
- embEllipse_perimeter, [452](#)
- embEllipse_setDiameterMajor, [452](#)
- embEllipse_setDiameterMinor, [453](#)
- embEllipse_setRadiusMajor, [453](#)
- embEllipse_setRadiusMinor, [453](#)
- embEllipse_setSize, [453](#)
- embEllipse_updatePath, [453](#)
- embEllipse_width, [453](#)
- ellipse_objectQuadrant0
 - ellipse.c, [452](#)
- ellipse_objectQuadrant180
 - ellipse.c, [452](#)
- ellipse_objectQuadrant270
 - ellipse.c, [452](#)
- ellipse_objectQuadrant90
 - ellipse.c, [452](#)
- ELLIPSETOEND
 - embroidery_internal.h, [357](#)
- ELLIPSETORAD
 - embroidery_internal.h, [357](#)
- Elna, [415](#)
- Eltac, [416](#)
- em2_dev_script, [39](#)
 - d, [39](#)
 - header, [39](#)
 - s, [39](#)
- EMB_ARC
 - embroidery.h, [312](#)
- EMB_ARRAY
 - embroidery.h, [312](#)
- EMB_BIG_ENDIAN
 - embroidery_internal.h, [357](#)
- EMB_CIRCLE
 - embroidery.h, [312](#)
- emb_constant_pi
 - embroidermodder.h, [237](#)
- EMB_DIM_DIAMETER
 - embroidery.h, [312](#)
- EMB_DIM_LEADER
 - embroidery.h, [312](#)
- EMB_ELLIPSE
 - embroidery.h, [312](#)
- emb_error
 - embroidery.h, [336](#)
 - main.c, [471](#)
- EMB_FLAG
 - embroidery.h, [312](#)
- EMB_FORMAT_100
 - embroidery.h, [312](#)
- EMB_FORMAT_100
 - embroidery.h, [312](#)
- embroidery.h, [312](#)
- EMB_FORMAT_ART
 - embroidery.h, [312](#)
- EMB_FORMAT_BMC
 - embroidery.h, [312](#)
- EMB_FORMAT_BRO
 - embroidery.h, [312](#)
- EMB_FORMAT_CND
 - embroidery.h, [312](#)
- EMB_FORMAT_COL
 - embroidery.h, [312](#)
- EMB_FORMAT_CSD
 - embroidery.h, [313](#)
- EMB_FORMAT_CSV
 - embroidery.h, [313](#)
- EMB_FORMAT_DAT
 - embroidery.h, [313](#)
- EMB_FORMAT_DEM
 - embroidery.h, [313](#)
- EMB_FORMAT_DSB
 - embroidery.h, [313](#)
- EMB_FORMAT_DST
 - embroidery.h, [313](#)
- EMB_FORMAT_DSZ
 - embroidery.h, [313](#)
- EMB_FORMAT_DXF
 - embroidery.h, [313](#)
- EMB_FORMAT_EDR
 - embroidery.h, [313](#)
- EMB_FORMAT_EMD
 - embroidery.h, [313](#)
- EMB_FORMAT_EXP
 - embroidery.h, [313](#)
- EMB_FORMAT_EXY
 - embroidery.h, [313](#)
- EMB_FORMAT_EYS
 - embroidery.h, [313](#)
- EMB_FORMAT_FXY
 - embroidery.h, [313](#)
- EMB_FORMAT_GC
 - embroidery.h, [313](#)
- EMB_FORMAT_GNC
 - embroidery.h, [313](#)
- EMB_FORMAT_GT
 - embroidery.h, [313](#)
- EMB_FORMAT_HUS
 - embroidery.h, [313](#)
- EMB_FORMAT_INB
 - embroidery.h, [314](#)
- EMB_FORMAT_INF
 - embroidery.h, [314](#)
- EMB_FORMAT_JEF
 - embroidery.h, [314](#)
- EMB_FORMAT_KSM
 - embroidery.h, [314](#)
- EMB_FORMAT_MAX
 - embroidery.h, [314](#)
- EMB_FORMAT_MIT

embroidery.h, 314
EMB_FORMAT_NEW
embroidery.h, 314
EMB_FORMAT_OFM
embroidery.h, 314
EMB_FORMAT_PCD
embroidery.h, 314
EMB_FORMAT_PCM
embroidery.h, 314
EMB_FORMAT_PCQ
embroidery.h, 314
EMB_FORMAT_PCS
embroidery.h, 314
EMB_FORMAT_PEC
embroidery.h, 314
EMB_FORMAT_PEL
embroidery.h, 314
EMB_FORMAT_PEM
embroidery.h, 314
EMB_FORMAT_PES
embroidery.h, 314
EMB_FORMAT_PHB
embroidery.h, 314
EMB_FORMAT_PHC
embroidery.h, 314
EMB_FORMAT_PLT
embroidery.h, 315
EMB_FORMAT_RGB
embroidery.h, 315
EMB_FORMAT_SEW
embroidery.h, 315
EMB_FORMAT_SHV
embroidery.h, 315
EMB_FORMAT_SST
embroidery.h, 315
EMB_FORMAT_STX
embroidery.h, 315
EMB_FORMAT_SVG
embroidery.h, 315
EMB_FORMAT_T01
embroidery.h, 315
EMB_FORMAT_T09
embroidery.h, 315
EMB_FORMAT_TAP
embroidery.h, 315
EMB_FORMAT_THR
embroidery.h, 315
EMB_FORMAT_TXT
embroidery.h, 315
EMB_FORMAT_U00
embroidery.h, 315
EMB_FORMAT_U01
embroidery.h, 315
EMB_FORMAT_VIP
embroidery.h, 315
EMB_FORMAT_VP3
embroidery.h, 315
EMB_FORMAT_XXX
embroidery.h, 315
EMB_FORMAT_ZSK
embroidery.h, 315
emb_identify_format
embroidery.h, 321
formats.c, 401
EMB_IMAGE
embroidery.h, 316
EMB_INT16_BIG
embroidery_internal.h, 357
EMB_INT16_LITTLE
embroidery_internal.h, 357
EMB_INT32_BIG
embroidery_internal.h, 357
EMB_INT32_LITTLE
embroidery_internal.h, 357
EMB_LINE
embroidery.h, 316
EMB_LITTLE_ENDIAN
embroidery_internal.h, 357
EMB_MAX
embroidery_internal.h, 357
EMB_MAX_LAYERS
embroidery.h, 316
EMB_MIN
embroidery_internal.h, 357
emb_optOut
embroidery_internal.h, 366
main.c, 466
EMB_PATH
embroidery.h, 316
EMB_POINT
embroidery.h, 316
EMB_POLYGON
embroidery.h, 316
EMB_POLYLINE
embroidery.h, 316
EMB_PUBLIC
embroidery.h, 316
emb_readline
embroidery_internal.h, 366
main.c, 467
EMB_RECT
embroidery.h, 316
emb_round
embroidery.h, 321
functions.c, 453
EMB_SPLINE
embroidery.h, 316
EMB_STITCH
embroidery.h, 316
EMB_TEXT_MULTI
embroidery.h, 316
EMB_TEXT_SINGLE
embroidery.h, 316
EMB_THREAD
embroidery.h, 316
EMB_VECTOR

- embroidery.h, 316
- emb_verbose
 - embroidery.h, 336
 - main.c, 471
- EmbAlignedDim
 - embroidery.h, 319
- EmbAlignedDim_, 76
 - position, 76
- EmbAngularDim
 - embroidery.h, 319
- EmbAngularDim_, 76
 - position, 77
- EmbArc
 - embroidery.h, 319
- EmbArc_, 77
 - end, 77
 - mid, 77
 - start, 77
- embArc_arcLength
 - arc.c, 448
- embArc_area
 - arc.c, 448
- embArc_chord
 - arc.c, 448
- embArc_clockwise
 - arc.c, 448
 - embroidery.h, 321
- embArc_endAngle
 - arc.c, 448
- embArc_gripEdit
 - arc.c, 448
- embArc_includedAngle
 - arc.c, 448
- embArc_init
 - arc.c, 448
 - embroidery.h, 321
- embArc_mouseSnapPoint
 - arc.c, 449
- embArc_paint
 - arc.c, 449
- embArc_print
 - main.c, 467
- embArc_setCenter
 - arc.c, 449
- embArc_setEndAngle
 - arc.c, 449
- embArc_setRadius
 - arc.c, 449
- embArc_setStartAngle
 - arc.c, 449
- embArc_startAngle
 - arc.c, 449
- embArc_updatePath
 - arc.c, 449
- embArc_updateRubber
 - arc.c, 449
- EmbArcLengthDim
 - embroidery.h, 319
- EmbArcLengthDim_, 78
 - position, 78
- EmbArray
 - embroidery.h, 319
- EmbArray_, 78
 - count, 78
 - geometry, 78
 - length, 78
 - stitch, 79
 - thread, 79
 - type, 79
- embArray_addArc
 - array.c, 301
 - embroidery.h, 322
- embArray_addCircle
 - array.c, 301
 - embroidery.h, 322
- embArray_addEllipse
 - array.c, 301
 - embroidery.h, 322
- embArray_addFlag
 - array.c, 301
 - embroidery.h, 322
- embArray_addLine
 - array.c, 301
 - embroidery.h, 322
- embArray_addPath
 - array.c, 301
 - embroidery.h, 322
- embArray_addPoint
 - array.c, 301
 - embroidery.h, 322
- embArray_addPolygon
 - array.c, 301
 - embroidery.h, 322
- embArray_addPolyline
 - array.c, 301
 - embroidery.h, 322
- embArray_addRect
 - array.c, 301
 - embroidery.h, 322
- embArray_addStitch
 - array.c, 301
 - embroidery.h, 322
- embArray_addThread
 - embroidery.h, 323
- embArray_addVector
 - array.c, 302
 - embroidery.h, 323
- embArray_copy
 - array.c, 302
 - embroidery.h, 323
- embArray_create
 - array.c, 302
 - embroidery.h, 323
- embArray_free
 - array.c, 302
 - embroidery.h, 323

- embArray_resize
 - array.c, 302
 - embroidery.h, 323
- embBase_setColorRGB
 - arc.c, 449
- EmbBezier
 - embroidery.h, 319
- EmbBezier_, 79
 - control1, 79
 - control2, 79
 - end, 79
 - start, 79
- EmbBlock
 - embroidery.h, 319
- EmbBlock_, 80
 - position, 80
- EmbCircle
 - embroidery.h, 319
- EmbCircle_, 80
 - center, 80
 - radius, 80
- embCircle_area
 - circle.c, 451
- embCircle_circumference
 - circle.c, 451
- embCircle_init
 - circle.c, 451
 - embroidery.h, 323
- embCircle_prompt
 - arc.c, 449
- embCircle_setArea
 - arc.c, 449
- embCircle_setCircumference
 - arc.c, 450
- EmbColor
 - embroidery.h, 319
- EmbColor_, 81
 - b, 81
 - g, 81
 - r, 81
- embColor_create
 - embroidery.h, 323
- embColor_distance
 - embroidery.h, 323
 - main.c, 467
- embColor_fromHexStr
 - embroidery.h, 323
 - encoding.c, 392
- embColor_make
 - embroidery.h, 324
- embColor_read
 - embroidery_internal.h, 366
 - main.c, 467
- embColor_write
 - embroidery_internal.h, 366
 - main.c, 467
- embConstantPi
 - embroidery.h, 336
- main.c, 471
- EmbDetailsDialog, 81
 - ~EmbDetailsDialog, 82
 - boundingRect, 83
 - buttonBox, 83
 - colorChanges, 83
 - colorTotal, 83
 - createHistogram, 83
 - createMainWidget, 83
 - EmbDetailsDialog, 82
 - getInfo, 83
 - mainWidget, 83
 - stitchesJump, 83
 - stitchesReal, 84
 - stitchesTotal, 84
 - stitchesTrim, 84
- EmbDiameterDim
 - embroidery.h, 319
- EmbDiameterDim_, 84
 - position, 84
- EmbEllipse
 - embroidery.h, 319
- EmbEllipse_, 84
 - center, 85
 - radius, 85
 - rotation, 85
- embEllipse_area
 - ellipse.c, 452
 - embroidery.h, 324
- embEllipse_click
 - arc.c, 450
- embEllipse_diameterX
 - ellipse.c, 452
 - embroidery.h, 324
- embEllipse_diameterY
 - ellipse.c, 452
 - embroidery.h, 324
- embEllipse_height
 - ellipse.c, 452
 - embroidery.h, 324
- embEllipse_init
 - ellipse.c, 452
 - embroidery.h, 324
- embEllipse_main
 - arc.c, 450
- embEllipse_make
 - embroidery.h, 324
- embEllipse_perimeter
 - ellipse.c, 452
 - embroidery.h, 324
- embEllipse_setDiameterMajor
 - ellipse.c, 452
- embEllipse_setDiameterMinor
 - ellipse.c, 453
- embEllipse_setRadiusMajor
 - ellipse.c, 453
- embEllipse_setRadiusMinor
 - ellipse.c, 453

- embEllipse_setSize
 - ellipse.c, [453](#)
- embEllipse_updatePath
 - ellipse.c, [453](#)
- embEllipse_width
 - ellipse.c, [453](#)
 - embroidery.h, [324](#)
- EmbFlag
 - embroidery.h, [319](#)
- embFormat_getExtension
 - formats.c, [401](#)
- EMBFORMAT_MAXDESC
 - embroidery.h, [316](#)
- EMBFORMAT_MAXEXT
 - embroidery.h, [316](#)
- EMBFORMAT_OBJECTONLY
 - embroidery.h, [316](#)
- EMBFORMAT_STCHANDOBJ
 - embroidery.h, [317](#)
- EMBFORMAT_STITCHONLY
 - embroidery.h, [317](#)
- EMBFORMAT_UNSUPPORTED
 - embroidery.h, [317](#)
- EmbFormatList
 - embroidery.h, [319](#)
- EmbFormatList_, [85](#)
 - check_for_color_file, [85](#)
 - color_only, [85](#)
 - description, [85](#)
 - extension, [86](#)
 - reader_state, [86](#)
 - type, [86](#)
 - write_external_color_file, [86](#)
 - writer_state, [86](#)
- EmbGeometry
 - embroidery.h, [319](#)
- EmbGeometry_, [86](#)
 - arc, [87](#)
 - circle, [87](#)
 - color, [87](#)
 - ellipse, [87](#)
 - flag, [87](#)
 - line, [87](#)
 - lineType, [87](#)
 - object, [87](#)
 - path, [87](#)
 - point, [87](#)
 - polygon, [87](#)
 - polyline, [88](#)
 - rect, [88](#)
 - spline, [88](#)
 - stitch, [88](#)
 - thread, [88](#)
 - type, [88](#)
 - vector, [88](#)
- embGeometry_boundingBox
 - embroidery.h, [324](#)
 - geometry.c, [446](#)
- embGeometry_free
 - embroidery.h, [324](#)
 - geometry.c, [446](#)
- embGeometry_init
 - embroidery.h, [325](#)
 - geometry.c, [446](#)
- embGeometry_move
 - embroidery.h, [325](#)
 - geometry.c, [446](#)
- embGeometry_vulcanize
 - embroidery.h, [325](#)
 - geometry.c, [446](#)
- EmblImage
 - embroidery.h, [320](#)
- EmblImage_, [88](#)
 - data, [89](#)
 - dimensions, [89](#)
 - height, [89](#)
 - name, [89](#)
 - path, [89](#)
 - position, [89](#)
 - width, [89](#)
- emblImage_create
 - embroidery.h, [325](#)
- emblImage_free
 - embroidery.h, [325](#)
- emblImage_read
 - embroidery.h, [325](#)
- emblImage_write
 - embroidery.h, [325](#)
- EmblInfiniteLine
 - embroidery.h, [320](#)
- EmblInfiniteLine_, [89](#)
 - position, [90](#)
- emblInt_read
 - embroidery_internal.h, [367](#)
 - encoding.c, [393](#)
- emblInt_write
 - embroidery_internal.h, [367](#)
 - encoding.c, [393](#)
- Embird, [371](#), [409](#), [414](#)
- EmbLayer
 - embroidery.h, [320](#)
- EmbLayer_, [90](#)
 - geometry, [90](#)
 - name, [90](#)
- EmbLeaderDim
 - embroidery.h, [320](#)
- EmbLeaderDim_, [90](#)
 - position, [91](#)
- EmbLine
 - embroidery.h, [320](#)
- EmbLine_, [91](#)
 - color, [91](#)
 - end, [91](#)
 - lineType, [91](#)
 - start, [91](#)
- embLine_intersectionPoint

- embroidery.h, [325](#)
- line.c, [454](#)
- embLine_make
 - embroidery.h, [325](#)
- embLine_normalVector
 - embroidery.h, [326](#)
 - line.c, [454](#)
- embLine_toVector
 - line.c, [454](#)
- EmbLinearDim
 - embroidery.h, [320](#)
- EmbLinearDim_, [92](#)
 - position, [92](#)
- EmbOrdinateDim
 - embroidery.h, [320](#)
- EmbOrdinateDim_, [92](#)
 - position, [92](#)
- EmbPath
 - embroidery.h, [320](#)
- EmbPath_, [92](#)
 - color, [93](#)
 - flagList, [93](#)
 - lineType, [93](#)
 - pointList, [93](#)
- EmbPattern
 - embroidery.h, [320](#)
- EmbPattern_, [93](#)
 - currentColorIndex, [94](#)
 - dstJumpsPerTrim, [94](#)
 - geometry, [94](#)
 - home, [94](#)
 - hoop_height, [94](#)
 - hoop_width, [94](#)
 - layer, [94](#)
 - stitch_list, [94](#)
 - thread_list, [94](#)
- embPattern_addCircleAbs
 - embroidery.h, [326](#)
 - pattern.c, [472](#)
- embPattern_addEllipseAbs
 - embroidery.h, [326](#)
 - pattern.c, [472](#)
- embPattern_addLineAbs
 - embroidery.h, [326](#)
 - pattern.c, [473](#)
- embPattern_addPathAbs
 - embroidery.h, [326](#)
 - pattern.c, [473](#)
- embPattern_addPointAbs
 - embroidery.h, [326](#)
 - pattern.c, [473](#)
- embPattern_addPolygonAbs
 - embroidery.h, [326](#)
 - pattern.c, [473](#)
- embPattern_addPolylineAbs
 - embroidery.h, [326](#)
- embPattern_addPolylineObjectAbs
 - pattern.c, [473](#)
- embPattern_addRectAbs
 - embroidery.h, [326](#)
 - pattern.c, [473](#)
- embPattern_addStitchAbs
 - embroidery.h, [326](#)
 - pattern.c, [473](#)
- embPattern_addStitchRel
 - embroidery.h, [327](#)
 - pattern.c, [473](#)
- embPattern_addThread
 - embroidery.h, [327](#)
 - pattern.c, [473](#)
- embPattern_calcBoundingBox
 - embroidery.h, [327](#)
 - pattern.c, [474](#)
- embPattern_center
 - embroidery.h, [327](#)
 - pattern.c, [474](#)
- embPattern_changeColor
 - embroidery.h, [327](#)
 - pattern.c, [474](#)
- embPattern_color_count
 - embroidery.h, [327](#)
 - pattern.c, [474](#)
- embPattern_combine
 - embroidery.h, [327](#)
 - fill.c, [395](#)
- embPattern_combineJumpStitches
 - embroidery.h, [327](#)
 - pattern.c, [474](#)
- embPattern_convertGeometry
 - embroidery.h, [327](#)
 - fill.c, [395](#)
- embPattern_copyPolylinesToStitch_list
 - pattern.c, [474](#)
- embPattern_copyPolylinesToStitchList
 - embroidery.h, [328](#)
- embPattern_copystitch_listToPolylines
 - pattern.c, [474](#)
- embPattern_copyStitchListToPolylines
 - embroidery.h, [328](#)
- embPattern_correctForMaxStitchLength
 - embroidery.h, [328](#)
 - pattern.c, [474](#)
- embPattern_create
 - embroidery.h, [328](#)
 - pattern.c, [474](#)
- embPattern_crossstitch
 - embroidery.h, [328](#)
 - fill.c, [395](#)
- embPattern_designDetails
 - embroidery.h, [328](#)
 - pattern.c, [474](#)
- embPattern_end
 - embroidery.h, [328](#)
 - pattern.c, [475](#)
- embPattern_fixColorCount
 - embroidery.h, [328](#)

- pattern.c, 475
- embPattern_flip
 - embroidery.h, 329
 - pattern.c, 475
- embPattern_flipHorizontal
 - embroidery.h, 329
 - pattern.c, 475
- embPattern_flipVertical
 - embroidery.h, 329
 - pattern.c, 475
- embPattern_free
 - embroidery.h, 329
 - pattern.c, 475
- embPattern_hideStitchesOverLength
 - embroidery.h, 329
 - pattern.c, 475
- embPattern_horizontal_fill
 - embroidery.h, 329
 - fill.c, 395
- embPattern_jumpStitches
 - embroidery.h, 329
 - pattern.c, 475
- embPattern_lengthHistogram
 - embroidery.h, 329
 - pattern.c, 475
- embPattern_loadExternalColorFile
 - embroidery.h, 329
 - pattern.c, 475
- embPattern_maximumStitchLength
 - embroidery.h, 329
 - pattern.c, 475
- embPattern_minimumStitchLength
 - embroidery.h, 329
 - pattern.c, 475
- embPattern_movePolylinesToStitch_list
 - pattern.c, 476
- embPattern_movePolylinesToStitchList
 - embroidery.h, 330
- embPattern_movestitch_listToPolylines
 - pattern.c, 476
- embPattern_moveStitchListToPolylines
 - embroidery.h, 330
- embPattern_read
 - embroidery.h, 330
 - formats.c, 401
- embPattern_readAuto
 - embroidery.h, 330
 - formats.c, 401
- embPattern_realStitches
 - embroidery.h, 330
 - pattern.c, 476
- embPattern_render
 - embroidery.h, 330
- embPattern_scale
 - embroidery.h, 330
 - pattern.c, 476
- embPattern_simulate
 - embroidery.h, 330
- embPattern_stitchArc
 - fill.c, 395
- embPattern_stitchCircle
 - fill.c, 395
- embPattern_stitchEllipse
 - fill.c, 396
- embPattern_stitchPath
 - fill.c, 396
- embPattern_stitchPolygon
 - fill.c, 396
- embPattern_stitchPolyline
 - fill.c, 396
- embPattern_stitchRect
 - fill.c, 397
- embPattern_stitchText
 - fill.c, 397
- embPattern_totalStitchLength
 - embroidery.h, 330
 - pattern.c, 476
- embPattern_trimStitches
 - embroidery.h, 331
 - pattern.c, 476
- embPattern_write
 - embroidery.h, 331
 - formats.c, 401
- embPattern_writeAuto
 - embroidery.h, 331
 - formats.c, 401
- EmbPoint
 - embroidery.h, 320
- EmbPoint_, 94
 - color, 95
 - lineType, 95
 - position, 95
- EmbPolygon
 - embroidery.h, 320
- embPolygon_reduceByDistance
 - fill.c, 397
- embPolygon_reduceByNth
 - fill.c, 397
- EmbPolyline
 - embroidery.h, 320
- EmbRadiusDim
 - embroidery.h, 320
- EmbRadiusDim_, 95
 - position, 95
- EmbRay
 - embroidery.h, 320
- EmbRay_, 96
 - position, 96
- EmbReal
 - embroidery.h, 320
- EmbRect
 - embroidery.h, 320
- EmbRect_, 96
 - bottom, 96
 - left, 96
 - radius, 96

- right, [97](#)
- rotation, [97](#)
- top, [97](#)
- embRect_area
 - embroidery.h, [331](#)
 - rect.c, [455](#)
- embRect_bottomLeft
 - arc.c, [450](#)
- embRect_bottomRight
 - arc.c, [450](#)
- embRect_init
 - embroidery.h, [331](#)
 - rect.c, [455](#)
- embroidermodder.cpp
 - _appVer_, [222](#)
 - exitApp, [222](#)
 - main, [221](#)
 - usage, [222](#)
 - version, [222](#)
- embroidermodder.h
 - _mainWin, [236](#)
 - actionHash, [237](#)
 - activeScene, [228](#)
 - activeView, [228](#)
 - actuator, [228](#)
 - add_polyline, [229](#)
 - BOOL_TYPE, [226](#)
 - checkBoxes, [237](#)
 - comboBoxes, [237](#)
 - Command, [226](#)
 - config, [237](#)
 - config_tables, [237](#)
 - construct_command, [229](#)
 - contains, [229](#)
 - convert_args_to_type, [229](#)
 - create_menu, [229](#)
 - debug_message, [229](#)
 - degrees_, [230](#)
 - dialog, [237](#)
 - Dictionary, [226](#)
 - dockPropEdit, [237](#)
 - dockUndoEdit, [237](#)
 - doubleSpinBoxes, [237](#)
 - emb_constant_pi, [237](#)
 - fileExtension, [230](#)
 - FUNCTION_TYPE, [226](#)
 - get_bool, [230](#)
 - get_int, [230](#)
 - get_qstr, [230](#)
 - get_real, [230](#)
 - get_str, [230](#)
 - get_str_list, [231](#)
 - get_uint, [231](#)
 - groupBoxes, [237](#)
 - INT_TYPE, [226](#)
 - labels, [237](#)
 - lineEdits, [237](#)
 - make_checkbox, [231](#)
 - make_spinbox, [231](#)
 - make_ui_element, [231](#)
 - mdiArea, [237](#)
 - menuHash, [237](#)
 - Node, [226](#)
 - node_bool, [231](#)
 - node_int, [231](#)
 - node_qstr, [232](#)
 - node_real, [232](#)
 - node_str, [232](#)
 - node_str_list, [232](#)
 - node_uint, [232](#)
 - NodeList, [227](#)
 - OBJ_COLOR, [227](#)
 - OBJ_KEYS, [227](#)
 - OBJ_LAYER, [227](#)
 - OBJ_LTYPE, [227](#)
 - OBJ_LWT, [227](#)
 - OBJ_NAME, [227](#)
 - OBJ_RUBBER, [227](#)
 - OBJ_TYPE, [227](#)
 - OBJ_TYPE_ARC, [227](#)
 - OBJ_TYPE_BASE, [227](#)
 - OBJ_TYPE_BLOCK, [227](#)
 - OBJ_TYPE_CIRCLE, [227](#)
 - OBJ_TYPE_DIMALIGNED, [227](#)
 - OBJ_TYPE_DIMANGULAR, [227](#)
 - OBJ_TYPE_DIMARCLENGTH, [227](#)
 - OBJ_TYPE_DIMDIAMETER, [227](#)
 - OBJ_TYPE_DIMLEADER, [227](#)
 - OBJ_TYPE_DIMLINEAR, [227](#)
 - OBJ_TYPE_DIMORDINATE, [227](#)
 - OBJ_TYPE_DIMRADIUS, [227](#)
 - OBJ_TYPE_ELLIPSE, [228](#)
 - OBJ_TYPE_ELLIPSEARC, [228](#)
 - OBJ_TYPE_GRID, [228](#)
 - OBJ_TYPE_HATCH, [228](#)
 - OBJ_TYPE_IMAGE, [228](#)
 - OBJ_TYPE_INFINITELINE, [228](#)
 - OBJ_TYPE_LINE, [228](#)
 - OBJ_TYPE_NULL, [227](#)
 - OBJ_TYPE_PATH, [228](#)
 - OBJ_TYPE_POINT, [228](#)
 - OBJ_TYPE_POLYGON, [228](#)
 - OBJ_TYPE_POLYLINE, [228](#)
 - OBJ_TYPE_RAY, [228](#)
 - OBJ_TYPE_RECTANGLE, [228](#)
 - OBJ_TYPE_RUBBER, [228](#)
 - OBJ_TYPE_SLOT, [228](#)
 - OBJ_TYPE_SPLINE, [228](#)
 - OBJ_TYPE_TEXTMULTI, [228](#)
 - OBJ_TYPE_TEXTSINGLE, [228](#)
 - OBJ_TYPE_UNKNOWN, [228](#)
 - OBJ_TYPE_VALUES, [227](#)
 - operator*, [233](#)
 - operator+, [233](#)
 - operator-, [233](#)
 - prompt, [237](#)

- radians__, 233
- read_configuration, 233
- read_settings, 233
- read_string_setting, 233
- REAL_TYPE, 226
- rotate_vector, 234
- run_script, 234
- run_script_file, 234
- scripts, 237
- set_enabled, 234
- set_visibility, 235
- settings, 237
- spinBoxes, 238
- statusbar, 238
- String, 227
- STRING_LIST_TYPE, 226
- STRING_TYPE, 226
- StringList, 227
- subMenuHash, 238
- to_EmbVector, 235
- to_qlist, 235
- to_QPointF, 235
- to_string_vector, 235
- to_vector, 235
- tokenize, 236
- toolbarHash, 238
- toolButtons, 238
- translate_str, 236
- UNKNOWN_TYPE, 226
- validFileFormat, 236
- VECTOR_TYPE, 226
- write_settings, 236
- embroidermodder2/cmdprompt.cpp, 221
- embroidermodder2/em2_dev_script.py, 221
- embroidermodder2/embdetails-dialog.cpp, 221
- embroidermodder2/embroidermodder.cpp, 221
- embroidermodder2/embroidermodder.h, 222, 238
- embroidermodder2/imagewidget.cpp, 256
- embroidermodder2/interface.cpp, 256
- embroidermodder2/layer-manager.cpp, 263
- embroidermodder2/mainwindow-menus.cpp, 263
- embroidermodder2/mainwindow-toolbars.cpp, 264
- embroidermodder2/mainwindow.cpp, 264
- embroidermodder2/mdiarea.cpp, 294
- embroidermodder2/mdiwindow.cpp, 294
- embroidermodder2/objects.cpp, 294
- embroidermodder2/preview-dialog.cpp, 295
- embroidermodder2/property-editor.cpp, 295
- embroidermodder2/README.md, 297
- embroidermodder2/selectbox.cpp, 297
- embroidermodder2/settings-dialog.cpp, 297
- embroidermodder2/statusbar.cpp, 299
- embroidermodder2/undo-commands.cpp, 299
- embroidermodder2/undo-editor.cpp, 299
- embroidermodder2/view.cpp, 300
- embroidery.h
 - _dxfColorTable, 336
 - Arc_Polyester, 311
 - Arc_Rayon, 312
 - black_thread, 336
 - CHUNK_SIZE, 312
 - CoatsAndClark_Rayon, 312
 - convert, 321
 - degrees, 321
 - dxf_color, 312
 - EMB_ARC, 312
 - EMB_ARRAY, 312
 - EMB_CIRCLE, 312
 - EMB_DIM_DIAMETER, 312
 - EMB_DIM_LEADER, 312
 - EMB_ELLIPSE, 312
 - emb_error, 336
 - EMB_FLAG, 312
 - EMB_FORMAT_100, 312
 - EMB_FORMAT_10O, 312
 - EMB_FORMAT_ART, 312
 - EMB_FORMAT_BMC, 312
 - EMB_FORMAT_BRO, 312
 - EMB_FORMAT_CND, 312
 - EMB_FORMAT_COL, 312
 - EMB_FORMAT_CSD, 313
 - EMB_FORMAT_CSV, 313
 - EMB_FORMAT_DAT, 313
 - EMB_FORMAT_DEM, 313
 - EMB_FORMAT_DSB, 313
 - EMB_FORMAT_DST, 313
 - EMB_FORMAT_DSZ, 313
 - EMB_FORMAT_DXF, 313
 - EMB_FORMAT_EDR, 313
 - EMB_FORMAT_EMD, 313
 - EMB_FORMAT_EXP, 313
 - EMB_FORMAT_EXY, 313
 - EMB_FORMAT_EYS, 313
 - EMB_FORMAT_FXY, 313
 - EMB_FORMAT_GC, 313
 - EMB_FORMAT_GNC, 313
 - EMB_FORMAT_GT, 313
 - EMB_FORMAT_HUS, 313
 - EMB_FORMAT_INB, 314
 - EMB_FORMAT_INF, 314
 - EMB_FORMAT_JEF, 314
 - EMB_FORMAT_KSM, 314
 - EMB_FORMAT_MAX, 314
 - EMB_FORMAT_MIT, 314
 - EMB_FORMAT_NEW, 314
 - EMB_FORMAT_OFM, 314
 - EMB_FORMAT_PCD, 314
 - EMB_FORMAT_PCM, 314
 - EMB_FORMAT_PCQ, 314
 - EMB_FORMAT_PCS, 314
 - EMB_FORMAT_PEC, 314
 - EMB_FORMAT_PEL, 314
 - EMB_FORMAT_PEM, 314
 - EMB_FORMAT_PES, 314
 - EMB_FORMAT_PHB, 314
 - EMB_FORMAT_PHC, 314

EMB_FORMAT_PLT, [315](#)
EMB_FORMAT_RGB, [315](#)
EMB_FORMAT_SEW, [315](#)
EMB_FORMAT_SHV, [315](#)
EMB_FORMAT_SST, [315](#)
EMB_FORMAT_STX, [315](#)
EMB_FORMAT_SVG, [315](#)
EMB_FORMAT_T01, [315](#)
EMB_FORMAT_T09, [315](#)
EMB_FORMAT_TAP, [315](#)
EMB_FORMAT_THR, [315](#)
EMB_FORMAT_TXT, [315](#)
EMB_FORMAT_U00, [315](#)
EMB_FORMAT_U01, [315](#)
EMB_FORMAT_VIP, [315](#)
EMB_FORMAT_VP3, [315](#)
EMB_FORMAT_XXX, [315](#)
EMB_FORMAT_ZSK, [315](#)
emb_identify_format, [321](#)
EMB_IMAGE, [316](#)
EMB_LINE, [316](#)
EMB_MAX_LAYERS, [316](#)
EMB_PATH, [316](#)
EMB_POINT, [316](#)
EMB_POLYGON, [316](#)
EMB_POLYLINE, [316](#)
EMB_PUBLIC, [316](#)
EMB_RECT, [316](#)
emb_round, [321](#)
EMB_SPLINE, [316](#)
EMB_STITCH, [316](#)
EMB_TEXT_MULTI, [316](#)
EMB_TEXT_SINGLE, [316](#)
EMB_THREAD, [316](#)
EMB_VECTOR, [316](#)
emb_verbose, [336](#)
EmbAlignedDim, [319](#)
EmbAngularDim, [319](#)
EmbArc, [319](#)
embArc_clockwise, [321](#)
embArc_init, [321](#)
EmbArcLengthDim, [319](#)
EmbArray, [319](#)
embArray_addArc, [322](#)
embArray_addCircle, [322](#)
embArray_addEllipse, [322](#)
embArray_addFlag, [322](#)
embArray_addLine, [322](#)
embArray_addPath, [322](#)
embArray_addPoint, [322](#)
embArray_addPolygon, [322](#)
embArray_addPolyline, [322](#)
embArray_addRect, [322](#)
embArray_addStitch, [322](#)
embArray_addThread, [323](#)
embArray_addVector, [323](#)
embArray_copy, [323](#)
embArray_create, [323](#)
embArray_free, [323](#)
embArray_resize, [323](#)
EmbBezier, [319](#)
EmbBlock, [319](#)
EmbCircle, [319](#)
embCircle_init, [323](#)
EmbColor, [319](#)
embColor_create, [323](#)
embColor_distance, [323](#)
embColor_fromHexStr, [323](#)
embColor_make, [324](#)
embConstantPi, [336](#)
EmbDiameterDim, [319](#)
EmbEllipse, [319](#)
embEllipse_area, [324](#)
embEllipse_diameterX, [324](#)
embEllipse_diameterY, [324](#)
embEllipse_height, [324](#)
embEllipse_init, [324](#)
embEllipse_make, [324](#)
embEllipse_perimeter, [324](#)
embEllipse_width, [324](#)
EmbFlag, [319](#)
EMBFORMAT_MAXDESC, [316](#)
EMBFORMAT_MAXEXT, [316](#)
EMBFORMAT_OBJECTONLY, [316](#)
EMBFORMAT_STCHANDOBJ, [317](#)
EMBFORMAT_STITCHONLY, [317](#)
EMBFORMAT_UNSUPPORTED, [317](#)
EmbFormatList, [319](#)
EmbGeometry, [319](#)
embGeometry_boundingRect, [324](#)
embGeometry_free, [324](#)
embGeometry_init, [325](#)
embGeometry_move, [325](#)
embGeometry_vulcanize, [325](#)
EmbImage, [320](#)
embImage_create, [325](#)
embImage_free, [325](#)
embImage_read, [325](#)
embImage_write, [325](#)
EmbInfiniteLine, [320](#)
EmbLayer, [320](#)
EmbLeaderDim, [320](#)
EmbLine, [320](#)
embLine_intersectionPoint, [325](#)
embLine_make, [325](#)
embLine_normalVector, [326](#)
EmbLinearDim, [320](#)
EmbOrdinateDim, [320](#)
EmbPath, [320](#)
EmbPattern, [320](#)
embPattern_addCircleAbs, [326](#)
embPattern_addEllipseAbs, [326](#)
embPattern_addLineAbs, [326](#)
embPattern_addPathAbs, [326](#)
embPattern_addPointAbs, [326](#)
embPattern_addPolygonAbs, [326](#)

- embPattern_addPolylineAbs, 326
- embPattern_addRectAbs, 326
- embPattern_addStitchAbs, 326
- embPattern_addStitchRel, 327
- embPattern_addThread, 327
- embPattern_calcBoundingBox, 327
- embPattern_center, 327
- embPattern_changeColor, 327
- embPattern_color_count, 327
- embPattern_combine, 327
- embPattern_combineJumpStitches, 327
- embPattern_convertGeometry, 327
- embPattern_copyPolylinesToStitchList, 328
- embPattern_copyStitchListToPolylines, 328
- embPattern_correctForMaxStitchLength, 328
- embPattern_create, 328
- embPattern_crossstitch, 328
- embPattern_designDetails, 328
- embPattern_end, 328
- embPattern_fixColorCount, 328
- embPattern_flip, 328
- embPattern_flipHorizontal, 329
- embPattern_flipVertical, 329
- embPattern_free, 329
- embPattern_hideStitchesOverLength, 329
- embPattern_horizontal_fill, 329
- embPattern_jumpStitches, 329
- embPattern_lengthHistogram, 329
- embPattern_loadExternalColorFile, 329
- embPattern_maximumStitchLength, 329
- embPattern_minimumStitchLength, 329
- embPattern_movePolylinesToStitchList, 330
- embPattern_moveStitchListToPolylines, 330
- embPattern_read, 330
- embPattern_readAuto, 330
- embPattern_realStitches, 330
- embPattern_render, 330
- embPattern_scale, 330
- embPattern_simulate, 330
- embPattern_totalStitchLength, 330
- embPattern_trimStitches, 331
- embPattern_write, 331
- embPattern_writeAuto, 331
- EmbPoint, 320
- EmbPolygon, 320
- EmbPolyline, 320
- EmbRadiusDim, 320
- EmbRay, 320
- EmbReal, 320
- EmbRect, 320
- embRect_area, 331
- embRect_init, 331
- EmbSatinOutline, 320
- embSatinOutline_generateSatinOutline, 331
- embSatinOutline_renderStitches, 331
- EmbSpline, 320
- EmbStitch, 321
- EmbTextMulti, 321
- EmbTextSingle, 321
- EmbThread, 321
- embThread_findNearestColor, 331
- embThread_findNearestThread, 332
- embThread_getRandom, 332
- EmbTime, 321
- embTime_initNow, 332
- embTime_time, 332
- EmbVector, 321
- embVector_add, 332
- embVector_angle, 332
- embVector_average, 333
- embVector_cross, 333
- embVector_distance, 333
- embVector_dot, 333
- embVector_length, 333
- embVector_multiply, 333
- embVector_normalize, 334
- embVector_relativeX, 334
- embVector_relativeY, 334
- embVector_subtract, 334
- embVector_transpose_product, 334
- embVector_unit, 334
- END, 317
- Exquisite_Polyester, 317
- formatTable, 336
- Fufu_Polyester, 317
- Fufu_Rayon, 317
- full_test_matrix, 334
- getArcCenter, 334
- getArcDataFromBulge, 335
- getCircleCircleIntersections, 335
- getCircleTangentPoints, 335
- Hemingworth_Polyester, 317
- hilbert_curve, 335
- hus_thread, 317
- husThreads, 336
- Isacord_Polyester, 317
- Isafil_Rayon, 317
- jef_thread, 317
- jefThreads, 336
- JUMP, 317
- L_system, 321
- LIBEMBROIDERY_EMBEDDED_VERSION, 317
- lindenmayer_system, 335
- Madeira_Polyester, 317
- Madeira_Rayon, 317
- Marathon_Polyester, 317
- Marathon_Rayon, 317
- MAX_STITCHES, 318
- MAX_THREADS, 318
- Metro_Polyester, 318
- NORMAL, 318
- numberOfFormats, 318
- Pantone, 318
- pcm_thread, 318
- pcmThreads, 337
- pec_thread, 318

- pecThreadCount, [337](#)
- pecThreads, [337](#)
- radians, [335](#)
- report, [335](#)
- RobisonAnton_Polyester, [318](#)
- RobisonAnton_Rayon, [318](#)
- SEQUIN, [318](#)
- shv_thread, [318](#)
- shvThreadCount, [337](#)
- shvThreads, [337](#)
- Sigma_Polyester, [318](#)
- STOP, [318](#)
- Sulky_Rayon, [318](#)
- SVG_Colors, [318](#)
- testMain, [336](#)
- thread_color, [321](#)
- ThreadArt_Polyester, [318](#)
- ThreadArt_Rayon, [318](#)
- threadColor, [336](#)
- threadColorName, [336](#)
- threadColorNum, [336](#)
- ThreaDelight_Polyester, [319](#)
- TRIM, [319](#)
- vipDecodingTable, [337](#)
- Z102_Isacord_Polyester, [319](#)
- embroidery_internal.h
 - bcf_difat_create, [362](#)
 - bcf_directory, [360](#)
 - bcf_directory_entry, [360](#)
 - bcf_directory_free, [362](#)
 - bcf_file, [360](#)
 - bcf_file_difat, [360](#)
 - bcf_file_difat_free, [362](#)
 - bcf_file_fat, [360](#)
 - bcf_file_fat_free, [362](#)
 - bcf_file_free, [362](#)
 - bcf_file_header, [360](#)
 - bcfFile_read, [362](#)
 - bcfFileFat_create, [362](#)
 - bcfFileHeader_isValid, [362](#)
 - bcfFileHeader_read, [363](#)
 - binaryReadString, [363](#)
 - binaryReadUnicodeString, [363](#)
 - binaryWriteInt, [363](#)
 - binaryWriteIntBE, [363](#)
 - binaryWriteShort, [363](#)
 - binaryWriteUInt, [363](#)
 - binaryWriteUIntBE, [363](#)
 - binaryWriteUShort, [364](#)
 - binaryWriteUShortBE, [364](#)
 - BULGETOCONTROL, [352](#)
 - BULGETOEND, [353](#)
 - check_header_present, [364](#)
 - CompoundFileDirectory, [364](#)
 - CompoundFileDirectoryEntry, [364](#)
 - CompoundFileSector_DIFAT_Sector, [353](#)
 - CompoundFileSector_EndOfChain, [353](#)
 - CompoundFileSector_FAT_Sector, [353](#)
 - CompoundFileSector_FreeSector, [353](#)
 - CompoundFileSector_MaxRegSector, [353](#)
 - CompoundFileStreamId_MaxRegularStreamId, [353](#)
 - CompoundFileStreamId_NoStream, [353](#)
 - compress, [361](#)
 - compress_get_bits, [364](#)
 - compress_get_position, [364](#)
 - compress_get_token, [365](#)
 - compress_load_block, [365](#)
 - compress_load_character_huffman, [365](#)
 - compress_load_character_length_huffman, [365](#)
 - compress_load_distance_huffman, [365](#)
 - compress_pop, [365](#)
 - compress_read_variable_length, [365](#)
 - copy_trim, [365](#)
 - create_test_file_1, [365](#)
 - create_test_file_2, [365](#)
 - create_test_file_3, [365](#)
 - CSV_EXPECT, [361](#)
 - CSV_EXPECT_COMMA, [361](#)
 - CSV_EXPECT_NULL, [361](#)
 - CSV_EXPECT_QUOTE1, [361](#)
 - CSV_EXPECT_QUOTE2, [361](#)
 - CSV_MODE, [361](#)
 - CSV_MODE_COMMENT, [361](#)
 - CSV_MODE_NULL, [361](#)
 - CSV_MODE_STITCH, [362](#)
 - CSV_MODE_THREAD, [361](#)
 - CSV_MODE_VARIABLE, [361](#)
 - CUBICTOCONTROL1, [353](#)
 - CUBICTOCONTROL2, [353](#)
 - CUBICTOEND, [353](#)
 - decode_t01_record, [365](#)
 - decode_tajima_ternary, [366](#)
 - decodeNewStitch, [366](#)
 - DXF_VERSION_2000, [353](#)
 - DXF_VERSION_2002, [353](#)
 - DXF_VERSION_2004, [353](#)
 - DXF_VERSION_2006, [353](#)
 - DXF_VERSION_2007, [353](#)
 - DXF_VERSION_2009, [353](#)
 - DXF_VERSION_2010, [354](#)
 - DXF_VERSION_2013, [354](#)
 - DXF_VERSION_R10, [354](#)
 - DXF_VERSION_R11, [354](#)
 - DXF_VERSION_R12, [354](#)
 - DXF_VERSION_R13, [354](#)
 - DXF_VERSION_R14, [354](#)
 - DXF_VERSION_R15, [354](#)
 - DXF_VERSION_R18, [354](#)
 - DXF_VERSION_R21, [354](#)
 - DXF_VERSION_R24, [354](#)
 - DXF_VERSION_R27, [354](#)
 - ELEMENT_A, [354](#)
 - ELEMENT_ANIMATE, [354](#)
 - ELEMENT_ANIMATECOLOR, [354](#)
 - ELEMENT_ANIMATEMOTION, [354](#)

ELEMENT_ANIMATETRANSFORM, 354
ELEMENT_ANIMATION, 354
ELEMENT_AUDIO, 355
ELEMENT_CIRCLE, 355
ELEMENT_DEFS, 355
ELEMENT_DESC, 355
ELEMENT_DISCARD, 355
ELEMENT_ELLIPSE, 355
ELEMENT_FONT, 355
ELEMENT_FONT_FACE, 355
ELEMENT_FONT_FACE_SRC, 355
ELEMENT_FONT_FACE_URI, 355
ELEMENT_FOREIGN_OBJECT, 355
ELEMENT_G, 355
ELEMENT_GLYPH, 355
ELEMENT_HANDLER, 355
ELEMENT_HKERN, 355
ELEMENT_IMAGE, 355
ELEMENT_LINE, 355
ELEMENT_LINEAR_GRADIENT, 355
ELEMENT_LISTENER, 356
ELEMENT_METADATA, 356
ELEMENT_MISSING_GLYPH, 356
ELEMENT_MPATH, 356
ELEMENT_PATH, 356
ELEMENT_POLYGON, 356
ELEMENT_POLYLINE, 356
ELEMENT_PREFETCH, 356
ELEMENT_RADIAL_GRADIENT, 356
ELEMENT_RECT, 356
ELEMENT_SCRIPT, 356
ELEMENT_SET, 356
ELEMENT_SOLID_COLOR, 356
ELEMENT_STOP, 356
ELEMENT_SVG, 356
ELEMENT_SWITCH, 356
ELEMENT_TBREAK, 356
ELEMENT_TEXT, 356
ELEMENT_TEXT_AREA, 357
ELEMENT_TITLE, 357
ELEMENT_TSPAN, 357
ELEMENT_USE, 357
ELEMENT_VIDEO, 357
ELEMENT_XML, 357
ELLIPSETOEND, 357
ELLIPSETORAD, 357
EMB_BIG_ENDIAN, 357
EMB_INT16_BIG, 357
EMB_INT16_LITTLE, 357
EMB_INT32_BIG, 357
EMB_INT32_LITTLE, 357
EMB_LITTLE_ENDIAN, 357
EMB_MAX, 357
EMB_MIN, 357
emb_optOut, 366
emb_readline, 366
embColor_read, 366
embColor_write, 366
emblnt_read, 367
emblnt_write, 367
encode_t01_record, 367
encode_tajima_ternary, 367
ENDIAN_HOST, 357
entriesInDifatSector, 367
fpad, 367
fread_int16, 367
fread_int32_be, 368
fread_uint16, 368
GetFile, 368
GREEN_TERM_COLOR, 358
HOOP_110X110, 358
HOOP_126X110, 358
HOOP_140X200, 358
HOOP_230X200, 358
HOOP_50X50, 358
huffman, 361
huffman_build_table, 368
huffman_table_lookup, 368
hus_compress, 368
hus_decompress, 368
imageWithFrame, 384
LINETO, 358
loadFatFromSector, 369
mitDecodeStitch, 369
mitEncodeStitch, 369
MOVETO, 358
N_PES_VERSIONS, 358
numberOfEntriesInDifatSector, 369
ObjectTypeRootEntry, 358
ObjectTypeStorage, 358
ObjectTypeStream, 358
ObjectTypeUnknown, 358
PES0001, 358
PES0020, 358
PES0022, 358
PES0030, 358
PES0040, 359
PES0050, 359
PES0055, 359
PES0056, 359
PES0060, 359
PES0070, 359
PES0080, 359
PES0090, 359
PES0100, 359
pfaffDecode, 369
pfaffEncode, 369
printArcResults, 369
QUADTOCONTROL, 359
QUADTOEND, 359
read100, 370
read10o, 370
readArt, 370
readBmc, 370
readBro, 370
readCnd, 370

readCol, [370](#)
readCsd, [370](#)
readCsv, [370](#)
readDat, [370](#)
readDem, [370](#)
readDescriptions, [371](#)
readDsb, [371](#)
readDst, [371](#)
readDsz, [371](#)
readDxf, [371](#)
readEdr, [371](#)
readEmd, [371](#)
readExp, [371](#)
readExy, [371](#)
readEys, [371](#)
readFeatherPatterns, [372](#)
readFullSector, [372](#)
readFxy, [372](#)
readGc, [372](#)
readGnc, [372](#)
readGt, [372](#)
readHoopName, [372](#)
readHus, [372](#)
readImageString, [372](#)
readInb, [373](#)
readInf, [373](#)
readJef, [373](#)
readKsm, [373](#)
readMax, [373](#)
readMit, [373](#)
readMotifPatterns, [373](#)
readNew, [373](#)
readNextSector, [373](#)
readOfm, [373](#)
readPcd, [374](#)
readPcm, [374](#)
readPcq, [374](#)
readPcs, [374](#)
readPec, [374](#)
readPecStitches, [374](#)
readPel, [374](#)
readPem, [374](#)
readPes, [375](#)
readPESHeaderV10, [375](#)
readPESHeaderV5, [375](#)
readPESHeaderV6, [375](#)
readPESHeaderV7, [375](#)
readPESHeaderV8, [375](#)
readPESHeaderV9, [375](#)
readPhb, [375](#)
readPhc, [375](#)
readPlt, [375](#)
readProgrammableFills, [376](#)
readRgb, [376](#)
readSew, [376](#)
readShv, [376](#)
readSst, [376](#)
readStx, [376](#)
readSvg, [376](#)
readT01, [376](#)
readT09, [376](#)
readTap, [376](#)
readThr, [377](#)
readThreads, [377](#)
readTxt, [377](#)
readU00, [377](#)
readU01, [377](#)
readVip, [377](#)
readVp3, [377](#)
readXxx, [377](#)
readZsk, [377](#)
RED_TERM_COLOR, [359](#)
RESET_TERM_COLOR, [359](#)
safe_free, [377](#)
stringInArray, [378](#)
StxThread, [361](#)
SubDescriptor, [361](#)
SVG_ATTRIBUTE, [359](#)
SVG_CATCH_ALL, [359](#)
SVG_CREATOR_EMBROIDERMODDER, [359](#)
SVG_CREATOR_ILLUSTRATOR, [359](#)
SVG_CREATOR_INKSCAPE, [359](#)
SVG_CREATOR_NULL, [360](#)
SVG_ELEMENT, [360](#)
SVG_EXPECT_ATTRIBUTE, [360](#)
SVG_EXPECT_ELEMENT, [360](#)
SVG_EXPECT_NULL, [360](#)
SVG_EXPECT_VALUE, [360](#)
SVG_MEDIA_PROPERTY, [360](#)
SVG_NULL, [360](#)
SVG_PROPERTY, [360](#)
SvgAttribute, [361](#)
testEmbCircle, [378](#)
testEmbCircle_2, [378](#)
testEmbFormat, [378](#)
testGeomArc, [378](#)
testTangentPoints, [378](#)
testThreadColor, [378](#)
ThredExtension, [361](#)
ThredHeader, [361](#)
VipHeader, [361](#)
vp3Hoop, [361](#)
write100, [378](#)
write10o, [378](#)
write_24bit, [378](#)
writeArt, [379](#)
writeBmc, [379](#)
writeBro, [379](#)
writeCnd, [379](#)
writeCol, [379](#)
writeCsd, [379](#)
writeCsv, [379](#)
writeDat, [379](#)
writeDem, [379](#)
writeDsb, [379](#)
writeDst, [379](#)

- writeDsz, [380](#)
- writeDxf, [380](#)
- writeEdr, [380](#)
- writeEmd, [380](#)
- writeExp, [380](#)
- writeExy, [380](#)
- writeEys, [380](#)
- writeFxy, [380](#)
- writeGc, [380](#)
- writeGnc, [380](#)
- writeGt, [380](#)
- writeHus, [381](#)
- writeInb, [381](#)
- writeInf, [381](#)
- writeJef, [381](#)
- writeKsm, [381](#)
- writeMax, [381](#)
- writeMit, [381](#)
- writeNew, [381](#)
- writeOfm, [381](#)
- writePcd, [381](#)
- writePcm, [381](#)
- writePcq, [382](#)
- writePcs, [382](#)
- writePec, [382](#)
- writePecStitches, [382](#)
- writePel, [382](#)
- writePem, [382](#)
- writePes, [382](#)
- writePhb, [382](#)
- writePhc, [382](#)
- writePlt, [382](#)
- writeRgb, [383](#)
- writeSew, [383](#)
- writeShv, [383](#)
- writeSst, [383](#)
- writeStx, [383](#)
- writeSvg, [383](#)
- writeT01, [383](#)
- writeT09, [383](#)
- writeTap, [383](#)
- writeThr, [383](#)
- writeTxt, [383](#)
- writeU00, [384](#)
- writeU01, [384](#)
- writeVip, [384](#)
- writeVp3, [384](#)
- writeXxx, [384](#)
- writeZsk, [384](#)
- YELLOW_TERM_COLOR, [360](#)
- EmbSatinOutline
 - embroidery.h, [320](#)
- EmbSatinOutline_, [97](#)
 - length, [97](#)
 - side1, [97](#)
 - side2, [97](#)
- embSatinOutline_generateSatinOutline
 - embroidery.h, [331](#)
- main.c, [467](#)
- embSatinOutline_renderStitches
 - embroidery.h, [331](#)
 - main.c, [467](#)
- EmbSpline
 - embroidery.h, [320](#)
- EmbSpline_, [98](#)
 - beziers, [98](#)
- EmbStitch
 - embroidery.h, [321](#)
- EmbStitch_, [98](#)
 - color, [98](#)
 - flags, [98](#)
 - x, [98](#)
 - y, [99](#)
- EmbTextMulti
 - embroidery.h, [321](#)
- EmbTextMulti_, [99](#)
 - position, [99](#)
 - text, [99](#)
- EmbTextSingle
 - embroidery.h, [321](#)
- EmbTextSingle_, [99](#)
 - position, [100](#)
 - text, [100](#)
- EmbThread
 - embroidery.h, [321](#)
- EmbThread_, [100](#)
 - catalogNumber, [100](#)
 - color, [100](#)
 - description, [100](#)
- embThread_findNearestColor
 - embroidery.h, [331](#)
 - main.c, [468](#)
- embThread_findNearestThread
 - embroidery.h, [332](#)
 - main.c, [468](#)
- embThread_getRandom
 - embroidery.h, [332](#)
 - main.c, [468](#)
- EmbTime
 - embroidery.h, [321](#)
- EmbTime_, [101](#)
 - day, [101](#)
 - hour, [101](#)
 - minute, [101](#)
 - month, [101](#)
 - second, [101](#)
 - year, [101](#)
- embTime_initNow
 - embroidery.h, [332](#)
 - main.c, [468](#)
- embTime_time
 - embroidery.h, [332](#)
 - main.c, [468](#)
- EmbVector
 - embroidery.h, [321](#)
- EmbVector_, [102](#)

- x, [102](#)
 - y, [102](#)
- embVector_add
 - embroidery.h, [332](#)
 - vector.c, [457](#)
- embVector_angle
 - embroidery.h, [332](#)
 - vector.c, [457](#)
- embVector_average
 - embroidery.h, [333](#)
 - vector.c, [457](#)
- embVector_cross
 - embroidery.h, [333](#)
 - vector.c, [457](#)
- embVector_distance
 - embroidery.h, [333](#)
 - vector.c, [458](#)
- embVector_dot
 - embroidery.h, [333](#)
 - vector.c, [458](#)
- embVector_length
 - embroidery.h, [333](#)
 - vector.c, [458](#)
- embVector_multiply
 - embroidery.h, [333](#)
 - vector.c, [458](#)
- embVector_normalize
 - embroidery.h, [334](#)
 - vector.c, [458](#)
- embVector_print
 - main.c, [468](#)
- embVector_relativeX
 - embroidery.h, [334](#)
 - vector.c, [458](#)
- embVector_relativeY
 - embroidery.h, [334](#)
 - vector.c, [459](#)
- embVector_subtract
 - embroidery.h, [334](#)
 - vector.c, [459](#)
- embVector_transpose_product
 - embroidery.h, [334](#)
 - vector.c, [459](#)
- embVector_unit
 - embroidery.h, [334](#)
 - vector.c, [459](#)
- emd, [9](#), [415](#)
- emdDecode
 - format_emd.c, [415](#)
- encode_record
 - format_dst.c, [413](#)
- encode_t01_record
 - embroidery_internal.h, [367](#)
 - encoding.c, [393](#)
- encode_tajima_ternary
 - embroidery_internal.h, [367](#)
 - encoding.c, [393](#)
- encode_tap_record
 - format_tap.c, [439](#)
- encoding.c
 - decode_t01_record, [392](#)
 - decode_tajima_ternary, [392](#)
 - decodeNewStitch, [392](#)
 - embColor_fromHexStr, [392](#)
 - embInt_read, [393](#)
 - embInt_write, [393](#)
 - encode_t01_record, [393](#)
 - encode_tajima_ternary, [393](#)
 - mitDecodeStitch, [393](#)
 - mitEncodeStitch, [393](#)
 - pfaffDecode, [393](#)
 - pfaffEncode, [394](#)
 - reverse_byte_order, [394](#)
 - write_24bit, [394](#)
- END
 - embroidery.h, [317](#)
- end
 - EmbArc_, [77](#)
 - EmbBezier_, [79](#)
 - EmbLine_, [91](#)
- end_action
 - mainwindow.cpp, [280](#)
- endCommand
 - CmdPromptInput, [69](#)
- ENDIAN_HOST
 - embroidery_internal.h, [357](#)
- enterEvent
 - View, [211](#)
- entriesInDifatSector
 - embroidery_internal.h, [367](#)
 - main.c, [469](#)
- error_action
 - mainwindow.cpp, [280](#)
- escapePressed
 - CmdPrompt, [54](#)
 - CmdPromptInput, [69](#)
 - MainWindow, [144](#)
 - MdiWindow, [163](#)
 - View, [211](#)
- event
 - Application, [50](#)
- eventFilter
 - CmdPromptInput, [69](#)
 - PropertyEditor, [172](#)
- exitApp
 - embroidermodder.cpp, [222](#)
- exp, [9](#), [415](#)
- expDecode
 - format_exp.c, [415](#)
- Exquisite_Polyester
 - embroidery.h, [317](#)
- extension
 - EmbFormatList_, [86](#)
- extensions
 - settings-dialog.cpp, [298](#)
- extern/libembroidery/src/array.c, [300](#)

- extern/libembroidery/src/compress.c, 302
- extern/libembroidery/src/embroidery.h, 304, 337
- extern/libembroidery/src/embroidery_internal.h, 344, 384
- extern/libembroidery/src/encoding.c, 391
- extern/libembroidery/src/fill.c, 394
- extern/libembroidery/src/formats.c, 399
- extern/libembroidery/src/formats/format_100.c, 403
- extern/libembroidery/src/formats/format_10o.c, 403
- extern/libembroidery/src/formats/format_art.c, 404
- extern/libembroidery/src/formats/format_bmc.c, 404
- extern/libembroidery/src/formats/format_bro.c, 405
- extern/libembroidery/src/formats/format_cnd.c, 405
- extern/libembroidery/src/formats/format_col.c, 406
- extern/libembroidery/src/formats/format_csd.c, 407
- extern/libembroidery/src/formats/format_csv.c, 408
- extern/libembroidery/src/formats/format_dat.c, 409
- extern/libembroidery/src/formats/format_dem.c, 410
- extern/libembroidery/src/formats/format_dsb.c, 410
- extern/libembroidery/src/formats/format_dst.c, 411
- extern/libembroidery/src/formats/format_dsz.c, 413
- extern/libembroidery/src/formats/format_dxf.c, 413
- extern/libembroidery/src/formats/format_edr.c, 414
- extern/libembroidery/src/formats/format_emd.c, 415
- extern/libembroidery/src/formats/format_exp.c, 415
- extern/libembroidery/src/formats/format_exy.c, 416
- extern/libembroidery/src/formats/format_eyc.c, 416
- extern/libembroidery/src/formats/format_fxy.c, 417
- extern/libembroidery/src/formats/format_gc.c, 417
- extern/libembroidery/src/formats/format_gnc.c, 418
- extern/libembroidery/src/formats/format_gt.c, 418
- extern/libembroidery/src/formats/format_hus.c, 419
- extern/libembroidery/src/formats/format_inb.c, 420
- extern/libembroidery/src/formats/format_inf.c, 420
- extern/libembroidery/src/formats/format_jef.c, 421
- extern/libembroidery/src/formats/format_ksm.c, 422
- extern/libembroidery/src/formats/format_max.c, 422
- extern/libembroidery/src/formats/format_mit.c, 423
- extern/libembroidery/src/formats/format_new.c, 424
- extern/libembroidery/src/formats/format_ofm.c, 424
- extern/libembroidery/src/formats/format_pcd.c, 425
- extern/libembroidery/src/formats/format_pcm.c, 426
- extern/libembroidery/src/formats/format_pcq.c, 426
- extern/libembroidery/src/formats/format_pcs.c, 427
- extern/libembroidery/src/formats/format_pec.c, 427
- extern/libembroidery/src/formats/format_pel.c, 429
- extern/libembroidery/src/formats/format_pem.c, 429
- extern/libembroidery/src/formats/format_pes.c, 430
- extern/libembroidery/src/formats/format_phb.c, 432
- extern/libembroidery/src/formats/format_phc.c, 433
- extern/libembroidery/src/formats/format_plt.c, 433
- extern/libembroidery/src/formats/format_rgb.c, 434
- extern/libembroidery/src/formats/format_sew.c, 434
- extern/libembroidery/src/formats/format_shv.c, 435
- extern/libembroidery/src/formats/format_sst.c, 435
- extern/libembroidery/src/formats/format_stx.c, 436
- extern/libembroidery/src/formats/format_svg.c, 436
- extern/libembroidery/src/formats/format_t01.c, 438
- extern/libembroidery/src/formats/format_t09.c, 438
- extern/libembroidery/src/formats/format_tap.c, 439
- extern/libembroidery/src/formats/format_thr.c, 439
- extern/libembroidery/src/formats/format_txt.c, 440
- extern/libembroidery/src/formats/format_u00.c, 440
- extern/libembroidery/src/formats/format_u01.c, 441
- extern/libembroidery/src/formats/format_vip.c, 441
- extern/libembroidery/src/formats/format_vp3.c, 443
- extern/libembroidery/src/formats/format_xxx.c, 444
- extern/libembroidery/src/formats/format_zsk.c, 445
- extern/libembroidery/src/geometry.c, 445
- extern/libembroidery/src/geometry/arc.c, 447
- extern/libembroidery/src/geometry/circle.c, 450
- extern/libembroidery/src/geometry/ellipse.c, 451
- extern/libembroidery/src/geometry/functions.c, 453
- extern/libembroidery/src/geometry/line.c, 454
- extern/libembroidery/src/geometry/path.c, 454
- extern/libembroidery/src/geometry/polygon.c, 454
- extern/libembroidery/src/geometry/polyline.c, 454
- extern/libembroidery/src/geometry/rect.c, 455
- extern/libembroidery/src/geometry/text.c, 455
- extern/libembroidery/src/geometry/vector.c, 457
- extern/libembroidery/src/image.c, 459
- extern/libembroidery/src/main.c, 460
- extern/libembroidery/src/pattern.c, 471
- extern/libembroidery/src/thread-color.c, 476
- exy, 9, 416
- eyc, 372, 417
- F10Pressed
 - CmdPrompt, 54
 - CmdPromptInput, 70
- F11Pressed
 - CmdPrompt, 54
 - CmdPromptInput, 70
- F12Pressed
 - CmdPrompt, 54
 - CmdPromptInput, 70
- F1Pressed
 - CmdPrompt, 54
 - CmdPromptInput, 70
- F2Pressed
 - CmdPrompt, 54
 - CmdPromptInput, 70
- F3Pressed
 - CmdPrompt, 54
 - CmdPromptInput, 70
- F4Pressed
 - CmdPrompt, 54
 - CmdPromptInput, 70
- F5Pressed
 - CmdPrompt, 54
 - CmdPromptInput, 70
- F6Pressed
 - CmdPrompt, 55
 - CmdPromptInput, 70
- F7Pressed
 - CmdPrompt, 55
 - CmdPromptInput, 70

- F8Pressed
 - CmdPrompt, 55
 - CmdPromptInput, 71
- F9Pressed
 - CmdPrompt, 55
 - CmdPromptInput, 71
- factor
 - UndoableCommand, 204
- fat
 - _bcf_file, 42
- fatEntries
 - _bcf_file_fat, 44
- fatEntryCount
 - _bcf_file_fat, 44
- fatSectorCount
 - _bcf_file_difat, 43
- fatSectorEntries
 - _bcf_file_difat, 43
- fieldEdited
 - PropertyEditor, 172
- fieldNewText
 - property-editor.cpp, 296
- fieldNoText
 - property-editor.cpp, 296
- fieldOffText
 - property-editor.cpp, 296
- fieldOldText
 - property-editor.cpp, 296
- fieldOnText
 - property-editor.cpp, 296
- fieldVariesText
 - property-editor.cpp, 296
- fieldYesText
 - property-editor.cpp, 296
- fileExtension
 - embroidermodder.h, 230
 - mdiwindow.cpp, 294
- fileWasLoaded
 - MdiWindow, 167
- fill.c
 - dragon_curve, 395
 - embPattern_combine, 395
 - embPattern_convertGeometry, 395
 - embPattern_crossstitch, 395
 - embPattern_horizontal_fill, 395
 - embPattern_stitchArc, 395
 - embPattern_stitchCircle, 395
 - embPattern_stitchEllipse, 396
 - embPattern_stitchPath, 396
 - embPattern_stitchPolygon, 396
 - embPattern_stitchPolyline, 396
 - embPattern_stitchRect, 397
 - embPattern_stitchText, 397
 - embPolygon_reduceByDistance, 397
 - embPolygon_reduceByNth, 397
 - generate_dragon_curve, 397
 - greedy_algorithm, 397
 - hilbert_curve, 397
 - hilbert_curve_l_system, 398
 - join_short_stitches, 398
 - lindenmayer_system, 398
 - rules, 398
 - save_points_to_pattern, 398
 - threshold_method, 398
- filled
 - Geometry, 129
- findIndex
 - Geometry, 110
- findMdiWindow
 - MainWindow, 144
- firstDifatSectorLocation
 - _bcf_file_header, 45
- firstDirectorySectorLocation
 - _bcf_file_header, 45
- firstMiniFATSectorLocation
 - _bcf_file_header, 45
- flag
 - EmbGeometry_, 87
- FLAG_CIRCLE
 - main.c, 462
- FLAG_CIRCLE_SHORT
 - main.c, 463
- FLAG_COMBINE
 - main.c, 463
- FLAG_CROSS_STITCH
 - main.c, 463
- FLAG_ELLIPSE
 - main.c, 463
- FLAG_ELLIPSE_SHORT
 - main.c, 463
- FLAG_FILL
 - main.c, 463
- FLAG_FILL_SHORT
 - main.c, 463
- FLAG_FORMATS
 - main.c, 463
- FLAG_FORMATS_SHORT
 - main.c, 463
- FLAG_FULL_TEST_SUITE
 - main.c, 463
- FLAG_HELP
 - main.c, 463
- FLAG_HELP_SHORT
 - main.c, 463
- FLAG_HILBERT_CURVE
 - main.c, 463
- FLAG_LINE
 - main.c, 463
- FLAG_LINE_SHORT
 - main.c, 463
- FLAG_POLYGON
 - main.c, 463
- FLAG_POLYGON_SHORT
 - main.c, 463
- FLAG_POLYLINE
 - main.c, 463

- FLAG_POLYLINE_SHORT
 - main.c, [464](#)
- FLAG_QUIET
 - main.c, [464](#)
- FLAG_QUIET_SHORT
 - main.c, [464](#)
- FLAG_RENDER
 - main.c, [464](#)
- FLAG_RENDER_SHORT
 - main.c, [464](#)
- FLAG_SATIN
 - main.c, [464](#)
- FLAG_SATIN_SHORT
 - main.c, [464](#)
- FLAG_SIERPINSKI_TRIANGLE
 - main.c, [464](#)
- FLAG_SIMULATE
 - main.c, [464](#)
- FLAG_STITCH
 - main.c, [464](#)
- FLAG_STITCH_SHORT
 - main.c, [464](#)
- FLAG_TEST
 - main.c, [464](#)
- FLAG_TO
 - main.c, [464](#)
- FLAG_TO_SHORT
 - main.c, [464](#)
- FLAG_VERBOSE
 - main.c, [464](#)
- FLAG_VERBOSE_SHORT
 - main.c, [464](#)
- FLAG_VERSION
 - main.c, [464](#)
- FLAG_VERSION_SHORT
 - main.c, [464](#)
- flagList
 - EmbPath_, [93](#)
- flags
 - EmbStitch_, [98](#)
- Flared
 - Geometry, [107](#)
- Fletching
 - Geometry, [107](#)
- floatingChanged
 - CmdPrompt, [55](#)
- floatingChangedToolBar
 - MainWindow, [144](#)
- focusWidget
 - PropertyEditor, [174](#)
 - UndoEditor, [206](#)
- forceRepaint
 - MdiArea, [157](#)
 - SelectBox, [187](#)
- format_100.c
 - read100, [403](#)
 - write100, [403](#)
- format_10o.c
 - read10o, [404](#)
 - write10o, [404](#)
- format_art.c
 - readArt, [404](#)
 - writeArt, [404](#)
- format_bmc.c
 - readBmc, [405](#)
 - writeBmc, [405](#)
- format_bro.c
 - readBro, [405](#)
 - writeBro, [405](#)
- format_cnd.c
 - readCnd, [406](#)
 - writeCnd, [406](#)
- format_col.c
 - readCol, [407](#)
 - writeCol, [407](#)
- format_csd.c
 - _subMask, [408](#)
 - _xorMask, [408](#)
 - BuildDecryptionTable, [407](#)
 - csd_decryptArray, [408](#)
 - CsdSubMaskSize, [407](#)
 - CsdXorMaskSize, [407](#)
 - DecodeCsdByte, [408](#)
 - readCsd, [408](#)
 - writeCsd, [408](#)
- format_csv.c
 - csvStitchFlagToStr, [409](#)
 - csvStrToStitchFlag, [409](#)
 - readCsv, [409](#)
 - writeCsv, [409](#)
- format_dat.c
 - readDat, [409](#)
 - writeDat, [409](#)
- format_dem.c
 - readDem, [410](#)
 - writeDem, [410](#)
- format_dsb.c
 - readDsb, [411](#)
 - writeDsb, [411](#)
- format_dst.c
 - cci, [412](#)
 - decode_record_flags, [412](#)
 - encode_record, [413](#)
 - readDst, [413](#)
 - set_dst_variable, [413](#)
 - writeDst, [413](#)
- format_dsz.c
 - readDsz, [413](#)
 - writeDsz, [413](#)
- format_dxf.c
 - readDxf, [414](#)
 - readLine, [414](#)
 - writeDxf, [414](#)
- format_edr.c
 - readEdr, [414](#)
 - writeEdr, [414](#)

format_emd.c
 emdDecode, 415
 readEmd, 415
 writeEmd, 415
format_exp.c
 expDecode, 415
 readExp, 415
 writeExp, 416
format_exy.c
 decode_exy_flags, 416
 readExy, 416
 writeExy, 416
format_eys.c
 readEys, 416
 writeEys, 417
format_fxy.c
 readFxy, 417
 writeFxy, 417
format_gc.c
 readGc, 417
 writeGc, 418
format_gnc.c
 readGnc, 418
 writeGnc, 418
format_gt.c
 readGt, 418
 writeGt, 419
format_hus.c
 husCompressData, 419
 husDecodeByte, 419
 husDecodeStitchType, 419
 husDecompressData, 419
 husEncodeByte, 419
 husEncodeStitchType, 419
 readHus, 420
 writeHus, 420
format_inb.c
 readInb, 420
 writeInb, 420
format_inf.c
 readInf, 420
 writeInf, 421
format_jef.c
 jefDecode, 421
 jefEncode, 421
 jefGetHoopSize, 421
 jefSetHoopFromId, 421
 read_hoop, 421
 readJef, 422
 writeJef, 422
format_ksm.c
 ksmEncode, 422
 readKsm, 422
 writeKsm, 422
format_max.c
 max_header, 423
 readMax, 423
 writeMax, 423
format_mit.c
 readMit, 423
 writeMit, 424
format_new.c
 readNew, 424
 writeNew, 424
format_ofm.c
 ofmDecode, 425
 ofmReadBlockHeader, 425
 ofmReadClass, 425
 ofmReadColorChange, 425
 ofmReadExpanded, 425
 ofmReadLibrary, 425
 ofmReadThreads, 425
 readOfm, 425
 writeOfm, 425
format_pcd.c
 readPcd, 426
 writePcd, 426
format_pcm.c
 readPcm, 426
 writePcm, 426
format_pcq.c
 readPcq, 427
 writePcq, 427
format_pcs.c
 readPcs, 427
 writePcs, 427
format_pec.c
 pecEncode, 428
 pecEncodeJump, 428
 pecEncodeStop, 428
 readPec, 428
 readPecStitches, 428
 writeImage, 428
 writePec, 428
 writePecStitches, 428
format_pel.c
 readPel, 429
 writePel, 429
format_pem.c
 readPem, 429
 writePem, 429
format_pes.c
 pes_version, 432
 pes_version_strings, 432
 pesWriteEmbOneSection, 430
 pesWriteSewSegSection, 430
 readDescriptions, 430
 readFeatherPatterns, 430
 readHoopName, 430
 readImageString, 431
 readMotifPatterns, 431
 readPes, 431
 readPESHeaderV10, 431
 readPESHeaderV5, 431
 readPESHeaderV6, 431
 readPESHeaderV7, 431

- readPESHeaderV8, [431](#)
- readPESHeaderV9, [431](#)
- readProgrammableFills, [431](#)
- readThreads, [431](#)
- writePes, [432](#)
- format_phb.c
 - readPhb, [432](#)
 - writePhb, [432](#)
- format_phc.c
 - readPhc, [433](#)
 - writePhc, [433](#)
- format_plt.c
 - readPlt, [433](#)
 - writePlt, [433](#)
- format_rgb.c
 - readRgb, [434](#)
 - writeRgb, [434](#)
- format_sew.c
 - readSew, [434](#)
 - sewDecode, [434](#)
 - writeSew, [434](#)
- format_shv.c
 - readShv, [435](#)
 - shvDecode, [435](#)
 - shvDecodeShort, [435](#)
 - writeShv, [435](#)
- format_sst.c
 - readSst, [435](#)
 - writeSst, [436](#)
- format_stx.c
 - readStx, [436](#)
 - stxReadThread, [436](#)
 - writeStx, [436](#)
- format_svg.c
 - attributeList, [437](#)
 - current_element_id, [437](#)
 - currentAttribute, [437](#)
 - currentValue, [437](#)
 - n_attributes, [437](#)
 - readSvg, [437](#)
 - svgCreator, [437](#)
 - svgExpect, [437](#)
 - svgMultiValue, [437](#)
 - writeSvg, [437](#)
- format_t01.c
 - readT01, [438](#)
 - writeT01, [438](#)
- format_t09.c
 - readT09, [438](#)
 - writeT09, [438](#)
- format_tap.c
 - decode_tap_record_flags, [439](#)
 - encode_tap_record, [439](#)
 - readTap, [439](#)
 - writeTap, [439](#)
- format_thr.c
 - readThr, [439](#)
 - writeThr, [440](#)
- format_txt.c
 - readTxt, [440](#)
 - writeTxt, [440](#)
- format_u00.c
 - readU00, [440](#)
 - writeU00, [441](#)
- format_u01.c
 - readU01, [441](#)
 - writeU01, [441](#)
- format_vip.c
 - readVip, [442](#)
 - vipCompressData, [442](#)
 - vipDecodeByte, [442](#)
 - vipDecodeStitchType, [442](#)
 - vipDecodingTable, [442](#)
 - vipDecompressData, [442](#)
 - vipEncodeByte, [442](#)
 - vipEncodeStitchType, [442](#)
 - writeVip, [442](#)
- format_vp3.c
 - readVp3, [443](#)
 - vp3Decode, [443](#)
 - vp3DecodeInt16, [443](#)
 - vp3PatchByteCount, [443](#)
 - vp3ReadHoopSection, [443](#)
 - vp3ReadString, [444](#)
 - vp3WriteString, [444](#)
 - vp3WriteStringLen, [444](#)
 - writeVp3, [444](#)
- format_xxx.c
 - readXxx, [444](#)
 - writeXxx, [444](#)
 - xxxDecodeByte, [444](#)
 - xxxEncodeDesign, [445](#)
 - xxxEncodeStitch, [445](#)
 - xxxEncodeStop, [445](#)
- format_zsk.c
 - readZsk, [445](#)
 - writeZsk, [445](#)
- formatFilterOpen
 - MainWindow, [154](#)
- formatFilterSave
 - MainWindow, [154](#)
- formats.c
 - binaryWriteInt, [400](#)
 - binaryWriteIntBE, [400](#)
 - binaryWriteShort, [400](#)
 - binaryWriteUInt, [400](#)
 - binaryWriteUIntBE, [400](#)
 - binaryWriteUShort, [400](#)
 - binaryWriteUShortBE, [400](#)
 - emb_identify_format, [401](#)
 - embFormat_getExtension, [401](#)
 - embPattern_read, [401](#)
 - embPattern_readAuto, [401](#)
 - embPattern_write, [401](#)
 - embPattern_writeAuto, [401](#)
 - formatTable, [402](#)

- fpad, 402
- fread_int16, 402
- fread_int32_be, 402
- fread_uint16, 402
- imageWithFrame, 403
- safe_free, 402
- formatTable
 - embroidery.h, 336
 - formats.c, 402
- formatType
 - SaveObject, 186
- Fortron, 372, 417
- fourier_series
 - objects.cpp, 295
- fpad
 - embroidery_internal.h, 367
 - formats.c, 402
- fread_int16
 - embroidery_internal.h, 367
 - formats.c, 402
- fread_int32_be
 - embroidery_internal.h, 368
 - formats.c, 402
- fread_uint16
 - embroidery_internal.h, 368
 - formats.c, 402
- fromCenter
 - UndoableCommand, 204
- fromTransform
 - UndoableCommand, 204
- Fufu_Polyester
 - embroidery.h, 317
- Fufu_Rayon
 - embroidery.h, 317
- full_test_matrix
 - embroidery.h, 334
- FUNCTION_TYPE
 - embroidermodder.h, 226
- functions.c
 - degrees, 453
 - emb_round, 453
 - radians, 453
- fx, 9, 372, 417
- g
 - EmbColor_, 81
- general_props
 - settings-dialog.cpp, 298
- generate_dragon_curve
 - fill.c, 397
- Geometry, 102
 - ~Geometry, 109
 - allGripPoints, 109
 - arcEndPoint, 128
 - arcMidPoint, 128
 - arcStartPoint, 128
 - ArrowStyle, 107
 - arrowStyleAngle, 128
 - arrowStyleLength, 128
 - arrowStylePath, 128
 - boundingRect, 109
 - Box, 107
 - calculateArcData, 109
 - circle_click, 110
 - Closed, 107
 - curved, 128
 - Dot, 107
 - drawRubberLine, 110
 - filled, 129
 - findIndex, 110
 - Flared, 107
 - Fletching, 107
 - Geometry, 107–109
 - gripEdit, 110
 - gripIndex, 129
 - init, 111
 - init_arc, 111
 - init_circle, 111
 - init_ellipse, 111
 - init_line, 111
 - init_path, 112
 - init_point, 112
 - init_rect, 112
 - init_text_single, 112
 - lineStyle, 107
 - lineStyleAngle, 129
 - lineStyleLength, 129
 - lineStylePath, 129
 - lwtPen, 129
 - mouseSnapPoint, 112
 - NoArrow, 107
 - NoLine, 107
 - normalPath, 129
 - objectAngle, 112
 - objectArcLength, 113
 - objectArea, 113
 - objectBottomLeft, 113
 - objectBottomRight, 113
 - objectCenter, 113
 - objectChord, 113
 - objectCircumference, 113
 - objectClockwise, 114
 - objectCopyPath, 114
 - objectDelta, 114
 - objectDiameter, 114
 - objectDiameterMajor, 114
 - objectDiameterMinor, 114
 - objectEndAngle, 114
 - objectEndPoint, 114
 - objectEndPoint1, 115
 - objectEndPoint2, 115
 - objectHeight, 115
 - objectIncludedAngle, 115
 - objectLength, 115
 - objectLineType, 115
 - objectLineWeight, 116
 - objectMidPoint, 116

- objectPos, 116
- objectQuadrant0, 116
- objectQuadrant180, 116
- objectQuadrant270, 116
- objectQuadrant90, 116
- objectRadius, 116
- objectRadiusMajor, 116
- objectRadiusMinor, 116
- objectRubberPoint, 116
- objectRubberText, 117
- objectSavePath, 117
- objectSavePathList, 117
- objectStartAngle, 117
- objectStartPoint, 117
- objectTopLeft, 118
- objectTopRight, 118
- objectWidth, 118
- objectX, 118
- objectX1, 118
- objectX2, 118
- objectY, 118
- objectY1, 118
- objectY2, 118
- objID, 129
- objLine, 129
- objPen, 129
- objRubberMode, 129
- objRubberPoints, 130
- objRubberTexts, 130
- objText, 130
- objTextBackward, 130
- objTextFont, 130
- objTextJustify, 130
- objTextPath, 130
- objTextUpsideDown, 130
- Open, 107
- paint, 118
- properties, 130
- realRender, 119
- rect, 119
- script_click, 119
- script_context, 119
- script_main, 119
- script_prompt, 120
- setLine, 120
- setObjectArea, 120
- setObjectCenter, 120
- setObjectCenterX, 120
- setObjectCenterY, 120
- setObjectCircumference, 120
- setObjectDiameter, 122
- setObjectDiameterMajor, 122
- setObjectDiameterMinor, 122
- setObjectEndAngle, 122
- setObjectEndPoint, 122
- setObjectEndPoint1, 123
- setObjectEndPoint2, 123
- setObjectLineWeight, 123
- setObjectMidPoint, 123
- setObjectPos, 123, 124
- setObjectRadius, 124
- setObjectRadiusMajor, 124
- setObjectRadiusMinor, 124
- setObjectRect, 124
- setObjectSize, 124
- setObjectStartAngle, 124
- setObjectStartPoint, 125
- setObjectText, 125
- setObjectTextBackward, 125
- setObjectTextBold, 125
- setObjectTextFont, 125
- setObjectTextItalic, 125
- setObjectTextJustify, 125
- setObjectTextOverline, 125
- setObjectTextSize, 126
- setObjectTextStrikeOut, 126
- setObjectTextStyle, 126
- setObjectTextUnderline, 126
- setObjectTextUpsideDown, 126
- setObjectX, 126
- setObjectY, 126
- setRect, 126
- subPathList, 127
- Tick, 107
- Type, 130
- type, 127
- updateArcRect, 127
- updateLeader, 127
- updatePath, 127
- updateRubber, 128
- vulcanize, 128
- x_values, 130
- y_values, 131
- geometry
 - EmbArray_, 78
 - EmbLayer_, 90
 - EmbPattern_, 94
- geometry.c
 - embGeometry_boundingRect, 446
 - embGeometry_free, 446
 - embGeometry_init, 446
 - embGeometry_move, 446
 - embGeometry_vulcanize, 446
- get_bool
 - embroidermodder.h, 230
 - interface.cpp, 258
- get_int
 - embroidermodder.h, 230
 - interface.cpp, 258
- get_n_reals
 - interface.cpp, 258
- get_qstr
 - embroidermodder.h, 230
 - interface.cpp, 258
- get_real
 - embroidermodder.h, 230

- interface.cpp, 258
- get_str
 - embroidermodder.h, 230
 - interface.cpp, 258
- get_str_list
 - embroidermodder.h, 231
 - interface.cpp, 258
- get_trim_bounds
 - main.c, 469
- get_uint
 - embroidermodder.h, 231
 - interface.cpp, 258
- getArcCenter
 - arc.c, 450
 - embroidery.h, 334
- getArcDataFromBulge
 - arc.c, 450
 - embroidery.h, 335
- getCircleCircleIntersections
 - circle.c, 451
 - embroidery.h, 335
- getCircleTangentPoints
 - circle.c, 451
 - embroidery.h, 335
- getCurrentColor
 - MainWindow, 145
- getCurrentLayer
 - MainWindow, 145
- getCurrentLineType
 - MainWindow, 145
- getCurrentLineWeight
 - MainWindow, 145
- GetFile
 - embroidery_internal.h, 368
 - main.c, 469
- getFileSeparator
 - MainWindow, 145
- getInfo
 - EmbDetailsDialog, 83
- getShortCurrentFile
 - MdiWindow, 163
- getUndoStack
 - View, 211
- gnc, 9, 372, 418
- Gold Thread, 372, 419
- Great Notions, 372, 418
- greedy_algorithm
 - fill.c, 397
- GREEN_TERM_COLOR
 - embroidery_internal.h, 358
- gridColor
 - View, 216
- gridPath
 - View, 216
- gripBaseObj
 - View, 216
- gripColorCool
 - View, 216
- gripColorHot
 - View, 217
- gripEdit
 - Geometry, 110
- gripIndex
 - Geometry, 129
- grippingActive
 - View, 217
- gripSize
 - View, 217
- group_box_data
 - property-editor.cpp, 296
- group_box_types
 - property-editor.cpp, 296
- groupBoxes
 - embroidermodder.h, 237
 - mainwindow.cpp, 293
- gscene
 - MdiWindow, 167
 - SaveObject, 186
 - View, 217
- gt, 9, 372, 419
- gview
 - MdiWindow, 167
 - UndoableCommand, 204
- handleMoved
 - CmdPromptHandle, 61
- handlePressed
 - CmdPromptHandle, 61
- handleReleased
 - CmdPromptHandle, 61
- Happy, 439
- hashDeletedObjects
 - View, 217
- haveExtraDIFATSectors
 - main.c, 469
- header
 - _bcf_file, 43
 - em2_dev_script, 39
- height
 - _vp3Hoop, 47
 - EmblImage_, 89
- help_action
 - mainwindow.cpp, 280
- Hemingworth_Polyester
 - embroidery.h, 317
- hex_code
 - thread_color_, 200
- hideAllGroups
 - PropertyEditor, 172
- hideUnimplemented
 - MainWindow, 145
- hilbert_curve
 - embroidery.h, 335
 - fill.c, 397
- hilbert_curve_l_system
 - fill.c, 398
- historyAppended

- CmdPrompt, 55
- CmdPromptHistory, 64
- home
 - EmbPattern_, 94
- HOOP_110X110
 - embroidery_internal.h, 358
- HOOP_126X110
 - embroidery_internal.h, 358
- HOOP_140X200
 - embroidery_internal.h, 358
- HOOP_230X200
 - embroidery_internal.h, 358
- HOOP_50X50
 - embroidery_internal.h, 358
- hoop_height
 - EmbPattern_, 94
- hoop_padding, 131
 - bottom, 131
 - left, 131
 - right, 131
 - top, 131
- hoop_width
 - EmbPattern_, 94
- hoopSize
 - ThredHeader_, 201
- hoopX
 - ThredExtension_, 200
- hoopY
 - ThredExtension_, 200
- hour
 - EmbTime_, 101
- Huffman, 131
 - default_value, 132
 - lengths, 132
 - nlengths, 132
 - ntable, 132
 - table, 132
 - table_width, 132
- huffman
 - embroidery_internal.h, 361
- huffman_build_table
 - compress.c, 304
 - embroidery_internal.h, 368
- huffman_lookup
 - compress.c, 304
- huffman_lookup_data
 - compress.c, 304
- huffman_table_lookup
 - embroidery_internal.h, 368
- hus, 9, 419
- hus_compress
 - compress.c, 304
 - embroidery_internal.h, 368
- hus_decompress
 - compress.c, 304
 - embroidery_internal.h, 368
- hus_thread
 - embroidery.h, 317
- husCompressData
 - format_hus.c, 419
- husDecodeByte
 - format_hus.c, 419
- husDecodeStitchType
 - format_hus.c, 419
- husDecompressData
 - format_hus.c, 419
- husEncodeByte
 - format_hus.c, 419
- husEncodeStitchType
 - format_hus.c, 419
- Husqvarna Viking, 419, 435
- husThreads
 - embroidery.h, 336
 - thread-color.c, 477
- i
 - Node_, 168
- icon_action
 - mainwindow.cpp, 280
- iconDir
 - PropertyEditor, 174
 - UndoEditor, 206
- iconResize
 - MainWindow, 146
- iconSize
 - PropertyEditor, 175
 - UndoEditor, 206
- id
 - UndoableCommand, 203
- image.c
 - image_diff, 459
 - writelnImage, 460
- image_diff
 - image.c, 459
- ImageWidget, 132
 - ~ImageWidget, 133
 - ImageWidget, 133
 - img, 134
 - load, 133
 - paintEvent, 134
 - save, 134
- imageWithFrame
 - embroidery_internal.h, 384
 - formats.c, 403
- img
 - ImageWidget, 134
- imgWidget
 - PreviewDialog, 169
- inb, 9, 373, 420
- Inbro, 373, 420
- include_action
 - mainwindow.cpp, 281
- inf, 373, 421
- init
 - Geometry, 111
- init_action
 - mainwindow.cpp, 281

- init_arc
 - Geometry, 111
- init_circle
 - Geometry, 111
- init_ellipse
 - Geometry, 111
- init_line
 - Geometry, 111
- init_path
 - Geometry, 112
- init_point
 - Geometry, 112
- init_rect
 - Geometry, 112
- init_text_single
 - Geometry, 112
- input_data
 - Compress, 76
- input_length
 - Compress, 76
- INT_TYPE
 - embroidermodder.h, 226
- interface.cpp
 - add_to_path, 257
 - debug_message, 257
 - degrees__, 257
 - get_bool, 258
 - get_int, 258
 - get_n_reals, 258
 - get_qstr, 258
 - get_real, 258
 - get_str, 258
 - get_str_list, 258
 - get_uint, 258
 - make_checkbox, 258
 - make_spinbox, 259
 - make_ui_element, 259
 - node_bool, 259
 - node_int, 259
 - node_qstr, 259
 - node_real, 259
 - node_str, 260
 - node_str_list, 260
 - node_uint, 260
 - operator*, 260
 - operator+, 261
 - operator-, 261
 - radians__, 261
 - set_enabled, 261
 - set_visibility, 261
 - to_EmbVector, 262
 - to_qlist, 262
 - to_QPointF, 262
 - to_string_vector, 262
 - to_vector, 262
 - tokenize, 263
 - translate_str, 263
- is_int_action
 - mainwindow.cpp, 281
- Isacord_Polyester
 - embroidery.h, 317
- Isafil_Rayon
 - embroidery.h, 317
- isBlinking
 - CmdPromptInput, 73
- isCommandActive
 - MainWindow, 146
- isLwtEnabled
 - View, 211
- isRealEnabled
 - View, 211
- isShiftPressed
 - MainWindow, 146
- Janome, 421, 434
- jef, 9, 421
- jef_thread
 - embroidery.h, 317
- jefDecode
 - format_jef.c, 421
- jefEncode
 - format_jef.c, 421
- jefGetHoopSize
 - format_jef.c, 421
- jefSetHoopFromId
 - format_jef.c, 421
- jefThreads
 - embroidery.h, 336
 - thread-color.c, 477
- join_short_stitches
 - fill.c, 398
- JUMP
 - embroidery.h, 317
- ksm, 9, 422
- ksmEncode
 - format_ksm.c, 422
- L_system
 - embroidery.h, 321
- labels
 - embroidermodder.h, 237
 - mainwindow.cpp, 293
- labelTipOfTheDay
 - mainwindow.cpp, 293
- lastCmd
 - CmdPromptInput, 73
- layer
 - EmbPattern_, 94
- layer_manager_action
 - mainwindow.cpp, 281
- layer_previous_action
 - mainwindow.cpp, 281
- LayerManager, 134
 - ~LayerManager, 135
 - addLayer, 135
 - LayerManager, 135

- layerModel, 136
- layerModelSorted, 136
- treeView, 136
- layerModel
 - LayerManager, 136
- layerModelSorted
 - LayerManager, 136
- layerSelector
 - MainWindow, 154
- layerSelectorIndexChanged
 - MainWindow, 146
- layoutState
 - MainWindow, 154
- left
 - _vp3Hoop, 48
 - EmbRect_, 96
 - hoop_padding, 131
- left2
 - _vp3Hoop, 48
- leftBrush
 - SelectBox, 188
- leftBrushColor
 - SelectBox, 188
- leftPen
 - SelectBox, 188
- leftPenColor
 - SelectBox, 188
- leftSiblingId
 - _bcf_directory_entry, 41
- length
 - EmbArray_, 78
 - EmbSatinOutline_, 97
 - ThredHeader_, 201
- lengths
 - Huffman, 132
- LIBEMBROIDERY_EMBEDDED_VERSION
 - embroidery.h, 317
- lindenmayer_system
 - embroidery.h, 335
 - fill.c, 398
- line
 - EmbGeometry_, 87
- line.c
 - embLine_intersectionPoint, 454
 - embLine_normalVector, 454
 - embLine_toVector, 454
- lineEdits
 - embroidermodder.h, 237
 - mainwindow.cpp, 293
- lineStyle
 - Geometry, 107
- lineStyleAngle
 - Geometry, 129
- lineStyleLength
 - Geometry, 129
- lineStylePath
 - Geometry, 129
- LINETO
 - embroidery_internal.h, 358
- lineType
 - EmbGeometry_, 87
 - EmbLine_, 91
 - EmbPath_, 93
 - EmbPoint_, 95
- linetypeSelector
 - MainWindow, 154
- linetypeSelectorIndexChanged
 - MainWindow, 146
- lineweightSelector
 - MainWindow, 154
- lineweightSelectorIndexChanged
 - MainWindow, 146
- listMdiWin
 - MainWindow, 154
- load
 - ImageWidget, 133
- load_group_box_data_from_table
 - property-editor.cpp, 296
- loadFatFromSector
 - embroidery_internal.h, 369
 - main.c, 469
- loadFile
 - MdiWindow, 163
- loadFormats
 - MainWindow, 146
- loadRulerSettings
 - View, 211
- logPromptInput
 - MainWindow, 147
 - MdiWindow, 163
- LSYSTEM, 136
 - alphabet, 136
 - axiom, 137
 - constants, 137
 - rules, 137
- lwtPen
 - Geometry, 129
- Madeira_Polyester
 - embroidery.h, 317
- Madeira_Rayon
 - embroidery.h, 317
- magicCode
 - VipHeader_, 220
- main
 - embroidermodder.cpp, 221
- main.c
 - bcf_difat_create, 465
 - bcf_directory_free, 465
 - bcf_file_free, 465
 - bcfFile_read, 465
 - bcfFileFat_create, 465
 - bcfFileHeader_read, 465
 - binaryReadString, 465
 - binaryReadUnicodeString, 466
 - black_thread, 471
 - check_header_present, 466

CompoundFileDirectory, 466
CompoundFileDirectoryEntry, 466
copy_trim, 466
difatEntriesInHeader, 471
emb_error, 471
emb_optOut, 466
emb_readline, 467
emb_verbose, 471
embArc_print, 467
embColor_distance, 467
embColor_read, 467
embColor_write, 467
embConstantPi, 471
embSatinOutline_generateSatinOutline, 467
embSatinOutline_renderStitches, 467
embThread_findNearestColor, 468
embThread_findNearestThread, 468
embThread_getRandom, 468
embTime_initNow, 468
embTime_time, 468
embVector_print, 468
entriesInDifatSector, 469
FLAG_CIRCLE, 462
FLAG_CIRCLE_SHORT, 463
FLAG_COMBINE, 463
FLAG_CROSS_STITCH, 463
FLAG_ELLIPSE, 463
FLAG_ELLIPSE_SHORT, 463
FLAG_FILL, 463
FLAG_FILL_SHORT, 463
FLAG_FORMATS, 463
FLAG_FORMATS_SHORT, 463
FLAG_FULL_TEST_SUITE, 463
FLAG_HELP, 463
FLAG_HELP_SHORT, 463
FLAG_HILBERT_CURVE, 463
FLAG_LINE, 463
FLAG_LINE_SHORT, 463
FLAG_POLYGON, 463
FLAG_POLYGON_SHORT, 463
FLAG_POLYLINE, 463
FLAG_POLYLINE_SHORT, 464
FLAG_QUIET, 464
FLAG_QUIET_SHORT, 464
FLAG_RENDER, 464
FLAG_RENDER_SHORT, 464
FLAG_SATIN, 464
FLAG_SATIN_SHORT, 464
FLAG_SIERPINSKI_TRIANGLE, 464
FLAG_SIMULATE, 464
FLAG_STITCH, 464
FLAG_STITCH_SHORT, 464
FLAG_TEST, 464
FLAG_TO, 464
FLAG_TO_SHORT, 464
FLAG_VERBOSE, 464
FLAG_VERBOSE_SHORT, 464
FLAG_VERSION, 464
FLAG_VERSION_SHORT, 464
get_trim_bounds, 469
GetFile, 469
haveExtraDIFATSectors, 469
loadFatFromSector, 469
NUM_FLAGS, 465
parseDIFATSectors, 469
parseDirectoryEntryName, 469
parseTime, 469
readFullSector, 470
readNextSector, 470
sectorSize, 470
seekToSector, 470
sizeofChainingEntryAtEndOfDifatSector, 471
sizeofDifatEntry, 471
sizeofDirectoryEntry, 471
sizeofFatEntry, 471
stringInArray, 470
WHITESPACE, 471
write_24bit, 470
mainWidget
 EmbDetailsDialog, 83
MainWindow, 137
 ~MainWindow, 141
 about, 141
 activeCommand, 141
 activeMdiWindow, 141
 activeUndoStack, 141
 buttonTipOfTheDayClicked, 142
 checkForUpdates, 142
 closeEvent, 142
 closeToolBar, 142
 colorSelector, 154
 colorSelectorIndexChanged, 142
 create_icon, 143
 create_toolbar, 143
 createAllActions, 143
 createAllMenus, 143
 createAllToolbars, 144
 cutCopyObjectList, 154
 deletePressed, 144
 docIndex, 154
 escapePressed, 144
 findMdiWindow, 144
 floatingChangedToolBar, 144
 formatFilterOpen, 154
 formatFilterSave, 154
 getCurrentColor, 145
 getCurrentLayer, 145
 getCurrentLineType, 145
 getCurrentLineWeight, 145
 getFileSeparator, 145
 hideUnimplemented, 145
 iconResize, 146
 isCommandActive, 146
 isShiftPressed, 146
 layerSelector, 154
 layerSelectorIndexChanged, 146

layoutState, 154
linetypeSelector, 154
linetypeSelectorIndexChanged, 146
lineweightSelector, 154
lineweightSelectorIndexChanged, 146
listMdiWin, 154
loadFormats, 146
logPromptInput, 147
MainWindow, 141
myFileSeparator, 155
newFile, 147
numOfDocs, 155
onCloseMdiWin, 147
onCloseWindow, 147
onWindowActivated, 147
openFile, 147
openFilesSelected, 148
openrecentfile, 148
pickAddModeToggled, 148
platformString, 148
promptHistoryAppended, 148
promptInputNext, 149
promptInputPrevious, 149
quit, 149
recentMenuAboutToShow, 149
resizeEvent, 149
saveasfile, 149
savefile, 149
setShiftPressed, 149
setShiftReleased, 150
setTextFont, 150
setTextSize, 150
settingsPrompt, 150
setUndoCleanIcon, 150
shiftKeyPressedState, 155
stub_testing, 150
textFontSelector, 155
textFontSelectorCurrentFontChanged, 151
textSizeSelector, 155
textSizeSelectorIndexChanged, 151
tipOfTheDay, 151
toggleGrid, 151
toggleLwt, 151
toggleRuler, 151
updateAllViewBackgroundColors, 151
updateAllViewCrossHairColors, 152
updateAllViewGridColors, 152
updateAllViewRulerColors, 152
updateAllViewScrollBars, 152
updateAllViewSelectBoxColors, 153
updateMenuToolbarStatusbar, 153
updatePickAddMode, 153
windowMenuAboutToShow, 153
windowMenuActivated, 153
mainwindow-menus.cpp
 create_menu, 263
mainwindow.cpp
 _mainWin, 292
 about_action, 271
 actionHash, 292
 activeScene, 271
 activeView, 271
 actuator, 271
 add_arc_action, 271
 add_circle_action, 272
 add_dim_leader_action, 272
 add_ellipse_action, 272
 add_geometry_action, 272
 add_horizontal_dimension_action, 272
 add_image_action, 272
 add_infinite_line_action, 272
 add_line_action, 272
 add_path_action, 273
 add_point_action, 273
 add_polygon_action, 273
 add_polyline_action, 273
 add_ray_action, 274
 add_rectangle_action, 274
 add_regular_polygon_action, 274
 add_rounded_rectangle_action, 274
 add_rubber_action, 274
 add_slot_action, 274
 add_text_multi_action, 275
 add_text_single_action, 275
 add_to_selection_action, 275
 add_triangle_action, 275
 add_vertical_dimension_action, 276
 alert_action, 276
 allow_rubber_action, 276
 append_history_action, 276
 append_prompt_history_action, 276
 blink_prompt_action, 277
 calculate_angle_action, 277
 calculate_distance_action, 277
 changelog_action, 277
 checkboxes, 292
 checkBoxTipOfTheDay, 292
 clear_rubber_action, 277
 clear_selection_action, 277
 comboBoxes, 292
 command_map, 292
 config, 292
 config_tables, 292
 construct_command, 278
 convert_args_to_type, 278
 copy_action, 278
 copy_selected_action, 278
 cut_action, 278
 cut_selected_action, 279
 day_vision_action, 279
 debug_action, 279
 delete_selected_action, 279
 design_details_action, 279
 dialog, 292
 disable_action, 279
 do_nothing_action, 279

dockPropEdit, 292
dockUndoEdit, 293
doubleSpinBoxes, 293
end_action, 280
error_action, 280
groupBoxes, 293
help_action, 280
icon_action, 280
include_action, 281
init_action, 281
is_int_action, 281
labels, 293
labelTipOfTheDay, 293
layer_manager_action, 281
layer_previous_action, 281
lineEdits, 293
make_layer_active_action, 282
mdiArea, 293
menuHash, 293
messagebox_action, 282
mirror_selected_action, 282
mouse_x_action, 282
mouse_y_action, 282
move_selected_action, 282
new_action, 282
night_vision_action, 283
no_argument_debug, 283
num_selected_action, 283
OBJ_LTYPE_CENTER, 269
OBJ_LTYPE_CONT, 269
OBJ_LTYPE_DOT, 269
OBJ_LTYPE_FISHBONE, 269
OBJ_LTYPE_HIDDEN, 269
OBJ_LTYPE_PHANTOM, 269
OBJ_LTYPE_RUNNING, 269
OBJ_LTYPE_SATIN, 269
OBJ_LTYPE_VALUES, 269
OBJ_LTYPE_ZIGZAG, 269
OBJ_LWT_01, 270
OBJ_LWT_02, 270
OBJ_LWT_03, 270
OBJ_LWT_04, 270
OBJ_LWT_05, 270
OBJ_LWT_06, 270
OBJ_LWT_07, 270
OBJ_LWT_08, 270
OBJ_LWT_09, 270
OBJ_LWT_10, 270
OBJ_LWT_11, 270
OBJ_LWT_12, 270
OBJ_LWT_13, 270
OBJ_LWT_14, 270
OBJ_LWT_15, 270
OBJ_LWT_16, 270
OBJ_LWT_17, 270
OBJ_LWT_18, 270
OBJ_LWT_19, 270
OBJ_LWT_20, 270
OBJ_LWT_21, 270
OBJ_LWT_22, 270
OBJ_LWT_23, 270
OBJ_LWT_24, 270
OBJ_LWT_BYBLOCK, 270
OBJ_LWT_BYLAYER, 270
OBJ_LWT_DEFAULT, 270
OBJ_LWT_VALUES, 269
OBJ_SNAP_APPINTERSECTION, 270
OBJ_SNAP_CENTER, 270
OBJ_SNAP_ENDPOINT, 270
OBJ_SNAP_EXTENSION, 270
OBJ_SNAP_INSERTION, 270
OBJ_SNAP_INTERSECTION, 270
OBJ_SNAP_MIDPOINT, 270
OBJ_SNAP_NEAREST, 270
OBJ_SNAP_NODE, 270
OBJ_SNAP_NULL, 270
OBJ_SNAP_PARALLEL, 270
OBJ_SNAP_PERPENDICULAR, 270
OBJ_SNAP_QUADRANT, 270
OBJ_SNAP_TANGENT, 270
OBJ_SNAP_VALUES, 270
open_action, 283
pan_action, 283
paste_action, 284
paste_selected_action, 284
perpendicular_distance_action, 284
platform_action, 284
platformString, 284
preview_off_action, 284
preview_on_action, 285
print_action, 285
print_area_action, 285
prompt, 293
qsnap_x_action, 285
qsnap_y_action, 286
quit_action, 286
read_configuration, 286
read_string_list_setting, 286
read_string_setting, 286
redo_action, 286
rotate_selected_action, 287
rubber_action, 287
rubber_modes, 293
run_script, 287
run_script_file, 287
scale_selected_action, 287
scripts, 293
select_all_action, 287
set_background_color_action, 288
set_crosshair_color_action, 288
set_cursor_shape_action, 288
set_grid_color_action, 288
set_prompt_prefix_action, 289
set_rubber_filter_action, 289
set_rubber_mode_action, 289
set_rubber_point_action, 289

- set_rubber_text_action, 289
- SetRubberText, 289
- SetTextAngle_action, 289
- settings, 293
- settings_dialog_action, 289
- spare_rubber_action, 290
- spinBoxes, 293
- statusbar, 294
- subMenuHash, 294
- tip_of_the_day_action, 290
- todo_action, 290
- toolbarHash, 294
- toolButtons, 294
- undo_action, 290
- validFileFormat, 291
- validRGB, 291
- version_action, 291
- vulcanize_action, 291
- whats_this_action, 291
- window_action, 291
- wizardTipOfTheDay, 294
- zoom_action, 292
- majorVersion
 - _bcf_file_header, 45
- make_checkbox
 - embroidermodder.h, 231
 - interface.cpp, 258
- make_editing_copy
 - settings-dialog.cpp, 297
- make_layer_active_action
 - mainwindow.cpp, 282
- make_spinbox
 - embroidermodder.h, 231
 - interface.cpp, 259
- make_ui_element
 - embroidermodder.h, 231
 - interface.cpp, 259
- manufacturer_code
 - thread_color_, 200
- mapSignal
 - PropertyEditor, 173
- Marathon_Polyester
 - embroidery.h, 317
- Marathon_Rayon
 - embroidery.h, 317
- max, 423
- max_header
 - format_max.c, 423
- MAX_STITCHES
 - embroidery.h, 318
- MAX_THREADS
 - embroidery.h, 318
- maxNumberOfDirectoryEntries
 - _bcf_directory, 40
- MdiArea, 155
 - ~MdiArea, 157
 - bgColor, 159
 - bgLogo, 159
 - bgTexture, 159
 - cascade, 157
 - forceRepaint, 157
 - MdiArea, 156
 - mouseDoubleClickEvent, 157
 - paintEvent, 157
 - setBackgroundColor, 157
 - setBackgroundLogo, 158
 - setBackgroundTexture, 158
 - tile, 158
 - useBackgroundColor, 158
 - useBackgroundLogo, 158
 - useBackgroundTexture, 159
 - useColor, 159
 - useLogo, 159
 - useTexture, 159
 - zoomExtentsAllSubWindows, 159
- mdiArea
 - embroidermodder.h, 237
 - mainwindow.cpp, 293
 - MdiWindow, 167
- MdiWindow, 160
 - ~MdiWindow, 161
 - closeEvent, 162
 - curColor, 166
 - curFile, 166
 - curLayer, 166
 - curLineType, 167
 - curLineWeight, 167
 - currentColorChanged, 162
 - currentLayerChanged, 162
 - currentLinetypeChanged, 162
 - currentLineweightChanged, 162
 - deletePressed, 163
 - designDetails, 163
 - escapePressed, 163
 - fileWasLoaded, 167
 - getShortCurrentFile, 163
 - gscene, 167
 - gview, 167
 - loadFile, 163
 - logPromptInput, 163
 - mdiArea, 167
 - MdiWindow, 161
 - myIndex, 167
 - onWindowActivated, 163
 - print, 164
 - printer, 167
 - promptHistory, 167
 - promptHistoryAppended, 164
 - promptInputList, 167
 - promptInputNext, 164
 - promptInputNum, 167
 - promptInputPrevious, 164
 - promptInputPrevNext, 164
 - saveBMC, 164
 - saveFile, 165
 - sendCloseMdiWin, 165

- setCurrentFile, [165](#)
- setViewBackgroundColor, [165](#)
- setViewCrossHairColor, [165](#)
- setViewGridColor, [165](#)
- setViewRulerColor, [166](#)
- setViewSelectBoxColors, [166](#)
- showViewScrollBars, [166](#)
- sizeHint, [166](#)
- updateColorLinetypeLineweight, [166](#)
- mdiwindow.cpp
 - fileExtension, [294](#)
- Mega 2560 or another board with equal or, [16](#)
- Melco, [406](#), [410](#), [415](#), [425](#)
- menuHash
 - embroidermodder.h, [237](#)
 - mainwindow.cpp, [293](#)
- mergeWith
 - UndoableCommand, [203](#)
- messagebox_action
 - mainwindow.cpp, [282](#)
- Metro_Polyester
 - embroidery.h, [318](#)
- mid
 - EmbArc_, [77](#)
- miniSectorShift
 - _bcf_file_header, [45](#)
- miniStreamCutoffSize
 - _bcf_file_header, [45](#)
- minorVersion
 - _bcf_file_header, [45](#)
- minute
 - EmbTime_, [101](#)
- mirror
 - UndoableCommand, [203](#)
- mirror_selected_action
 - mainwindow.cpp, [282](#)
- mirrorLine
 - UndoableCommand, [204](#)
- mirrorSelected
 - View, [211](#)
- mit, [373](#), [424](#)
- mitDecodeStitch
 - embroidery_internal.h, [369](#)
 - encoding.c, [393](#)
- mitEncodeStitch
 - embroidery_internal.h, [369](#)
 - encoding.c, [393](#)
- Mitsubishi, [373](#), [424](#)
- modifiedTime
 - _bcf_directory_entry, [41](#)
- modifierName
 - ThredExtension_, [200](#)
- month
 - EmbTime_, [101](#)
- mouse_x_action
 - mainwindow.cpp, [282](#)
- mouse_y_action
 - mainwindow.cpp, [282](#)
- mouseDoubleClickEvent
 - MdiArea, [157](#)
 - View, [212](#)
- mouseMoveEvent
 - CmdPromptHandle, [61](#)
 - View, [212](#)
- mousePressEvent
 - CmdPromptHandle, [61](#)
 - View, [212](#)
- mouseReleaseEvent
 - CmdPromptHandle, [61](#)
 - View, [212](#)
- mouseSnapPoint
 - Geometry, [112](#)
- move_selected_action
 - mainwindow.cpp, [282](#)
- moveAction
 - View, [212](#)
- movePoint
 - View, [217](#)
- moveResizeHistory
 - CmdPromptSplitter, [74](#)
- moveSelected
 - View, [212](#)
- MOVETO
 - embroidery_internal.h, [358](#)
- moveY
 - CmdPromptHandle, [62](#)
- movingActive
 - View, [217](#)
- myFileSeparator
 - MainWindow, [155](#)
- myIndex
 - MdiWindow, [167](#)
- n_attributes
 - format_svg.c, [437](#)
- N_PES_VERSIONS
 - embroidery_internal.h, [358](#)
- name
 - EmblImage_, [89](#)
 - Emblayer_, [90](#)
 - SvgAttribute_, [199](#)
 - thread_color_, [200](#)
- navType
 - UndoableCommand, [204](#)
- negativeXHoopSize
 - VipHeader_, [220](#)
- negativeYHoopSize
 - VipHeader_, [220](#)
- new, [373](#), [424](#)
- new_action
 - mainwindow.cpp, [282](#)
- newFile
 - MainWindow, [147](#)
- next
 - _bcf_directory_entry, [41](#)
- night_vision_action
 - mainwindow.cpp, [283](#)

- nlengths
 - Huffman, 132
- no_argument_debug
 - mainwindow.cpp, 283
- NoArrow
 - Geometry, 107
- Node
 - embroidermodder.h, 226
- Node_, 168
 - b, 168
 - i, 168
 - r, 168
 - s, 168
 - sl, 168
 - type, 168
- node_bool
 - embroidermodder.h, 231
 - interface.cpp, 259
- node_int
 - embroidermodder.h, 231
 - interface.cpp, 259
- node_qstr
 - embroidermodder.h, 232
 - interface.cpp, 259
- node_real
 - embroidermodder.h, 232
 - interface.cpp, 259
- node_str
 - embroidermodder.h, 232
 - interface.cpp, 260
- node_str_list
 - embroidermodder.h, 232
 - interface.cpp, 260
- node_uint
 - embroidermodder.h, 232
 - interface.cpp, 260
- NodeList
 - embroidermodder.h, 227
- NoLine
 - Geometry, 107
- NORMAL
 - embroidery.h, 318
- normalPath
 - Geometry, 129
- ntable
 - Huffman, 132
- NUM_FLAGS
 - main.c, 465
- num_selected_action
 - mainwindow.cpp, 283
- numberOfBytesRemaining
 - _vp3Hoop, 48
- numberOfColors
 - _vp3Hoop, 48
 - VipHeader_, 220
- numberOfDifatSectors
 - _bcf_file_header, 45
- numberOfDirectorySectors
 - _bcf_file_header, 46
- numberOfEntriesInDifatSector
 - embroidery_internal.h, 369
- numberOfEntriesInFatSector
 - _bcf_file_fat, 44
- numberOfFATSectors
 - _bcf_file_header, 46
- numberOfFormats
 - embroidery.h, 318
- numberOfMiniFatSectors
 - _bcf_file_header, 46
- numberOfStitches
 - VipHeader_, 220
- numOfDocs
 - MainWindow, 155
- numSelected
 - View, 212
- numStiches
 - ThredHeader_, 201
- OBJ_COLOR
 - embroidermodder.h, 227
- OBJ_KEYS
 - embroidermodder.h, 227
- OBJ_LAYER
 - embroidermodder.h, 227
- OBJ_LTYPE
 - embroidermodder.h, 227
- OBJ_LTYPE_CENTER
 - mainwindow.cpp, 269
- OBJ_LTYPE_CONT
 - mainwindow.cpp, 269
- OBJ_LTYPE_DOT
 - mainwindow.cpp, 269
- OBJ_LTYPE_FISHBONE
 - mainwindow.cpp, 269
- OBJ_LTYPE_HIDDEN
 - mainwindow.cpp, 269
- OBJ_LTYPE_PHANTOM
 - mainwindow.cpp, 269
- OBJ_LTYPE_RUNNING
 - mainwindow.cpp, 269
- OBJ_LTYPE_SATIN
 - mainwindow.cpp, 269
- OBJ_LTYPE_VALUES
 - mainwindow.cpp, 269
- OBJ_LTYPE_ZIGZAG
 - mainwindow.cpp, 269
- OBJ_LWT
 - embroidermodder.h, 227
- OBJ_LWT_01
 - mainwindow.cpp, 270
- OBJ_LWT_02
 - mainwindow.cpp, 270
- OBJ_LWT_03
 - mainwindow.cpp, 270
- OBJ_LWT_04
 - mainwindow.cpp, 270
- OBJ_LWT_05

- mainwindow.cpp, [270](#)
- OBJ_LWT_06
 - mainwindow.cpp, [270](#)
- OBJ_LWT_07
 - mainwindow.cpp, [270](#)
- OBJ_LWT_08
 - mainwindow.cpp, [270](#)
- OBJ_LWT_09
 - mainwindow.cpp, [270](#)
- OBJ_LWT_10
 - mainwindow.cpp, [270](#)
- OBJ_LWT_11
 - mainwindow.cpp, [270](#)
- OBJ_LWT_12
 - mainwindow.cpp, [270](#)
- OBJ_LWT_13
 - mainwindow.cpp, [270](#)
- OBJ_LWT_14
 - mainwindow.cpp, [270](#)
- OBJ_LWT_15
 - mainwindow.cpp, [270](#)
- OBJ_LWT_16
 - mainwindow.cpp, [270](#)
- OBJ_LWT_17
 - mainwindow.cpp, [270](#)
- OBJ_LWT_18
 - mainwindow.cpp, [270](#)
- OBJ_LWT_19
 - mainwindow.cpp, [270](#)
- OBJ_LWT_20
 - mainwindow.cpp, [270](#)
- OBJ_LWT_21
 - mainwindow.cpp, [270](#)
- OBJ_LWT_22
 - mainwindow.cpp, [270](#)
- OBJ_LWT_23
 - mainwindow.cpp, [270](#)
- OBJ_LWT_24
 - mainwindow.cpp, [270](#)
- OBJ_LWT_BYBLOCK
 - mainwindow.cpp, [270](#)
- OBJ_LWT_BYLAYER
 - mainwindow.cpp, [270](#)
- OBJ_LWT_DEFAULT
 - mainwindow.cpp, [270](#)
- OBJ_LWT_VALUES
 - mainwindow.cpp, [269](#)
- OBJ_NAME
 - embroidermodder.h, [227](#)
- OBJ_RUBBER
 - embroidermodder.h, [227](#)
- OBJ_SNAP_APPINTERSECTION
 - mainwindow.cpp, [270](#)
- OBJ_SNAP_CENTER
 - mainwindow.cpp, [270](#)
- OBJ_SNAP_ENDPOINT
 - mainwindow.cpp, [270](#)
- OBJ_SNAP_EXTENSION
 - mainwindow.cpp, [270](#)
- OBJ_SNAP_INSERTION
 - mainwindow.cpp, [270](#)
- OBJ_SNAP_INTERSECTION
 - mainwindow.cpp, [270](#)
- OBJ_SNAP_MIDPOINT
 - mainwindow.cpp, [270](#)
- OBJ_SNAP_NEAREST
 - mainwindow.cpp, [270](#)
- OBJ_SNAP_NODE
 - mainwindow.cpp, [270](#)
- OBJ_SNAP_NULL
 - mainwindow.cpp, [270](#)
- OBJ_SNAP_PARALLEL
 - mainwindow.cpp, [270](#)
- OBJ_SNAP_PERPENDICULAR
 - mainwindow.cpp, [270](#)
- OBJ_SNAP_QUADRANT
 - mainwindow.cpp, [270](#)
- OBJ_SNAP_TANGENT
 - mainwindow.cpp, [270](#)
- OBJ_SNAP_VALUES
 - mainwindow.cpp, [270](#)
- OBJ_TYPE
 - embroidermodder.h, [227](#)
- OBJ_TYPE_ARC
 - embroidermodder.h, [227](#)
- OBJ_TYPE_BASE
 - embroidermodder.h, [227](#)
- OBJ_TYPE_BLOCK
 - embroidermodder.h, [227](#)
- OBJ_TYPE_CIRCLE
 - embroidermodder.h, [227](#)
- OBJ_TYPE_DIMALIGNED
 - embroidermodder.h, [227](#)
- OBJ_TYPE_DIMANGULAR
 - embroidermodder.h, [227](#)
- OBJ_TYPE_DIMARCLENGTH
 - embroidermodder.h, [227](#)
- OBJ_TYPE_DIMDIAMETER
 - embroidermodder.h, [227](#)
- OBJ_TYPE_DIMLEADER
 - embroidermodder.h, [227](#)
- OBJ_TYPE_DIMLINEAR
 - embroidermodder.h, [227](#)
- OBJ_TYPE_DIMORDINATE
 - embroidermodder.h, [227](#)
- OBJ_TYPE_DIMRADIUS
 - embroidermodder.h, [227](#)
- OBJ_TYPE_ELLIPSE
 - embroidermodder.h, [228](#)
- OBJ_TYPE_ELLIPSEARC
 - embroidermodder.h, [228](#)
- OBJ_TYPE_GRID
 - embroidermodder.h, [228](#)
- OBJ_TYPE_HATCH
 - embroidermodder.h, [228](#)
- OBJ_TYPE_IMAGE

- embroidermodder.h, 228
- OBJ_TYPE_INFINITELINE
 - embroidermodder.h, 228
- OBJ_TYPE_LINE
 - embroidermodder.h, 228
- OBJ_TYPE_NULL
 - embroidermodder.h, 227
- OBJ_TYPE_PATH
 - embroidermodder.h, 228
- OBJ_TYPE_POINT
 - embroidermodder.h, 228
- OBJ_TYPE_POLYGON
 - embroidermodder.h, 228
- OBJ_TYPE_POLYLINE
 - embroidermodder.h, 228
- OBJ_TYPE_RAY
 - embroidermodder.h, 228
- OBJ_TYPE_RECTANGLE
 - embroidermodder.h, 228
- OBJ_TYPE_RUBBER
 - embroidermodder.h, 228
- OBJ_TYPE_SLOT
 - embroidermodder.h, 228
- OBJ_TYPE_SPLINE
 - embroidermodder.h, 228
- OBJ_TYPE_TEXTMULTI
 - embroidermodder.h, 228
- OBJ_TYPE_TEXTSINGLE
 - embroidermodder.h, 228
- OBJ_TYPE_UNKNOWN
 - embroidermodder.h, 228
- OBJ_TYPE_VALUES
 - embroidermodder.h, 227
- object
 - EmbGeometry_, 87
 - UndoableCommand, 204
- object_names
 - property-editor.cpp, 296
- objectAngle
 - Geometry, 112
- objectArcLength
 - Geometry, 113
- objectArea
 - Geometry, 113
- objectBottomLeft
 - Geometry, 113
- objectBottomRight
 - Geometry, 113
- objectCenter
 - Geometry, 113
- objectChord
 - Geometry, 113
- objectCircumference
 - Geometry, 113
- objectClockwise
 - Geometry, 114
- objectCopyPath
 - Geometry, 114
- objectDelta
 - Geometry, 114
- objectDiameter
 - Geometry, 114
- objectDiameterMajor
 - Geometry, 114
- objectDiameterMinor
 - Geometry, 114
- objectEndAngle
 - Geometry, 114
- objectEndPoint
 - Geometry, 114
- objectEndPoint1
 - Geometry, 115
- objectEndPoint2
 - Geometry, 115
- objectHeight
 - Geometry, 115
- objectIncludedAngle
 - Geometry, 115
- objectLength
 - Geometry, 115
- objectLineType
 - Geometry, 115
- objectLineWeight
 - Geometry, 116
- objectMidPoint
 - Geometry, 116
- objectPos
 - Geometry, 116
- objectQuadrant0
 - Geometry, 116
- objectQuadrant180
 - Geometry, 116
- objectQuadrant270
 - Geometry, 116
- objectQuadrant90
 - Geometry, 116
- objectRadius
 - Geometry, 116
- objectRadiusMajor
 - Geometry, 116
- objectRadiusMinor
 - Geometry, 116
- objectRubberPoint
 - Geometry, 116
- objectRubberText
 - Geometry, 117
- objects.cpp
 - add_polyline, 295
 - closest_point, 295
 - fourier_series, 295
 - rotate_vector, 295
- objectSavePath
 - Geometry, 117
- objectSavePathList
 - Geometry, 117
- objectStartAngle

- Geometry, [117](#)
- objectStartPoint
 - Geometry, [117](#)
- objectTopLeft
 - Geometry, [118](#)
- objectTopRight
 - Geometry, [118](#)
- objectType
 - _bcf_directory_entry, [41](#)
- ObjectTypeRootEntry
 - embroidery_internal.h, [358](#)
- ObjectTypeStorage
 - embroidery_internal.h, [358](#)
- ObjectTypeStream
 - embroidery_internal.h, [358](#)
- ObjectTypeUnknown
 - embroidery_internal.h, [358](#)
- objectWidth
 - Geometry, [118](#)
- objectX
 - Geometry, [118](#)
- objectX1
 - Geometry, [118](#)
- objectX2
 - Geometry, [118](#)
- objectY
 - Geometry, [118](#)
- objectY1
 - Geometry, [118](#)
- objectY2
 - Geometry, [118](#)
- objID
 - Geometry, [129](#)
- objLine
 - Geometry, [129](#)
- objPen
 - Geometry, [129](#)
- objRubberMode
 - Geometry, [129](#)
- objRubberPoints
 - Geometry, [130](#)
- objRubberTexts
 - Geometry, [130](#)
- objText
 - Geometry, [130](#)
- objTextBackward
 - Geometry, [130](#)
- objTextFont
 - Geometry, [130](#)
- objTextJustify
 - Geometry, [130](#)
- objTextPath
 - Geometry, [130](#)
- objTextUpsideDown
 - Geometry, [130](#)
- ofm, [425](#)
- ofmDecode
 - format_ofm.c, [425](#)
- ofmReadBlockHeader
 - format_ofm.c, [425](#)
- ofmReadClass
 - format_ofm.c, [425](#)
- ofmReadColorChange
 - format_ofm.c, [425](#)
- ofmReadExpanded
 - format_ofm.c, [425](#)
- ofmReadLibrary
 - format_ofm.c, [425](#)
- ofmReadThreads
 - format_ofm.c, [425](#)
- onCloseMdiWin
 - MainWindow, [147](#)
- onCloseWindow
 - MainWindow, [147](#)
- onWindowActivated
 - MainWindow, [147](#)
 - MdiWindow, [163](#)
- Open
 - Geometry, [107](#)
- open_action
 - mainwindow.cpp, [283](#)
- openFile
 - MainWindow, [147](#)
- openFilesSelected
 - MainWindow, [148](#)
- openrecentfile
 - MainWindow, [148](#)
- opensave_props
 - settings-dialog.cpp, [299](#)
- operator*
 - embroidermodder.h, [233](#)
 - interface.cpp, [260](#)
- operator+
 - embroidermodder.h, [233](#)
 - interface.cpp, [261](#)
- operator-
 - embroidermodder.h, [233](#)
 - interface.cpp, [261](#)
- originPath
 - View, [217](#)
- paint
 - Geometry, [118](#)
- paintEvent
 - ImageWidget, [134](#)
 - MdiArea, [157](#)
 - SelectBox, [187](#)
- pan_action
 - mainwindow.cpp, [283](#)
- panDistance
 - View, [217](#)
- panDown
 - View, [212](#)
- panLeft
 - View, [212](#)
- panningActive
 - View, [217](#)

- panningPointActive
 - View, 217
- panningRealTimeActive
 - View, 217
- panPoint
 - View, 212
- panRealTime
 - View, 212
- panRight
 - View, 212
- panStart
 - View, 212
- panStartX
 - View, 217
- panStartY
 - View, 217
- Pantone
 - embroidery.h, 318
- panUp
 - View, 212
- parseDIFATSectors
 - main.c, 469
- parseDirectoryEntryName
 - main.c, 469
- parseTime
 - main.c, 469
- paste
 - View, 213
- paste_action
 - mainwindow.cpp, 284
- paste_selected_action
 - mainwindow.cpp, 284
- pasteClip
 - CmdPromptInput, 71
- pasteDelta
 - View, 217
- pasteObjectItemGroup
 - View, 217
- pastePressed
 - CmdPrompt, 55
 - CmdPromptInput, 71
- pastingActive
 - View, 217
- path
 - EmbGeometry_, 87
 - EmblImage_, 89
- pattern.c
 - convert, 472
 - embPattern_addCircleAbs, 472
 - embPattern_addEllipseAbs, 472
 - embPattern_addLineAbs, 473
 - embPattern_addPathAbs, 473
 - embPattern_addPointAbs, 473
 - embPattern_addPolygonAbs, 473
 - embPattern_addPolylineObjectAbs, 473
 - embPattern_addRectAbs, 473
 - embPattern_addStitchAbs, 473
 - embPattern_addStitchRel, 473
 - embPattern_addThread, 473
 - embPattern_calcBoundingBox, 474
 - embPattern_center, 474
 - embPattern_changeColor, 474
 - embPattern_color_count, 474
 - embPattern_combineJumpStitches, 474
 - embPattern_copyPolylinesToStitch_list, 474
 - embPattern_copystitch_listToPolylines, 474
 - embPattern_correctForMaxStitchLength, 474
 - embPattern_create, 474
 - embPattern_designDetails, 474
 - embPattern_end, 475
 - embPattern_fixColorCount, 475
 - embPattern_flip, 475
 - embPattern_flipHorizontal, 475
 - embPattern_flipVertical, 475
 - embPattern_free, 475
 - embPattern_hideStitchesOverLength, 475
 - embPattern_jumpStitches, 475
 - embPattern_lengthHistogram, 475
 - embPattern_loadExternalColorFile, 475
 - embPattern_maximumStitchLength, 475
 - embPattern_minimumStitchLength, 475
 - embPattern_movePolylinesToStitch_list, 476
 - embPattern_movestitch_listToPolylines, 476
 - embPattern_realStitches, 476
 - embPattern_scale, 476
 - embPattern_totalStitchLength, 476
 - embPattern_trimStitches, 476
- pcd, 9, 374, 426
- pcm, 9, 374, 426
- pcm_thread
 - embroidery.h, 318
- pcmThreads
 - embroidery.h, 337
 - thread-color.c, 477
- pcq, 9, 374, 427
- pcs, 9, 374, 427
- pec, 9, 374, 428
- pec_thread
 - embroidery.h, 318
- pecEncode
 - format_pec.c, 428
- pecEncodeJump
 - format_pec.c, 428
- pecEncodeStop
 - format_pec.c, 428
- pecThreadCount
 - embroidery.h, 337
 - thread-color.c, 477
- pecThreads
 - embroidery.h, 337
 - thread-color.c, 478
- pel, 9, 374, 429
- pem, 9, 375, 429
- perpendicular_distance_action
 - mainwindow.cpp, 284
- pes, 9, 432

- PES0001
 - embroidery_internal.h, [358](#)
- PES0020
 - embroidery_internal.h, [358](#)
- PES0022
 - embroidery_internal.h, [358](#)
- PES0030
 - embroidery_internal.h, [358](#)
- PES0040
 - embroidery_internal.h, [359](#)
- PES0050
 - embroidery_internal.h, [359](#)
- PES0055
 - embroidery_internal.h, [359](#)
- PES0056
 - embroidery_internal.h, [359](#)
- PES0060
 - embroidery_internal.h, [359](#)
- PES0070
 - embroidery_internal.h, [359](#)
- PES0080
 - embroidery_internal.h, [359](#)
- PES0090
 - embroidery_internal.h, [359](#)
- PES0100
 - embroidery_internal.h, [359](#)
- pes_version
 - format_pes.c, [432](#)
- pes_version_strings
 - format_pes.c, [432](#)
- pesWriteEmbOneSection
 - format_pes.c, [430](#)
- pesWriteSewSegSection
 - format_pes.c, [430](#)
- Pfaff, [337](#), [374](#), [376](#), [422](#), [423](#), [426](#), [427](#), [438](#), [443](#), [444](#)
- pfaffDecode
 - embroidery_internal.h, [369](#)
 - encoding.c, [393](#)
- pfaffEncode
 - embroidery_internal.h, [369](#)
 - encoding.c, [394](#)
- phb, [9](#), [375](#), [432](#)
- phc, [9](#), [375](#), [433](#)
- pickAdd
 - PropertyEditor, [175](#)
- pickAddModeToggled
 - MainWindow, [148](#)
 - PropertyEditor, [173](#)
- pickBoxSize
 - View, [217](#)
- pivot
 - UndoableCommand, [204](#)
- platform_action
 - mainwindow.cpp, [284](#)
- platformString
 - MainWindow, [148](#)
 - mainwindow.cpp, [284](#)
- plt, [376](#), [433](#)
- point
 - EmbGeometry_, [87](#)
- pointList
 - EmbPath_, [93](#)
- polygon
 - EmbGeometry_, [87](#)
- polyline
 - EmbGeometry_, [88](#)
- position
 - EmbAlignedDim_, [76](#)
 - EmbAngularDim_, [77](#)
 - EmbArcLengthDim_, [78](#)
 - EmbBlock_, [80](#)
 - EmbDiameterDim_, [84](#)
 - EmbImage_, [89](#)
 - EmbInfiniteLine_, [90](#)
 - EmbLeaderDim_, [91](#)
 - EmbLinearDim_, [92](#)
 - EmbOrdinateDim_, [92](#)
 - EmbPoint_, [95](#)
 - EmbRadiusDim_, [95](#)
 - EmbRay_, [96](#)
 - EmbTextMulti_, [99](#)
 - EmbTextSingle_, [100](#)
- positiveXHoopSize
 - VipHeader_, [220](#)
- positiveYHoopSize
 - VipHeader_, [220](#)
- precisionAngle
 - PropertyEditor, [175](#)
- precisionLength
 - PropertyEditor, [175](#)
- prefix
 - CmdPromptInput, [73](#)
- pressPoint
 - View, [218](#)
- pressResizeHistory
 - CmdPromptSplitter, [74](#)
- pressY
 - CmdPromptHandle, [62](#)
- preview
 - settings-dialog.cpp, [299](#)
- preview_off_action
 - mainwindow.cpp, [284](#)
- preview_on_action
 - mainwindow.cpp, [285](#)
- previewActive
 - View, [218](#)
- previewData
 - View, [218](#)
- PreviewDialog, [169](#)
 - ~PreviewDialog, [169](#)
 - imgWidget, [169](#)
 - PreviewDialog, [169](#)
- previewMode
 - View, [218](#)
- previewObjectItemGroup
 - View, [218](#)

- previewObjectList
 - View, 218
- previewOff
 - View, 213
- previewOn
 - View, 213
- previewPoint
 - View, 218
- print
 - MdiWindow, 164
- print_action
 - mainwindow.cpp, 285
- print_area_action
 - mainwindow.cpp, 285
- printArcResults
 - embroidery_internal.h, 369
- printer
 - MdiWindow, 167
- privacy_policy.md, 478
- processInput
 - CmdPromptInput, 71
- prompt
 - embroidermodder.h, 237
 - mainwindow.cpp, 293
- prompt_props
 - settings-dialog.cpp, 299
- promptDivider
 - CmdPrompt, 59
- promptHistory
 - CmdPrompt, 59
 - MdiWindow, 167
- promptHistoryAppended
 - MainWindow, 148
 - MdiWindow, 164
- promptInput
 - CmdPrompt, 59
- promptInputList
 - MdiWindow, 167
- promptInputNext
 - MainWindow, 149
 - MdiWindow, 164
- promptInputNum
 - MdiWindow, 167
- promptInputPrevious
 - MainWindow, 149
 - MdiWindow, 164
- promptInputPrevNext
 - MdiWindow, 164
- promptSplitter
 - CmdPrompt, 59
- promptVBoxLayout
 - CmdPrompt, 59
- properties
 - Geometry, 130
- property-editor.cpp
 - comboBoxTextSingleFont, 296
 - fieldNewText, 296
 - fieldNoText, 296
 - fieldOffText, 296
 - fieldOldText, 296
 - fieldOnText, 296
 - fieldVariesText, 296
 - fieldYesText, 296
 - group_box_data, 296
 - group_box_types, 296
 - load_group_box_data_from_table, 296
 - object_names, 296
- PropertyEditor, 170
 - ~PropertyEditor, 171
 - clearAllFields, 171
 - comboBoxSelected, 174
 - createComboBoxSelected, 171
 - createGroupBox, 172
 - createLineEdit, 172
 - createToolButton, 172
 - createToolButtonPickAdd, 172
 - createToolButtonQSelect, 172
 - eventFilter, 172
 - fieldEdited, 172
 - focusWidget, 174
 - hideAllGroups, 172
 - iconDir, 174
 - iconSize, 175
 - mapSignal, 173
 - pickAdd, 175
 - pickAddModeToggled, 173
 - precisionAngle, 175
 - precisionLength, 175
 - PropertyEditor, 171
 - propertyEditorButtonStyle, 175
 - selectedItemList, 175
 - setSelectedItems, 173
 - showGroups, 173
 - showOneType, 173
 - signalMapper, 175
 - togglePickAddMode, 173
 - toolButtonPickAdd, 175
 - toolButtonQSelect, 175
 - updateComboBoxBoolIfVaries, 173
 - updateComboBoxStrIfVaries, 174
 - updateFontComboBoxStrIfVaries, 174
 - updateLineEditNumIfVaries, 174
 - updateLineEditStrIfVaries, 174
 - updatePickAddModeButton, 174
- propertyEditorButtonStyle
 - PropertyEditor, 175
- qsnap_x_action
 - mainwindow.cpp, 285
- qsnap_y_action
 - mainwindow.cpp, 286
- qSnapActive
 - View, 218
- qsnapApertureSize
 - View, 218
- qsnapLocatorColor
 - View, 218

- qsnapLocatorSize
 - View, 218
- qSnapToggle
 - View, 218
- QUADTOCONTROL
 - embroidery_internal.h, 359
- QUADTOEND
 - embroidery_internal.h, 359
- quick_snap_props
 - settings-dialog.cpp, 299
- quit
 - MainWindow, 149
- quit_action
 - mainwindow.cpp, 286
- r
 - EmbColor_, 81
 - Node_, 168
- radians
 - embroidery.h, 335
 - functions.c, 453
- radians__
 - embroidermodder.h, 233
 - interface.cpp, 261
- radius
 - EmbCircle_, 80
 - EmbEllipse_, 85
 - EmbRect_, 96
- rapidFireEnabled
 - CmdPromptInput, 73
- rapidMoveActive
 - View, 218
- read100
 - embroidery_internal.h, 370
 - format_100.c, 403
- read10o
 - embroidery_internal.h, 370
 - format_10o.c, 404
- read_configuration
 - embroidermodder.h, 233
 - mainwindow.cpp, 286
- read_hoop
 - format_jef.c, 421
- read_settings
 - embroidermodder.h, 233
 - settings-dialog.cpp, 298
- read_string_list_setting
 - mainwindow.cpp, 286
- read_string_setting
 - embroidermodder.h, 233
 - mainwindow.cpp, 286
- readArt
 - embroidery_internal.h, 370
 - format_art.c, 404
- readBmc
 - embroidery_internal.h, 370
 - format_bmc.c, 405
- readBro
 - embroidery_internal.h, 370
- format_bro.c, 405
- readCnd
 - embroidery_internal.h, 370
 - format_cnd.c, 406
- readCol
 - embroidery_internal.h, 370
 - format_col.c, 407
- readCsd
 - embroidery_internal.h, 370
 - format_csd.c, 408
- readCsv
 - embroidery_internal.h, 370
 - format_csv.c, 409
- readDat
 - embroidery_internal.h, 370
 - format_dat.c, 409
- readDem
 - embroidery_internal.h, 370
 - format_dem.c, 410
- readDescriptions
 - embroidery_internal.h, 371
 - format_pes.c, 430
- readDsb
 - embroidery_internal.h, 371
 - format_dsb.c, 411
- readDst
 - embroidery_internal.h, 371
 - format_dst.c, 413
- readDsz
 - embroidery_internal.h, 371
 - format_dsz.c, 413
- readDxf
 - embroidery_internal.h, 371
 - format_dxf.c, 414
- readEdr
 - embroidery_internal.h, 371
 - format_edr.c, 414
- readEmd
 - embroidery_internal.h, 371
 - format_emd.c, 415
- reader_state
 - EmbFormatList_, 86
- readExp
 - embroidery_internal.h, 371
 - format_exp.c, 415
- readExy
 - embroidery_internal.h, 371
 - format_exy.c, 416
- readEys
 - embroidery_internal.h, 371
 - format_eyes.c, 416
- readFeatherPatterns
 - embroidery_internal.h, 372
 - format_pes.c, 430
- readFullSector
 - embroidery_internal.h, 372
 - main.c, 470
- readFxy

embroidery_internal.h, 372
format_fxy.c, 417
readGc
embroidery_internal.h, 372
format_gc.c, 417
readGnc
embroidery_internal.h, 372
format_gnc.c, 418
readGt
embroidery_internal.h, 372
format_gt.c, 418
readHoopName
embroidery_internal.h, 372
format_pes.c, 430
readHus
embroidery_internal.h, 372
format_hus.c, 420
readImageString
embroidery_internal.h, 372
format_pes.c, 431
readInb
embroidery_internal.h, 373
format_inb.c, 420
readInf
embroidery_internal.h, 373
format_inf.c, 420
readJef
embroidery_internal.h, 373
format_jef.c, 422
readKsm
embroidery_internal.h, 373
format_ksm.c, 422
readLine
format_dxf.c, 414
readMax
embroidery_internal.h, 373
format_max.c, 423
readMit
embroidery_internal.h, 373
format_mit.c, 423
readMotifPatterns
embroidery_internal.h, 373
format_pes.c, 431
readNew
embroidery_internal.h, 373
format_new.c, 424
readNextSector
embroidery_internal.h, 373
main.c, 470
readOfm
embroidery_internal.h, 373
format_ofm.c, 425
readPcd
embroidery_internal.h, 374
format_pcd.c, 426
readPcm
embroidery_internal.h, 374
format_pcm.c, 426
readPcq
embroidery_internal.h, 374
format_pcq.c, 427
readPcs
embroidery_internal.h, 374
format_pcs.c, 427
readPec
embroidery_internal.h, 374
format_pec.c, 428
readPecStitches
embroidery_internal.h, 374
format_pec.c, 428
readPel
embroidery_internal.h, 374
format_pel.c, 429
readPem
embroidery_internal.h, 374
format_pem.c, 429
readPes
embroidery_internal.h, 375
format_pes.c, 431
readPESHeaderV10
embroidery_internal.h, 375
format_pes.c, 431
readPESHeaderV5
embroidery_internal.h, 375
format_pes.c, 431
readPESHeaderV6
embroidery_internal.h, 375
format_pes.c, 431
readPESHeaderV7
embroidery_internal.h, 375
format_pes.c, 431
readPESHeaderV8
embroidery_internal.h, 375
format_pes.c, 431
readPESHeaderV9
embroidery_internal.h, 375
format_pes.c, 431
readPhb
embroidery_internal.h, 375
format_phb.c, 432
readPhc
embroidery_internal.h, 375
format_phc.c, 433
readPlt
embroidery_internal.h, 375
format_plt.c, 433
readProgrammableFills
embroidery_internal.h, 376
format_pes.c, 431
readRgb
embroidery_internal.h, 376
format_rgb.c, 434
readSew
embroidery_internal.h, 376
format_sew.c, 434
readShv

- embroidery_internal.h, 376
- format_shv.c, 435
- readSst
 - embroidery_internal.h, 376
 - format_sst.c, 435
- readStx
 - embroidery_internal.h, 376
 - format_stx.c, 436
- readSvg
 - embroidery_internal.h, 376
 - format_svg.c, 437
- readT01
 - embroidery_internal.h, 376
 - format_t01.c, 438
- readT09
 - embroidery_internal.h, 376
 - format_t09.c, 438
- readTap
 - embroidery_internal.h, 376
 - format_tap.c, 439
- readThr
 - embroidery_internal.h, 377
 - format_thr.c, 439
- readThreads
 - embroidery_internal.h, 377
 - format_pes.c, 431
- readTxt
 - embroidery_internal.h, 377
 - format_txt.c, 440
- readU00
 - embroidery_internal.h, 377
 - format_u00.c, 440
- readU01
 - embroidery_internal.h, 377
 - format_u01.c, 441
- readVip
 - embroidery_internal.h, 377
 - format_vip.c, 442
- readVp3
 - embroidery_internal.h, 377
 - format_vp3.c, 443
- readXxx
 - embroidery_internal.h, 377
 - format_xxx.c, 444
- readZsk
 - embroidery_internal.h, 377
 - format_zsk.c, 445
- REAL_TYPE
 - embroidermodder.h, 226
- realRender
 - Geometry, 119
- recalculateLimits
 - View, 213
- recentMenuAboutToShow
 - MainWindow, 149
- rect
 - EmbGeometry_, 88
 - Geometry, 119
- rect.c
 - embRect_area, 455
 - embRect_init, 455
- RED_TERM_COLOR
 - embroidery_internal.h, 359
- redo
 - UndoableCommand, 203
 - UndoEditor, 205
- redo_action
 - mainwindow.cpp, 286
- redoPressed
 - CmdPrompt, 55
 - CmdPromptInput, 71
- redoText
 - UndoEditor, 205
- rejectChanges
 - Settings_Dialog, 196
- releasePoint
 - View, 218
- releaseResizeHistory
 - CmdPromptSplitter, 75
- releaseY
 - CmdPromptHandle, 62
- repeatAction
 - View, 213
- report
 - embroidery.h, 335
- reserved
 - ThredExtension_, 200
 - ThredHeader_, 201
- reserved1
 - _bcf_file_header, 46
- reserved2
 - _bcf_file_header, 46
- RESET_TERM_COLOR
 - embroidery_internal.h, 359
- resizeEvent
 - MainWindow, 149
- resizeHistory
 - CmdPromptHistory, 64
- reverse_byte_order
 - encoding.c, 394
- rgb, 9, 376, 434
- right
 - _vp3Hoop, 48
 - EmbRect_, 97
 - hoop_padding, 131
- right2
 - _vp3Hoop, 48
- rightBrush
 - SelectBox, 188
- rightBrushColor
 - SelectBox, 188
- rightPen
 - SelectBox, 188
- rightPenColor
 - SelectBox, 188
- rightSiblingId

- _bcf_directory_entry, 41
- RobisonAnton_Polyester
 - embroidery.h, 318
- RobisonAnton_Rayon
 - embroidery.h, 318
- rotate
 - UndoableCommand, 203
- rotate_selected_action
 - mainwindow.cpp, 287
- rotate_vector
 - embroidermodder.h, 234
 - objects.cpp, 295
- rotateAction
 - View, 213
- rotateSelected
 - View, 213
- rotation
 - EmbEllipse_, 85
 - EmbRect_, 97
- roundToMultiple
 - View, 213
- rubber_action
 - mainwindow.cpp, 287
- rubber_modes
 - mainwindow.cpp, 293
- rubberRoomList
 - View, 218
- rulerColor
 - View, 218
- rulerMetric
 - View, 218
- rulerPixelSize
 - View, 218
- rules
 - fill.c, 398
 - LSYSTEM, 137
- run_script
 - embroidermodder.h, 234
 - mainwindow.cpp, 287
- run_script_file
 - embroidermodder.h, 234
 - mainwindow.cpp, 287
- runCommand
 - CmdPrompt, 55
 - CmdPromptInput, 71
- s
 - em2_dev_script, 39
 - Node_, 168
- safe_free
 - embroidery_internal.h, 377
 - formats.c, 402
- save
 - ImageWidget, 134
 - SaveObject, 185
- save_points_to_pattern
 - fill.c, 398
- saveasfile
 - MainWindow, 149
- saveBMC
 - MdiWindow, 164
- saveFile
 - MdiWindow, 165
- savefile
 - MainWindow, 149
- saveHistory
 - CmdPrompt, 56
- SaveObject, 176
 - ~SaveObject, 177
 - addArc, 178
 - addBlock, 178
 - addCircle, 178
 - addDimAligned, 178
 - addDimAngular, 179
 - addDimArcLength, 179
 - addDimDiameter, 179
 - addDimLeader, 179
 - addDimLinear, 180
 - addDimOrdinate, 180
 - addDimRadius, 180
 - addEllipse, 181
 - addEllipseArc, 181
 - addGrid, 181
 - addHatch, 181
 - addImage, 182
 - addInfiniteLine, 182
 - addLine, 182
 - addPath, 182
 - addPoint, 183
 - addPolygon, 183
 - addPolyline, 183
 - addRay, 184
 - addRectangle, 184
 - addSlot, 184
 - addSpline, 184
 - addTextMulti, 185
 - addTextSingle, 185
 - formatType, 186
 - gscene, 186
 - save, 185
 - SaveObject, 177
 - toPolyline, 185
- scale_selected_action
 - mainwindow.cpp, 287
- scaleAction
 - View, 213
- scaleSelected
 - View, 213
- sceneGripPoint
 - View, 219
- sceneMousePoint
 - View, 219
- sceneMovePoint
 - View, 219
- scenePressPoint
 - View, 219
- sceneReleasePoint

- View, 219
- script_click
 - Geometry, 119
- script_context
 - Geometry, 119
- script_main
 - Geometry, 119
- script_prompt
 - Geometry, 120
- scripts
 - embroidermodder.h, 237
 - mainwindow.cpp, 293
- second
 - EmbTime_, 101
- sectionName
 - StxThread_, 198
- sectorShift
 - _bcf_file_header, 46
- sectorSize
 - _bcf_file_difat, 43
 - main.c, 470
- seekToSector
 - main.c, 470
- select_all_action
 - mainwindow.cpp, 287
- selectAll
 - View, 213
- selectAllPressed
 - CmdPrompt, 56
 - CmdPromptInput, 71
- SelectBox, 186
 - alpha, 188
 - boxDir, 188
 - dirBrush, 188
 - dirPen, 188
 - forceRepaint, 187
 - leftBrush, 188
 - leftBrushColor, 188
 - leftPen, 188
 - leftPenColor, 188
 - paintEvent, 187
 - rightBrush, 188
 - rightBrushColor, 188
 - rightPen, 188
 - rightPenColor, 188
 - SelectBox, 187
 - setColors, 188
 - setDirection, 188
- selectBox
 - View, 219
- selected_items
 - View, 213
- selectedItemList
 - PropertyEditor, 175
- selectingActive
 - View, 219
- selectionChanged
 - View, 213
- sendCloseMdiWin
 - MdiWindow, 165
- SEQUIN
 - embroidery.h, 318
- set_background_color_action
 - mainwindow.cpp, 288
- set_crosshair_color_action
 - mainwindow.cpp, 288
- set_cursor_shape_action
 - mainwindow.cpp, 288
- set_dst_variable
 - format_dst.c, 413
- set_enabled
 - embroidermodder.h, 234
 - interface.cpp, 261
- set_grid_color_action
 - mainwindow.cpp, 288
- set_object_color
 - arc.c, 450
- set_prompt_prefix_action
 - mainwindow.cpp, 289
- set_rubber_filter_action
 - mainwindow.cpp, 289
- set_rubber_mode_action
 - mainwindow.cpp, 289
- set_rubber_point_action
 - mainwindow.cpp, 289
- set_rubber_text_action
 - mainwindow.cpp, 289
- set_visibility
 - embroidermodder.h, 235
 - interface.cpp, 261
- setBackgroundColor
 - MdiArea, 157
 - View, 214
- setBackgroundLogo
 - MdiArea, 158
- setBackgroundTexture
 - MdiArea, 158
- setColors
 - SelectBox, 188
- setCornerButton
 - View, 214
- setCrossHairColor
 - View, 214
- setCrossHairSize
 - View, 214
- setCurrentFile
 - MdiWindow, 165
- setCurrentText
 - CmdPrompt, 56
- setDirection
 - SelectBox, 188
- setGridColor
 - View, 214
- setHistory
 - CmdPrompt, 56
- setLine

- Geometry, [120](#)
- setMainWin
 - Application, [50](#)
- setMouseCoord
 - StatusBar, [197](#)
- setObjectArea
 - Geometry, [120](#)
- setObjectCenter
 - Geometry, [120](#)
- setObjectCenterX
 - Geometry, [120](#)
- setObjectCenterY
 - Geometry, [120](#)
- setObjectCircumference
 - Geometry, [120](#)
- setObjectDiameter
 - Geometry, [122](#)
- setObjectDiameterMajor
 - Geometry, [122](#)
- setObjectDiameterMinor
 - Geometry, [122](#)
- setObjectEndAngle
 - Geometry, [122](#)
- setObjectEndPoint
 - Geometry, [122](#)
- setObjectEndPoint1
 - Geometry, [123](#)
- setObjectEndPoint2
 - Geometry, [123](#)
- setObjectLineWeight
 - Geometry, [123](#)
- setObjectMidPoint
 - Geometry, [123](#)
- setObjectPos
 - Geometry, [123](#), [124](#)
- setObjectRadius
 - Geometry, [124](#)
- setObjectRadiusMajor
 - Geometry, [124](#)
- setObjectRadiusMinor
 - Geometry, [124](#)
- setObjectRect
 - Geometry, [124](#)
- setObjectSize
 - Geometry, [124](#)
- setObjectStartAngle
 - Geometry, [124](#)
- setObjectStartPoint
 - Geometry, [125](#)
- setObjectText
 - Geometry, [125](#)
- setObjectTextBackward
 - Geometry, [125](#)
- setObjectTextBold
 - Geometry, [125](#)
- setObjectTextFont
 - Geometry, [125](#)
- setObjectTextItalic
 - Geometry, [125](#)
- setObjectTextJustify
 - Geometry, [125](#)
- setObjectTextOverline
 - Geometry, [125](#)
- setObjectTextSize
 - Geometry, [126](#)
- setObjectTextStrikeOut
 - Geometry, [126](#)
- setObjectTextStyle
 - Geometry, [126](#)
- setObjectTextUnderline
 - Geometry, [126](#)
- setObjectTextUpsideDown
 - Geometry, [126](#)
- setObjectX
 - Geometry, [126](#)
- setObjectY
 - Geometry, [126](#)
- setPrefix
 - CmdPrompt, [56](#)
- setPromptBackgroundColor
 - CmdPrompt, [56](#)
- setPromptFontFamily
 - CmdPrompt, [57](#)
- setPromptFontSize
 - CmdPrompt, [57](#)
- setPromptFontStyle
 - CmdPrompt, [57](#)
- setPromptTextColor
 - CmdPrompt, [57](#)
- setRect
 - Geometry, [126](#)
- setRubberMode
 - View, [214](#)
- setRubberPoint
 - View, [214](#)
- SetRubberText
 - mainwindow.cpp, [289](#)
- setRubberText
 - View, [214](#)
- setRulerColor
 - View, [214](#)
- setSelectBoxColors
 - View, [214](#)
- setSelectedItems
 - PropertyEditor, [173](#)
- setShiftPressed
 - MainWindow, [149](#)
- setShiftReleased
 - MainWindow, [150](#)
- SetTextAngle_action
 - mainwindow.cpp, [289](#)
- setTextFont
 - MainWindow, [150](#)
- setTextSize
 - MainWindow, [150](#)
- settings

- embroidermodder.h, 237
- mainwindow.cpp, 293
- settings-dialog.cpp
 - accept_, 298
 - display_props, 298
 - extensions, 298
 - general_props, 298
 - make_editing_copy, 297
 - opensave_props, 299
 - preview, 299
 - prompt_props, 299
 - quick_snap_props, 299
 - read_settings, 298
 - write_settings, 298
- Settings_Dialog, 189
 - ~Settings_Dialog, 190
 - acceptChanges, 191
 - addColorsToComboBox, 191
 - buttonBox, 196
 - buttonCustomFilterClearAll, 191
 - buttonCustomFilterClearAllClicked, 191
 - buttonCustomFilterSelectAll, 191
 - buttonCustomFilterSelectAllClicked, 191
 - buttonQSnapClearAll, 191
 - buttonQSnapClearAllClicked, 191
 - buttonQSnapSelectAll, 191
 - buttonQSnapSelectAllClicked, 191
 - checkBoxCustomFilterStateChanged, 191
 - checkBoxGeneralMdiBGUseColorStateChanged, 191
 - checkBoxGeneralMdiBGUseLogoStateChanged, 191
 - checkBoxGeneralMdiBGUseTextureStateChanged, 191
 - checkBoxGridCenterOnOriginStateChanged, 192
 - checkBoxGridColorMatchCrossHairStateChanged, 192
 - checkBoxGridLoadFromFileStateChanged, 192
 - checkBoxLwtRealRenderStateChanged, 192
 - checkBoxLwtShowLwtStateChanged, 192
 - checkBoxPromptSaveHistoryAsHtmlStateChanged, 192
 - checkBoxRulerShowOnLoadStateChanged, 192
 - checkBoxShowScrollBarsStateChanged, 192
 - chooseDisplayBackgroundColor, 192
 - chooseDisplayCrossHairColor, 192
 - chooseDisplaySelectBoxLeftColor, 192
 - chooseDisplaySelectBoxLeftFill, 192
 - chooseDisplaySelectBoxRightColor, 193
 - chooseDisplaySelectBoxRightFill, 193
 - chooseGeneralMdiBackgroundColor, 193
 - chooseGeneralMdiBackgroundLogo, 193
 - chooseGeneralMdiBackgroundTexture, 193
 - chooseGridColor, 193
 - choosePromptBackgroundColor, 193
 - choosePromptTextColor, 193
 - chooseRulerColor, 193
 - comboBoxGridTypeCurrentIndexChanged, 193
 - comboBoxIconSizeCurrentIndexChanged, 193
 - comboBoxPromptFontFamilyCurrentIndexChanged, 194
 - comboBoxPromptFontStyleCurrentIndexChanged, 194
 - comboBoxQSnapLocatorColorCurrentIndexChanged, 194
 - comboBoxRulerMetricCurrentIndexChanged, 194
 - comboBoxScrollBarWidgetCurrentIndexChanged, 194
 - comboBoxSelectionCoolGripColorCurrentIndexChanged, 194
 - comboBoxSelectionHotGripColorCurrentIndexChanged, 194
 - create_checkbox, 194
 - create_float_spinbox, 194
 - createTabDisplay, 194
 - createTabFilePaths, 194
 - createTabGeneral, 195
 - createTabGridRuler, 195
 - createTabLineWeight, 195
 - createTabOpenSave, 195
 - createTabOrthoPolar, 195
 - createTabPrinting, 195
 - createTabPrompt, 195
 - createTabQuickSnap, 195
 - createTabQuickTrack, 195
 - createTabSelection, 195
 - createTabSnap, 195
 - currentDisplayBackgroundColorChanged, 195
 - currentDisplayCrossHairColorChanged, 195
 - currentDisplaySelectBoxLeftColorChanged, 195
 - currentDisplaySelectBoxLeftFillChanged, 195
 - currentDisplaySelectBoxRightColorChanged, 195
 - currentDisplaySelectBoxRightFillChanged, 196
 - currentGeneralMdiBackgroundColorChanged, 196
 - currentGridColorChanged, 196
 - currentPromptBackgroundColorChanged, 196
 - currentPromptTextColorChanged, 196
 - currentRulerColorChanged, 196
 - rejectChanges, 196
 - Settings_Dialog, 190
 - spinBoxDisplaySelectBoxAlphaValueChanged, 196
 - spinBoxPromptFontSizeValueChanged, 196
 - spinBoxRulerPixelSizeValueChanged, 196
 - tabWidget, 196
- settings_dialog_action
 - mainwindow.cpp, 289
- settingsPrompt
 - MainWindow, 150
- setUndoCleanIcon
 - MainWindow, 150
- setViewBackgroundColor
 - MdiWindow, 165
- setViewCrossHairColor
 - MdiWindow, 165
- setViewGridColor

- MdiWindow, 165
- setViewRulerColor
 - MdiWindow, 166
- setViewSelectBoxColors
 - MdiWindow, 166
- sew, 9, 434
- sewDecode
 - format_sew.c, 434
- shiftKeyPressedState
 - MainWindow, 155
- shiftPressed
 - CmdPrompt, 58
 - CmdPromptInput, 71
- shiftReleased
 - CmdPrompt, 58
 - CmdPromptInput, 71
- showGroups
 - PropertyEditor, 173
- showOneType
 - PropertyEditor, 173
- showScrollBars
 - View, 214
- showSettings
 - CmdPrompt, 58
 - CmdPromptInput, 72
- showViewScrollBars
 - MdiWindow, 166
- shv, 9, 435
- shv_thread
 - embroidery.h, 318
- shvDecode
 - format_shv.c, 435
- shvDecodeShort
 - format_shv.c, 435
- shvThreadCount
 - embroidery.h, 337
 - thread-color.c, 478
- shvThreads
 - embroidery.h, 337
 - thread-color.c, 478
- side1
 - EmbSatinOutline_, 97
- side2
 - EmbSatinOutline_, 97
- Sierra Expanded, 372, 417
- Sigma_Polyester
 - embroidery.h, 318
- signalMapper
 - PropertyEditor, 175
- signature
 - _bcf_file_header, 46
- sigVersion
 - ThredHeader_, 201
- Singer, 407, 445
- sizeHint
 - MdiWindow, 166
- sizeOfChainingEntryAtEndOfDifatSector
 - main.c, 471
- sizeOfDifatEntry
 - main.c, 471
- sizeOfDirectoryEntry
 - main.c, 471
- sizeOfFatEntry
 - main.c, 471
- sl
 - Node_, 168
- someInt
 - SubDescriptor_, 199
- someNum
 - SubDescriptor_, 199
- someOtherInt
 - SubDescriptor_, 199
- spare_rubber_action
 - mainwindow.cpp, 290
- spareRubber
 - View, 214
- spareRubberList
 - View, 219
- spinBoxDisplaySelectBoxAlphaValueChanged
 - Settings_Dialog, 196
- spinBoxes
 - embroidermodder.h, 238
 - mainwindow.cpp, 293
- spinBoxPromptFontSizeValueChanged
 - Settings_Dialog, 196
- spinBoxRulerPixelSizeValueChanged
 - Settings_Dialog, 196
- spline
 - EmbGeometry_, 88
- sst, 9, 376, 436
- start
 - EmbArc_, 77
 - EmbBezier_, 79
 - EmbLine_, 91
- startBlinking
 - CmdPrompt, 58
- startCommand
 - CmdPrompt, 58
 - CmdPromptInput, 72
- startGripping
 - View, 214
- startingSectorLocation
 - _bcf_directory_entry, 42
- startResizeHistory
 - CmdPromptHistory, 65
- state
 - View, 219
- stateBits
 - _bcf_directory_entry, 42
- StatusBar, 197
 - buttons, 198
 - context_menu_action, 197
 - context_menu_event, 197
 - setMouseCoord, 197
 - StatusBar, 197
 - statusBarMouseCoord, 198

- toggle, 197
- statusbar
 - embroidermodder.h, 238
 - mainwindow.cpp, 294
- statusBarMouseCoord
 - StatusBar, 198
- stitch
 - EmbArray_, 79
 - EmbGeometry_, 88
- stitch_list
 - EmbPattern_, 94
- stitchesJump
 - EmbDetailsDialog, 83
- stitchesReal
 - EmbDetailsDialog, 84
- stitchesTotal
 - EmbDetailsDialog, 84
- stitchesTrim
 - EmbDetailsDialog, 84
- stitchGranularity
 - ThredExtension_, 200
- STOP
 - embroidery.h, 318
- stopBlinking
 - CmdPrompt, 58
 - CmdPromptInput, 72
- stopGripping
 - View, 215
- stopResizeHistory
 - CmdPromptHistory, 65
- streamSize
 - _bcf_directory_entry, 42
- streamSizeHigh
 - _bcf_directory_entry, 42
- String
 - embroidermodder.h, 227
- STRING_LIST_TYPE
 - embroidermodder.h, 226
- STRING_TYPE
 - embroidermodder.h, 226
- stringInArray
 - embroidery_internal.h, 378
 - main.c, 470
- StringList
 - embroidermodder.h, 227
- stringVal
 - VipHeader_, 220
- stub_testing
 - MainWindow, 150
- stx, 436
- stxColor
 - StxThread_, 198
- stxReadThread
 - format_stx.c, 436
- StxThread
 - embroidery_internal.h, 361
- StxThread_, 198
 - colorCode, 198
 - sectionName, 198
 - stxColor, 198
 - subDescriptors, 198
- styleHash
 - CmdPrompt, 59
- SubDescriptor
 - embroidery_internal.h, 361
- SubDescriptor_, 198
 - colorCode, 199
 - colorName, 199
 - someInt, 199
 - someNum, 199
 - someOtherInt, 199
- subDescriptors
 - StxThread_, 198
- subMenuHash
 - embroidermodder.h, 238
 - mainwindow.cpp, 294
- subPathList
 - Geometry, 127
- Sulky_Rayon
 - embroidery.h, 318
- Sunstar, 376, 436
- svg, 9, 437
- SVG_ATTRIBUTE
 - embroidery_internal.h, 359
- SVG_CATCH_ALL
 - embroidery_internal.h, 359
- SVG_Colors
 - embroidery.h, 318
- SVG_CREATOR_EMBROIDERMODDER
 - embroidery_internal.h, 359
- SVG_CREATOR_ILLUSTRATOR
 - embroidery_internal.h, 359
- SVG_CREATOR_INKSCAPE
 - embroidery_internal.h, 359
- SVG_CREATOR_NULL
 - embroidery_internal.h, 360
- SVG_ELEMENT
 - embroidery_internal.h, 360
- SVG_EXPECT_ATTRIBUTE
 - embroidery_internal.h, 360
- SVG_EXPECT_ELEMENT
 - embroidery_internal.h, 360
- SVG_EXPECT_NULL
 - embroidery_internal.h, 360
- SVG_EXPECT_VALUE
 - embroidery_internal.h, 360
- SVG_MEDIA_PROPERTY
 - embroidery_internal.h, 360
- SVG_NULL
 - embroidery_internal.h, 360
- SVG_PROPERTY
 - embroidery_internal.h, 360
- SvgAttribute
 - embroidery_internal.h, 361
- SvgAttribute_, 199

- name, 199
 - value, 199
- svgCreator
 - format_svg.c, 437
- svgExpect
 - format_svg.c, 437
- svgMultiValue
 - format_svg.c, 437
- t01, 376, 438
- t09, 376, 438
- table
 - Huffman, 132
- table_width
 - Huffman, 132
- tabPressed
 - CmdPrompt, 58
 - CmdPromptInput, 72
- tabWidget
 - Settings_Dialog, 196
- Tajima, 411
- tap, 9, 439
- tempBaseObj
 - View, 219
- testEmbCircle
 - embroidery_internal.h, 378
- testEmbCircle_2
 - embroidery_internal.h, 378
- testEmbFormat
 - embroidery_internal.h, 378
- testGeomArc
 - embroidery_internal.h, 378
- testMain
 - embroidery.h, 336
- testTangentPoints
 - embroidery_internal.h, 378
- testThreadColor
 - embroidery_internal.h, 378
- text
 - EmbTextMulti_, 99
 - EmbTextSingle_, 100
- text.c
 - textSingle_gripEdit, 455
 - textSingle_mouseSnapPoint, 455
 - textSingle_paint, 456
 - textSingle_setJustify, 456
 - textSingle_setTextBackward, 456
 - textSingle_setTextBold, 456
 - textSingle_setTextFont, 456
 - textSingle_setTextItalic, 456
 - textSingle_setTextOverline, 456
 - textSingle_setTextSize, 456
 - textSingle_setTextStrikeOut, 456
 - textSingle_setTextStyle, 456
 - textSingle_setTextUnderline, 456
 - textSingle_setTextUpsideDown, 456
 - textSingle_updateRubber, 456
- textFontSelector
 - MainWindow, 155
- textFontSelectorCurrentFontChanged
 - MainWindow, 151
- textSingle_gripEdit
 - text.c, 455
- textSingle_mouseSnapPoint
 - text.c, 455
- textSingle_paint
 - text.c, 456
- textSingle_setJustify
 - text.c, 456
- textSingle_setTextBackward
 - text.c, 456
- textSingle_setTextBold
 - text.c, 456
- textSingle_setTextFont
 - text.c, 456
- textSingle_setTextItalic
 - text.c, 456
- textSingle_setTextOverline
 - text.c, 456
- textSingle_setTextSize
 - text.c, 456
- textSingle_setTextStrikeOut
 - text.c, 456
- textSingle_setTextStyle
 - text.c, 456
- textSingle_setTextUnderline
 - text.c, 456
- textSingle_setTextUpsideDown
 - text.c, 456
- textSingle_updateRubber
 - text.c, 456
- textSizeSelector
 - MainWindow, 155
- textSizeSelectorIndexChanged
 - MainWindow, 151
- thr, 377, 440
- thread
 - EmbArray_, 79
 - EmbGeometry_, 88
- thread-color.c
 - _dxfColorTable, 477
 - brand_codes, 477
 - brand_codes_files, 477
 - husThreads, 477
 - jefThreads, 477
 - pcmThreads, 477
 - pecThreadCount, 477
 - pecThreads, 478
 - shvThreadCount, 478
 - shvThreads, 478
 - threadColor, 477
 - threadColorName, 477
 - threadColorNum, 477
- thread_color
 - embroidery.h, 321
- thread_color_, 199
- hex_code, 200

- manufacturer_code, 200
- name, 200
- thread_list
 - EmbPattern_, 94
- ThreadArt_Polyester
 - embroidery.h, 318
- ThreadArt_Rayon
 - embroidery.h, 318
- threadColor
 - embroidery.h, 336
 - thread-color.c, 477
- threadColorName
 - embroidery.h, 336
 - thread-color.c, 477
- threadColorNum
 - embroidery.h, 336
 - thread-color.c, 477
- ThreaDelight_Polyester
 - embroidery.h, 319
- threadLength
 - _vp3Hoop, 48
- ThreadWorks, 377, 440
- ThredExtension
 - embroidery_internal.h, 361
- ThredExtension_, 200
 - auxFormat, 200
 - creatorName, 200
 - hoopX, 200
 - hoopY, 200
 - modifierName, 200
 - reserved, 200
 - stitchGranularity, 200
- ThredHeader
 - embroidery_internal.h, 361
- ThredHeader_, 201
 - hoopSize, 201
 - length, 201
 - numStiches, 201
 - reserved, 201
 - sigVersion, 201
- threshold_method
 - fill.c, 398
- Tick
 - Geometry, 107
- tile
 - MdiArea, 158
- tip_of_the_day_action
 - mainwindow.cpp, 290
- tipOfTheDay
 - MainWindow, 151
- tmpHeight
 - CmdPromptHistory, 65
- to_EmbVector
 - embroidermodder.h, 235
 - interface.cpp, 262
- to_qlist
 - embroidermodder.h, 235
 - interface.cpp, 262
- to_QPointF
 - embroidermodder.h, 235
 - interface.cpp, 262
- to_string_vector
 - embroidermodder.h, 235
 - interface.cpp, 262
- to_vector
 - embroidermodder.h, 235
 - interface.cpp, 262
- toCenter
 - UndoableCommand, 204
- todo_action
 - mainwindow.cpp, 290
- toggle
 - StatusBar, 197
- toggleGrid
 - MainWindow, 151
 - View, 215
- toggleLwt
 - MainWindow, 151
 - View, 215
- toggleOrtho
 - View, 215
- togglePickAddMode
 - PropertyEditor, 173
- togglePolar
 - View, 215
- toggleQSnap
 - View, 215
- toggleQTrack
 - View, 215
- toggleReal
 - View, 215
- toggleRuler
 - MainWindow, 151
 - View, 215
- toggleSnap
 - View, 215
- tokenize
 - embroidermodder.h, 236
 - interface.cpp, 263
- toolbarHash
 - embroidermodder.h, 238
 - mainwindow.cpp, 294
- toolButtonPickAdd
 - PropertyEditor, 175
- toolButtonQSelect
 - PropertyEditor, 175
- toolButtons
 - embroidermodder.h, 238
 - mainwindow.cpp, 294
- top
 - _vp3Hoop, 48
 - EmbRect_, 97
 - hoop_padding, 131
- top2
 - _vp3Hoop, 48
- toPolyline

- SaveObject, 185
- toTransform
 - UndoableCommand, 204
- Toyota, 403
- transactionSignatureNumber
 - _bcf_file_header, 46
- translate_str
 - embroidermodder.h, 236
 - interface.cpp, 263
- treeView
 - LayerManager, 136
- TRIM
 - embroidery.h, 319
- txt, 377, 440
- Type
 - Geometry, 130
- type
 - EmbArray_, 79
 - EmbFormatList_, 86
 - EmbGeometry_, 88
 - Geometry, 127
 - Node_, 168
- u00, 377, 441
- u01, 9, 377, 441
- undo
 - UndoableCommand, 203
 - UndoEditor, 205
- undo_action
 - mainwindow.cpp, 290
- UndoableCommand, 201
 - after, 203
 - angle, 203
 - before, 203
 - command, 203
 - delta, 204
 - done, 204
 - factor, 204
 - fromCenter, 204
 - fromTransform, 204
 - gview, 204
 - id, 203
 - mergeWith, 203
 - mirror, 203
 - mirrorLine, 204
 - navType, 204
 - object, 204
 - pivot, 204
 - redo, 203
 - rotate, 203
 - toCenter, 204
 - toTransform, 204
 - undo, 203
 - UndoableCommand, 202, 203
- UndoEditor, 204
 - ~UndoEditor, 205
 - addStack, 205
 - canRedo, 205
 - canUndo, 205
 - focusWidget, 206
 - iconDir, 206
 - iconSize, 206
 - redo, 205
 - redoText, 205
 - undo, 205
 - UndoEditor, 205
 - undoGroup, 206
 - undoText, 205
 - undoView, 206
 - updateCleanIcon, 205
- undoGroup
 - UndoEditor, 206
- undoPressed
 - CmdPrompt, 58
 - CmdPromptInput, 72
- undoStack
 - View, 219
- undoText
 - UndoEditor, 205
- undoView
 - UndoEditor, 206
- unknown
 - VipHeader_, 220
- unknown2
 - _vp3Hoop, 48
- unknown3
 - _vp3Hoop, 48
- unknown4
 - _vp3Hoop, 49
- UNKNOWN_TYPE
 - embroidermodder.h, 226
- updateAllViewBackgroundColors
 - MainWindow, 151
- updateAllViewCrossHairColors
 - MainWindow, 152
- updateAllViewGridColors
 - MainWindow, 152
- updateAllViewRulerColors
 - MainWindow, 152
- updateAllViewScrollBars
 - MainWindow, 152
- updateAllViewSelectBoxColors
 - MainWindow, 153
- updateArcRect
 - Geometry, 127
- updateCleanIcon
 - UndoEditor, 205
- updateColorLinetypeLineweight
 - MdiWindow, 166
- updateComboBoxBoolIfVaries
 - PropertyEditor, 173
- updateComboBoxStrIfVaries
 - PropertyEditor, 174
- updateCurrentText
 - CmdPromptInput, 72
- updateFontComboBoxStrIfVaries
 - PropertyEditor, 174

- updateLeader
 - Geometry, [127](#)
- updateLineEditNumIfVaries
 - PropertyEditor, [174](#)
- updateLineEditStrIfVaries
 - PropertyEditor, [174](#)
- updateMenuToolBarStatusBar
 - MainWindow, [153](#)
- updateMouseCoords
 - View, [215](#)
- updatePath
 - Geometry, [127](#)
- updatePickAddMode
 - MainWindow, [153](#)
- updatePickAddModeButton
 - PropertyEditor, [174](#)
- updateRubber
 - Geometry, [128](#)
- updateStyle
 - CmdPrompt, [58](#)
- upPressed
 - CmdPrompt, [58](#)
 - CmdPromptInput, [72](#)
- usage
 - embroidermodder.cpp, [222](#)
- useBackgroundColor
 - MdiArea, [158](#)
- useBackgroundLogo
 - MdiArea, [158](#)
- useBackgroundTexture
 - MdiArea, [159](#)
- useColor
 - MdiArea, [159](#)
- useLogo
 - MdiArea, [159](#)
- useTexture
 - MdiArea, [159](#)
- validFileFormat
 - embroidermodder.h, [236](#)
 - mainwindow.cpp, [291](#)
- validRGB
 - mainwindow.cpp, [291](#)
- value
 - SvgAttribute_, [199](#)
- vector
 - EmbGeometry_, [88](#)
- vector.c
 - embVector_add, [457](#)
 - embVector_angle, [457](#)
 - embVector_average, [457](#)
 - embVector_cross, [457](#)
 - embVector_distance, [458](#)
 - embVector_dot, [458](#)
 - embVector_length, [458](#)
 - embVector_multiply, [458](#)
 - embVector_normalize, [458](#)
 - embVector_relativeX, [458](#)
 - embVector_relativeY, [459](#)
 - embVector_subtract, [459](#)
 - embVector_transpose_product, [459](#)
 - embVector_unit, [459](#)
- VECTOR_TYPE
 - embroidermodder.h, [226](#)
- version
 - embroidermodder.cpp, [222](#)
- version_action
 - mainwindow.cpp, [291](#)
- View, [206](#)
 - ~View, [209](#)
 - addObject, [209](#)
 - addToRubberRoom, [209](#)
 - alignScenePointWithViewPoint, [209](#)
 - allowRubber, [210](#)
 - allowZoomIn, [210](#)
 - allowZoomOut, [210](#)
 - center, [210](#)
 - centerAt, [210](#)
 - clearRubberRoom, [210](#)
 - clearSelection, [210](#)
 - contextMenuEvent, [210](#)
 - copy, [210](#)
 - copySelected, [210](#)
 - cornerButtonClicked, [210](#)
 - createGrid, [210](#)
 - createGridIso, [210](#)
 - createGridPolar, [210](#)
 - createGridRect, [210](#)
 - createObjectList, [210](#)
 - createOrigin, [211](#)
 - createRulerTextPath, [211](#)
 - crosshairColor, [216](#)
 - crosshairSize, [216](#)
 - cut, [211](#)
 - cutCopyMousePoint, [216](#)
 - deleteObject, [211](#)
 - deletePressed, [211](#)
 - deleteSelected, [211](#)
 - drawBackground, [211](#)
 - drawForeground, [211](#)
 - enterEvent, [211](#)
 - escapePressed, [211](#)
 - getUndoStack, [211](#)
 - gridColor, [216](#)
 - gridPath, [216](#)
 - gripBaseObj, [216](#)
 - gripColorCool, [216](#)
 - gripColorHot, [217](#)
 - grippingActive, [217](#)
 - gripSize, [217](#)
 - gscene, [217](#)
 - hashDeletedObjects, [217](#)
 - isLwtEnabled, [211](#)
 - isRealEnabled, [211](#)
 - loadRulerSettings, [211](#)
 - mirrorSelected, [211](#)
 - mouseDoubleClickEvent, [212](#)

mouseMoveEvent, 212
mousePressEvent, 212
mouseReleaseEvent, 212
moveAction, 212
movePoint, 217
moveSelected, 212
movingActive, 217
numSelected, 212
originPath, 217
panDistance, 217
panDown, 212
panLeft, 212
panningActive, 217
panningPointActive, 217
panningRealTimeActive, 217
panPoint, 212
panRealTime, 212
panRight, 212
panStart, 212
panStartX, 217
panStartY, 217
panUp, 212
paste, 213
pasteDelta, 217
pasteObjectItemGroup, 217
pastingActive, 217
pickBoxSize, 217
pressPoint, 218
previewActive, 218
previewData, 218
previewMode, 218
previewObjectItemGroup, 218
previewObjectList, 218
previewOff, 213
previewOn, 213
previewPoint, 218
qSnapActive, 218
qsnapApertureSize, 218
qsnapLocatorColor, 218
qsnapLocatorSize, 218
qSnapToggle, 218
rapidMoveActive, 218
recalculateLimits, 213
releasePoint, 218
repeatAction, 213
rotateAction, 213
rotateSelected, 213
roundToMultiple, 213
rubberRoomList, 218
rulerColor, 218
rulerMetric, 218
rulerPixelSize, 218
scaleAction, 213
scaleSelected, 213
sceneGripPoint, 219
sceneMousePoint, 219
sceneMovePoint, 219
scenePressPoint, 219
sceneReleasePoint, 219
selectAll, 213
selectBox, 219
selected_items, 213
selectingActive, 219
selectionChanged, 213
setBackgroundColor, 214
setCornerButton, 214
setCrossHairColor, 214
setCrossHairSize, 214
setGridColor, 214
setRubberMode, 214
setRubberPoint, 214
setRubberText, 214
setRulerColor, 214
setSelectBoxColors, 214
showScrollBars, 214
spareRubber, 214
spareRubberList, 219
startGripping, 214
state, 219
stopGripping, 215
tempBaseObj, 219
toggleGrid, 215
toggleLwt, 215
toggleOrtho, 215
togglePolar, 215
toggleQSnap, 215
toggleQTrack, 215
toggleReal, 215
toggleRuler, 215
toggleSnap, 215
undoStack, 219
updateMouseCoords, 215
View, 209
viewMousePoint, 219
vulcanizeObject, 215
vulcanizeRubberRoom, 215
wheelEvent, 215
willOverflowInt32, 216
willUnderflowInt32, 216
zoomExtents, 216
zoomIn, 216
zoomOut, 216
zoomSelected, 216
zoomToPoint, 216
zoomWindow, 216
zoomWindowActive, 219
view.cpp
 contains, 300
viewMousePoint
 View, 219
vip, 9, 337, 443
vipCompressData
 format_vip.c, 442
vipDecodeByte
 format_vip.c, 442
vipDecodeStitchType

- format_vip.c, [442](#)
- vipDecodingTable
 - embroidery.h, [337](#)
 - format_vip.c, [442](#)
- vipDecompressData
 - format_vip.c, [442](#)
- vipEncodeByte
 - format_vip.c, [442](#)
- vipEncodeStitchType
 - format_vip.c, [442](#)
- VipHeader
 - embroidery_internal.h, [361](#)
- VipHeader_, [219](#)
 - attributeOffset, [220](#)
 - colorLength, [220](#)
 - magicCode, [220](#)
 - negativeXHoopSize, [220](#)
 - negativeYHoopSize, [220](#)
 - numberOfColors, [220](#)
 - numberOfStitches, [220](#)
 - postitiveXHoopSize, [220](#)
 - postitiveYHoopSize, [220](#)
 - stringVal, [220](#)
 - unknown, [220](#)
 - xOffset, [220](#)
 - yOffset, [220](#)
- vp3, [9](#), [444](#)
- vp3Decode
 - format_vp3.c, [443](#)
- vp3DecodeInt16
 - format_vp3.c, [443](#)
- vp3Hoop
 - embroidery_internal.h, [361](#)
- vp3PatchByteCount
 - format_vp3.c, [443](#)
- vp3ReadHoopSection
 - format_vp3.c, [443](#)
- vp3ReadString
 - format_vp3.c, [444](#)
- vp3WriteString
 - format_vp3.c, [444](#)
- vp3WriteStringLen
 - format_vp3.c, [444](#)
- vulcanize
 - Geometry, [128](#)
- vulcanize_action
 - mainwindow.cpp, [291](#)
- vulcanizeObject
 - View, [215](#)
- vulcanizeRubberRoom
 - View, [215](#)
- whats_this_action
 - mainwindow.cpp, [291](#)
- wheelEvent
 - View, [215](#)
- WHITESPACE
 - main.c, [471](#)
- width
 - _vp3Hoop, [49](#)
 - EmblImage_, [89](#)
- willOverflowInt32
 - View, [216](#)
- willUnderflowInt32
 - View, [216](#)
- window_action
 - mainwindow.cpp, [291](#)
- windowMenuAboutToShow
 - MainWindow, [153](#)
- windowMenuActivated
 - MainWindow, [153](#)
- wizardTipOfTheDay
 - mainwindow.cpp, [294](#)
- write100
 - embroidery_internal.h, [378](#)
 - format_100.c, [403](#)
- write10o
 - embroidery_internal.h, [378](#)
 - format_10o.c, [404](#)
- write_24bit
 - embroidery_internal.h, [378](#)
 - encoding.c, [394](#)
 - main.c, [470](#)
- write_external_color_file
 - EmbFormatList_, [86](#)
- write_settings
 - embroidermodder.h, [236](#)
 - settings-dialog.cpp, [298](#)
- writeArt
 - embroidery_internal.h, [379](#)
 - format_art.c, [404](#)
- writeBmc
 - embroidery_internal.h, [379](#)
 - format_bmc.c, [405](#)
- writeBro
 - embroidery_internal.h, [379](#)
 - format_bro.c, [405](#)
- writeCnd
 - embroidery_internal.h, [379](#)
 - format_cnd.c, [406](#)
- writeCol
 - embroidery_internal.h, [379](#)
 - format_col.c, [407](#)
- writeCsd
 - embroidery_internal.h, [379](#)
 - format_csd.c, [408](#)
- writeCsv
 - embroidery_internal.h, [379](#)
 - format_csv.c, [409](#)
- writeDat
 - embroidery_internal.h, [379](#)
 - format_dat.c, [409](#)
- writeDem
 - embroidery_internal.h, [379](#)
 - format_dem.c, [410](#)
- writeDsb
 - embroidery_internal.h, [379](#)

- format_dsb.c, [411](#)
- writeDst
 - embroidery_internal.h, [379](#)
 - format_dst.c, [413](#)
- writeDsz
 - embroidery_internal.h, [380](#)
 - format_dsz.c, [413](#)
- writeDxf
 - embroidery_internal.h, [380](#)
 - format_dxf.c, [414](#)
- writeEdr
 - embroidery_internal.h, [380](#)
 - format_edr.c, [414](#)
- writeEmd
 - embroidery_internal.h, [380](#)
 - format_emd.c, [415](#)
- writeExp
 - embroidery_internal.h, [380](#)
 - format_exp.c, [416](#)
- writeExy
 - embroidery_internal.h, [380](#)
 - format_exy.c, [416](#)
- writeEys
 - embroidery_internal.h, [380](#)
 - format_ey.c, [417](#)
- writeFxy
 - embroidery_internal.h, [380](#)
 - format_fxy.c, [417](#)
- writeGc
 - embroidery_internal.h, [380](#)
 - format_gc.c, [418](#)
- writeGnc
 - embroidery_internal.h, [380](#)
 - format_gnc.c, [418](#)
- writeGt
 - embroidery_internal.h, [380](#)
 - format_gt.c, [419](#)
- writeHus
 - embroidery_internal.h, [381](#)
 - format_hus.c, [420](#)
- writelnImage
 - format_pec.c, [428](#)
 - image.c, [460](#)
- writelnb
 - embroidery_internal.h, [381](#)
 - format_inb.c, [420](#)
- writelnf
 - embroidery_internal.h, [381](#)
 - format_inf.c, [421](#)
- writeJef
 - embroidery_internal.h, [381](#)
 - format_jef.c, [422](#)
- writeKsm
 - embroidery_internal.h, [381](#)
 - format_ksm.c, [422](#)
- writeMax
 - embroidery_internal.h, [381](#)
 - format_max.c, [423](#)
- writeMit
 - embroidery_internal.h, [381](#)
 - format_mit.c, [424](#)
- writeNew
 - embroidery_internal.h, [381](#)
 - format_new.c, [424](#)
- writeOfm
 - embroidery_internal.h, [381](#)
 - format_ofm.c, [425](#)
- writePcd
 - embroidery_internal.h, [381](#)
 - format_pcd.c, [426](#)
- writePcm
 - embroidery_internal.h, [381](#)
 - format_pcm.c, [426](#)
- writePcq
 - embroidery_internal.h, [382](#)
 - format_pcq.c, [427](#)
- writePcs
 - embroidery_internal.h, [382](#)
 - format_pcs.c, [427](#)
- writePec
 - embroidery_internal.h, [382](#)
 - format_pec.c, [428](#)
- writePecStitches
 - embroidery_internal.h, [382](#)
 - format_pec.c, [428](#)
- writePel
 - embroidery_internal.h, [382](#)
 - format_pel.c, [429](#)
- writePem
 - embroidery_internal.h, [382](#)
 - format_pem.c, [429](#)
- writePes
 - embroidery_internal.h, [382](#)
 - format_pes.c, [432](#)
- writePhb
 - embroidery_internal.h, [382](#)
 - format_phb.c, [432](#)
- writePhc
 - embroidery_internal.h, [382](#)
 - format_phc.c, [433](#)
- writePlt
 - embroidery_internal.h, [382](#)
 - format_plt.c, [433](#)
- writer_state
 - EmbFormatList_, [86](#)
- writeRgb
 - embroidery_internal.h, [383](#)
 - format_rgb.c, [434](#)
- writeSew
 - embroidery_internal.h, [383](#)
 - format_sew.c, [434](#)
- writeShv
 - embroidery_internal.h, [383](#)
 - format_shv.c, [435](#)
- writeSst
 - embroidery_internal.h, [383](#)

- format_sst.c, [436](#)
- writeStx
 - embroidery_internal.h, [383](#)
 - format_stx.c, [436](#)
- writeSvg
 - embroidery_internal.h, [383](#)
 - format_svg.c, [437](#)
- writeT01
 - embroidery_internal.h, [383](#)
 - format_t01.c, [438](#)
- writeT09
 - embroidery_internal.h, [383](#)
 - format_t09.c, [438](#)
- writeTap
 - embroidery_internal.h, [383](#)
 - format_tap.c, [439](#)
- writeThr
 - embroidery_internal.h, [383](#)
 - format_thr.c, [440](#)
- writeTxt
 - embroidery_internal.h, [383](#)
 - format_txt.c, [440](#)
- writeU00
 - embroidery_internal.h, [384](#)
 - format_u00.c, [441](#)
- writeU01
 - embroidery_internal.h, [384](#)
 - format_u01.c, [441](#)
- writeVip
 - embroidery_internal.h, [384](#)
 - format_vip.c, [442](#)
- writeVp3
 - embroidery_internal.h, [384](#)
 - format_vp3.c, [444](#)
- writeXxx
 - embroidery_internal.h, [384](#)
 - format_xxx.c, [444](#)
- writeZsk
 - embroidery_internal.h, [384](#)
 - format_zsk.c, [445](#)
- x
 - EmbStitch_, [98](#)
 - EmbVector_, [102](#)
- x_values
 - Geometry, [130](#)
- xOffset
 - _vp3Hoop, [49](#)
 - VipHeader_, [220](#)
- xxx, [9](#), [445](#)
- xxxDecodeByte
 - format_xxx.c, [444](#)
- xxxEncodeDesign
 - format_xxx.c, [445](#)
- xxxEncodeStitch
 - format_xxx.c, [445](#)
- xxxEncodeStop
 - format_xxx.c, [445](#)
- y
 - EmbStitch_, [99](#)
 - EmbVector_, [102](#)
- y_values
 - Geometry, [131](#)
- year
 - EmbTime_, [101](#)
- YELLOW_TERM_COLOR
 - embroidery_internal.h, [360](#)
- yOffset
 - _vp3Hoop, [49](#)
 - VipHeader_, [220](#)
- Z102_Isacord_Polyester
 - embroidery.h, [319](#)
- zoom_action
 - mainwindow.cpp, [292](#)
- zoomExtents
 - View, [216](#)
- zoomExtentsAllSubWindows
 - MdiArea, [159](#)
- zoomIn
 - View, [216](#)
- zoomOut
 - View, [216](#)
- zoomSelected
 - View, [216](#)
- zoomToPoint
 - View, [216](#)
- zoomWindow
 - View, [216](#)
- zoomWindowActive
 - View, [219](#)
- zsk, [9](#), [445](#)
- ZSK USA, [371](#), [413](#), [445](#)