

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	32
12.1 Namespace List	32
13 Hierarchical Index	33
13.1 Class Hierarchy	33
14 Class Index	35
14.1 Class List	35
15 File Index	38
15.1 File List	38
16 Namespace Documentation	42
16.1 em2_dev_script Namespace Reference	42
16.1.1 Detailed Description	42
16.1.2 Variable Documentation	42
17 Class Documentation	43
17.1 _bcf_directory Struct Reference	43
17.1.1 Detailed Description	43

17.1.2 Member Data Documentation	43
17.2 _bcf_directory_entry Struct Reference	43
17.2.1 Member Data Documentation	44
17.3 _bcf_file Struct Reference	45
17.3.1 Member Data Documentation	45
17.4 _bcf_file_difat Struct Reference	46
17.4.1 Member Data Documentation	46
17.5 _bcf_file_fat Struct Reference	46
17.5.1 Member Data Documentation	47
17.6 _bcf_file_header Struct Reference	47
17.6.1 Detailed Description	48
17.6.2 Member Data Documentation	48
17.7 _vp3Hoop Struct Reference	49
17.7.1 Member Data Documentation	50
17.8 Application Class Reference	52
17.8.1 Detailed Description	52
17.8.2 Constructor & Destructor Documentation	52
17.8.3 Member Function Documentation	53
17.8.4 Member Data Documentation	53
17.9 CmdPrompt Class Reference	53
17.9.1 Detailed Description	55
17.9.2 Constructor & Destructor Documentation	55
17.9.3 Member Function Documentation	56
17.9.4 Member Data Documentation	62
17.10 CmdPromptHandle Class Reference	62
17.10.1 Detailed Description	63
17.10.2 Constructor & Destructor Documentation	63
17.10.3 Member Function Documentation	64
17.10.4 Member Data Documentation	65
17.11 CmdPromptHistory Class Reference	65
17.11.1 Detailed Description	66
17.11.2 Constructor & Destructor Documentation	66
17.11.3 Member Function Documentation	66
17.11.4 Member Data Documentation	68
17.12 CmdPromptInput Class Reference	68
17.12.1 Constructor & Destructor Documentation	70
17.12.2 Member Function Documentation	70
17.12.3 Member Data Documentation	75
17.13 CmdPromptSplitter Class Reference	76
17.13.1 Detailed Description	77
17.13.2 Constructor & Destructor Documentation	77
17.13.3 Member Function Documentation	77

17.14 Compress Struct Reference	78
17.14.1 Member Data Documentation	78
17.15 EmbAlignedDim_ Struct Reference	79
17.15.1 Member Data Documentation	79
17.16 EmbAngularDim_ Struct Reference	79
17.16.1 Member Data Documentation	80
17.17 EmbArc_ Struct Reference	80
17.17.1 Detailed Description	80
17.17.2 Member Data Documentation	80
17.18 EmbArcLengthDim_ Struct Reference	81
17.18.1 Member Data Documentation	81
17.19 EmbArray_ Struct Reference	81
17.19.1 Member Data Documentation	81
17.20 EmbBezier_ Struct Reference	82
17.20.1 Member Data Documentation	82
17.21 EmbBlock_ Struct Reference	83
17.21.1 Member Data Documentation	83
17.22 EmbCircle_ Struct Reference	83
17.22.1 Member Data Documentation	83
17.23 EmbColor_ Struct Reference	84
17.23.1 Detailed Description	84
17.23.2 Member Data Documentation	84
17.24 EmbDetailsDialog Class Reference	84
17.24.1 Detailed Description	85
17.24.2 Constructor & Destructor Documentation	85
17.24.3 Member Function Documentation	86
17.24.4 Member Data Documentation	86
17.25 EmbDiameterDim_ Struct Reference	87
17.25.1 Member Data Documentation	87
17.26 EmbEllipse_ Struct Reference	87
17.26.1 Member Data Documentation	88
17.27 EmbFormatList_ Struct Reference	88
17.27.1 Member Data Documentation	88
17.28 EmbGeometry_ Struct Reference	89
17.28.1 Member Data Documentation	90
17.29 EmbImage_ Struct Reference	91
17.29.1 Member Data Documentation	92
17.30 EmbInfiniteLine_ Struct Reference	92
17.30.1 Member Data Documentation	93
17.31 EmbLayer_ Struct Reference	93
17.31.1 Member Data Documentation	93
17.32 EmbLeaderDim_ Struct Reference	93

17.32.1 Member Data Documentation	94
17.33 EmbLine_ Struct Reference	94
17.33.1 Member Data Documentation	94
17.34 EmbLinearDim_ Struct Reference	95
17.34.1 Member Data Documentation	95
17.35 EmbOrdinateDim_ Struct Reference	95
17.35.1 Member Data Documentation	95
17.36 EmbPath_ Struct Reference	95
17.36.1 Member Data Documentation	96
17.37 EmbPattern_ Struct Reference	96
17.37.1 Member Data Documentation	97
17.38 EmbPoint_ Struct Reference	97
17.38.1 Member Data Documentation	98
17.39 EmbRadiusDim_ Struct Reference	98
17.39.1 Member Data Documentation	98
17.40 EmbRay_ Struct Reference	99
17.40.1 Member Data Documentation	99
17.41 EmbRect_ Struct Reference	99
17.41.1 Member Data Documentation	99
17.42 EmbSatinOutline_ Struct Reference	100
17.42.1 Member Data Documentation	100
17.43 EmbSpline_ Struct Reference	101
17.43.1 Member Data Documentation	101
17.44 EmbStitch_ Struct Reference	101
17.44.1 Member Data Documentation	101
17.45 EmbTextMulti_ Struct Reference	102
17.45.1 Member Data Documentation	102
17.46 EmbTextSingle_ Struct Reference	102
17.46.1 Member Data Documentation	103
17.47 EmbThread_ Struct Reference	103
17.47.1 Member Data Documentation	103
17.48 EmbTime_ Struct Reference	104
17.48.1 Member Data Documentation	104
17.49 EmbVector_ Struct Reference	105
17.49.1 Detailed Description	105
17.49.2 Member Data Documentation	105
17.50 Geometry Class Reference	105
17.50.1 Detailed Description	110
17.50.2 Member Enumeration Documentation	110
17.50.3 Constructor & Destructor Documentation	111
17.50.4 Member Function Documentation	114
17.50.5 Member Data Documentation	138

17.51 hoop_padding Struct Reference	140
17.51.1 Member Data Documentation	141
17.52 Huffman Struct Reference	141
17.52.1 Member Data Documentation	141
17.53 ImageWidget Class Reference	142
17.53.1 Detailed Description	142
17.53.2 Constructor & Destructor Documentation	142
17.53.3 Member Function Documentation	143
17.53.4 Member Data Documentation	144
17.54 LayerManager Class Reference	144
17.54.1 Detailed Description	144
17.54.2 Constructor & Destructor Documentation	144
17.54.3 Member Function Documentation	145
17.54.4 Member Data Documentation	145
17.55 LSYSTEM Struct Reference	146
17.55.1 Member Data Documentation	146
17.56 MainWindow Class Reference	146
17.56.1 Detailed Description	150
17.56.2 Constructor & Destructor Documentation	150
17.56.3 Member Function Documentation	150
17.56.4 Member Data Documentation	163
17.57 MdiArea Class Reference	165
17.57.1 Constructor & Destructor Documentation	166
17.57.2 Member Function Documentation	166
17.57.3 Member Data Documentation	168
17.58 MdiWindow Class Reference	169
17.58.1 Constructor & Destructor Documentation	170
17.58.2 Member Function Documentation	171
17.58.3 Member Data Documentation	177
17.59 Node_ Struct Reference	179
17.59.1 Member Data Documentation	179
17.60 PreviewDialog Class Reference	180
17.60.1 Constructor & Destructor Documentation	180
17.60.2 Member Data Documentation	180
17.61 PropertyEditor Class Reference	181
17.61.1 Constructor & Destructor Documentation	182
17.61.2 Member Function Documentation	182
17.61.3 Member Data Documentation	185
17.62 SaveObject Class Reference	187
17.62.1 Constructor & Destructor Documentation	188
17.62.2 Member Function Documentation	189
17.62.3 Member Data Documentation	197

17.63 SelectBox Class Reference	197
17.63.1 Constructor & Destructor Documentation	198
17.63.2 Member Function Documentation	198
17.63.3 Member Data Documentation	199
17.64 Settings_Dialog Class Reference	200
17.64.1 Constructor & Destructor Documentation	201
17.64.2 Member Function Documentation	202
17.64.3 Member Data Documentation	207
17.65 StatusBar Class Reference	208
17.65.1 Detailed Description	208
17.65.2 Constructor & Destructor Documentation	208
17.65.3 Member Function Documentation	208
17.65.4 Member Data Documentation	209
17.66 StxThread_ Struct Reference	209
17.66.1 Member Data Documentation	209
17.67 SubDescriptor_ Struct Reference	209
17.67.1 Member Data Documentation	210
17.68 SvgAttribute_ Struct Reference	210
17.68.1 Member Data Documentation	210
17.69 thread_color_ Struct Reference	210
17.69.1 Member Data Documentation	211
17.70 ThredExtension_ Struct Reference	211
17.70.1 Member Data Documentation	211
17.71 ThredHeader_ Struct Reference	212
17.71.1 Member Data Documentation	212
17.72 UndoableCommand Class Reference	212
17.72.1 Constructor & Destructor Documentation	213
17.72.2 Member Function Documentation	214
17.72.3 Member Data Documentation	214
17.73 UndoEditor Class Reference	215
17.73.1 Constructor & Destructor Documentation	216
17.73.2 Member Function Documentation	216
17.73.3 Member Data Documentation	217
17.74 View Class Reference	217
17.74.1 Constructor & Destructor Documentation	220
17.74.2 Member Function Documentation	220
17.74.3 Member Data Documentation	227
17.75 VipHeader_ Struct Reference	230
17.75.1 Member Data Documentation	231
18 File Documentation	232
18.1 CODE_OF_CONDUCT.md File Reference	232

18.2 extern/libembroidery/src/array.c File Reference	232
18.2.1 Function Documentation	232
18.3 extern/libembroidery/src/compress.c File Reference	234
18.3.1 Detailed Description	234
18.3.2 Function Documentation	234
18.3.3 Variable Documentation	236
18.4 extern/libembroidery/src/embroidery.h File Reference	236
18.4.1 Macro Definition Documentation	243
18.4.2 Typedef Documentation	250
18.4.3 Function Documentation	253
18.4.4 Variable Documentation	268
18.5 embroidery.h	269
18.6 extern/libembroidery/src/embroidery_internal.h File Reference	276
18.6.1 Macro Definition Documentation	284
18.6.2 Typedef Documentation	292
18.6.3 Enumeration Type Documentation	293
18.6.4 Function Documentation	293
18.6.5 Variable Documentation	316
18.7 embroidery_internal.h	316
18.8 extern/libembroidery/src/encoding.c File Reference	323
18.8.1 Detailed Description	324
18.8.2 Function Documentation	324
18.9 extern/libembroidery/src/fill.c File Reference	326
18.9.1 Function Documentation	327
18.9.2 Variable Documentation	330
18.10 extern/libembroidery/src/formats.c File Reference	331
18.10.1 Function Documentation	332
18.10.2 Variable Documentation	334
18.11 extern/libembroidery/src/formats/format_100.c File Reference	335
18.11.1 Detailed Description	335
18.11.2 Function Documentation	335
18.12 extern/libembroidery/src/formats/format_10o.c File Reference	335
18.12.1 Detailed Description	335
18.12.2 Function Documentation	336
18.13 extern/libembroidery/src/formats/format_art.c File Reference	336
18.13.1 Detailed Description	336
18.13.2 Function Documentation	336
18.14 extern/libembroidery/src/formats/format_bmc.c File Reference	336
18.14.1 Detailed Description	337
18.14.2 Function Documentation	337
18.15 extern/libembroidery/src/formats/format_bro.c File Reference	337
18.15.1 Detailed Description	337

18.15.2 Function Documentation	337
18.16 extern/libembroidery/src/formats/format_cnd.c File Reference	337
18.16.1 Detailed Description	338
18.16.2 Function Documentation	338
18.17 extern/libembroidery/src/formats/format_col.c File Reference	338
18.17.1 Detailed Description	338
18.17.2 Function Documentation	339
18.18 extern/libembroidery/src/formats/format_csd.c File Reference	339
18.18.1 Detailed Description	339
18.18.2 Macro Definition Documentation	339
18.18.3 Function Documentation	339
18.18.4 Variable Documentation	340
18.19 extern/libembroidery/src/formats/format_csv.c File Reference	340
18.19.1 Detailed Description	341
18.19.2 Function Documentation	341
18.20 extern/libembroidery/src/formats/format_dat.c File Reference	341
18.20.1 Function Documentation	341
18.21 extern/libembroidery/src/formats/format_dem.c File Reference	342
18.21.1 Detailed Description	342
18.21.2 Function Documentation	342
18.22 extern/libembroidery/src/formats/format_dsb.c File Reference	342
18.22.1 Detailed Description	342
18.22.2 Function Documentation	343
18.23 extern/libembroidery/src/formats/format_dst.c File Reference	343
18.23.1 Detailed Description	343
18.23.2 Macro Definition Documentation	344
18.23.3 Function Documentation	344
18.24 extern/libembroidery/src/formats/format_dsz.c File Reference	345
18.24.1 Function Documentation	345
18.25 extern/libembroidery/src/formats/format_dxf.c File Reference	345
18.25.1 Function Documentation	346
18.26 extern/libembroidery/src/formats/format_edr.c File Reference	346
18.26.1 Function Documentation	346
18.27 extern/libembroidery/src/formats/format_emd.c File Reference	347
18.27.1 Detailed Description	347
18.27.2 Function Documentation	347
18.28 extern/libembroidery/src/formats/format_exp.c File Reference	347
18.28.1 Function Documentation	347
18.29 extern/libembroidery/src/formats/format_exy.c File Reference	348
18.29.1 Function Documentation	348
18.30 extern/libembroidery/src/formats/format_eyz.c File Reference	348
18.30.1 Function Documentation	348

18.31	extern/libembroidery/src/formats/format_fxy.c File Reference	349
18.31.1	Function Documentation	349
18.32	extern/libembroidery/src/formats/format_gc.c File Reference	349
18.32.1	Function Documentation	349
18.33	extern/libembroidery/src/formats/format_gnc.c File Reference	350
18.33.1	Function Documentation	350
18.34	extern/libembroidery/src/formats/format_gt.c File Reference	350
18.34.1	Function Documentation	350
18.35	extern/libembroidery/src/formats/format_hus.c File Reference	351
18.35.1	Function Documentation	351
18.36	extern/libembroidery/src/formats/format_inb.c File Reference	352
18.36.1	Function Documentation	352
18.37	extern/libembroidery/src/formats/format_inf.c File Reference	352
18.37.1	Function Documentation	352
18.38	extern/libembroidery/src/formats/format_jef.c File Reference	353
18.38.1	Function Documentation	353
18.39	extern/libembroidery/src/formats/format_ksm.c File Reference	354
18.39.1	Function Documentation	354
18.40	extern/libembroidery/src/formats/format_max.c File Reference	354
18.40.1	Function Documentation	355
18.40.2	Variable Documentation	355
18.41	extern/libembroidery/src/formats/format_mit.c File Reference	355
18.41.1	Function Documentation	355
18.42	extern/libembroidery/src/formats/format_new.c File Reference	356
18.42.1	Function Documentation	356
18.43	extern/libembroidery/src/formats/format_ofm.c File Reference	356
18.43.1	Function Documentation	357
18.44	extern/libembroidery/src/formats/format_pcd.c File Reference	357
18.44.1	Function Documentation	358
18.45	extern/libembroidery/src/formats/format_pcm.c File Reference	358
18.45.1	Function Documentation	358
18.46	extern/libembroidery/src/formats/format_pcq.c File Reference	358
18.46.1	Function Documentation	359
18.47	extern/libembroidery/src/formats/format_pcs.c File Reference	359
18.47.1	Function Documentation	359
18.48	extern/libembroidery/src/formats/format_pec.c File Reference	359
18.48.1	Function Documentation	360
18.49	extern/libembroidery/src/formats/format_pel.c File Reference	361
18.49.1	Function Documentation	361
18.50	extern/libembroidery/src/formats/format_pem.c File Reference	361
18.50.1	Function Documentation	361
18.51	extern/libembroidery/src/formats/format_pes.c File Reference	362

18.51.1 Function Documentation	362
18.51.2 Variable Documentation	364
18.52 extern/libembroidery/src/formats/format_phb.c File Reference	364
18.52.1 Function Documentation	364
18.53 extern/libembroidery/src/formats/format_phc.c File Reference	365
18.53.1 Function Documentation	365
18.54 extern/libembroidery/src/formats/format_plt.c File Reference	365
18.54.1 Function Documentation	365
18.55 extern/libembroidery/src/formats/format_rgb.c File Reference	366
18.55.1 Function Documentation	366
18.56 extern/libembroidery/src/formats/format_sew.c File Reference	366
18.56.1 Function Documentation	366
18.57 extern/libembroidery/src/formats/format_shv.c File Reference	367
18.57.1 Function Documentation	367
18.58 extern/libembroidery/src/formats/format_sst.c File Reference	367
18.58.1 Function Documentation	367
18.59 extern/libembroidery/src/formats/format_stx.c File Reference	368
18.59.1 Function Documentation	368
18.60 extern/libembroidery/src/formats/format_svg.c File Reference	368
18.60.1 Function Documentation	369
18.60.2 Variable Documentation	369
18.61 extern/libembroidery/src/formats/format_t01.c File Reference	370
18.61.1 Function Documentation	370
18.62 extern/libembroidery/src/formats/format_t09.c File Reference	370
18.62.1 Function Documentation	370
18.63 extern/libembroidery/src/formats/format_tap.c File Reference	371
18.63.1 Function Documentation	371
18.64 extern/libembroidery/src/formats/format_thr.c File Reference	371
18.64.1 Function Documentation	371
18.65 extern/libembroidery/src/formats/format_txt.c File Reference	372
18.65.1 Function Documentation	372
18.66 extern/libembroidery/src/formats/format_u00.c File Reference	372
18.66.1 Function Documentation	372
18.67 extern/libembroidery/src/formats/format_u01.c File Reference	373
18.67.1 Function Documentation	373
18.68 extern/libembroidery/src/formats/format_vip.c File Reference	373
18.68.1 Function Documentation	374
18.68.2 Variable Documentation	374
18.69 extern/libembroidery/src/formats/format_vp3.c File Reference	375
18.69.1 Function Documentation	375
18.70 extern/libembroidery/src/formats/format_xxx.c File Reference	376
18.70.1 Function Documentation	376

18.71	extern/libembroidery/src/formats/format_zsk.c File Reference	377
18.71.1	Detailed Description	377
18.71.2	Function Documentation	377
18.72	extern/libembroidery/src/geometry.c File Reference	377
18.72.1	Function Documentation	378
18.73	extern/libembroidery/src/geometry/arc.c File Reference	379
18.73.1	Function Documentation	379
18.74	extern/libembroidery/src/geometry/circle.c File Reference	382
18.74.1	Function Documentation	383
18.75	extern/libembroidery/src/geometry/ellipse.c File Reference	383
18.75.1	Function Documentation	384
18.76	extern/libembroidery/src/geometry/functions.c File Reference	385
18.76.1	Function Documentation	385
18.77	extern/libembroidery/src/geometry/line.c File Reference	386
18.77.1	Function Documentation	386
18.78	extern/libembroidery/src/geometry/path.c File Reference	386
18.79	extern/libembroidery/src/geometry/polygon.c File Reference	386
18.80	extern/libembroidery/src/geometry/polyline.c File Reference	386
18.81	extern/libembroidery/src/geometry/rect.c File Reference	387
18.81.1	Function Documentation	387
18.82	extern/libembroidery/src/geometry/text.c File Reference	387
18.82.1	Function Documentation	387
18.83	extern/libembroidery/src/geometry/vector.c File Reference	389
18.83.1	Function Documentation	389
18.84	extern/libembroidery/src/image.c File Reference	391
18.84.1	Detailed Description	391
18.84.2	Function Documentation	391
18.85	extern/libembroidery/src/main.c File Reference	392
18.85.1	Macro Definition Documentation	394
18.85.2	Function Documentation	397
18.85.3	Variable Documentation	403
18.86	extern/libembroidery/src/pattern.c File Reference	403
18.86.1	Detailed Description	404
18.86.2	Function Documentation	404
18.87	extern/libembroidery/src/thread-color.c File Reference	408
18.87.1	Function Documentation	409
18.87.2	Variable Documentation	409
18.88	privacy_policy.md File Reference	410
18.89	src/cmdprompt.cpp File Reference	410
18.89.1	Detailed Description	410
18.90	src/em2_dev_script.py File Reference	410
18.91	src/embdetails-dialog.cpp File Reference	410

18.92 src/embroidermodder.cpp File Reference	410
18.92.1 Function Documentation	411
18.92.2 Variable Documentation	411
18.93 src/embroidermodder.h File Reference	411
18.93.1 Detailed Description	415
18.93.2 Macro Definition Documentation	418
18.93.3 Typedef Documentation	419
18.93.4 Enumeration Type Documentation	419
18.93.5 Function Documentation	420
18.93.6 Variable Documentation	429
18.94 embroidermodder.h	431
18.95 src/imagewidget.cpp File Reference	448
18.96 src/interface.cpp File Reference	449
18.96.1 Detailed Description	450
18.96.2 Function Documentation	450
18.97 src/layer-manager.cpp File Reference	457
18.97.1 Detailed Description	457
18.98 src/mainwindow-menus.cpp File Reference	457
18.98.1 Function Documentation	457
18.99 src/mainwindow-toolbars.cpp File Reference	457
18.100 src/mainwindow.cpp File Reference	457
18.100.1 Enumeration Type Documentation	463
18.100.2 Function Documentation	464
18.100.3 Variable Documentation	485
18.101 src/mdiarea.cpp File Reference	487
18.102 src/mdiwindow.cpp File Reference	487
18.102.1 Function Documentation	487
18.103 src/objects.cpp File Reference	488
18.103.1 Function Documentation	488
18.104 src/preview-dialog.cpp File Reference	489
18.105 src/property-editor.cpp File Reference	489
18.105.1 Function Documentation	489
18.105.2 Variable Documentation	489
18.106 src/README.md File Reference	490
18.107 src/selectbox.cpp File Reference	490
18.108 src/settings-dialog.cpp File Reference	491
18.108.1 Function Documentation	491
18.108.2 Variable Documentation	491
18.109 src/statusbar.cpp File Reference	493
18.110 src/undo-commands.cpp File Reference	493
18.111 src/undo-editor.cpp File Reference	493
18.111.1 Detailed Description	493

18.112 src/view.cpp File Reference	493
18.112.1 Detailed Description	493
18.112.2 Function Documentation	493
Bibliography	495
Index	497

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`ed 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 `binaryRead`↵
`Int16` 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

File **embroidermodder.h**

Notify user of data loss if not saving to an object format.

Import Raster Image

SNAP/ORTHO/POLAR

Layer Manager + LayerSwitcher DockWidget

Reading DXF

Writing DXF DONE - Up and Down keys cycle thru commands in the command prompt

Amount of Thread & Machine Time Estimation (also allow customizable times for setup, color changes, manually trimming jump threads, etc...that way a realistic total time can be estimated)

Otto Theme Icons - whatsthis icon doesn't scale well, needs redone

embroidermodder2.ico 16 x 16 looks horrible

Load/Save Menu/Toolbars configurations into settings.ini

automate changelog and write to a javascript file for the docs: `git log --pretty=tformat:' %s'`

Custom Filter Bug - doesn't save changes in some cases

Cannot open file with # in name when opening multiple files (works fine when opening the single file)

Closing Settings Dialog with the X in the window saves settings rather than discards them WIP - Advanced Printing

Filling Algorithms (varga)

Otto Theme Icons - beta (rt) - Units, Render, Selectors

QDoc Comments

Review KDE4 Thumbnailer

Documentation for libembroidery & formats

HTML Help files

Update language translations

CAD Command review: line

CAD Command review: circle

CAD Command review: rectangle

CAD Command review: polygon

CAD Command review: polyline

CAD Command review: point

CAD Command review: ellipse

CAD Command review: arc

CAD Command review: distance

CAD Command review: locatepoint

CAD Command review: move

CAD Command review: rgb

CAD Command review: rotate

CAD Command review: scale

CAD Command review: singlelinetext

CAD Command review: star

Clean up all compiler warning messages, right now theres plenty :P

tar.gz archive

zip archive

Debian Package (rt)

NSIS Installer (rt)

Mac Bundle?

press release

libembroidery.mk for MXE project (refer to qt submodule packages for qmake based building. Also refer to plibc.mk for example of how write an update macro for github.)

libembroidery safeguard for all writers - check if the last stitch is an END stitch. If not, add an end stitch in the writer and modify the header data if necessary.

Cut/Copy - Allow Post-selection

CAD Command: Array

CAD Command: Offset

CAD Command: Extend

CAD Command: Trim

CAD Command: BreakAtPoint

CAD Command: Break2Points
CAD Command: Fillet
CAD Command: Chamfer
CAD Command: Split
CAD Command: Area
CAD Command: Time
CAD Command: PickAdd
CAD Command: Product
CAD Command: Program
CAD Command: ZoomFactor
CAD Command: GripHot
CAD Command: GripColor & GripCool
CAD Command: GripSize
CAD Command: Highlight
CAD Command: Units
CAD Command: Grid
CAD Command: Find
CAD Command: Divide
CAD Command: ZoomWindow (Move out of [view.cpp](#))
Command: Web (Generates Spiderweb patterns)
Command: Guilloche (Generates Guilloche patterns)
Command: Celtic Knots
Command: Knotted Wreath
Lego Mindstorms NXT/EV3 ports and/or commands.
native function that flashes the command prompt to get users attention when using the prompt is required for a command.
libembroidery-composer like app that combines multiple files into one.
Settings Dialog, it would be nice to have it notify you when switching tabs that a setting has been changed. Adding an Apply button is what would make sense for this to happen.
Keyboard Zooming/Panning
G-Code format?
3D Raised Embroidery
Gradient Filling Algorithms
Stitching Simulation
RPM packages?
Reports?
Record and Playback Commands
Settings option for reversing zoom scrolling direction
Qt GUI for libembroidery-convert
EPS format? Look at using Ghostscript as an optional add-on to libembroidery...
optional compile option for including LGPL/GPL libs etc... with warning to user about license requirements.
Realistic Visualization - Bump Mapping/OpenGL/Gradients?
Stippling Fill
User Designed Custom Fill
Honeycomb Fill

Hilburt Curve Fill

Sierpinski Triangle fill

Circle Grid Fill

Spiral Fill

Offset Fill

Brick Fill

Trim jumps over a certain length.

FAQ about setting high number of jumps for more controlled trimming.

Minimum stitch length option. (Many machines also have this option too)

Add 'Design Details' functionality to libembroidery-convert

Add 'Batch convert many to one format' functionality to libembroidery-convert

EmbroideryFLOSS - Color picker that displays catalog numbers and names.

emscripten/javascript port of libembroidery

Fix emb-outline files

Fix thread-color files

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

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

Smoothieboard experiments

looping test that reads 10 times while running valgrind. See [embPattern_loadExternalColorFile\(\)](#) Arduino leak note for more info.

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](#)

Support for Singer FHE, CHE (Compucon) 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.

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

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

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.

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

(Arduino) Smoothieboard experiments

(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 thread-color files

(Arduino) Fix emb-outline files

Member Geometry::calculateArcData (EmbArc arc)

convert this to update and make it Type sensitive.

Member Geometry::init_text_single (QString str, EmbVector position, QRgb rgb, Qt::PenStyle lineType)

set the justification properly.

pass in proper linewidth

Page Ideas

Bibliography style to plainnat.

Serif font for printed docs.

US letter paper version of 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 ()

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?

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

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::addTextSingle](#) (EmbPattern *pattern, QGraphicsItem *item)

saving polygons, polylines and paths must be stable before we go here.

This needs to work like a path, not a polyline. Improve this.

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); //

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	42
--------------------------------	----

13 Hierarchical Index

13.1 Class Hierarchy

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

_bcf_directory	43
_bcf_directory_entry	43
_bcf_file	45
_bcf_file_difat	46
_bcf_file_fat	46
_bcf_file_header	47
_vp3Hoop	49
Compress	78
EmbAlignedDim_	79
EmbAngularDim_	79
EmbArc_	80
EmbArcLengthDim_	81
EmbArray_	81
EmbBezier_	82
EmbBlock_	83
EmbCircle_	83
EmbColor_	84
EmbDiameterDim_	87
EmbEllipse_	87
EmbFormatList_	88
EmbGeometry_	89
EmblImage_	91
EmblInfiniteLine_	92
EmbLayer_	93
EmbLeaderDim_	93
EmbLine_	94

EmbLinearDim_	95
EmbOrdinateDim_	95
EmbPath_	95
EmbPattern_	96
EmbPoint_	97
EmbRadiusDim_	98
EmbRay_	99
EmbRect_	99
EmbSatinOutline_	100
EmbSpline_	101
EmbStitch_	101
EmbTextMulti_	102
EmbTextSingle_	102
EmbThread_	103
EmbTime_	104
EmbVector_	105
hoop_padding	140
Huffman	141
LSYSTEM	146
Node_	179
QApplication	
Application	52
QDialog	
EmbDetailsDialog	84
LayerManager	144
Settings_Dialog	200
QDockWidget	
PropertyEditor	181
UndoEditor	215
QFileDialog	
PreviewDialog	180
QGraphicsPathItem	
Geometry	105
QGraphicsView	

View	217
QLineEdit	
CmdPromptInput	68
QMainWindow	
MainWindow	146
QMdiArea	
MdiArea	165
QMdiSubWindow	
MdiWindow	169
QObject	
SaveObject	187
QRubberBand	
SelectBox	197
QSplitter	
CmdPromptSplitter	76
QSplitterHandle	
CmdPromptHandle	62
QStatusBar	
StatusBar	208
QTextBrowser	
CmdPromptHistory	65
QUndoCommand	
UndoableCommand	212
QWidget	
CmdPrompt	53
ImageWidget	142
StxThread_	209
SubDescriptor_	209
SvgAttribute_	210
thread_color_	210
ThredExtension_	211
ThredHeader_	212
VipHeader_	230

14 Class Index

14.1 Class List

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

_bcf_directory	43
_bcf_directory_entry	43
_bcf_file	45
_bcf_file_difat	46
_bcf_file_fat	46
_bcf_file_header	47
_vp3Hoop	49
Application	52
CmdPrompt	53
CmdPromptHandle	62
CmdPromptHistory	
The Command Prompt History class	65
CmdPromptInput	68
CmdPromptSplitter	76
Compress	78
EmbAlignedDim_	79
EmbAngularDim_	79
EmbArc_	
Absolute position (not relative)	80
EmbArcLengthDim_	81
EmbArray_	81
EmbBezier_	82
EmbBlock_	83
EmbCircle_	83
EmbColor_	84
EmbDetailsDialog	
84	
EmbDiameterDim_	87
EmbEllipse_	87
EmbFormatList_	88
EmbGeometry_	89
EmblImage_	91

EmblInfiniteLine_	92
EmbLayer_	93
EmbLeaderDim_	93
EmbLine_	94
EmbLinearDim_	95
EmbOrdinateDim_	95
EmbPath_	95
EmbPattern_	96
EmbPoint_	97
EmbRadiusDim_	98
EmbRay_	99
EmbRect_	99
EmbSatinOutline_	100
EmbSpline_	101
EmbStitch_	101
EmbTextMulti_	102
EmbTextSingle_	102
EmbThread_	103
EmbTime_	104
EmbVector_	105
Geometry	
The Geometry class	105
hoop_padding	140
Huffman	141
ImageWidget	
142	
LayerManager	
144	
LSYSTEM	146
MainWindow	
The MainWindow class	146
MdiArea	165
MdiWindow	169
Node_	179

PreviewDialog	180
PropertyEditor	181
SaveObject	187
SelectBox	197
Settings_Dialog	200
StatusBar	208
StxThread_	209
SubDescriptor_	209
SvgAttribute_	210
thread_color_	210
ThredExtension_	211
ThredHeader_	212
UndoableCommand	212
UndoEditor	215
View	217
VipHeader_	230

15 File Index

15.1 File List

Here is a list of all files with brief descriptions:

extern/libembroidery/src/array.c	232
extern/libembroidery/src/compress.c	234
extern/libembroidery/src/embroidery.h	236
extern/libembroidery/src/embroidery_internal.h	276
extern/libembroidery/src/encoding.c	323
extern/libembroidery/src/fill.c	326
extern/libembroidery/src/formats.c	331
extern/libembroidery/src/geometry.c	377
extern/libembroidery/src/image.c	391
extern/libembroidery/src/main.c	392
extern/libembroidery/src/pattern.c	403

extern/libembroidery/src/ thread-color.c	408
extern/libembroidery/src/formats/ format_100.c	335
extern/libembroidery/src/formats/ format_10o.c	335
extern/libembroidery/src/formats/ format_art.c	336
extern/libembroidery/src/formats/ format_bmc.c	336
extern/libembroidery/src/formats/ format_bro.c	337
extern/libembroidery/src/formats/ format_cnd.c	337
extern/libembroidery/src/formats/ format_col.c	338
extern/libembroidery/src/formats/ format_csd.c	339
extern/libembroidery/src/formats/ format_csv.c	340
extern/libembroidery/src/formats/ format_dat.c	341
extern/libembroidery/src/formats/ format_dem.c	342
extern/libembroidery/src/formats/ format_dsb.c	342
extern/libembroidery/src/formats/ format_dst.c	343
extern/libembroidery/src/formats/ format_dsz.c	345
extern/libembroidery/src/formats/ format_dxf.c	345
extern/libembroidery/src/formats/ format_edr.c	346
extern/libembroidery/src/formats/ format_emd.c	347
extern/libembroidery/src/formats/ format_exp.c	347
extern/libembroidery/src/formats/ format_exy.c	348
extern/libembroidery/src/formats/ format_eyc.c	348
extern/libembroidery/src/formats/ format_fxy.c	349
extern/libembroidery/src/formats/ format_gc.c	349
extern/libembroidery/src/formats/ format_gnc.c	350
extern/libembroidery/src/formats/ format_gt.c	350
extern/libembroidery/src/formats/ format_hus.c	351
extern/libembroidery/src/formats/ format_inb.c	352
extern/libembroidery/src/formats/ format_inf.c	352
extern/libembroidery/src/formats/ format_jef.c	353
extern/libembroidery/src/formats/ format_ksm.c	354
extern/libembroidery/src/formats/ format_max.c	354
extern/libembroidery/src/formats/ format_mit.c	355

extern/libembroidery/src/formats/ format_new.c	356
extern/libembroidery/src/formats/ format_ofm.c	356
extern/libembroidery/src/formats/ format_pcd.c	357
extern/libembroidery/src/formats/ format_pcm.c	358
extern/libembroidery/src/formats/ format_pcq.c	358
extern/libembroidery/src/formats/ format_pcs.c	359
extern/libembroidery/src/formats/ format_pec.c	359
extern/libembroidery/src/formats/ format_pel.c	361
extern/libembroidery/src/formats/ format_pem.c	361
extern/libembroidery/src/formats/ format_pes.c	362
extern/libembroidery/src/formats/ format_phb.c	364
extern/libembroidery/src/formats/ format_phc.c	365
extern/libembroidery/src/formats/ format_plt.c	365
extern/libembroidery/src/formats/ format_rgb.c	366
extern/libembroidery/src/formats/ format_sew.c	366
extern/libembroidery/src/formats/ format_shv.c	367
extern/libembroidery/src/formats/ format_sst.c	367
extern/libembroidery/src/formats/ format_stx.c	368
extern/libembroidery/src/formats/ format_svg.c	368
extern/libembroidery/src/formats/ format_t01.c	370
extern/libembroidery/src/formats/ format_t09.c	370
extern/libembroidery/src/formats/ format_tap.c	371
extern/libembroidery/src/formats/ format_thr.c	371
extern/libembroidery/src/formats/ format_txt.c	372
extern/libembroidery/src/formats/ format_u00.c	372
extern/libembroidery/src/formats/ format_u01.c	373
extern/libembroidery/src/formats/ format_vip.c	373
extern/libembroidery/src/formats/ format_vp3.c	375
extern/libembroidery/src/formats/ format_xxx.c	376
extern/libembroidery/src/formats/ format_zsk.c	377
extern/libembroidery/src/geometry/ arc.c	379
extern/libembroidery/src/geometry/ circle.c	382

extern/libembroidery/src/geometry/ellipse.c	383
extern/libembroidery/src/geometry/functions.c	385
extern/libembroidery/src/geometry/line.c	386
extern/libembroidery/src/geometry/path.c	386
extern/libembroidery/src/geometry/polygon.c	386
extern/libembroidery/src/geometry/polyline.c	386
extern/libembroidery/src/geometry/rect.c	387
extern/libembroidery/src/geometry/text.c	387
extern/libembroidery/src/geometry/vector.c	389
src/cmdprompt.cpp	410
src/em2_dev_script.py	410
src/embdetails-dialog.cpp	410
src/embroidermodder.cpp	410
src/embroidermodder.h	411
src/imagewidget.cpp	448
src/interface.cpp	449
src/layer-manager.cpp	457
src/mainwindow-menus.cpp	457
src/mainwindow-toolbars.cpp	457
src/mainwindow.cpp	457
src/mdiarea.cpp	487
src/mdiwindow.cpp	487
src/objects.cpp	488
src/preview-dialog.cpp	489
src/property-editor.cpp	489
src/selectbox.cpp	490
src/settings-dialog.cpp	491
src/statusbar.cpp	493
src/undo-commands.cpp	493
src/undo-editor.cpp	493
src/view.cpp	493

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` `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:

- [src/embroidermodder.h](#)
- [src/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:

- [src/embroidermodder.h](#)
- [src/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:

- [src/embroidermodder.h](#)
- [src/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:

- [src/embroidermodder.h](#)
- [src/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:

- [src/embroidermodder.h](#)
- [src/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:

- src/[embroidermodder.h](#)
- src/[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:

- [src/embroidermodder.h](#)
- [src/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 EmbInfiniteLine_ 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::Geometry.](#)
- [Geometry](#) ([EmbArc](#) arc, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem *parent=0)
Construct a new [Geometry](#)::[Geometry](#) object.
- [Geometry](#) ([EmbCircle](#) circle, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem *parent=0)
Construct a new [Geometry](#)::[Geometry](#) object.
- [Geometry](#) ([EmbLine](#) line, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem *parent=0)
- [Geometry](#) ([EmbEllipse](#) ellipse, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem *parent=0)
Construct a new [Geometry](#)::[Geometry](#) object.
- [Geometry](#) ([EmbRect](#) rect, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem *parent=0)
Construct a new [Geometry](#) object.
- [Geometry](#) (QString str, [EmbVector](#) position, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem *parent=0)
Construct a new [Geometry](#) object.
- [Geometry](#) ([EmbLine](#) line, int Type_, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem *parent)
Construct a new [Geometry](#)::[Geometry](#) object.
- [Geometry](#) (QPainterPath p, int type_, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem *parent=0)
Construct a new [Geometry](#)::[Geometry](#) object.
- [Geometry](#) ([EmbVector](#) pos, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem *parent=0)
Construct a new [Geometry](#)::[Geometry](#) object.
- void [init_arc](#) ([EmbArc](#) arc, QRgb rgb, Qt::PenStyle lineType)
[Geometry::init.](#)
- void [init_circle](#) ([EmbCircle](#) circle, QRgb rgb, Qt::PenStyle lineType)
[Geometry::init_circle.](#)
- void [init_line](#) ([EmbLine](#) line, QRgb rgb, Qt::PenStyle lineType)
[Geometry::init_line.](#)
- void [init_ellipse](#) ([EmbEllipse](#) ellipse, QRgb rgb, Qt::PenStyle lineType)
[Geometry::init_ellipse.](#)
- void [init_rect](#) ([EmbRect](#) rect, QRgb rgb, Qt::PenStyle lineType)
[Geometry::init_line.](#)
- void [init_text_single](#) (QString str, [EmbVector](#) position, QRgb rgb, Qt::PenStyle lineType)
[Geometry::init_line.](#)
- void [init_path](#) (QPainterPath p, QRgb rgb, Qt::PenStyle lineType)
[Geometry::init_line.](#)
- void [init_point](#) ([EmbVector](#) pos, QRgb rgb, Qt::PenStyle lineType)
[Geometry::init_line.](#)
- 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)
Verify the string is a valid option, otherwise default to "Left".
- 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)`

[Geometry::Geometry.](#)

Parameters

<i>*obj</i>	
<i>*parent</i>	

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

Construct a new [Geometry::Geometry](#) object.

Parameters

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

17.50.3.4 Geometry() [4/11] `Geometry (`

```
EmbCircle circle,  
QRgb rgb,  
Qt::PenStyle lineType,  
QGraphicsItem * parent = 0 )
```

Construct a new [Geometry](#)::[Geometry](#) object.

Parameters

<i>circle</i>	
<i>rgb</i>	
<i>lineType</i>	
<i>parent</i>	

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)

Construct a new [Geometry](#)::[Geometry](#) object.

Parameters

<i>ellipse</i>	
<i>rgb</i>	
<i>lineType</i>	
<i>parent</i>	

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

Construct a new [Geometry](#) object.

Parameters

<i>rect</i>	
<i>rgb</i>	
<i>lineType</i>	
<i>parent</i>	

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

Construct a new [Geometry](#) object.

Parameters

<i>str</i>	
<i>v</i>	
<i>rgb</i>	
<i>lineType</i>	
<i>parent</i>	

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

Construct a new [Geometry::Geometry](#) object.

Parameters

<i>line</i>	
<i>Type_</i>	
<i>rgb</i>	
<i>lineType</i>	
<i>parent</i>	

17.50.3.10 Geometry() [10/11] [Geometry](#) (
 QPainterPath *p*,

```
int Type_,
QRgb rgb,
Qt::PenStyle lineType,
QGraphicsItem * parent = 0 )
```

Construct a new [Geometry::Geometry](#) object.

Parameters

<i>p</i>	
<i>Type_</i>	
<i>rgb</i>	
<i>lineType</i>	
<i>parent</i>	

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

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

Construct a new [Geometry::Geometry](#) object.

Parameters

<i>vector</i>	
<i>rgb</i>	
<i>lineType</i>	
<i>parent</i>	

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::ItemIsMovable`. 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)`

[Geometry::init_circle.](#)

Parameters

<i>circle</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.

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

[Geometry::init_ellipse.](#)

Parameters

<i>ellipse</i>	
<i>rgb</i>	
<i>lineType</i>	

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)`

[Geometry::init_line.](#)

Parameters

<i>ellipse</i>	
<i>rgb</i>	
<i>lineType</i>	

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.

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

[Geometry::init_line.](#)

Parameters

<i>ellipse</i>	
<i>rgb</i>	
<i>lineType</i>	

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.

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

[Geometry::init_line.](#)

Parameters

<i>ellipse</i>	
<i>rgb</i>	
<i>lineType</i>	

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.

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

[Geometry::init_line.](#)

Parameters

<i>ellipse</i>	
<i>rgb</i>	
<i>lineType</i>	

Warning

DO NOT enable `QGraphicsItem::ItemIsMovable`. 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.

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

[Geometry::init_line.](#)

Parameters

<i>ellipse</i>	
<i>rgb</i>	
<i>lineType</i>	

Warning

DO NOT enable `QGraphicsItem::ItemIsMovable`. 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.

Todo set the justification properly.
 pass in proper linewidth

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 ()

Returns

[QPointF](#)

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

Returns

[QPointF](#)

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 ()

Returns

[EmbReal](#)

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 ()`

Returns

`QPointF`

17.50.4.44 objectQuadrant180() `QPointF objectQuadrant180 ()`

Returns

`QPointF`

17.50.4.45 objectQuadrant270() `QPointF objectQuadrant270 ()`

Returns

`QPointF`

17.50.4.46 objectQuadrant90() `QPointF objectQuadrant90 ()`

Returns

`QPointF`

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 ()`

Returns

The top left corner location as a QPointF.

17.50.4.57 objectTopRight() `QPointF objectTopRight ()`

Returns

QPointF

17.50.4.58 objectWidth() `EmbReal objectWidth ()`

Returns

EmbReal

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)`

Parameters

<i>str</i>	
------------	--

17.50.4.70 script_main() `void script_main (`
`void)`

17.50.4.71 script_prompt() `void script_prompt (`
`String str)`

Parameters

<i>str</i>	
------------	--

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)`

Parameters

<i>val</i>	
------------	--

17.50.4.99 setObjectTextBold() `void setObjectTextBold (`
`bool val)`

Parameters

<i>val</i>	
------------	--

17.50.4.100 setObjectTextFont() `void setObjectTextFont (`
`QString font)`

Parameters

<i>font</i>	
-------------	--

17.50.4.101 setObjectTextItalic() `void setObjectTextItalic (`
`bool val)`

Parameters

<i>val</i>	
------------	--

17.50.4.102 setObjectTextJustify() `void setObjectTextJustify (`
`QString justify)`

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

Parameters

<i>justify</i>	
----------------	--

17.50.4.103 setObjectTextOverline() `void setObjectTextOverline (bool val)`

Parameters

<i>val</i>	
------------	--

17.50.4.104 setObjectTextSize() `void setObjectTextSize (EmbReal size)`

Parameters

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

17.50.4.105 setObjectTextStrikeOut() `void setObjectTextStrikeOut (bool val)`

Parameters

<i>val</i>	
------------	--

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

Parameters

<i>bold</i>	
<i>italic</i>	
<i>under</i>	
<i>strike</i>	
<i>over</i>	

17.50.4.107 setObjectTextUnderline() `void setObjectTextUnderline (bool val)`

Parameters

<i>val</i>	
------------	--

17.50.4.108 setObjectTextUpsideDown() `void setObjectTextUpsideDown (bool val)`

Parameters

<i>val</i>	
------------	--

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 ()`

Returns

`std::vector<QPainterPath>`

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:

- [src/embroidermodder.h](#)
- [src/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

<i>fileName</i>	
-----------------	--

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

<i>fileName</i>	
-----------------	--

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:

- `src/embroidermodder.h`
- `src/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` `QStandardItemModel* 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:

- [src/embroidermodder.h](#)
- [src/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, [StringList](#) 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>	
----------------	--

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:

- [src/embroidermodder.h](#)
- [src/mainwindow-menus.cpp](#)
- [src/mainwindow-toolbars.cpp](#)
- [src/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 (QPainter * 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:

- [src/embroidermodder.h](#)
- [src/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`)
Construct a new `MdiWindow` object.
- `~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`
- `QRgb 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`)

Construct a new `MdiWindow` object.

Parameters

<i>theIndex</i>	
<i>parent</i>	
<i>wflags</i>	

Warning

DO NOT SET THE QMDISUBWINDOW (this) FOCUSPROXY TO THE PROMPT AS IT WILL CAUSE THE WINDOW MENU TO NOT SWITCH WINDOWS PROPERLY! ALTHOUGH IT SEEMS THAT SETTING INTERNAL WIDGETS FOCUSPROXY IS OK.

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]

Parameters

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

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]

Parameters

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

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

fileName	
----------	--

17.58.2.22 setViewBackgroundColor void setViewBackgroundColor (
 QRgb color) [slot]

Parameters

color	
-------	--

17.58.2.23 setViewCrossHairColor void setViewCrossHairColor (
 QRgb color) [slot]

Parameters

color	
-------	--

17.58.2.24 setViewGridColor void setViewGridColor (
 QRgb color) [slot]

Parameters

color	
-------	--

17.58.2.25 setViewRulerColor void setViewRulerColor (
 QRgb color) [slot]

Parameters

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

17.58.2.26 setViewSelectBoxColors `void setViewSelectBoxColors (`
 `QRgb colorL,`
 `QRgb fillL,`
 `QRgb colorR,`
 `QRgb fillR,`
 `int alpha)` [slot]

Parameters

<i>colorL</i>	
<i>fillL</i>	
<i>colorR</i>	
<i>fillR</i>	
<i>alpha</i>	

17.58.2.27 showViewScrollBars `void showViewScrollBars (`
 `bool val)` [slot]

Parameters

<i>val</i>	
------------	--

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:

- [src/embroidermodder.h](#)
- [src/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:

- [src/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:

- [src/embroidermodder.h](#)
- [src/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](#) ()
- QToolButton * [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](#) ()
- QToolButton * [createToolButtonQSelect](#) ()
- QToolButton * [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](#)
- QToolButton * [toolButtonQSelect](#)
- QToolButton * [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:

- [src/embroidermodder.h](#)
- [src/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>	

Todo saving polygons, polylines and paths must be stable before we go here.

Todo This needs to work like a path, not a polyline. Improve this.

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

Todo proper lineType embPattern_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:

- src/[embroidermodder.h](#)
- src/[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:

- src/[embroidermodder.h](#)
- src/[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> |  |
|----------------|--|

**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 createTabFilesPaths()** `QWidget * createTabFilesPaths ( )`

**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:

- `src/embroidermodder.h`
- `src/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:

- `src/embroidermodder.h`
- `src/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:

- [src/embroidermodder.h](#)
- [src/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:

- src/[embroidermodder.h](#)
- src/[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 [setBackgroundColor](#) (QRgb color)
- void [setSelectBoxColors](#) (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 &centerPoint)
- 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:

- [src/embroidermodder.h](#)
- [src/view.cpp](#)

## 17.75 VipHeader\_ Struct Reference

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

## Public Attributes

- int [magicCode](#)
- int [numberOfStitches](#)
- int [numberOfColors](#)
- short [postitiveXHoopSize](#)
- short [postitiveYHoopSize](#)
- 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 postitiveXHoopSize**   short postitiveXHoopSize

**17.75.1.9 postitiveYHoopSize**   short postitiveYHoopSize

**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 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)

#### 18.2.1 Function Documentation

**18.2.1.1 [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.

**18.2.1.2 [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.

**18.2.1.3 [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.

**18.2.1.4 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.

**18.2.1.5 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.

**18.2.1.6 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.

**18.2.1.7 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.

**18.2.1.8 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.

**18.2.1.9 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.

**18.2.1.10 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.

**18.2.1.11 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.

**18.2.1.12 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.

**18.2.1.13 embArray\_copy()** `void embArray_copy (`  
    `EmbArray * dst,`  
    `EmbArray * src )`

Copies all entries in the EmbArray struct from *src* to *dst*.

**18.2.1.14 embArray\_create()** `EmbArray * embArray_create (`  
    `int type )`

Allocates memory for an EmbArray of the type determined by the argument *type*.

**18.2.1.15 embArray\_free()** void embArray\_free (   
 EmbArray \* a )

Free the memory of EmbArray *a*, recursively if necessary.

**18.2.1.16 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.

## 18.3 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]

### 18.3.1 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/HUSandVIPFileFormatInfo.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.

### 18.3.2 Function Documentation

**18.3.2.1 compress\_get\_bits()** int compress\_get\_bits (   
 compress \* c,   
 int length )

*c length* Returns .



**18.3.2.2 compress\_get\_position()** `int compress_get_position (`  
`compress * c )`

`c` . Returns the position as an int.

**18.3.2.3 compress\_get\_token()** `int compress_get_token (`  
`compress * c )`

`c` . Returns the token as an int.

**18.3.2.4 compress\_init()** `void compress_init ( )`

**18.3.2.5 compress\_load\_block()** `void compress_load_block (`  
`compress * c )`

`c` . Returns nothing.

**18.3.2.6 compress\_load\_character\_huffman()** `void compress_load_character_huffman (`  
`compress * c )`

Load character table to compress struct `c`. Returns nothing.

**18.3.2.7 compress\_load\_character\_length\_huffman()** `void compress_load_character_length_huffman`  
`(`  
`compress * c )`

`c` . Returns.

**18.3.2.8 compress\_load\_distance\_huffman()** `void compress_load_distance_huffman (`  
`compress * c )`

`c` . Returns nothing.

**18.3.2.9 compress\_peek()** `int compress_peek (`  
`compress * c,`  
`int bit_count )`

`c bit_count`. Returns.

**18.3.2.10 compress\_pop()** `int compress_pop (`  
`compress * c,`  
`int bit_count )`

`c bit_count` . Returns.

**18.3.2.11 compress\_read\_variable\_length()** `int compress_read_variable_length (`  
`compress * c )`

`c`. Returns.

**18.3.2.12 huffman\_build\_table()** `void huffman_build_table (`  
`huffman * h )`

These next 2 functions represent the [Huffman](#) class in tartarize's code. `h`

**18.3.2.13 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`.

**18.3.2.14** `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).

**18.3.2.15** `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.

### 18.3.3 Variable Documentation

**18.3.3.1** `huffman_lookup_data` `int huffman_lookup_data[2]`

## 18.4 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)  
*t*
- EMB\_PUBLIC EmbTime embTime\_time (EmbTime \*t)  
*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 is↵ 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.*

## 18.4.1 Macro Definition Documentation

**18.4.1.1 Arc\_Polyester** `#define Arc_Polyester 0`

**18.4.1.2 Arc\_Rayon** `#define Arc_Rayon 1`

**18.4.1.3 CHUNK\_SIZE** `#define CHUNK_SIZE 128`

**18.4.1.4 CoatsAndClark\_Rayon** `#define CoatsAndClark_Rayon 2`

**18.4.1.5 dxf\_color** `#define dxf_color 29`

**18.4.1.6 EMB\_ARC** `#define EMB_ARC 1`

**18.4.1.7 EMB\_ARRAY** `#define EMB_ARRAY 0`

**18.4.1.8 EMB\_CIRCLE** `#define EMB_CIRCLE 2`

**18.4.1.9 EMB\_DIM\_DIAMETER** `#define EMB_DIM_DIAMETER 3`

**18.4.1.10 EMB\_DIM\_LEADER** `#define EMB_DIM_LEADER 4`

**18.4.1.11 EMB\_ELLIPSE** `#define EMB_ELLIPSE 5`

**18.4.1.12 EMB\_FLAG** `#define EMB_FLAG 6`

**18.4.1.13 EMB\_FORMAT\_100** `#define EMB_FORMAT_100 0`  
Format identifiers

**18.4.1.14 EMB\_FORMAT\_100** `#define EMB_FORMAT_100 1`

**18.4.1.15 EMB\_FORMAT\_ART** `#define EMB_FORMAT_ART 2`

**18.4.1.16 EMB\_FORMAT\_BMC** `#define EMB_FORMAT_BMC 3`

**18.4.1.17 EMB\_FORMAT\_BRO** `#define EMB_FORMAT_BRO 4`

**18.4.1.18 EMB\_FORMAT\_CND** `#define EMB_FORMAT_CND 5`

**18.4.1.19 EMB\_FORMAT\_COL** `#define EMB_FORMAT_COL 6`

**18.4.1.20 EMB\_FORMAT\_CSD** `#define EMB_FORMAT_CSD 7`

**18.4.1.21 EMB\_FORMAT\_CSV** `#define EMB_FORMAT_CSV 8`

**18.4.1.22 EMB\_FORMAT\_DAT** `#define EMB_FORMAT_DAT 9`

**18.4.1.23 EMB\_FORMAT\_DEM** `#define EMB_FORMAT_DEM 10`

- 18.4.1.24 EMB\_FORMAT\_DSB** `#define EMB_FORMAT_DSB 11`
- 18.4.1.25 EMB\_FORMAT\_DST** `#define EMB_FORMAT_DST 12`
- 18.4.1.26 EMB\_FORMAT\_DSZ** `#define EMB_FORMAT_DSZ 13`
- 18.4.1.27 EMB\_FORMAT\_DXF** `#define EMB_FORMAT_DXF 14`
- 18.4.1.28 EMB\_FORMAT\_EDR** `#define EMB_FORMAT_EDR 15`
- 18.4.1.29 EMB\_FORMAT\_EMD** `#define EMB_FORMAT_EMD 16`
- 18.4.1.30 EMB\_FORMAT\_EXP** `#define EMB_FORMAT_EXP 17`
- 18.4.1.31 EMB\_FORMAT\_EXY** `#define EMB_FORMAT_EXY 18`
- 18.4.1.32 EMB\_FORMAT\_EYS** `#define EMB_FORMAT_EYS 19`
- 18.4.1.33 EMB\_FORMAT\_FXY** `#define EMB_FORMAT_FXY 20`
- 18.4.1.34 EMB\_FORMAT\_GC** `#define EMB_FORMAT_GC 21`
- 18.4.1.35 EMB\_FORMAT\_GNC** `#define EMB_FORMAT_GNC 22`
- 18.4.1.36 EMB\_FORMAT\_GT** `#define EMB_FORMAT_GT 23`
- 18.4.1.37 EMB\_FORMAT\_HUS** `#define EMB_FORMAT_HUS 24`
- 18.4.1.38 EMB\_FORMAT\_INB** `#define EMB_FORMAT_INB 25`
- 18.4.1.39 EMB\_FORMAT\_INF** `#define EMB_FORMAT_INF 26`
- 18.4.1.40 EMB\_FORMAT\_JEF** `#define EMB_FORMAT_JEF 27`
- 18.4.1.41 EMB\_FORMAT\_KSM** `#define EMB_FORMAT_KSM 28`

- 18.4.1.42 EMB\_FORMAT\_MAX** #define EMB\_FORMAT\_MAX 29
  
- 18.4.1.43 EMB\_FORMAT\_MIT** #define EMB\_FORMAT\_MIT 30
  
- 18.4.1.44 EMB\_FORMAT\_NEW** #define EMB\_FORMAT\_NEW 31
  
- 18.4.1.45 EMB\_FORMAT\_OFM** #define EMB\_FORMAT\_OFM 32
  
- 18.4.1.46 EMB\_FORMAT\_PCD** #define EMB\_FORMAT\_PCD 33
  
- 18.4.1.47 EMB\_FORMAT\_PCM** #define EMB\_FORMAT\_PCM 34
  
- 18.4.1.48 EMB\_FORMAT\_PCQ** #define EMB\_FORMAT\_PCQ 35
  
- 18.4.1.49 EMB\_FORMAT\_PCS** #define EMB\_FORMAT\_PCS 36
  
- 18.4.1.50 EMB\_FORMAT\_PEC** #define EMB\_FORMAT\_PEC 37
  
- 18.4.1.51 EMB\_FORMAT\_PEL** #define EMB\_FORMAT\_PEL 38
  
- 18.4.1.52 EMB\_FORMAT\_PEM** #define EMB\_FORMAT\_PEM 39
  
- 18.4.1.53 EMB\_FORMAT\_PES** #define EMB\_FORMAT\_PES 40
  
- 18.4.1.54 EMB\_FORMAT\_PHB** #define EMB\_FORMAT\_PHB 41
  
- 18.4.1.55 EMB\_FORMAT\_PHC** #define EMB\_FORMAT\_PHC 42
  
- 18.4.1.56 EMB\_FORMAT\_PLT** #define EMB\_FORMAT\_PLT 43
  
- 18.4.1.57 EMB\_FORMAT\_RGB** #define EMB\_FORMAT\_RGB 44
  
- 18.4.1.58 EMB\_FORMAT\_SEW** #define EMB\_FORMAT\_SEW 45
  
- 18.4.1.59 EMB\_FORMAT\_SHV** #define EMB\_FORMAT\_SHV 46

- 18.4.1.60 **EMB\_FORMAT\_SST** `#define EMB_FORMAT_SST 47`
- 18.4.1.61 **EMB\_FORMAT\_STX** `#define EMB_FORMAT_STX 48`
- 18.4.1.62 **EMB\_FORMAT\_SVG** `#define EMB_FORMAT_SVG 49`
- 18.4.1.63 **EMB\_FORMAT\_T01** `#define EMB_FORMAT_T01 50`
- 18.4.1.64 **EMB\_FORMAT\_T09** `#define EMB_FORMAT_T09 51`
- 18.4.1.65 **EMB\_FORMAT\_TAP** `#define EMB_FORMAT_TAP 52`
- 18.4.1.66 **EMB\_FORMAT\_THR** `#define EMB_FORMAT_THR 53`
- 18.4.1.67 **EMB\_FORMAT\_TXT** `#define EMB_FORMAT_TXT 54`
- 18.4.1.68 **EMB\_FORMAT\_U00** `#define EMB_FORMAT_U00 55`
- 18.4.1.69 **EMB\_FORMAT\_U01** `#define EMB_FORMAT_U01 56`
- 18.4.1.70 **EMB\_FORMAT\_VIP** `#define EMB_FORMAT_VIP 57`
- 18.4.1.71 **EMB\_FORMAT\_VP3** `#define EMB_FORMAT_VP3 58`
- 18.4.1.72 **EMB\_FORMAT\_XXX** `#define EMB_FORMAT_XXX 59`
- 18.4.1.73 **EMB\_FORMAT\_ZSK** `#define EMB_FORMAT_ZSK 60`
- 18.4.1.74 **EMB\_IMAGE** `#define EMB_IMAGE 8`
- 18.4.1.75 **EMB\_LINE** `#define EMB_LINE 7`
- 18.4.1.76 **EMB\_MAX\_LAYERS** `#define EMB_MAX_LAYERS 10`
- 18.4.1.77 **EMB\_PATH** `#define EMB_PATH 9`

**18.4.1.78 EMB\_POINT** #define EMB\_POINT 10

**18.4.1.79 EMB\_POLYGON** #define EMB\_POLYGON 11

**18.4.1.80 EMB\_POLYLINE** #define EMB\_POLYLINE 12

**18.4.1.81 EMB\_PUBLIC** #define EMB\_PUBLIC

**18.4.1.82 EMB\_RECT** #define EMB\_RECT 13

**18.4.1.83 EMB\_SPLINE** #define EMB\_SPLINE 14

**18.4.1.84 EMB\_STITCH** #define EMB\_STITCH 15

**18.4.1.85 EMB\_TEXT\_MULTI** #define EMB\_TEXT\_MULTI 17

**18.4.1.86 EMB\_TEXT\_SINGLE** #define EMB\_TEXT\_SINGLE 16

**18.4.1.87 EMB\_THREAD** #define EMB\_THREAD 19

**18.4.1.88 EMB\_VECTOR** #define EMB\_VECTOR 18

**18.4.1.89 EMBFORMAT\_MAXDESC** #define EMBFORMAT\_MAXDESC 50

**18.4.1.90 EMBFORMAT\_MAXEXT** #define EMBFORMAT\_MAXEXT 3

**18.4.1.91 EMBFORMAT\_OBJECTONLY** #define EMBFORMAT\_OBJECTONLY 2

**18.4.1.92 EMBFORMAT\_STCHANDOBJ** #define EMBFORMAT\_STCHANDOBJ 3 /\* binary operation↔  
: 1+2=3 \*/

**18.4.1.93 EMBFORMAT\_STITCHONLY** #define EMBFORMAT\_STITCHONLY 1

**18.4.1.94 EMBFORMAT\_UNSUPPORTED** #define EMBFORMAT\_UNSUPPORTED 0

**18.4.1.95 END** #define END 16 /\*! end of program \*/

**18.4.1.96 Exquisite\_Polyester** `#define Exquisite_Polyester 3`

**18.4.1.97 Fufu\_Polyester** `#define Fufu_Polyester 4`

**18.4.1.98 Fufu\_Rayon** `#define Fufu_Rayon 5`

**18.4.1.99 Hemingworth\_Polyester** `#define Hemingworth_Polyester 6`

**18.4.1.100 hus\_thread** `#define hus_thread 24`

**18.4.1.101 Isacord\_Polyester** `#define Isacord_Polyester 7`

**18.4.1.102 Isafil\_Rayon** `#define Isafil_Rayon 8`

**18.4.1.103 jef\_thread** `#define jef_thread 25`

**18.4.1.104 JUMP** `#define JUMP 1 /*! move to (x, y) */`

**18.4.1.105 LIBEMBROIDERY\_EMBEDDED\_VERSION** `#define LIBEMBROIDERY_EMBEDDED_VERSION 0`

**18.4.1.106 Madeira\_Polyester** `#define Madeira_Polyester 11`

**18.4.1.107 Madeira\_Rayon** `#define Madeira_Rayon 12`

**18.4.1.108 Marathon\_Polyester** `#define Marathon_Polyester 9`

**18.4.1.109 Marathon\_Rayon** `#define Marathon_Rayon 10`

**18.4.1.110 MAX\_STITCHES** `#define MAX_STITCHES 1000000`

**18.4.1.111 MAX\_THREADS** `#define MAX_THREADS 256`

**18.4.1.112 Metro\_Polyester** `#define Metro_Polyester 13`

**18.4.1.113 NORMAL** `#define NORMAL 0 /*! stitch to (x, y) */`  
Machine codes for stitch flags

**18.4.1.114 numberOfFormats** `#define numberOfFormats 61`

**18.4.1.115 Pantone** `#define Pantone 14`

**18.4.1.116 pcm\_thread** `#define pcm_thread 26`

**18.4.1.117 pec\_thread** `#define pec_thread 27`

**18.4.1.118 RobisonAnton\_Polyester** `#define RobisonAnton_Polyester 15`

**18.4.1.119 RobisonAnton\_Rayon** `#define RobisonAnton_Rayon 16`

**18.4.1.120 SEQUIN** `#define SEQUIN 8 /*! sequin */`

**18.4.1.121 shv\_thread** `#define shv_thread 28`

**18.4.1.122 Sigma\_Polyester** `#define Sigma_Polyester 17`

**18.4.1.123 STOP** `#define STOP 4 /*! pause machine for thread change */`

**18.4.1.124 Sulky\_Rayon** `#define Sulky_Rayon 18`

**18.4.1.125 SVG\_Colors** `#define SVG_Colors 23`

**18.4.1.126 ThreadArt\_Polyester** `#define ThreadArt_Polyester 20`

**18.4.1.127 ThreadArt\_Rayon** `#define ThreadArt_Rayon 19`

**18.4.1.128 ThreaDelight\_Polyester** `#define ThreaDelight_Polyester 21`

**18.4.1.129 TRIM** `#define TRIM 2 /*! trim + move to (x, y) */`

**18.4.1.130 Z102\_Isacord\_Polyester** `#define Z102_Isacord_Polyester 22`

## **18.4.2 Typedef Documentation**



**18.4.2.1 EmbAlignedDim** typedef struct [EmbAlignedDim\\_](#) EmbAlignedDim

**18.4.2.2 EmbAngularDim** typedef struct [EmbAngularDim\\_](#) EmbAngularDim

**18.4.2.3 EmbArc** typedef struct [EmbArc\\_](#) EmbArc  
absolute position (not relative)

**18.4.2.4 EmbArcLengthDim** typedef struct [EmbArcLengthDim\\_](#) EmbArcLengthDim

**18.4.2.5 EmbArray** typedef struct [EmbArray\\_](#) EmbArray  
The basic array type.

**18.4.2.6 EmbBezier** typedef struct [EmbBezier\\_](#) EmbBezier

**18.4.2.7 EmbBlock** typedef struct [EmbBlock\\_](#) EmbBlock

**18.4.2.8 EmbCircle** typedef struct [EmbCircle\\_](#) EmbCircle

**18.4.2.9 EmbColor** typedef struct [EmbColor\\_](#) EmbColor  
EmbColor uses the light primaries: red, green, blue in that order.

**18.4.2.10 EmbDiameterDim** typedef struct [EmbDiameterDim\\_](#) EmbDiameterDim

**18.4.2.11 EmbEllipse** typedef struct [EmbEllipse\\_](#) EmbEllipse

**18.4.2.12 EmbFlag** typedef int [EmbFlag](#)

**18.4.2.13 EmbFormatList** typedef struct [EmbFormatList\\_](#) EmbFormatList

**18.4.2.14 EmbGeometry** typedef struct [EmbGeometry\\_](#) EmbGeometry

**18.4.2.15 EmbImage** typedef struct [EmbImage\\_](#) EmbImage

**18.4.2.16 EmbInfiniteLine** typedef struct [EmbInfiniteLine\\_](#) EmbInfiniteLine

**18.4.2.17 EmbLayer** typedef struct [EmbLayer\\_](#) EmbLayer

**18.4.2.18 EmbLeaderDim** typedef struct [EmbLeaderDim\\_](#) EmbLeaderDim

**18.4.2.19 EmbLine** typedef struct [EmbLine\\_](#) EmbLine

**18.4.2.20 EmbLinearDim** typedef struct [EmbLinearDim\\_](#) EmbLinearDim

**18.4.2.21 EmbOrdinateDim** typedef struct [EmbOrdinateDim\\_](#) EmbOrdinateDim

**18.4.2.22 EmbPath** typedef struct [EmbPath\\_](#) EmbPath

**18.4.2.23 EmbPattern** typedef struct [EmbPattern\\_](#) EmbPattern

**18.4.2.24 EmbPoint** typedef struct [EmbPoint\\_](#) EmbPoint

**18.4.2.25 EmbPolygon** typedef [EmbPath](#) EmbPolygon

**18.4.2.26 EmbPolyline** typedef [EmbPath](#) EmbPolyline

**18.4.2.27 EmbRadiusDim** typedef struct [EmbRadiusDim\\_](#) EmbRadiusDim

**18.4.2.28 EmbRay** typedef struct [EmbRay\\_](#) EmbRay

**18.4.2.29 EmbReal** typedef float [EmbReal](#)

**18.4.2.30 EmbRect** typedef struct [EmbRect\\_](#) EmbRect

**18.4.2.31 EmbSatinOutline** typedef struct [EmbSatinOutline\\_](#) EmbSatinOutline

**18.4.2.32 EmbSpline** typedef struct [EmbSpline\\_](#) EmbSpline

**18.4.2.33 EmbStitch** typedef struct [EmbStitch\\_](#) EmbStitch

**18.4.2.34 EmbTextMulti** typedef struct [EmbTextMulti\\_](#) EmbTextMulti

**18.4.2.35 EmbTextSingle** typedef struct [EmbTextSingle\\_](#) EmbTextSingle

**18.4.2.36 EmbThread** typedef struct [EmbThread\\_](#) EmbThread

**18.4.2.37 EmbTime** typedef struct `EmbTime_ EmbTime`

**18.4.2.38 EmbVector** typedef struct `EmbVector_ EmbVector`

The basic type to represent points absolutely or represent directions.  
Positive y is up, units are in mm.

**18.4.2.39 L\_system** typedef struct `LSYSTEM L_system`

**18.4.2.40 thread\_color** typedef struct `thread_color_ thread_color`

### 18.4.3 Function Documentation

**18.4.3.1 convert()** `EMB_PUBLIC int convert (`  
     `const char * inf,`  
     `const char * outf )`

**18.4.3.2 degrees()** `EMB_PUBLIC EmbReal degrees (`  
     `EmbReal radian )`

**18.4.3.3 emb\_identify\_format()** `EMB_PUBLIC int emb_identify_format (`  
     `const char * fileName )`

*fileName*

Returns

int

**18.4.3.4 emb\_round()** `EMB_PUBLIC int emb_round (`  
     `EmbReal x )`

**18.4.3.5 embArc\_clockwise()** `EMB_PUBLIC char embArc_clockwise (`  
     `EmbArc arc )`

**18.4.3.6 embArc\_init()** `EMB_PUBLIC EmbArc embArc_init (`  
     `void )`

**18.4.3.7 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.

**18.4.3.8 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.

**18.4.3.9 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.

**18.4.3.10 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.

**18.4.3.11 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.

**18.4.3.12 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.

**18.4.3.13 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.

**18.4.3.14 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.

**18.4.3.15 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.

**18.4.3.16 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.

**18.4.3.17 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.

**18.4.3.18 embArray\_addThread()** `EMB_PUBLIC int embArray_addThread (`  
    `EmbArray * g,`  
    `EmbThread p )`

**18.4.3.19 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.

**18.4.3.20 embArray\_copy()** `EMB_PUBLIC void embArray_copy (`  
     `EmbArray * dst,`  
     `EmbArray * src )`

Copies all entries in the EmbArray struct from *src* to *dst*.

**18.4.3.21 embArray\_create()** `EMB_PUBLIC EmbArray * embArray_create (`  
     `int type )`

Allocates memory for an EmbArray of the type determined by the argument *type*.

**18.4.3.22 embArray\_free()** `EMB_PUBLIC void embArray_free (`  
     `EmbArray * a )`

Free the memory of EmbArray *a*, recursively if necessary.

**18.4.3.23 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.

**18.4.3.24 embCircle\_init()** `EMB_PUBLIC EmbCircle embCircle_init (`  
     `void )`

**18.4.3.25 embColor\_create()** `EMB_PUBLIC EmbColor * embColor_create (`  
     `unsigned char r,`  
     `unsigned char g,`  
     `unsigned char b )`

**18.4.3.26 embColor\_distance()** `EMB_PUBLIC int embColor_distance (`  
     `EmbColor a,`  
     `EmbColor b )`

*a b*

Returns

int

**18.4.3.27 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.

**18.4.3.28 embColor\_make()** `EMB_PUBLIC EmbColor embColor_make (`  
     `unsigned char r,`  
     `unsigned char g,`  
     `unsigned char b )`

**18.4.3.29 embEllipse\_area()** `EMB_PUBLIC EmbReal embEllipse_area (`  
     `EmbEllipse ellipse )`

**18.4.3.30 embEllipse\_diameterX()** `EMB_PUBLIC EmbReal embEllipse_diameterX ( EmbEllipse ellipse )`

**18.4.3.31 embEllipse\_diameterY()** `EMB_PUBLIC EmbReal embEllipse_diameterY ( EmbEllipse ellipse )`

**18.4.3.32 embEllipse\_height()** `EMB_PUBLIC EmbReal embEllipse_height ( EmbEllipse ellipse )`

**18.4.3.33 embEllipse\_init()** `EMB_PUBLIC EmbEllipse embEllipse_init ( void )`

**18.4.3.34 embEllipse\_make()** `EMB_PUBLIC EmbEllipse embEllipse_make ( EmbReal cx, EmbReal cy, EmbReal rx, EmbReal ry )`

**18.4.3.35 embEllipse\_perimeter()** `EMB_PUBLIC EmbReal embEllipse_perimeter ( EmbEllipse ellipse )`

**18.4.3.36 embEllipse\_width()** `EMB_PUBLIC EmbReal embEllipse_width ( EmbEllipse ellipse )`

**18.4.3.37 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).

**18.4.3.38 embGeometry\_free()** `EMB_PUBLIC void embGeometry_free ( EmbGeometry * obj )`

Free the memory occupied by a non-stitch geometry object.  
*obj* Pointer to geometry memory.

**18.4.3.39 embGeometry\_init()** `EMB_PUBLIC EmbGeometry * embGeometry_init ( int type_in )`

Our generic object interface backends to each individual type.  
*type\_in*

#### Returns

EmbGeometry\*

**18.4.3.40 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.

**18.4.3.41 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.

**18.4.3.42 emblImage\_create()** `EMB_PUBLIC EmbImage emblImage_create (`  
     `int ,`  
     `int )`

**18.4.3.43 emblImage\_free()** `EMB_PUBLIC void emblImage_free (`  
     `EmbImage * image )`

**18.4.3.44 emblImage\_read()** `EMB_PUBLIC void emblImage_read (`  
     `EmbImage * image,`  
     `char * fname )`

**18.4.3.45 emblImage\_write()** `EMB_PUBLIC int emblImage_write (`  
     `EmbImage * image,`  
     `char * fname )`

**18.4.3.46 embLine\_intersectionPoint()** `EMB_PUBLIC EmbVector embLine_intersectionPoint (`  
     `EmbLine line1,`  
     `EmbLine line2 )`

**18.4.3.47 embLine\_make()** `EMB_PUBLIC EmbLine embLine_make (`  
     `EmbReal x1,`  
     `EmbReal y1,`  
     `EmbReal x2,`  
     `EmbReal y2 )`

**18.4.3.48 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)

**18.4.3.49 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.

**18.4.3.50 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.

**18.4.3.51 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.

**18.4.3.52 embPattern\_addPathAbs()** `EMB_PUBLIC void embPattern_addPathAbs (`  
    `EmbPattern * p,`  
    `EmbPath obj )`

**18.4.3.53 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.

**18.4.3.54 embPattern\_addPolygonAbs()** `EMB_PUBLIC void embPattern_addPolygonAbs (`  
    `EmbPattern * p,`  
    `EmbPolygon obj )`

**18.4.3.55 embPattern\_addPolylineAbs()** `EMB_PUBLIC void embPattern_addPolylineAbs (`  
    `EmbPattern * p,`  
    `EmbPolyline obj )`

**18.4.3.56 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.

**18.4.3.57 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.

**18.4.3.58 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.



**18.4.3.59 embPattern\_addThread()** `EMB_PUBLIC int embPattern_addThread (`  
`EmbPattern * pattern,`  
`EmbThread thread )`

*pattern thread*

Returns

int

**18.4.3.60 embPattern\_calcBoundingBox()** `EMB_PUBLIC EmbRect embPattern_calcBoundingBox (`  
`EmbPattern * p )`

Returns an EmbRect that encapsulates all stitches and objects in the pattern (*p*).

**18.4.3.61 embPattern\_center()** `EMB_PUBLIC void embPattern_center (`  
`EmbPattern * p )`

Center the pattern *p*.

**18.4.3.62 embPattern\_changeColor()** `EMB_PUBLIC void embPattern_changeColor (`  
`EmbPattern * p,`  
`int index )`

Change the currentColorIndex of pattern *p* to *index*.

**18.4.3.63 embPattern\_color\_count()** `EMB_PUBLIC int embPattern_color_count (`  
`EmbPattern * pattern,`  
`EmbColor startColor )`

**18.4.3.64 embPattern\_combine()** `EMB_PUBLIC EmbPattern * embPattern_combine (`  
`EmbPattern * p1,`  
`EmbPattern * p2 )`

*p1 p2*

Returns

EmbPattern\*

**18.4.3.65 embPattern\_combineJumpStitches()** `EMB_PUBLIC void embPattern_combineJumpStitches (`  
`EmbPattern * p )`

*p*

**18.4.3.66 embPattern\_convertGeometry()** `EMB_PUBLIC void embPattern_convertGeometry (`  
`EmbPattern * p )`

*p*

**18.4.3.67 embPattern\_copyPolylinesToStitchList()** `EMB_PUBLIC void embPattern_copyPolylinesTo↵`  
`StitchList (`  
`EmbPattern * pattern )`

**18.4.3.68 embPattern\_copyStitchListToPolylines()** `EMB_PUBLIC void embPattern_copyStitchListTo↵`  
`Polylines (`  
`EmbPattern * pattern )`

**18.4.3.69** `embPattern_correctForMaxStitchLength()` `EMB_PUBLIC` `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.

**18.4.3.70** `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*`

**18.4.3.71** `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.

**18.4.3.72** `embPattern_designDetails()` `EMB_PUBLIC` `void embPattern_designDetails (`  
`EmbPattern * p )`

**18.4.3.73** `embPattern_end()` `EMB_PUBLIC` `void embPattern_end (`  
`EmbPattern * p )`

**18.4.3.74** `embPattern_fixColorCount()` `EMB_PUBLIC` `void embPattern_fixColorCount (`  
`EmbPattern * p )`

*p*

**18.4.3.75** `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.

**18.4.3.76** `embPattern_flipHorizontal()` `EMB_PUBLIC` `void embPattern_flipHorizontal (`  
`EmbPattern * p )`

Flips the entire pattern (*p*) horizontally about the y-axis.

**18.4.3.77** `embPattern_flipVertical()` `EMB_PUBLIC` `void embPattern_flipVertical (`  
`EmbPattern * p )`

Flips the entire pattern (*p*) vertically about the x-axis.

**18.4.3.78 embPattern\_free()** `EMB_PUBLIC void embPattern_free (`  
`EmbPattern * p )`

Frees all memory allocated in the pattern (*p*).

**18.4.3.79 embPattern\_hideStitchesOverLength()** `EMB_PUBLIC void embPattern_hideStitchesOverLength`  
`(`  
`EmbPattern * p,`  
`int length )`

*p length*

**18.4.3.80 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.

**18.4.3.81 embPattern\_jumpStitches()** `EMB_PUBLIC int embPattern_jumpStitches (`  
`EmbPattern * pattern )`

**18.4.3.82 embPattern\_lengthHistogram()** `EMB_PUBLIC void embPattern_lengthHistogram (`  
`EmbPattern * pattern,`  
`int * bin,`  
`int NUMBINS )`

**18.4.3.83 embPattern\_loadExternalColorFile()** `EMB_PUBLIC void embPattern_loadExternalColorFile (`  
`EmbPattern * p,`  
`const char * fileName )`

TODO: Description needed.

**18.4.3.84 embPattern\_maximumStitchLength()** `EMB_PUBLIC EmbReal embPattern_maximumStitchLength`  
`(`  
`EmbPattern * pattern )`

**18.4.3.85 embPattern\_minimumStitchLength()** `EMB_PUBLIC EmbReal embPattern_minimumStitchLength (`  
`EmbPattern * pattern )`

**18.4.3.86 embPattern\_movePolylinesToStitchList()** `EMB_PUBLIC void embPattern_movePolylinesTo↵`  
`StitchList (`  
`EmbPattern * pattern )`

**18.4.3.87 embPattern\_moveStitchListToPolylines()** `EMB_PUBLIC void embPattern_moveStitchListTo↵`  
`Polylines (`  
`EmbPattern * pattern )`

**18.4.3.88 embPattern\_read()** `EMB_PUBLIC` char embPattern\_read (   
     EmbPattern \* pattern,   
     const char \* fileName,   
     int format )

*pattern fileName format*

Returns

char

**18.4.3.89 embPattern\_readAuto()** `EMB_PUBLIC` char embPattern\_readAuto (   
     EmbPattern \* pattern,   
     const char \* fileName )

*pattern fileName*

Returns

char

**18.4.3.90 embPattern\_realStitches()** `EMB_PUBLIC` int embPattern\_realStitches (   
     EmbPattern \* pattern )

**18.4.3.91 embPattern\_render()** `EMB_PUBLIC` int embPattern\_render (   
     EmbPattern \* pattern,   
     char \* fname )

**18.4.3.92 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.

**18.4.3.93 embPattern\_simulate()** `EMB_PUBLIC` int embPattern\_simulate (   
     EmbPattern \* pattern,   
     char \* fname )

**18.4.3.94 embPattern\_totalStitchLength()** `EMB_PUBLIC` EmbReal embPattern\_totalStitchLength (   
     EmbPattern \* pattern )

*pattern*

Returns

float

**18.4.3.95 embPattern\_trimStitches()** `EMB_PUBLIC` int embPattern\_trimStitches (   
     EmbPattern \* pattern )

**18.4.3.96 embPattern\_write()** `EMB_PUBLIC` char embPattern\_write (   
     EmbPattern \* pattern,   
     const char \* fileName,   
     int format )

*pattern fileName format*

Returns

char

**18.4.3.97 embPattern\_writeAuto()** `EMB_PUBLIC` char embPattern\_writeAuto (   
     EmbPattern \* pattern,   
     const char \* fileName )

*pattern fileName*

Returns

char

**18.4.3.98 embRect\_area()** `EMB_PUBLIC` EmbReal embRect\_area (   
     EmbRect rect )

**18.4.3.99 embRect\_init()** `EMB_PUBLIC` EmbRect embRect\_init (   
     void )

**18.4.3.100 embSatinOutline\_generateSatinOutline()** `EMB_PUBLIC` void embSatinOutline\_generate↵   
 SatinOutline (   
     EmbArray \* lines,   
     EmbReal thickness,   
     EmbSatinOutline \* result )

*lines thickness result*

**18.4.3.101 embSatinOutline\_renderStitches()** `EMB_PUBLIC` EmbArray \* embSatinOutline\_renderStitches   
 (   
     EmbSatinOutline \* result,   
     EmbReal density )

*result density*

Returns

EmbArray\*

**18.4.3.102 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.

**18.4.3.103 embThread\_findNearestThread()** `EMB_PUBLIC int embThread_findNearestThread (`  
    `EmbColor color,`  
    `EmbThread * thread_list,`  
    `int n_threads )`  
*color thread\_list n\_threads*

**Returns**

int

**18.4.3.104 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.

**18.4.3.105 embTime\_initNow()** `EMB_PUBLIC void embTime_initNow (`  
    `EmbTime * t )`  
*t*

**18.4.3.106 embTime\_time()** `EMB_PUBLIC EmbTime embTime_time (`  
    `EmbTime * t )`  
*t*

**Returns**

EmbTime

**18.4.3.107 embVector\_add()** `EMB_PUBLIC EmbVector embVector_add (`  
    `EmbVector a,`  
    `EmbVector b )`

The sum of vectors *a* and *b* returned as a vector.

Equivalent to:

$$\mathbf{c} = \mathbf{a} + \mathbf{b} = \begin{pmatrix} a_x + b_x \\ a_y + b_y \end{pmatrix}$$

**18.4.3.108 embVector\_angle()** `EMB_PUBLIC EmbReal embVector_angle (`  
    `EmbVector v )`

The angle, measured anti-clockwise from the x-axis, of a vector *v*.

**18.4.3.109 embVector\_average()** `EMB_PUBLIC EmbVector embVector_average ( EmbVector a, EmbVector b )`

The average of vectors *v1* and *v2* returned as a vector.  
Equivalent to:

$$\mathbf{c} = \frac{\mathbf{a} + \mathbf{b}}{2} = \begin{pmatrix} \frac{a_x + b_x}{2} \\ \frac{a_y + b_y}{2} \end{pmatrix}$$

**18.4.3.110 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:

$$|c| = \left| \begin{pmatrix} a_x \\ a_y \\ 0 \end{pmatrix} \times \begin{pmatrix} b_x \\ b_y \\ 0 \end{pmatrix} \right| = \left| \begin{pmatrix} 0 \\ 0 \\ a_x b_y - a_y b_x \end{pmatrix} \right| = a_x b_y - a_y b_x$$

**18.4.3.111 embVector\_distance()** `EMB_PUBLIC EmbReal embVector_distance ( EmbVector a, EmbVector b )`

The distance between *a* and *b* returned as a real value.

$$d = |\mathbf{a} - \mathbf{b}| = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}$$

**18.4.3.112 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:

$$c = \mathbf{a} \cdot \mathbf{b} = a_x b_x + a_y b_y$$

**18.4.3.113 embVector\_length()** `EMB_PUBLIC EmbReal embVector_length ( EmbVector vector )`

The length or absolute value of the vector *vector*.  
Equivalent to:

$$|v| = \sqrt{v_x^2 + v_y^2}$$

**18.4.3.114 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.

**18.4.3.115 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.

**18.4.3.116 embVector\_relativeX()** `EMB_PUBLIC EmbReal embVector_relativeX (`  
`EmbVector a1,`  
`EmbVector a2,`  
`EmbVector a3 )`

The x-component of the vector

**18.4.3.117 embVector\_relativeY()** `EMB_PUBLIC EmbReal embVector_relativeY (`  
`EmbVector a1,`  
`EmbVector a2,`  
`EmbVector a3 )`

The y-component of the vector

**18.4.3.118 embVector\_subtract()** `EMB_PUBLIC EmbVector embVector_subtract (`  
`EmbVector v1,`  
`EmbVector v2 )`

The difference between vectors *v1* and *v2* returned as *result*.

Equivalent to:

$$\mathbf{c} = \mathbf{a} - \mathbf{b} = \begin{pmatrix} a_x - b_x \\ a_y - b_y \end{pmatrix}$$

**18.4.3.119 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  $\begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} xa \\ yb \end{pmatrix}$

**18.4.3.120 embVector\_unit()** `EMB_PUBLIC EmbVector embVector_unit (`  
`EmbReal alpha )`

The unit vector in the direction *angle*.

$$\mathbf{a}_\alpha = \begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \end{pmatrix}$$

**18.4.3.121 full\_test\_matrix()** `EMB_PUBLIC int full_test_matrix (`  
`char * fname )`

**18.4.3.122 getArcCenter()** `EMB_PUBLIC void getArcCenter (`  
`EmbArc arc,`  
`EmbVector * arcCenter )`



**18.4.3.123 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)

```

**18.4.3.124 getCircleCircleIntersections()** `EMB_PUBLIC` int getCircleCircleIntersections (

```

 EmbCircle c0,
 EmbCircle c1,
 EmbVector * v0,
 EmbVector * v1)

```

**18.4.3.125 getCircleTangentPoints()** `EMB_PUBLIC` int getCircleTangentPoints (

```

 EmbCircle c,
 EmbVector p,
 EmbVector * v0,
 EmbVector * v1)

```

**18.4.3.126 hilbert\_curve()** `EMB_PUBLIC` int hilbert\_curve (

```

 EmbPattern * pattern,
 int iterations)

```

*pattern iterations*

[https://en.wikipedia.org/wiki/Hilbert\\_curve](https://en.wikipedia.org/wiki/Hilbert_curve)

Using the Lindenmayer System, so we can save work across different functions.

**18.4.3.127 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.

**18.4.3.128 radians()** `EMB_PUBLIC` EmbReal radians (

```

 EmbReal degree)

```

**18.4.3.129 report()** `EMB_PUBLIC` void report (

```

 int result,
 char * label)

```

**18.4.3.130 testMain()** `EMB_PUBLIC` void testMain (  
    int level )

**18.4.3.131 threadColor()** `EMB_PUBLIC` int threadColor (  
    const char \* name,  
    int brand )

**18.4.3.132 threadColorName()** `EMB_PUBLIC` const char \* threadColorName (  
    unsigned int color,  
    int brand )

**18.4.3.133 threadColorNum()** `EMB_PUBLIC` int threadColorNum (  
    unsigned int color,  
    int brand )

#### 18.4.4 Variable Documentation

**18.4.4.1 \_dxfColorTable** const unsigned char \_dxfColorTable[][3] [extern]

**18.4.4.2 black\_thread** `EmbThread` black\_thread [extern]

**18.4.4.3 emb\_error** int emb\_error [extern]  
Error code storage for optional control flow blocking.

**18.4.4.4 emb\_verbose** int emb\_verbose [extern]  
Verbosity level.

**18.4.4.5 embConstantPi** const `EmbReal` embConstantPi [extern]

**18.4.4.6 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). \*

**18.4.4.7 husThreads** const `EmbThread` husThreads[] [extern]

**18.4.4.8 jefThreads** const `EmbThread` jefThreads[] [extern]

**18.4.4.9 pcmThreads** const `EmbThread` pcmThreads[] [extern]

**18.4.4.10** `pecThreadCount` `const int pecThreadCount` `[extern]`

**18.4.4.11** `pecThreads` `const EmbThread pecThreads[]` `[extern]`

**18.4.4.12** `shvThreadCount` `const int shvThreadCount` `[extern]`

**18.4.4.13** `shvThreads` `const EmbThread shvThreads[]` `[extern]`

**18.4.4.14** `vipDecodingTable` `const unsigned char vipDecodingTable[]` `[extern]`

**18.4.4.15** **Embroidery Format (.pcq)** The Pfaff vip format is stitch-only.

## 18.5 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_FYS 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
192 typedef struct EmbColor_
193 {
194 unsigned char r;
195 unsigned char g;
196 unsigned char b;
197 } EmbColor;
198
199
200 typedef struct EmbVector_
201 {
202 EmbReal x;
203 EmbReal y;
204 } EmbVector;
205
206
207 typedef struct EmbArray_ EmbArray;
208
209
210 typedef struct EmbImage_ {
211 EmbVector position;
212 EmbVector dimensions;
213 unsigned char* data;
214 int width;
215 int height;
216 char path[200];
217 char name[200];
218 } EmbImage;
219
220
221 typedef struct EmbBlock_ {
222 EmbVector position;
223 } EmbBlock;
224
225
226 typedef struct EmbAlignedDim_ {
227 EmbVector position;
228 } EmbAlignedDim;
229
230
231 typedef struct EmbAngularDim_ {
232 EmbVector position;
233 } EmbAngularDim;
234
235
236 typedef struct EmbArcLengthDim_ {
237 EmbVector position;
238 } EmbArcLengthDim;
239
240
241 typedef struct EmbDiameterDim_ {
242 EmbVector position;
243 } EmbDiameterDim;
244
245
246 typedef struct EmbLeaderDim_ {
247 EmbVector position;
248 } EmbLeaderDim;
249
250
251 typedef struct EmbLinearDim_ {
252 EmbVector position;
253 } EmbLinearDim;
254
255
256 typedef struct EmbOrdinateDim_ {
257 EmbVector position;
258 } EmbOrdinateDim;
259

```

```
290
295 typedef struct EmbRadiusDim_ {
296 EmbVector position;
297 } EmbRadiusDim;
298
303 typedef struct EmbInfiniteLine_ {
304 EmbVector position;
305 } EmbInfiniteLine;
306
311 typedef struct EmbRay_ {
312 EmbVector position;
313 } EmbRay;
314
319 typedef struct EmbTextMulti_ {
320 EmbVector position;
321 char text[200];
322 } EmbTextMulti;
323
328 typedef struct EmbTextSingle_ {
329 EmbVector position;
330 char text[200];
331 } EmbTextSingle;
332
337 typedef struct EmbTime_
338 {
339 unsigned int year;
340 unsigned int month;
341 unsigned int day;
342 unsigned int hour;
343 unsigned int minute;
344 unsigned int second;
345 } EmbTime;
346
351 typedef struct EmbPoint_
352 {
353 EmbVector position;
354 int lineType;
355 EmbColor color;
356 } EmbPoint;
357
362 typedef struct EmbLine_
363 {
364 EmbVector start;
365 EmbVector end;
366 int lineType;
367 EmbColor color;
368 } EmbLine;
369
374 typedef struct EmbPath_
375 {
376 EmbArray* pointList;
377 EmbArray* flagList;
378 int lineType;
379 EmbColor color;
380 } EmbPath;
381
386 typedef struct EmbStitch_
387 {
388 int flags;
389 EmbReal x;
390 EmbReal y;
391 int color;
392 } EmbStitch;
393
399 typedef struct EmbThread_
400 {
401 EmbColor color;
402 char description[50];
403 char catalogNumber[30];
404 } EmbThread;
405
410 typedef struct thread_color_ {
411 char name[22];
412 unsigned int hex_code;
413 int manufacturer_code;
414 } thread_color;
415
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
441 typedef struct EmbCircle_
442 {
443 EmbVector center;
444 EmbReal radius;
445 } EmbCircle;
446
447 typedef EmbPath EmbPolygon;
448
449 typedef EmbPath EmbPolyline;
450
451 typedef int EmbFlag;
452
453 typedef struct EmbSatinOutline_
454 {
455 int length;
456 EmbArray* side1;
457 EmbArray* side2;
458 } EmbSatinOutline;
459
460 typedef struct EmbEllipse_
461 {
462 EmbVector center;
463 EmbVector radius;
464 EmbReal rotation;
465 } EmbEllipse;
466
467 typedef struct EmbBezier_ {
468 EmbVector start;
469 EmbVector controll1;
470 EmbVector control2;
471 EmbVector end;
472 } EmbBezier;
473
474 typedef struct EmbSpline_ {
475 EmbArray *beziers;
476 } EmbSpline;
477
478 typedef struct LSYSTEM {
479 char axiom;
480 char *alphabet;
481 char *constants;
482 char **rules;
483 } L_system;
484
485 typedef struct EmbGeometry_ {
486 union {
487 EmbArc arc;
488 EmbCircle circle;
489 EmbColor color;
490 EmbEllipse ellipse;
491 EmbLine line;
492 EmbPath path;
493 EmbPoint point;
494 EmbPolygon polygon;
495 EmbPolyline polyline;
496 EmbRect rect;
497 EmbSpline spline;
498 EmbVector vector;
499 } object;
500 EmbStitch stitch;
501 EmbThread thread;
502 int flag;
503 int type;
504 int lineType;
505 EmbColor color;
506 } EmbGeometry;
507
508 struct EmbArray_ {
509 EmbGeometry *geometry;
510 EmbStitch *stitch;
511 EmbThread *thread;
512 int count;
513 int length;
514 int type;
515 };
516
517 typedef struct EmbLayer_
518 {
519 char name[100];
520 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);
720 EMB_PUBLIC void embPattern_addStitchRel(EmbPattern* p, EmbReal dx, EmbReal dy, int flags, int
721 isAutoColorIndex);
722 EMB_PUBLIC void embPattern_changeColor(EmbPattern* p, int index);
723 EMB_PUBLIC void embPattern_free(EmbPattern* p);
724 EMB_PUBLIC void embPattern_scale(EmbPattern* p, EmbReal scale);
725 EMB_PUBLIC EmbReal embPattern_totalStitchLength(EmbPattern *pattern);
726 EMB_PUBLIC EmbReal embPattern_minimumStitchLength(EmbPattern *pattern);
727 EMB_PUBLIC EmbReal embPattern_maximumStitchLength(EmbPattern *pattern);
728 EMB_PUBLIC void embPattern_lengthHistogram(EmbPattern *pattern, int *bin, int NUMBINS);
729 EMB_PUBLIC int embPattern_realStitches(EmbPattern *pattern);
730 EMB_PUBLIC int embPattern_jumpStitches(EmbPattern *pattern);
731 EMB_PUBLIC int embPattern_trimStitches(EmbPattern *pattern);
732 EMB_PUBLIC EmbRect embPattern_calcBoundingBox(EmbPattern* p);
733 EMB_PUBLIC void embPattern_flipHorizontal(EmbPattern* p);
734 EMB_PUBLIC void embPattern_flipVertical(EmbPattern* p);
735 EMB_PUBLIC void embPattern_flip(EmbPattern* p, int horz, int vert);
736 EMB_PUBLIC void embPattern_combineJumpStitches(EmbPattern* p);
737 EMB_PUBLIC void embPattern_correctForMaxStitchLength(EmbPattern* p, EmbReal maxStitchLength, EmbReal
738 maxJumpLength);
739 EMB_PUBLIC void embPattern_center(EmbPattern* p);
740 EMB_PUBLIC void embPattern_loadExternalColorFile(EmbPattern* p, const char* fileName);
741 EMB_PUBLIC void embPattern_convertGeometry(EmbPattern* p);
742 EMB_PUBLIC void embPattern_designDetails(EmbPattern *p);
743 EMB_PUBLIC EmbPattern *embPattern_combine(EmbPattern *p1, EmbPattern *p2);
744 EMB_PUBLIC int embPattern_color_count(EmbPattern *pattern, EmbColor startColor);
745 EMB_PUBLIC void embPattern_end(EmbPattern* p);
746 EMB_PUBLIC void embPattern_crossstitch(EmbPattern *pattern, EmbImage *, int threshold);
747 EMB_PUBLIC void embPattern_horizontal_fill(EmbPattern *pattern, EmbImage *, int threshold);
748 EMB_PUBLIC int embPattern_render(EmbPattern *pattern, char *fname);
749 EMB_PUBLIC int embPattern_simulate(EmbPattern *pattern, char *fname);
750

```

```

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
798 #ifdef __cplusplus
799 }
800 #endif /* __cplusplus */
801
802 #endif /* LIBEMBROIDERY_HEADER__ */
803

```

## 18.6 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 [emblInt\\_read](#) (FILE \*f, char \*label, void \*b, int mode)
- void [emblInt\\_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]

## 18.6.1 Macro Definition Documentation

**18.6.1.1 BULGETOCONTROL** `#define BULGETOCONTROL 2`

**18.6.1.2 BULGETOEND** `#define BULGETOEND 4`

**18.6.1.3 CompoundFileSector\_DIFAT\_Sector** `#define CompoundFileSector_DIFAT_Sector 0xFFFFFFFFC`

**18.6.1.4 CompoundFileSector\_EndOfChain** `#define CompoundFileSector_EndOfChain 0xFFFFFFFFE`

**18.6.1.5 CompoundFileSector\_FAT\_Sector** `#define CompoundFileSector_FAT_Sector 0xFFFFFFFFD`

**18.6.1.6 CompoundFileSector\_FreeSector** `#define CompoundFileSector_FreeSector 0xFFFFFFFFF`

**18.6.1.7 CompoundFileSector\_MaxRegSector** `#define CompoundFileSector_MaxRegSector 0xFFFFFFFFFA`  
Type of sector

**18.6.1.8 CompoundFileStreamId\_MaxRegularStreamId** `#define CompoundFileStreamId_MaxRegular↵  
StreamId 0xFFFFFFFFFA`  
Special values for Stream Identifiers All real stream Ids are less than this

**18.6.1.9 CompoundFileStreamId\_NoStream** `#define CompoundFileStreamId_NoStream 0xFFFFFFFFF`  
There is no valid stream Id

**18.6.1.10 CUBICTOCONTROL1** `#define CUBICTOCONTROL1 32`

**18.6.1.11 CUBICTOCONTROL2** `#define CUBICTOCONTROL2 64`

**18.6.1.12 CUBICTOEND** `#define CUBICTOEND 128`

**18.6.1.13 DXF\_VERSION\_2000** `#define DXF_VERSION_2000 "AC1015"`

**18.6.1.14 DXF\_VERSION\_2002** `#define DXF_VERSION_2002 "AC1015"`

**18.6.1.15 DXF\_VERSION\_2004** `#define DXF_VERSION_2004 "AC1018"`

**18.6.1.16 DXF\_VERSION\_2006** `#define DXF_VERSION_2006 "AC1018"`

**18.6.1.17 DXF\_VERSION\_2007** `#define DXF_VERSION_2007 "AC1021"`

**18.6.1.18 DXF\_VERSION\_2009** `#define DXF_VERSION_2009 "AC1021"`

**18.6.1.19 DXF\_VERSION\_2010** `#define DXF_VERSION_2010 "AC1024"`

**18.6.1.20 DXF\_VERSION\_2013** `#define DXF_VERSION_2013 "AC1027"`

**18.6.1.21 DXF\_VERSION\_R10** `#define DXF_VERSION_R10 "AC1006"`

**18.6.1.22 DXF\_VERSION\_R11** `#define DXF_VERSION_R11 "AC1009"`

**18.6.1.23 DXF\_VERSION\_R12** `#define DXF_VERSION_R12 "AC1009"`

**18.6.1.24 DXF\_VERSION\_R13** `#define DXF_VERSION_R13 "AC1012"`

**18.6.1.25 DXF\_VERSION\_R14** `#define DXF_VERSION_R14 "AC1014"`

**18.6.1.26 DXF\_VERSION\_R15** `#define DXF_VERSION_R15 "AC1015"`

**18.6.1.27 DXF\_VERSION\_R18** `#define DXF_VERSION_R18 "AC1018"`

**18.6.1.28 DXF\_VERSION\_R21** `#define DXF_VERSION_R21 "AC1021"`

**18.6.1.29 DXF\_VERSION\_R24** `#define DXF_VERSION_R24 "AC1024"`

**18.6.1.30 DXF\_VERSION\_R27** `#define DXF_VERSION_R27 "AC1027"`

**18.6.1.31 ELEMENT\_A** `#define ELEMENT_A 1`

**18.6.1.32 ELEMENT\_ANIMATE** `#define ELEMENT_ANIMATE 2`

**18.6.1.33 ELEMENT\_ANIMATECOLOR** `#define ELEMENT_ANIMATECOLOR 3`

**18.6.1.34 ELEMENT\_ANIMATEMOTION** `#define ELEMENT_ANIMATEMOTION 4`

**18.6.1.35 ELEMENT\_ANIMATETRANSFORM** `#define ELEMENT_ANIMATETRANSFORM 5`

**18.6.1.36 ELEMENT\_ANIMATION** `#define ELEMENT_ANIMATION 6`

**18.6.1.37 ELEMENT\_AUDIO** `#define ELEMENT_AUDIO 7`

**18.6.1.38 ELEMENT\_CIRCLE** `#define ELEMENT_CIRCLE 8`

**18.6.1.39 ELEMENT\_DEFS** `#define ELEMENT_DEFS 9`

**18.6.1.40 ELEMENT\_DESC** `#define ELEMENT_DESC 10`

**18.6.1.41 ELEMENT\_DISCARD** `#define ELEMENT_DISCARD 11`

**18.6.1.42 ELEMENT\_ELLIPSE** `#define ELEMENT_ELLIPSE 12`

**18.6.1.43 ELEMENT\_FONT** `#define ELEMENT_FONT 13`

**18.6.1.44 ELEMENT\_FONT\_FACE** `#define ELEMENT_FONT_FACE 14`

**18.6.1.45 ELEMENT\_FONT\_FACE\_SRC** `#define ELEMENT_FONT_FACE_SRC 15`

**18.6.1.46 ELEMENT\_FONT\_FACE\_URI** `#define ELEMENT_FONT_FACE_URI 16`

**18.6.1.47 ELEMENT\_FOREIGN\_OBJECT** `#define ELEMENT_FOREIGN_OBJECT 17`

**18.6.1.48 ELEMENT\_G** `#define ELEMENT_G 18`

**18.6.1.49 ELEMENT\_GLYPH** `#define ELEMENT_GLYPH 19`

**18.6.1.50 ELEMENT\_HANDLER** `#define ELEMENT_HANDLER 20`

**18.6.1.51 ELEMENT\_HKERN** `#define ELEMENT_HKERN 21`

**18.6.1.52 ELEMENT\_IMAGE** `#define ELEMENT_IMAGE 22`

**18.6.1.53 ELEMENT\_LINE** `#define ELEMENT_LINE 23`

**18.6.1.54 ELEMENT\_LINEAR\_GRADIENT** `#define ELEMENT_LINEAR_GRADIENT 24`

**18.6.1.55 ELEMENT\_LISTENER** `#define ELEMENT_LISTENER 25`

**18.6.1.56 ELEMENT\_METADATA** `#define ELEMENT_METADATA 26`

**18.6.1.57 ELEMENT\_MISSING\_GLYPH** `#define ELEMENT_MISSING_GLYPH 27`

**18.6.1.58 ELEMENT\_MPATH** `#define ELEMENT_MPATH 28`

**18.6.1.59 ELEMENT\_PATH** `#define ELEMENT_PATH 29`

**18.6.1.60 ELEMENT\_POLYGON** `#define ELEMENT_POLYGON 30`

**18.6.1.61 ELEMENT\_POLYLINE** `#define ELEMENT_POLYLINE 31`

**18.6.1.62 ELEMENT\_PREFETCH** `#define ELEMENT_PREFETCH 32`

**18.6.1.63 ELEMENT\_RADIAL\_GRADIENT** `#define ELEMENT_RADIAL_GRADIENT 33`

**18.6.1.64 ELEMENT\_RECT** `#define ELEMENT_RECT 34`

**18.6.1.65 ELEMENT\_SCRIPT** `#define ELEMENT_SCRIPT 35`

**18.6.1.66 ELEMENT\_SET** `#define ELEMENT_SET 36`

**18.6.1.67 ELEMENT\_SOLID\_COLOR** `#define ELEMENT_SOLID_COLOR 37`

**18.6.1.68 ELEMENT\_STOP** `#define ELEMENT_STOP 38`

**18.6.1.69 ELEMENT\_SVG** `#define ELEMENT_SVG 39`

**18.6.1.70 ELEMENT\_SWITCH** `#define ELEMENT_SWITCH 40`

**18.6.1.71 ELEMENT\_TBREAK** `#define ELEMENT_TBREAK 41`

**18.6.1.72 ELEMENT\_TEXT** `#define ELEMENT_TEXT 42`

**18.6.1.73 ELEMENT\_TEXT\_AREA** `#define ELEMENT_TEXT_AREA 43`

**18.6.1.74 ELEMENT\_TITLE** `#define ELEMENT_TITLE 44`

**18.6.1.75 ELEMENT\_TSPAN** `#define ELEMENT_TSPAN 45`

**18.6.1.76 ELEMENT\_USE** `#define ELEMENT_USE 46`

**18.6.1.77 ELEMENT\_VIDEO** `#define ELEMENT_VIDEO 47`

**18.6.1.78 ELEMENT\_XML** `#define ELEMENT_XML 0`

**18.6.1.79 ELLIPSETOEND** `#define ELLIPSETOEND 16`

**18.6.1.80 ELLIPSETORAD** `#define ELLIPSETORAD 8`

**18.6.1.81 EMB\_BIG\_ENDIAN** `#define EMB_BIG_ENDIAN 0`

**18.6.1.82 EMB\_INT16\_BIG** `#define EMB_INT16_BIG 2`

**18.6.1.83 EMB\_INT16\_LITTLE** `#define EMB_INT16_LITTLE 3`

**18.6.1.84 EMB\_INT32\_BIG** `#define EMB_INT32_BIG 4`

**18.6.1.85 EMB\_INT32\_LITTLE** `#define EMB_INT32_LITTLE 5`

**18.6.1.86 EMB\_LITTLE\_ENDIAN** `#define EMB_LITTLE_ENDIAN 1`

**18.6.1.87 EMB\_MAX** `#define EMB_MAX(  
A,  
B) ((A) > (B)) ? (A) : (B)`

**18.6.1.88 EMB\_MIN** `#define EMB_MIN(  
A,  
B) ((A) < (B)) ? (A) : (B)`

**18.6.1.89 ENDIAN\_HOST** `#define ENDIAN_HOST EMB\_LITTLE\_ENDIAN`

**18.6.1.90 GREEN\_TERM\_COLOR** `#define GREEN_TERM_COLOR "\x1B[0;32m"`

**18.6.1.91 HOOP\_110X110** `#define HOOP_110X110 1`

**18.6.1.92 HOOP\_126X110** #define HOOP\_126X110 0

**18.6.1.93 HOOP\_140X200** #define HOOP\_140X200 3

**18.6.1.94 HOOP\_230X200** #define HOOP\_230X200 4

**18.6.1.95 HOOP\_50X50** #define HOOP\_50X50 2

**18.6.1.96 LINETO** #define LINETO 0

**18.6.1.97 MOVETO** #define MOVETO 1

**18.6.1.98 N\_PES\_VERSIONS** #define N\_PES\_VERSIONS 13

**18.6.1.99 ObjectTypeRootEntry** #define ObjectTypeRootEntry 0x05  
the root entry

**18.6.1.100 ObjectTypeStorage** #define ObjectTypeStorage 0x01  
a directory type object

**18.6.1.101 ObjectTypeStream** #define ObjectTypeStream 0x02  
a file type object

**18.6.1.102 ObjectTypeUnknown** #define ObjectTypeUnknown 0x00  
Type of directory object Probably unallocated

**18.6.1.103 PES0001** #define PES0001 0

**18.6.1.104 PES0020** #define PES0020 1

**18.6.1.105 PES0022** #define PES0022 2

**18.6.1.106 PES0030** #define PES0030 3

**18.6.1.107 PES0040** #define PES0040 4

**18.6.1.108 PES0050** #define PES0050 5



**18.6.1.109 PES0055** `#define PES0055 6`

**18.6.1.110 PES0056** `#define PES0056 7`

**18.6.1.111 PES0060** `#define PES0060 8`

**18.6.1.112 PES0070** `#define PES0070 9`

**18.6.1.113 PES0080** `#define PES0080 10`

**18.6.1.114 PES0090** `#define PES0090 11`

**18.6.1.115 PES0100** `#define PES0100 12`

**18.6.1.116 QUADTOCONTROL** `#define QUADTOCONTROL 256`

**18.6.1.117 QUADTOEND** `#define QUADTOEND 512`

**18.6.1.118 RED\_TERM\_COLOR** `#define RED_TERM_COLOR "\x1B[0;31m"`

**18.6.1.119 RESET\_TERM\_COLOR** `#define RESET_TERM_COLOR "\033[0m"`

**18.6.1.120 SVG\_ATTRIBUTE** `#define SVG_ATTRIBUTE 4`

**18.6.1.121 SVG\_CATCH\_ALL** `#define SVG_CATCH_ALL 5`

**18.6.1.122 SVG\_CREATOR\_EMBROIDERMODDER** `#define SVG_CREATOR_EMBROIDERMODDER 1`

**18.6.1.123 SVG\_CREATOR\_ILLUSTRATOR** `#define SVG_CREATOR_ILLUSTRATOR 2`

**18.6.1.124 SVG\_CREATOR\_INKSCAPE** `#define SVG_CREATOR_INKSCAPE 3`

**18.6.1.125 SVG\_CREATOR\_NULL** `#define SVG_CREATOR_NULL 0`

**18.6.1.126 SVG\_ELEMENT** `#define SVG_ELEMENT 1`

**18.6.1.127 SVG\_EXPECT\_ATTRIBUTE** `#define SVG_EXPECT_ATTRIBUTE 2`

**18.6.1.128 SVG\_EXPECT\_ELEMENT** `#define SVG_EXPECT_ELEMENT 1`

**18.6.1.129 SVG\_EXPECT\_NULL** `#define SVG_EXPECT_NULL 0`

**18.6.1.130 SVG\_EXPECT\_VALUE** `#define SVG_EXPECT_VALUE 3`

**18.6.1.131 SVG\_MEDIA\_PROPERTY** `#define SVG_MEDIA_PROPERTY 3`

**18.6.1.132 SVG\_NULL** `#define SVG_NULL 0`

**18.6.1.133 SVG\_PROPERTY** `#define SVG_PROPERTY 2`

**18.6.1.134 YELLOW\_TERM\_COLOR** `#define YELLOW_TERM_COLOR "\x1B[1;33m"`

## 18.6.2 Typedef Documentation

**18.6.2.1 bcf\_directory** `typedef struct _bcf_directory bcf_directory`

**Todo** possibly add a directory tree in the future.

**18.6.2.2 bcf\_directory\_entry** `typedef struct _bcf_directory_entry bcf_directory_entry`

**18.6.2.3 bcf\_file** `typedef struct _bcf_file bcf_file`

**18.6.2.4 bcf\_file\_difat** `typedef struct _bcf_file_difat bcf_file_difat`

**18.6.2.5 bcf\_file\_fat** `typedef struct _bcf_file_fat bcf_file_fat`

**18.6.2.6 bcf\_file\_header** `typedef struct _bcf_file_header bcf_file_header`

**Todo** CLSID should be a separate type.

**18.6.2.7 compress** `typedef struct Compress compress`

**18.6.2.8 huffman** `typedef struct Huffman huffman`

**18.6.2.9 StxThread** typedef struct [StxThread\\_](#) StxThread

**18.6.2.10 SubDescriptor** typedef struct [SubDescriptor\\_](#) SubDescriptor

**18.6.2.11 SvgAttribute** typedef struct [SvgAttribute\\_](#) SvgAttribute

**18.6.2.12 ThredExtension** typedef struct [ThredExtension\\_](#) ThredExtension

**18.6.2.13 ThredHeader** typedef struct [ThredHeader\\_](#) ThredHeader

**18.6.2.14 VipHeader** typedef struct [VipHeader\\_](#) VipHeader

**18.6.2.15 vp3Hoop** typedef struct [\\_vp3Hoop](#) vp3Hoop

## 18.6.3 Enumeration Type Documentation

**18.6.3.1 CSV\_EXPECT** enum [CSV\\_EXPECT](#)

Enumerator

|                   |  |
|-------------------|--|
| CSV_EXPECT_NULL   |  |
| CSV_EXPECT_QUOTE1 |  |
| CSV_EXPECT_QUOTE2 |  |
| CSV_EXPECT_COMMA  |  |

**18.6.3.2 CSV\_MODE** enum [CSV\\_MODE](#)

Enumerator

|                   |  |
|-------------------|--|
| CSV_MODE_NULL     |  |
| CSV_MODE_COMMENT  |  |
| CSV_MODE_VARIABLE |  |
| CSV_MODE_THREAD   |  |
| CSV_MODE_STITCH   |  |

## 18.6.4 Function Documentation

**18.6.4.1 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\*

**18.6.4.2 bcf\_directory\_free()** void bcf\_directory\_free (   
 bcf\_directory \*\* dir )

*dir*

**18.6.4.3 bcf\_file\_difat\_free()** void bcf\_file\_difat\_free (   
 bcf\_file\_difat \* difat )

**18.6.4.4 bcf\_file\_fat\_free()** void bcf\_file\_fat\_free (   
 bcf\_file\_fat \*\* fat )

**18.6.4.5 bcf\_file\_free()** void bcf\_file\_free (   
 bcf\_file \* bcfFile )

*bcfFile*

**18.6.4.6 bcfFile\_read()** int bcfFile\_read (   
 FILE \* file,   
 bcf\_file \* bcfFile )

*file bcfFile*

## Returns

int

**18.6.4.7 bcfFileFat\_create()** bcf\_file\_fat \* bcfFileFat\_create (   
 const unsigned int sectorSize )

*sectorSize*

## Returns

bcf\_file\_fat\*

**18.6.4.8 bcfFileHeader\_isValid()** int bcfFileHeader\_isValid (   
 bcf\_file\_header header )

**18.6.4.9 bcfFileHeader\_read()** bcf\_file\_header bcfFileHeader\_read (   
 FILE \* file )

*file*

## Returns

bcf\_file\_header

**18.6.4.10 binaryReadString()** void binaryReadString (   
FILE \* *file*,  
char \* *buffer*,  
int *maxLength* )  
*file buffer maxLength*

**18.6.4.11 binaryReadUnicodeString()** void binaryReadUnicodeString (   
FILE \* *file*,  
char \* *buffer*,  
const int *stringLength* )  
*file buffer stringLength*

**18.6.4.12 binaryWriteInt()** void binaryWriteInt (   
FILE \* *f*,  
int *data* )  
*f data*

**Todo** replace with embInt\_read

**18.6.4.13 binaryWriteIntBE()** void binaryWriteIntBE (   
FILE \* *f*,  
int *data* )  
*f data*

**Todo** replace with embInt\_read

**18.6.4.14 binaryWriteShort()** void binaryWriteShort (   
FILE \* *f*,  
short *data* )  
*f data*

**Todo** replace with embInt\_read

**18.6.4.15 binaryWriteUInt()** void binaryWriteUInt (   
FILE \* *f*,  
unsigned int *data* )  
*f data*

**Todo** replace with embInt\_read

**18.6.4.16 binaryWriteUIntBE()** void binaryWriteUIntBE (   
FILE \* *f*,  
unsigned int *data* )  
*f data*

**Todo** replace with embInt\_read

**18.6.4.17 binaryWriteUShort()** void binaryWriteUShort (   
FILE \* *f*,  
unsigned short *data* )  
*f data*

**Todo** replace with embInt\_read

**18.6.4.18 binaryWriteUShortBE()** void binaryWriteUShortBE (   
FILE \* *f*,  
unsigned short *data* )  
*f data*

**Todo** replace with embInt\_read

**18.6.4.19 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.

**18.6.4.20 CompoundFileDirectory()** bcf\_directory \* CompoundFileDirectory (   
const unsigned int *maxNumberOfDirectoryEntries* )  
*maxNumberOfDirectoryEntries*

**Returns**

bcf\_directory\*

**18.6.4.21 CompoundFileDirectoryEntry()** bcf\_directory\_entry \* CompoundFileDirectoryEntry (   
FILE \* *file* )  
*file*

**Returns**

bcf\_directory\_entry\*

**18.6.4.22 compress\_get\_bits()** int compress\_get\_bits (   
compress \* *c*,  
int *length* )  
*c length* Returns .

**18.6.4.23 compress\_get\_position()** int compress\_get\_position (   
compress \* *c* )  
*c* . Returns the position as an int.

**18.6.4.24 compress\_get\_token()** int compress\_get\_token (  
     compress \* c )

c . Returns the token as an int.

**18.6.4.25 compress\_load\_block()** void compress\_load\_block (  
     compress \* c )

c . Returns nothing.

**18.6.4.26 compress\_load\_character\_huffman()** void compress\_load\_character\_huffman (  
     compress \* c )

Load character table to compress struct c. Returns nothing.

**18.6.4.27 compress\_load\_character\_length\_huffman()** void compress\_load\_character\_length\_huffman (  
     compress \* c )

c . Returns.

**18.6.4.28 compress\_load\_distance\_huffman()** void compress\_load\_distance\_huffman (  
     compress \* c )

c . Returns nothing.

**18.6.4.29 compress\_pop()** int compress\_pop (  
     compress \* c,  
     int bit\_count )

c bit\_count . Returns.

**18.6.4.30 compress\_read\_variable\_length()** int compress\_read\_variable\_length (  
     compress \* c )

c . Returns.

**18.6.4.31 copy\_trim()** char \* copy\_trim (  
     char const \* s )

s

Returns

char\*

**Todo** decription

**18.6.4.32 create\_test\_file\_1()** int create\_test\_file\_1 (  
     const char \* outf )

**18.6.4.33 create\_test\_file\_2()** int create\_test\_file\_2 (  
     const char \* outf )

**18.6.4.34 create\_test\_file\_3()** int create\_test\_file\_3 (  
     const char \* outf )

**18.6.4.35 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.

**18.6.4.36 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.

**18.6.4.37 decodeNewStitch()** `int decodeNewStitch (`  
    `unsigned char value )`

*value*

**Returns**

int

**18.6.4.38 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\*

**18.6.4.39 emb\_readline()** `int emb_readline (`  
    `FILE * file,`  
    `char * line,`  
    `int maxLength )`

*file line maxLength*

**Returns**

int

**18.6.4.40 embColor\_read()** `void embColor_read (`  
    `FILE * f,`  
    `EmbColor * c,`  
    `int toRead )`

*f c toRead*



**18.6.4.41 embColor\_write()** void embColor\_write (

```
FILE * f,
EmbColor c,
int toWrite)
```

*f c toWrite*

**18.6.4.42 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.

**18.6.4.43 embInt\_write()** void embInt\_write (

```
FILE * f,
char * label,
void * b,
int mode)
```

*f label b mode*

**18.6.4.44 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*.

**18.6.4.45 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.

**18.6.4.46 entriesInDifatSector()** unsigned int entriesInDifatSector (

```
bcf_file_difat * fat)
```

*fat*

Returns

unsigned int

**18.6.4.47 fpad()** void fpad (

```
FILE * file,
char c,
int n)
```

*f*

Returns

int

**18.6.4.48 fread\_int16()** short fread\_int16 (  
FILE \* *f* )

*f*

Returns

short

**18.6.4.49 fread\_int32\_be()** int fread\_int32\_be (  
FILE \* *f* )

*f*

Returns

int

**Todo** replace with embInt\_read

**18.6.4.50 fread\_uint16()** unsigned short fread\_uint16 (  
FILE \* *f* )

*f*

Returns

unsigned short

**Todo** replace with embInt\_read

**18.6.4.51 GetFile()** FILE \* GetFile (  
bcf\_file \* *bcfFile*,  
FILE \* *file*,  
char \* *fileToFind* )

Get the File object.

*bcfFile file fileToFind*

Returns

FILE\*

**18.6.4.52 huffman\_build\_table()** void huffman\_build\_table (  
huffman \* *h* )

These next 2 functions represent the **Huffman** class in tartarize's code. *h*

**18.6.4.53 huffman\_table\_lookup()** int \* huffman\_table\_lookup (  
huffman \* *h*,  
int *byte\_lookup*,  
int \* *lengths* )

**18.6.4.54 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).

**18.6.4.55 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.

**18.6.4.56 loadFatFromSector()** void loadFatFromSector (

```

 bcf_file_fat * fat,
 FILE * file)

```

*fat* *file*

**18.6.4.57 mitDecodeStitch()** int mitDecodeStitch (

```

 unsigned char value)

```

*value*

Returns

int

**18.6.4.58 mitEncodeStitch()** unsigned char mitEncodeStitch (

```

 EmbReal value)

```

*value*

Returns

unsigned char

**18.6.4.59 numberOfEntriesInDifatSector()** unsigned int numberOfEntriesInDifatSector (

```

 bcf_file_difat * fat)

```

**18.6.4.60 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.

**18.6.4.61 pfaffEncode()** void pfaffEncode (

```

 FILE * file,
 int dx,
 int dy,
 int flags)

```

*file* *dx* *dy* *flags*

**18.6.4.62 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)
```

**18.6.4.63 read100()** `char read100 (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.64 read10o()** `char read10o (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.65 readArt()** `char readArt (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.66 readBmc()** `char readBmc (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.67 readBro()** `char readBro (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.68 readCnd()** `char readCnd (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.69 readCol()** `char readCol (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.70 readCsd()** `char readCsd (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.71 readCsv()** `char readCsv (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.72 readDat()** `char readDat (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.73 readDem()** char readDem (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.74 readDescriptions()** void readDescriptions (  
    FILE \* file,  
    EmbPattern \* pattern )

**18.6.4.75 readDsb()** char readDsb (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.76 readDst()** char readDst (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.77 readDsz()** char readDsz (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.78 ZSK USA Embroidery Format (.dsz)** The ZSK USA dsz format is stitch-only.

**18.6.4.79 readDxf()** char readDxf (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.80 readEdr()** char readEdr (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.81 Embird Embroidery Format (.edr)** Stitch Only Format

**18.6.4.82 readEmd()** char readEmd (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.83 readExp()** char readExp (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.84 readExy()** char readExy (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.85 readEys()** char readEys (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.86 Sierra Expanded Embroidery Format (.eys)** Stitch Only Format.  
Smoothie G-Code Embroidery Format (.fxy)?

**18.6.4.87 readFeatherPatterns()** void readFeatherPatterns (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.6.4.88 readFullSector()** unsigned int readFullSector (  
FILE \* *file*,  
bcf\_file\_difat \* *bcfFile*,  
unsigned int \* *difatEntriesToRead* )  
*file bcfFile difatEntriesToRead*

Returns

unsigned int

**18.6.4.89 readFxy()** char readFxy (  
EmbPattern \* *pattern*,  
FILE \* *file* )

**18.6.4.90 Embroidery Format (.fxy)** Stitch Only Format.

**18.6.4.91 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

**18.6.4.92 readGnc()** char readGnc (  
EmbPattern \* *pattern*,  
FILE \* *file* )

**18.6.4.93 Great Notions Embroidery Format (.gnc)** Stitch Only Format.

**18.6.4.94 readGt()** char readGt (  
EmbPattern \* *pattern*,  
FILE \* *file* )

**18.6.4.95 Gold Thread Embroidery Format (.gt)** Stitch Only Format.

**18.6.4.96 readHoopName()** void readHoopName (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.6.4.97 readHus()** char readHus (  
EmbPattern \* *pattern*,  
FILE \* *file* )

**18.6.4.98 readImageString()** void readImageString (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.6.4.99 readInb()** char readInb (  
EmbPattern \* *pattern*,  
FILE \* *file* )

**18.6.4.100 Inbro Embroidery Format (.inb)** Stitch Only Format.

**18.6.4.101 readInf()** char readInf (  
EmbPattern \* *pattern*,  
FILE \* *file* )

**18.6.4.102 Embroidery Color Format (.inf)** Stitch Only Format.

**18.6.4.103 readJef()** char readJef (  
EmbPattern \* *pattern*,  
FILE \* *file* )

**18.6.4.104 readKsm()** char readKsm (  
EmbPattern \* *pattern*,  
FILE \* *file* )

**18.6.4.105 readMax()** char readMax (  
EmbPattern \* *pattern*,  
FILE \* *file* )

**18.6.4.106 readMit()** char readMit (  
EmbPattern \* *pattern*,  
FILE \* *file* )

**18.6.4.107 Mitsubishi Embroidery Format (.mit)** Stitch Only Format.

**18.6.4.108 readMotifPatterns()** void readMotifPatterns (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.6.4.109 readNew()** char readNew (  
EmbPattern \* *pattern*,  
FILE \* *file* )

**18.6.4.110 Ameco Embroidery Format (.new)** Stitch Only Format.

**18.6.4.111 readNextSector()** void readNextSector (  
FILE \* *file*,  
bcf\_directory \* *dir* )

*file dir*

**18.6.4.112 readOfm()** `char readOfm (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.113 readPcd()** `char readPcd (`  
    `EmbPattern * pattern,`  
    `const char * fileName,`  
    `FILE * file )`

**18.6.4.114 Pfaff PCD File Format (.pcd)** Stitch Only Format.

The format uses a signed 3 byte-length number type.

See the description here ([5](5)) for the overview of the format.

For an example of the format see ([11](11)).

**18.6.4.115 readPcm()** `char readPcm (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.116 Pfaff Embroidery Format (.pcm)** The Pfaff pcm format is stitch-only.

**18.6.4.117 readPcq()** `char readPcq (`  
    `EmbPattern * pattern,`  
    `const char * fileName,`  
    `FILE * file )`

**18.6.4.118 Embroidery Format (.pcq)** The Pfaff pcq format is stitch-only.

**18.6.4.119 readPcs()** `char readPcs (`  
    `EmbPattern * pattern,`  
    `const char * fileName,`  
    `FILE * file )`

**18.6.4.120 Embroidery Format (.pcq)** The Pfaff pcs format is stitch-only.

**18.6.4.121 readPec()** `char readPec (`  
    `EmbPattern * pattern,`  
    `const char * fileName,`  
    `FILE * file )`

**18.6.4.122 readPecStitches()** `void readPecStitches (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.123 Embroidery Format (.pec)** The Brother pec format is stitch-only.

**18.6.4.124 readPel()** `char readPel (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.125 Embroidery Format (.pec)** The Brother pel format is stitch-only.



**18.6.4.126 readPem()** `char readPem (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.127 Embroidery Format (.pec)** The Brother pem format is stitch-only.

**18.6.4.128 readPes()** `char readPes (`  
    `EmbPattern * pattern,`  
    `const char * fileName,`  
    `FILE * file )`

**18.6.4.129 readPESHeaderV10()** `void readPESHeaderV10 (`  
    `FILE * file,`  
    `EmbPattern * pattern )`

**18.6.4.130 readPESHeaderV5()** `void readPESHeaderV5 (`  
    `FILE * file,`  
    `EmbPattern * pattern )`

**18.6.4.131 readPESHeaderV6()** `void readPESHeaderV6 (`  
    `FILE * file,`  
    `EmbPattern * pattern )`

**18.6.4.132 readPESHeaderV7()** `void readPESHeaderV7 (`  
    `FILE * file,`  
    `EmbPattern * pattern )`

**18.6.4.133 readPESHeaderV8()** `void readPESHeaderV8 (`  
    `FILE * file,`  
    `EmbPattern * pattern )`

**18.6.4.134 readPESHeaderV9()** `void readPESHeaderV9 (`  
    `FILE * file,`  
    `EmbPattern * pattern )`

**18.6.4.135 readPhb()** `char readPhb (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.136 Embroidery Format (.pec)** The Brother phb format is stitch-only.

**18.6.4.137 readPhc()** `char readPhc (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.138 Embroidery Format (.pec)** The Brother phc format is stitch-only.

**18.6.4.139 readPlt()** `char readPlt (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.140 Embroidery Format (.plt)** The AutoCAD plt format is stitch-only.

**18.6.4.141 readProgrammableFills()** `void readProgrammableFills (`  
    `FILE * file,`  
    `EmbPattern * pattern )`

**18.6.4.142 readRgb()** `char readRgb (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.143 Color File (.rgb)** The RGB format is a color-only format to act as an external color file for other formats.

**18.6.4.144 readSew()** `char readSew (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.145 readShv()** `char readShv (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.146 readSst()** `char readSst (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.147 Embroidery Format (.sst)** The Sunstar sst format is stitch-only.

**18.6.4.148 readStx()** `char readStx (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.149 readSvg()** `char readSvg (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.150 readT01()** `char readT01 (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.151 Embroidery Format (.pcq)** The Pfaff t01 format is stitch-only.

**18.6.4.152 readT09()** `char readT09 (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.152.1 Embroidery Format (.pcq)** The Pfaff t09 format is stitch-only.

**18.6.4.153 readTap()** `char readTap (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.154 readThr()** `char readThr (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.155 Embroidery Format (.thr)** The ThreadWorks thr format is stitch-only.

**18.6.4.156 readThreads()** `void readThreads (`  
    `FILE * file,`  
    `EmbPattern * pattern )`

**18.6.4.157 readTxt()** `char readTxt (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.158 File (.txt)** The txt format is stitch-only and isn't associated with a specific company.

**18.6.4.159 readU00()** `char readU00 (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.160 Embroidery Format (.u00)** The Barudan u00 format is stitch-only.

**18.6.4.161 readU01()** `char readU01 (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.162 Embroidery Format (.u00)** The Barudan u01 format is stitch-only.

**18.6.4.163 readVip()** `char readVip (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.164 readVp3()** `char readVp3 (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.165 readXxx()** `char readXxx (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.166 readZsk()** `char readZsk (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.6.4.167 safe\_free()** void safe\_free (  
void \* data )  
*data*

**18.6.4.168 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.

**18.6.4.169 testEmbCircle()** int testEmbCircle (  
void )

**18.6.4.170 testEmbCircle\_2()** int testEmbCircle\_2 (  
void )

**18.6.4.171 testEmbFormat()** int testEmbFormat (  
void )

**18.6.4.172 testGeomArc()** int testGeomArc (  
void )

**18.6.4.173 testTangentPoints()** void testTangentPoints (  
EmbCircle c,  
EmbVector p,  
EmbVector \* t0,  
EmbVector \* t1 )

**18.6.4.174 testThreadColor()** int testThreadColor (  
void )

**18.6.4.175 write100()** char write100 (  
EmbPattern \* pattern,  
FILE \* file )

**18.6.4.176 write10o()** char write10o (  
EmbPattern \* pattern,  
FILE \* file )

**18.6.4.177 write\_24bit()** void write\_24bit (   
 FILE \* *file*,   
 int *x* )   
 *file x*

**18.6.4.178 writeArt()** char writeArt (   
 EmbPattern \* *pattern*,   
 FILE \* *file* )

**18.6.4.179 writeBmc()** char writeBmc (   
 EmbPattern \* *pattern*,   
 FILE \* *file* )

**18.6.4.180 writeBro()** char writeBro (   
 EmbPattern \* *pattern*,   
 FILE \* *file* )

**18.6.4.181 writeCnd()** char writeCnd (   
 EmbPattern \* *pattern*,   
 FILE \* *file* )

**18.6.4.182 writeCol()** char writeCol (   
 EmbPattern \* *pattern*,   
 FILE \* *file* )

**18.6.4.183 writeCsd()** char writeCsd (   
 EmbPattern \* *pattern*,   
 FILE \* *file* )

**18.6.4.184 writeCsv()** char writeCsv (   
 EmbPattern \* *pattern*,   
 FILE \* *file* )

**18.6.4.185 writeDat()** char writeDat (   
 EmbPattern \* *pattern*,   
 FILE \* *file* )

**18.6.4.186 writeDem()** char writeDem (   
 EmbPattern \* *pattern*,   
 FILE \* *file* )

**18.6.4.187 writeDsb()** char writeDsb (   
 EmbPattern \* *pattern*,   
 FILE \* *file* )

**18.6.4.188 writeDst()** char writeDst (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.189 writeDsz()** char writeDsz (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.190 writeDxf()** char writeDxf (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.191 writeEdr()** char writeEdr (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.192 writeEmd()** char writeEmd (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.193 writeExp()** char writeExp (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.194 writeExy()** char writeExy (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.195 writeEys()** char writeEys (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.196 writeFxy()** char writeFxy (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.197 writeGc()** char writeGc (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.198 writeGnc()** char writeGnc (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.199 writeGt()** char writeGt (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.200 writeHus()** char writeHus (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.201 writeInb()** char writeInb (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.202 writeInf()** char writeInf (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.203 writeJef()** char writeJef (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.204 writeKsm()** char writeKsm (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.205 writeMax()** char writeMax (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.206 writeMit()** char writeMit (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.207 writeNew()** char writeNew (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.208 writeOfm()** char writeOfm (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.209 writePcd()** char writePcd (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.210 writePcm()** char writePcm (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.211 writePcq()** char writePcq (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.212 writePcs()** char writePcs (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.213 writePec()** char writePec (  
    EmbPattern \* pattern,  
    const char \* fileName,  
    FILE \* file )

**18.6.4.214 writePecStitches()** void writePecStitches (  
    EmbPattern \* pattern,  
    FILE \* file,  
    const char \* filename )

**18.6.4.215 writePel()** char writePel (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.216 writePem()** char writePem (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.217 writePes()** char writePes (  
    EmbPattern \* pattern,  
    const char \* fileName,  
    FILE \* file )

**18.6.4.218 writePhb()** char writePhb (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.219 writePhc()** char writePhc (  
    EmbPattern \* pattern,  
    FILE \* file )



**18.6.4.220 writePlt()** char writePlt (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.221 writeRgb()** char writeRgb (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.222 writeSew()** char writeSew (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.223 writeShv()** char writeShv (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.224 writeSst()** char writeSst (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.225 writeStx()** char writeStx (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.226 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`.

**18.6.4.227 writeT01()** char writeT01 (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.228 writeT09()** char writeT09 (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.229 writeTap()** char writeTap (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.230 writeThr()** char writeThr (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.6.4.231 writeTxt()** char writeTxt (   
     EmbPattern \* pattern,   
     FILE \* file )

**18.6.4.232 writeU00()** char writeU00 (   
     EmbPattern \* pattern,   
     FILE \* file )

**18.6.4.233 writeU01()** char writeU01 (   
     EmbPattern \* pattern,   
     FILE \* file )

**18.6.4.234 writeVip()** char writeVip (   
     EmbPattern \* pattern,   
     FILE \* file )

**18.6.4.235 writeVp3()** char writeVp3 (   
     EmbPattern \* pattern,   
     FILE \* file )

**18.6.4.236 writeXxx()** char writeXxx (   
     EmbPattern \* pattern,   
     FILE \* file )

**18.6.4.237 writeZsk()** char writeZsk (   
     EmbPattern \* pattern,   
     FILE \* file )

## 18.6.5 Variable Documentation

**18.6.5.1 imageWithFrame** const char imageWithFrame[38][48] [extern]

## 18.7 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

```

## 18.8 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)

### 18.8.1 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.

### 18.8.2 Function Documentation

**18.8.2.1 `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.

**18.8.2.2 `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.

**18.8.2.3 `decodeNewStitch()`** int `decodeNewStitch` (  
     unsigned char *value* )

*value*

Returns

int

**18.8.2.4 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.

**18.8.2.5 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.

**18.8.2.6 embInt\_write()** `void embInt_write (`  
`FILE * f,`  
`char * label,`  
`void * b,`  
`int mode )`

*f label b mode*

**18.8.2.7 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*.

**18.8.2.8 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.

**18.8.2.9 mitDecodeStitch()** `int mitDecodeStitch (`  
`unsigned char value )`

*value*

Returns

int

**18.8.2.10 mitEncodeStitch()** `unsigned char mitEncodeStitch (`  
`EmbReal value )`

*value*

Returns

unsigned char

**18.8.2.11 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.

**18.8.2.12 pfaffEncode()** `void pfaffEncode (`  
     FILE \* *file*,  
     int *dx*,  
     int *dy*,  
     int *flags* )

*file dx dy flags*

**18.8.2.13 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.

**18.8.2.14 write\_24bit()** `void write_24bit (`  
     FILE \* *file*,  
     int *x* )

*file x*

## 18.9 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`

### 18.9.1 Function Documentation

**18.9.1.1 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.

**18.9.1.2 embPattern\_combine()** `EmbPattern * embPattern_combine (`  
     `EmbPattern * p1,`  
     `EmbPattern * p2 )`

*p1 p2*

Returns

`EmbPattern*`

**18.9.1.3 embPattern\_convertGeometry()** `void embPattern_convertGeometry (`  
     `EmbPattern * p )`

*p*

**18.9.1.4 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.

**18.9.1.5 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.

**18.9.1.6 embPattern\_stitchArc()** `void embPattern_stitchArc (`  
     `EmbPattern * p,`  
     `EmbArc arc,`  
     `int thread_index,`  
     `int style )`

*p arc thread\_index style*

**18.9.1.7 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.

**18.9.1.8 embPattern\_stitchEllipse()** void embPattern\_stitchEllipse (

```

 EmbPattern * p,
 EmbEllipse ellipse,
 int thread_index,
 int style)

```

*p ellipse thread\_index style*

**Todo** finish stitchEllipse

**18.9.1.9 embPattern\_stitchPath()** void embPattern\_stitchPath (

```

 EmbPattern * p,
 EmbPath path,
 int thread_index,
 int style)

```

*p rect thread\_index style*

**Todo** finish stitch path

**18.9.1.10 embPattern\_stitchPolygon()** void embPattern\_stitchPolygon (

```

 EmbPattern * p,
 EmbPolygon polygon,
 int thread_index,
 int style)

```

*p rect thread\_index style*

**Todo** finish stitch polygon

**18.9.1.11 embPattern\_stitchPolyline()** void embPattern\_stitchPolyline (

```

 EmbPattern * p,
 EmbPolyline polyline,
 int thread_index,
 int style)

```

*p rect thread\_index style*

**Todo** finish stitch polyline

**18.9.1.12 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.

**18.9.1.13 embPattern\_stitchText()** void embPattern\_stitchText (   
     EmbPattern \* p,   
     EmbRect rect,   
     int thread\_index,   
     int style )

*p rect thread\_index style*

**18.9.1.14 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".

**18.9.1.15 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.

**18.9.1.16 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

**18.9.1.17 greedy\_algorithm()** static void greedy\_algorithm (   
     int \* points,   
     int n\_points,   
     int width,   
     EmbReal bias ) [static]

*points n\_points width bias*

**18.9.1.18 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.

**18.9.1.19 hilbert\_curve()** `int hilbert_curve (`  
`EmbPattern * pattern,`  
`int iterations )`

*pattern iterations*

[https://en.wikipedia.org/wiki/Hilbert\\_curve](https://en.wikipedia.org/wiki/Hilbert_curve)

Using the Lindenmayer System, so we can save work across different functions.

**18.9.1.20 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.

**18.9.1.21 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.

**18.9.1.22 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*

**18.9.1.23 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.

## 18.9.2 Variable Documentation

**18.9.2.1 hilbert\_curve\_l\_system** `L_system` `hilbert_curve_l_system`

Initial value:

```
= {
 'A', "AB", "F+-", (char**)rules
}
```



**18.9.2.2 rules** `const char* rules[] = {"+BF-AFA-FB+", "-AF+BFB+FA-"}`

## 18.10 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`]

### 18.10.1 Function Documentation

**18.10.1.1 `binaryWriteInt()`** `void binaryWriteInt (`  
    `FILE * f,`  
    `int data )`  
*f data*

**Todo** replace with `emblnt_read`

**18.10.1.2 `binaryWriteIntBE()`** `void binaryWriteIntBE (`  
    `FILE * f,`  
    `int data )`  
*f data*

**Todo** replace with `emblnt_read`

**18.10.1.3 `binaryWriteShort()`** `void binaryWriteShort (`  
    `FILE * f,`  
    `short data )`  
*f data*

**Todo** replace with `emblnt_read`

**18.10.1.4 `binaryWriteUInt()`** `void binaryWriteUInt (`  
    `FILE * f,`  
    `unsigned int data )`  
*f data*

**Todo** replace with `emblnt_read`

**18.10.1.5 `binaryWriteUIntBE()`** `void binaryWriteUIntBE (`  
    `FILE * f,`  
    `unsigned int data )`  
*f data*

**Todo** replace with `emblnt_read`

**18.10.1.6 `binaryWriteUShort()`** `void binaryWriteUShort (`  
    `FILE * f,`  
    `unsigned short data )`  
*f data*

**Todo** replace with `emblnt_read`

**18.10.1.7 binaryWriteUShortBE()** void binaryWriteUShortBE (   
FILE \* *f*,  
unsigned short *data* )  
*f data*

**Todo** replace with embInt\_read

**18.10.1.8 emb\_identify\_format()** int emb\_identify\_format (   
const char \* *fileName* )  
*fileName*

Returns

int

**18.10.1.9 embFormat\_getExtension()** int embFormat\_getExtension (   
const char \* *fileName*,  
char \* *ending* )  
*fileName ending*

Returns

int

**18.10.1.10 embPattern\_read()** char embPattern\_read (   
EmbPattern \* *pattern*,  
const char \* *fileName*,  
int *format* )  
*pattern fileName format*

Returns

char

**18.10.1.11 embPattern\_readAuto()** char embPattern\_readAuto (   
EmbPattern \* *pattern*,  
const char \* *fileName* )  
*pattern fileName*

Returns

char

**18.10.1.12 embPattern\_write()** char embPattern\_write (   
EmbPattern \* *pattern*,  
const char \* *fileName*,  
int *format* )  
*pattern fileName format*

Returns

char

**18.10.1.13 embPattern\_writeAuto()** `char embPattern_writeAuto (`  
    `EmbPattern * pattern,`  
    `const char * fileName )`  
*pattern fileName*

Returns

char

**18.10.1.14 fpad()** `void fpad (`  
    `FILE * file,`  
    `char c,`  
    `int n )`

*f*

Returns

int

**18.10.1.15 fread\_int16()** `short fread_int16 (`  
    `FILE * f )`

*f*

Returns

short

**18.10.1.16 fread\_int32\_be()** `int fread_int32_be (`  
    `FILE * f )`

*f*

Returns

int

**Todo** replace with `emblnt_read`

**18.10.1.17 fread\_uint16()** `unsigned short fread_uint16 (`  
    `FILE * f )`

*f*

Returns

unsigned short

**Todo** replace with `emblnt_read`

**18.10.1.18 safe\_free()** `void safe_free (`  
    `void * data )`

*data*

## 18.10.2 Variable Documentation

**18.10.2.1 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). \*

**18.10.2.2 imageWithFrame** `const char imageWithFrame[38][48]`**18.11 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)

**18.11.1 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.

**18.11.2 Function Documentation**

**18.11.2.1 read100()** `char read100 (`  
     `EmbPattern * pattern,`  
     `FILE * file )`

**18.11.2.2 write100()** `char write100 (`  
     `EmbPattern * pattern,`  
     `FILE * file )`

**18.12 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)

**18.12.1 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.

## 18.12.2 Function Documentation

**18.12.2.1 read10o()** `char read10o (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.12.2.2 write10o()** `char write10o (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.13 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)

### 18.13.1 Detailed Description

The Bernina Embroidery Format (.art)  
We don't know much about this format.

**Todo** Find a source.

## 18.13.2 Function Documentation

**18.13.2.1 readArt()** `char readArt (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.13.2.2 writeArt()** `char writeArt (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.14 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)

### 18.14.1 Detailed Description

The Bitmap Cache Embroidery Format (.bmc)  
We don't know much about this format.

**Todo** Find a source.

### 18.14.2 Function Documentation

**18.14.2.1 readBmc()** `char readBmc (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.14.2.2 writeBmc()** `char writeBmc (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.15 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`)

### 18.15.1 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.

### 18.15.2 Function Documentation

**18.15.2.1 readBro()** `char readBro (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.15.2.2 writeBro()** `char writeBro (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.16 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)

### 18.16.1 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.

### 18.16.2 Function Documentation

**18.16.2.1 readCnd()** `char readCnd (`  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.16.2.2 writeCnd()** `char writeCnd (`  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.17 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)

### 18.17.1 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 seperated 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.

**18.17.1.0.1 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
```



### 18.17.2 Function Documentation

**18.17.2.1 readCol()** char readCol (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.17.2.2 writeCol()** char writeCol (  
    EmbPattern \* pattern,  
    FILE \* file )

## 18.18 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 []

### 18.18.1 Detailed Description

The Singer Embroidery Format (.csd)  
Stitch Only Format.

### 18.18.2 Macro Definition Documentation

**18.18.2.1 CsdSubMaskSize** #define CsdSubMaskSize 479

**18.18.2.2 CsdXorMaskSize** #define CsdXorMaskSize 501

### 18.18.3 Function Documentation

**18.18.3.1 BuildDecryptionTable()** void BuildDecryptionTable (  
int seed )

**18.18.3.2 DecodeCsdByte()** unsigned char DecodeCsdByte (  
long fileOffset,  
unsigned char val,  
int type )

**18.18.3.3 readCsd()** char readCsd (  
EmbPattern \* pattern,  
FILE \* file )

**18.18.3.4 writeCsd()** char writeCsd (  
EmbPattern \* pattern,  
FILE \* file )

## 18.18.4 Variable Documentation

**18.18.4.1 \_subMask** char \_subMask[CsdSubMaskSize]

**18.18.4.2 \_xorMask** char \_xorMask[CsdXorMaskSize]

**18.18.4.3 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
}
```

## 18.19 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)

### 18.19.1 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.

**18.19.1.0.1 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.

### 18.19.1.0.2 EmBird CSV Dialect

## 18.19.2 Function Documentation

**18.19.2.1 csvStitchFlagToStr()** char \* csvStitchFlagToStr (   
int flags )

**18.19.2.2 csvStrToStitchFlag()** int csvStrToStitchFlag (   
const char \* str )

**18.19.2.3 readCsv()** char readCsv (   
EmbPattern \* pattern,   
FILE \* file )

**18.19.2.4 writeCsv()** char writeCsv (   
EmbPattern \* pattern,   
FILE \* file )

## 18.20 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)

### 18.20.1 Function Documentation

**18.20.1.1 readDat()** char readDat (   
EmbPattern \* pattern,   
FILE \* file )

**18.20.1.2 writeDat()** `char writeDat (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.21 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)

#### 18.21.1 Detailed Description

The Melco Embroidery Format (.dem)  
Stitch Only Format

#### 18.21.2 Function Documentation

**18.21.2.1 readDem()** `char readDem (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.21.2.2 writeDem()** `char writeDem (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.22 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)

#### 18.22.1 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

### 18.22.2 Function Documentation

**18.22.2.1 readDsb()** `char readDsb (`  
     `EmbPattern * pattern,`  
     `FILE * file )`

**18.22.2.2 writeDsb()** `char writeDsb (`  
     `EmbPattern * pattern,`  
     `FILE * file )`

## 18.23 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)

### 18.23.1 Detailed Description

.DST (Tajima) embroidery file read/write routines Format comments are thanks to [tspilman@dalcoathletic.com](mailto:tspilman@dalcoathletic.com) who's notes appeared at <http://www.wotsit.org> under Tajima Format.

**18.23.1.1 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](mailto:tspilman@dalcoathletic.com) who's notes appeared at <http://www.wotsit.org> under Tajima Format.

Other references: [2], [1].

**18.23.1.1.1 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 %07d, 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.                                                                                                                                                                           |         |

**18.23.1.1.2 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.

## 18.23.2 Macro Definition Documentation

**18.23.2.1 cci** #define cci(  
    c1,  
    c2 ) (c1\*256+c2)

## 18.23.3 Function Documentation

**18.23.3.1 decode\_record\_flags()** int decode\_record\_flags (  
    unsigned char b2 )

**18.23.3.2 encode\_record()** void encode\_record (

```
FILE * file,
int x,
int y,
int flags)
```

**18.23.3.3 readDst()** char readDst (

```
EmbPattern * pattern,
FILE * file)
```

**18.23.3.4 set\_dst\_variable()** void set\_dst\_variable (

```
EmbPattern * pattern,
char * var,
char * val)
```

**18.23.3.5 writeDst()** char writeDst (

```
EmbPattern * pattern,
FILE * file)
```

## 18.24 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)

### 18.24.1 Function Documentation

**18.24.1.1 readDsz()** char readDsz (

```
EmbPattern * pattern,
FILE * file)
```

**18.24.1.2 ZSK USA Embroidery Format (.dsz)** The ZSK USA dsz format is stitch-only.

**18.24.1.3 writeDsz()** char writeDsz (

```
EmbPattern * pattern,
FILE * file)
```

## 18.25 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)

### 18.25.1 Function Documentation

**18.25.1.1 readDxf()** char readDxf (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.25.1.2 readLine()** void readLine (  
    FILE \* file,  
    char \* str )

**18.25.1.3 Drawing Exchange Format (.dxf)** Graphics format for drawing files designed and used by AutoDesk for their AutoCAD program. [[dxf\\_reference](#)]

**18.25.1.4 writeDxf()** char writeDxf (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.26 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)

### 18.26.1 Function Documentation

**18.26.1.1 readEdr()** char readEdr (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.26.1.2 Embird Embroidery Format (.edr)** Stitch Only Format

**18.26.1.3 writeEdr()** char writeEdr (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )



## 18.27 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)

#### 18.27.1 Detailed Description

The Elna Embroidery Format (.emd)  
Stitch Only Format.

#### 18.27.2 Function Documentation

**18.27.2.1 [emdDecode\(\)](#)** char emdDecode (  
    unsigned char *inputByte* )

**18.27.2.2 [readEmd\(\)](#)** char readEmd (  
    [EmbPattern](#) \* *pattern*,  
    FILE \* *file* )

**18.27.2.3 [writeEmd\(\)](#)** char writeEmd (  
    [EmbPattern](#) \* *pattern*,  
    FILE \* *file* )

## 18.28 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)

#### 18.28.1 Function Documentation

**18.28.1.1 [expDecode\(\)](#)** char expDecode (  
    unsigned char *a1* )

**18.28.1.2 Melco Embroidery Format (.exp)** Stitch Only Format.

**18.28.1.3 readExp()** `char readExp (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.28.1.4 writeExp()** `char writeExp (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.29 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)

#### 18.29.1 Function Documentation

**18.29.1.1 decode\_exy\_flags()** `int decode_exy_flags (`  
    `unsigned char b2 )`

**18.29.1.2 Eltac Embroidery Format (.exy)** Stitch Only Format.

**18.29.1.3 readExy()** `char readExy (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.29.1.4 writeExy()** `char writeExy (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.30 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)

#### 18.30.1 Function Documentation

**18.30.1.1 readEys()** `char readEys (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.30.1.2 Sierra Expanded Embroidery Format (.eys)** Stitch Only Format.  
Smoothie G-Code Embroidery Format (.fxy)?

**18.30.1.3 writeEys()** `char writeEys (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.31 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)

#### 18.31.1 Function Documentation

**18.31.1.1 readFxy()** `char readFxy (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.31.1.2 Embroidery Format (.fxy)** Stitch Only Format.

**18.31.1.3 writeFxy()** `char writeFxy (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.32 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)

#### 18.32.1 Function Documentation

**18.32.1.1 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

**18.32.1.2 writeGc()** `char writeGc (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.33 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`)

#### 18.33.1 Function Documentation

**18.33.1.1 readGnc()** `char readGnc (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.33.1.2 Great Notions Embroidery Format (.gnc)** Stitch Only Format.

**18.33.1.3 writeGnc()** `char writeGnc (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.34 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`)

#### 18.34.1 Function Documentation

**18.34.1.1 readGt()** `char readGt (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.34.1.2 Gold Thread Embroidery Format (.gt) Stitch Only Format.**

**18.34.1.3 writeGt()** char writeGt (  
     EmbPattern \* pattern,  
     FILE \* file )

**18.35 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)

**18.35.1 Function Documentation**

**18.35.1.1 husCompressData()** unsigned char \* husCompressData (  
     unsigned char \* input,  
     int decompressedInputSize,  
     int \* compressedSize )

**18.35.1.2 husDecodeByte()** int husDecodeByte (  
     unsigned char b )

**18.35.1.3 husDecodeStitchType()** int husDecodeStitchType (  
     unsigned char b )

**18.35.1.4 Husqvarna Viking Embroidery Format (.hus) Stitch Only Format.**

**18.35.1.5 husDecompressData()** unsigned char \* husDecompressData (  
     unsigned char \* input,  
     int compressedInputLength,  
     int decompressedContentLength )

**18.35.1.6 husEncodeByte()** unsigned char husEncodeByte (  
     EmbReal f )

**18.35.1.7 husEncodeStitchType()** unsigned char husEncodeStitchType (  
int st )

**18.35.1.8 readHus()** char readHus (  
EmbPattern \* pattern,  
FILE \* file )

**18.35.1.9 writeHus()** char writeHus (  
EmbPattern \* pattern,  
FILE \* file )

## 18.36 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)

#### 18.36.1 Function Documentation

**18.36.1.1 readInb()** char readInb (  
EmbPattern \* pattern,  
FILE \* file )

**18.36.1.2 Inbro Embroidery Format (.inb)** Stitch Only Format.

**18.36.1.3 writeInb()** char writeInb (  
EmbPattern \* pattern,  
FILE \* file )

## 18.37 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)

#### 18.37.1 Function Documentation

**18.37.1.1 readInf()** `char readInf (`  
     `EmbPattern * pattern,`  
     `FILE * file )`

**18.37.1.2 Embroidery Color Format (.inf)** Stitch Only Format.

**18.37.1.3 writeInf()** `char writeInf (`  
     `EmbPattern * pattern,`  
     `FILE * file )`

## 18.38 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)

### 18.38.1 Function Documentation

**18.38.1.1 jefDecode()** `char jefDecode (`  
     unsigned char *inputByte* )

**18.38.1.2 jefEncode()** `void jefEncode (`  
     unsigned char \* *b*,  
     char *dx*,  
     char *dy*,  
     int *flags* )

**18.38.1.3 jefGetHoopSize()** `int jefGetHoopSize (`  
     int *width*,  
     int *height* )

**18.38.1.4 Janome Embroidery Format (.jef)** Stitch Only Format.

**18.38.1.5 jefSetHoopFromId()** `void jefSetHoopFromId (`  
     EmbPattern \* *pattern*,  
     int *hoopCode* )

**18.38.1.6 read\_hoop()** void read\_hoop (   
FILE \* *file*,   
struct hoop\_padding \* *hoop*,   
char \* *label* )

**18.38.1.7 readJef()** char readJef (   
EmbPattern \* *pattern*,   
FILE \* *file* )

**18.38.1.8 writeJef()** char writeJef (   
EmbPattern \* *pattern*,   
FILE \* *file* )

## 18.39 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*)

### 18.39.1 Function Documentation

**18.39.1.1 ksmEncode()** void ksmEncode (   
unsigned char \* *b*,   
char *dx*,   
char *dy*,   
int *flags* )

**18.39.1.2 Pfaff professional Design format (.ksm)** Stitch Only Format.

**18.39.1.3 readKsm()** char readKsm (   
EmbPattern \* *pattern*,   
FILE \* *file* )

**18.39.1.4 writeKsm()** char writeKsm (   
EmbPattern \* *pattern*,   
FILE \* *file* )

## 18.40 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](#) []

### 18.40.1 Function Documentation

**18.40.1.1 readMax()** char readMax (  
[EmbPattern](#) \* pattern,  
 FILE \* file )

**18.40.1.2 writeMax()** char writeMax (  
[EmbPattern](#) \* pattern,  
 FILE \* file )

### 18.40.2 Variable Documentation

**18.40.2.1 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, 0x00, 0x00, 0x00, 0x00, 0x00, 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, 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, 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, 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
}
```

**18.40.2.2 Pfaff Embroidery Format (.max)** Stitch Only Format.

## 18.41 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)

### 18.41.1 Function Documentation

**18.41.1.1 readMit()** `char readMit (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.41.1.2 Mitsubishi Embroidery Format (.mit)** Stitch Only Format.

**18.41.1.3 writeMit()** `char writeMit (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.42 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)

#### 18.42.1 Function Documentation

**18.42.1.1 readNew()** `char readNew (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.42.1.2 Ameco Embroidery Format (.new)** Stitch Only Format.

**18.42.1.3 writeNew()** `char writeNew (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.43 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)

### 18.43.1 Function Documentation

**18.43.1.1 ofmDecode()** `EmbReal ofmDecode (`  
    `unsigned char b1,`  
    `unsigned char b2 )`

**18.43.1.2 ofmReadBlockHeader()** `void ofmReadBlockHeader (`  
    `FILE * file )`

**18.43.1.3 ofmReadClass()** `static int ofmReadClass (`  
    `FILE * file ) [static]`

**18.43.1.4 ofmReadColorChange()** `void ofmReadColorChange (`  
    `FILE * file,`  
    `EmbPattern * pattern )`

**18.43.1.5 ofmReadExpanded()** `void ofmReadExpanded (`  
    `FILE * file,`  
    `EmbPattern * p )`

**18.43.1.6 ofmReadLibrary()** `char * ofmReadLibrary (`  
    `FILE * file )`

**18.43.1.7 Melco Embroidery Format (.ofm)** Stitch Only Format.

**18.43.1.8 ofmReadThreads()** `void ofmReadThreads (`  
    `FILE * file,`  
    `EmbPattern * p )`

**18.43.1.9 readOfm()** `char readOfm (`  
    `EmbPattern * pattern,`  
    `FILE * fileCompound )`

**18.43.1.10 writeOfm()** `char writeOfm (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.44 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)

### 18.44.1 Function Documentation

**18.44.1.1 readPcd()** `char readPcd (`  
    [EmbPattern](#) \* *pattern*,  
    const char \* *fileName*,  
    FILE \* *file* )

#### 18.44.1.2 Pfaff PCD File Format (.pcd) Stitch Only Format.

The format uses a signed 3 byte-length number type.

See the description here ([5](5)) for the overview of the format.

For an example of the format see ([11](11)).

**18.44.1.3 writePcd()** `char writePcd (`  
    [EmbPattern](#) \* *pattern*,  
    FILE \* *file* )

## 18.45 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)

### 18.45.1 Function Documentation

**18.45.1.1 readPcm()** `char readPcm (`  
    [EmbPattern](#) \* *pattern*,  
    FILE \* *file* )

#### 18.45.1.2 Pfaff Embroidery Format (.pcm) The Pfaff pcm format is stitch-only.

**18.45.1.3 writePcm()** `char writePcm (`  
    [EmbPattern](#) \* *pattern*,  
    FILE \* *file* )

## 18.46 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)

### 18.46.1 Function Documentation

**18.46.1.1 readPcq()** char readPcq (  
    [EmbPattern](#) \* pattern,  
    const char \* fileName,  
    FILE \* file )

**18.46.1.2 Embroidery Format (.pcq)** The Pfaff pcq format is stitch-only.

**18.46.1.3 writePcq()** char writePcq (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.47 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)

### 18.47.1 Function Documentation

**18.47.1.1 readPcs()** char readPcs (  
    [EmbPattern](#) \* pattern,  
    const char \* fileName,  
    FILE \* file )

**18.47.1.2 Embroidery Format (.pcq)** The Pfaff pcs format is stitch-only.

**18.47.1.3 writePcs()** char writePcs (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.48 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)

### 18.48.1 Function Documentation

**18.48.1.1 `pecEncode()`** void `pecEncode` (  
FILE \* *file*,  
EmbPattern \* *p* )

**18.48.1.2 `pecEncodeJump()`** void `pecEncodeJump` (  
FILE \* *file*,  
int *x*,  
int *types* )

**18.48.1.3 `pecEncodeStop()`** void `pecEncodeStop` (  
FILE \* *file*,  
unsigned char *val* )

**18.48.1.4 `readPec()`** char `readPec` (  
EmbPattern \* *pattern*,  
const char \* *fileName*,  
FILE \* *file* )

**18.48.1.5 `readPecStitches()`** void `readPecStitches` (  
EmbPattern \* *pattern*,  
FILE \* *file* )

**18.48.1.6 Embroidery Format (.pec)** The Brother pec format is stitch-only.

**18.48.1.7 `writelImage()`** void `writeImage` (  
FILE \* *file*,  
unsigned char *image*[ ][48] )

Write a PES embedded *image* to the given *file* pointer.

**18.48.1.8 `writePec()`** char `writePec` (  
EmbPattern \* *pattern*,  
const char \* *fileName*,  
FILE \* *file* )

**18.48.1.9 writePecStitches()** void writePecStitches (  
    EmbPattern \* pattern,  
    FILE \* file,  
    const char \* fileName )

## 18.49 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)

#### 18.49.1 Function Documentation

**18.49.1.1 readPel()** char readPel (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.49.1.2 Embroidery Format (.pec)** The Brother pel format is stitch-only.

**18.49.1.3 writePel()** char writePel (  
    EmbPattern \* pattern,  
    FILE \* file )

## 18.50 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)

#### 18.50.1 Function Documentation

**18.50.1.1 readPem()** char readPem (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.50.1.2 Embroidery Format (.pec)** The Brother pem format is stitch-only.

**18.50.1.3 writePem()** char writePem (  
    EmbPattern \* pattern,  
    FILE \* file )

## 18.51 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

#### 18.51.1 Function Documentation

**18.51.1.1 [pesWriteEmbOneSection\(\)](#)** void [pesWriteEmbOneSection](#) (  
[EmbPattern](#) \* *pattern*,  
FILE \* *file* )

**18.51.1.2 [pesWriteSewSegSection\(\)](#)** void [pesWriteSewSegSection](#) (  
[EmbPattern](#) \* *pattern*,  
FILE \* *file* )

**18.51.1.3 [readDescriptions\(\)](#)** void [readDescriptions](#) (  
FILE \* *file*,  
[EmbPattern](#) \* *pattern* )

**18.51.1.4 [readFeatherPatterns\(\)](#)** void [readFeatherPatterns](#) (  
FILE \* *file*,  
[EmbPattern](#) \* *pattern* )



**18.51.1.5 readHoopName()** void readHoopName (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.51.1.6 readImageString()** void readImageString (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.51.1.7 readMotifPatterns()** void readMotifPatterns (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.51.1.8 readPes()** char readPes (  
EmbPattern \* *pattern*,  
const char \* *fileName*,  
FILE \* *file* )

**18.51.1.9 readPESHeaderV10()** void readPESHeaderV10 (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.51.1.10 readPESHeaderV5()** void readPESHeaderV5 (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.51.1.11 readPESHeaderV6()** void readPESHeaderV6 (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.51.1.12 readPESHeaderV7()** void readPESHeaderV7 (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.51.1.13 readPESHeaderV8()** void readPESHeaderV8 (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.51.1.14 readPESHeaderV9()** void readPESHeaderV9 (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.51.1.15 readProgrammableFills()** void readProgrammableFills (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.51.1.16 readThreads()** `void readThreads (`  
    `FILE * file,`  
    `EmbPattern * pattern )`

**18.51.1.17 writePes()** `char writePes (`  
    `EmbPattern * pattern,`  
    `const char * fileName,`  
    `FILE * file )`

## 18.51.2 Variable Documentation

**18.51.2.1 pes\_version** `int pes_version = PES0001`

**18.51.2.2 pes\_version\_strings** `const char* pes_version_strings[]`  
**Initial value:**

```
= {
 "#PES0001",
 "#PES0020",
 "#PES0022",
 "#PES0030",
 "#PES0040",
 "#PES0050",
 "#PES0055",
 "#PES0056",
 "#PES0060",
 "#PES0070",
 "#PES0080",
 "#PES0090",
 "#PES0100",
}
```

**18.51.2.3 Embroidery Format (.pec)** The Brother pes format is stitch-only.

## 18.52 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)`

### 18.52.1 Function Documentation

**18.52.1.1 readPhb()** `char readPhb (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.52.1.2 Embroidery Format (.pec)** The Brother phb format is stitch-only.

**18.52.1.3 writePhb()** `char writePhb (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.53 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)

#### 18.53.1 Function Documentation

**18.53.1.1 readPhc()** char readPhc (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.53.1.2 Embroidery Format (.pec)** The Brother phc format is stitch-only.

**18.53.1.3 writePhc()** char writePhc (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.54 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)

#### 18.54.1 Function Documentation

**18.54.1.1 readPlt()** char readPlt (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.54.1.2 Embroidery Format (.plt)** The AutoCAD plt format is stitch-only.

**18.54.1.3 writePlt()** char writePlt (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.55 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)

#### 18.55.1 Function Documentation

**18.55.1.1 readRgb()** char readRgb (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.55.1.2 Color File (.rgb)** The RGB format is a color-only format to act as an external color file for other formats.

**18.55.1.3 writeRgb()** char writeRgb (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.56 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)

#### 18.56.1 Function Documentation

**18.56.1.1 readSew()** char readSew (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.56.1.2 sewDecode()** char sewDecode (  
    unsigned char inputByte )

**18.56.1.3 Embroidery Format (.sew)** The Janome sew format is stitch-only.

**18.56.1.4 writeSew()** char writeSew (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.57 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)

### 18.57.1 Function Documentation

**18.57.1.1 [readShv\(\)](#)** char readShv (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.57.1.2 [shvDecode\(\)](#)** char shvDecode (  
    unsigned char inputByte )

**18.57.1.3 Viking Embroidery Format (.shv)** The Husqvarna Viking shv format is stitch-only.

**18.57.1.4 [shvDecodeShort\(\)](#)** short shvDecodeShort (  
    unsigned short inputByte )

**18.57.1.5 [writeShv\(\)](#)** char writeShv (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.58 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)

### 18.58.1 Function Documentation

**18.58.1.1 [readSst\(\)](#)** char readSst (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.58.1.2 Embroidery Format (.sst)** The Sunstar sst format is stitch-only.

**18.58.1.3 writeSst()** `char writeSst (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.59 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`)

### 18.59.1 Function Documentation

**18.59.1.1 readStx()** `char readStx (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.59.1.2 stxReadThread()** `int stxReadThread (`  
    `StxThread * thread,`  
    `FILE * file )`

**18.59.1.3 Stitch Embroidery Format (.stx)** The Data Stitch stx format is stitch-only.

**18.59.1.4 writeStx()** `char writeStx (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.60 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]

## 18.60.1 Function Documentation

**18.60.1.1 `readSvg()`** `char readSvg (`  
     `EmbPattern * pattern,`  
     `FILE * file )`

**18.60.1.2 `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`.

## 18.60.2 Variable Documentation

**18.60.2.1 `attributeList`** `SvgAttribute attributeList`[1000]

**18.60.2.2 `current_element_id`** `int current_element_id`

**18.60.2.3 `currentAttribute`** `char currentAttribute`[1000]

**18.60.2.4 `currentValue`** `char currentValue`[1000]

**18.60.2.5 `n_attributes`** `int n_attributes` = 0

**18.60.2.6 `svgCreator`** `int svgCreator`

**18.60.2.7 `Vector Graphics (.svg)`** The scalable vector graphics (SVG) format is a graphics format maintained by ...

**18.60.2.8 `svgExpect`** `int svgExpect`

**18.60.2.9 `svgMultiValue`** `int svgMultiValue`

## 18.61 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)

#### 18.61.1 Function Documentation

**18.61.1.1 readT01()** char readT01 (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.61.1.2 Embroidery Format (.pcq)** The Pfaff t01 format is stitch-only.

**18.61.1.3 writeT01()** char writeT01 (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.62 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)

#### 18.62.1 Function Documentation

**18.62.1.1 readT09()** char readT09 (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.62.1.1.1 Embroidery Format (.pcq)** The Pfaff t09 format is stitch-only.

**18.62.1.2 writeT09()** char writeT09 (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )



## 18.63 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)

### 18.63.1 Function Documentation

**18.63.1.1 [decode\\_tap\\_record\\_flags\(\)](#)** int [decode\\_tap\\_record\\_flags](#) (  
unsigned char *b2* )

**18.63.1.2 [encode\\_tap\\_record\(\)](#)** void [encode\\_tap\\_record](#) (  
FILE \* *file*,  
int *x*,  
int *y*,  
int *flags* )

**18.63.1.3 Embroidery Format (.tap)** The Happy tap format is stitch-only.

**18.63.1.4 [readTap\(\)](#)** char [readTap](#) (  
[EmbPattern](#) \* *pattern*,  
FILE \* *file* )

**18.63.1.5 [writeTap\(\)](#)** char [writeTap](#) (  
[EmbPattern](#) \* *pattern*,  
FILE \* *file* )

## 18.64 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)

### 18.64.1 Function Documentation

**18.64.1.1 readThr()** `char readThr (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.64.1.2 Embroidery Format (.thr)** The ThreadWorks thr format is stitch-only.

**18.64.1.3 writeThr()** `char writeThr (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.65 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)

#### 18.65.1 Function Documentation

**18.65.1.1 readTxt()** `char readTxt (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.65.1.2 File (.txt)** The txt format is stitch-only and isn't associated with a specific company.

**18.65.1.3 writeTxt()** `char writeTxt (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.66 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)

#### 18.66.1 Function Documentation

**18.66.1.1 readU00()** `char readU00 (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.66.1.2 Embroidery Format (.u00)** The Barudan u00 format is stitch-only.

**18.66.1.3 writeU00()** `char writeU00 (`  
     `EmbPattern * pattern,`  
     `FILE * file )`

## 18.67 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)

### 18.67.1 Function Documentation

**18.67.1.1 readU01()** `char readU01 (`  
     `EmbPattern * pattern,`  
     `FILE * file )`

**18.67.1.2 Embroidery Format (.u01)** The Barudan u01 format is stitch-only.

**18.67.1.3 writeU01()** `char writeU01 (`  
     `EmbPattern * pattern,`  
     `FILE * file )`

## 18.68 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](#) []

## 18.68.1 Function Documentation

**18.68.1.1 readVip()** char readVip (  
     EmbPattern \* pattern,  
     FILE \* file )

**18.68.1.2 vipCompressData()** unsigned char \* vipCompressData (  
     unsigned char \* input,  
     int decompressedInputSize,  
     int \* compressedSize )

**18.68.1.3 vipDecodeByte()** int vipDecodeByte (  
     unsigned char b )

**18.68.1.4 vipDecodeStitchType()** int vipDecodeStitchType (  
     unsigned char b )

**18.68.1.5 vipDecompressData()** unsigned char \* vipDecompressData (  
     unsigned char \* input,  
     int compressedInputLength,  
     int decompressedContentLength )

**18.68.1.6 vipEncodeByte()** unsigned char vipEncodeByte (  
     EmbReal f )

**18.68.1.7 vipEncodeStitchType()** unsigned char vipEncodeStitchType (  
     int st )

**18.68.1.8 writeVip()** char writeVip (  
     EmbPattern \* pattern,  
     FILE \* file )

## 18.68.2 Variable Documentation

**18.68.2.1 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
}

```

**18.68.2.2 Embroidery Format (.pcq)** The Pfaff vip format is stitch-only.

## 18.69 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)

### 18.69.1 Function Documentation

**18.69.1.1 readVp3()** char readVp3 (  
[EmbPattern](#) \* pattern,  
FILE \* file )

**18.69.1.2 vp3Decode()** int vp3Decode (  
unsigned char inputByte )

**18.69.1.3 vp3DecodeInt16()** short vp3DecodeInt16 (  
unsigned short inputByte )

**18.69.1.4 vp3PatchByteCount()** void vp3PatchByteCount (  
FILE \* file,  
int offset,  
int adjustment )

**18.69.1.5** `vp3ReadHoopSection()` `vp3Hoop` `vp3ReadHoopSection` (  
    FILE \* *file* )

**18.69.1.6** `vp3ReadString()` unsigned char \* `vp3ReadString` (  
    FILE \* *file* )

**18.69.1.7** **Embroidery Format (.pcq)** The Pfaff vp3 format is stitch-only.

**18.69.1.8** `vp3WriteString()` void `vp3WriteString` (  
    FILE \* *file*,  
    const char \* *str* )

**18.69.1.9** `vp3WriteStringLen()` void `vp3WriteStringLen` (  
    FILE \* *file*,  
    const char \* *str*,  
    int *len* )

**18.69.1.10** `writeVp3()` char `writeVp3` (  
    EmbPattern \* *pattern*,  
    FILE \* *file* )

## 18.70 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*)

### 18.70.1 Function Documentation

**18.70.1.1** `readXxx()` char `readXxx` (  
    EmbPattern \* *pattern*,  
    FILE \* *file* )

**18.70.1.2** `writeXxx()` char `writeXxx` (  
    EmbPattern \* *pattern*,  
    FILE \* *file* )

**18.70.1.3** `xxxDecodeByte()` char `xxxDecodeByte` (  
    unsigned char *inputByte* )

**18.70.1.4 Embroidery Format (.xxx)** The Singer xxx format is stitch-only.

**18.70.1.5 xxxEncodeDesign()** void xxxEncodeDesign (

```
FILE * file,
EmbPattern * p)
```

**18.70.1.6 xxxEncodeStitch()** void xxxEncodeStitch (

```
FILE * file,
EmbReal deltaX,
EmbReal deltaY,
int flags)
```

**18.70.1.7 xxxEncodeStop()** void xxxEncodeStop (

```
FILE * file,
EmbStitch s)
```

## 18.71 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)

#### 18.71.1 Detailed Description

The ZSK USA Embroidery Format (.zsk)  
The ZSK USA zsk format is stitch-only.

#### 18.71.2 Function Documentation

**18.71.2.1 readZsk()** char readZsk (

```
EmbPattern * pattern,
FILE * file)
```

**18.71.2.2 writeZsk()** char writeZsk (

```
EmbPattern * pattern,
FILE * file)
```

## 18.72 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.*

### 18.72.1 Function Documentation

**18.72.1.1 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).

**18.72.1.2 embGeometry\_free()** `void embGeometry_free ( EmbGeometry * obj )`

Free the memory occupied by a non-stitch geometry object.  
*obj* Pointer to geometry memory.

**18.72.1.3 embGeometry\_init()** `EmbGeometry * embGeometry_init ( int type_in )`

Our generic object interface backends to each individual type.  
*type\_in*

#### Returns

EmbGeometry\*

**18.72.1.4 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.

**18.72.1.5 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.



## 18.73 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)

### 18.73.1 Function Documentation

#### 18.73.1.1 [Arc\\_clockwise\(\)](#) char Arc\_clockwise ( )

**18.73.1.2 Base\_objectRubberPoint()** `EmbVector` Base\_objectRubberPoint (   
 `EmbGeometry` \* *obj*,  
const char \* *key* )

**18.73.1.3 Base\_objectRubberText()** const char \* Base\_objectRubberText (   
 `EmbGeometry` \* *obj*,  
const char \* *key* )

**18.73.1.4 Base\_setLineType()** void Base\_setLineType (   
 `EmbGeometry` \* *obj*,  
int *lineType* )

**18.73.1.5 Base\_setLineWeight()** void Base\_setLineWeight (   
 `EmbGeometry` \* *obj*,  
float *lineWeight* )

**18.73.1.6 clockwise()** char clockwise (   
 `EmbGeometry` \* *obj* )

**18.73.1.7 embArc\_arcLength()** float embArc\_arcLength (   
 `EmbArc` *arc* )

**18.73.1.8 embArc\_area()** float embArc\_area (   
 `EmbArc` *arc* )

**18.73.1.9 embArc\_chord()** float embArc\_chord (   
 `EmbArc` *arc* )

**18.73.1.10 embArc\_clockwise()** char embArc\_clockwise (   
 `EmbArc` *arc* )

**18.73.1.11 embArc\_endAngle()** float embArc\_endAngle (   
 `EmbArc` *arc* )

**18.73.1.12 embArc\_gripEdit()** void embArc\_gripEdit (   
 `EmbArc` \* *arc*,  
`EmbVector` *before*,  
`EmbVector` *after* )

**18.73.1.13 embArc\_includedAngle()** float embArc\_includedAngle (   
 `EmbArc` *arc* )

- 18.73.1.14** `embArc_init()` `EmbArc embArc_init (`  
`void )`
- 18.73.1.15** `embArc_mouseSnapPoint()` `EmbVector embArc_mouseSnapPoint (`  
`EmbArc arc,`  
`EmbVector mousePoint )`
- 18.73.1.16** `embArc_paint()` `void embArc_paint (`  
`void )`
- 18.73.1.17** `embArc_setCenter()` `void embArc_setCenter (`  
`EmbArc * arc,`  
`EmbVector point )`
- 18.73.1.18** `embArc_setEndAngle()` `void embArc_setEndAngle (`  
`EmbArc * arc,`  
`float angle )`
- 18.73.1.19** `embArc_setRadius()` `void embArc_setRadius (`  
`EmbArc * arc,`  
`float radius )`
- 18.73.1.20** `embArc_setStartAngle()` `void embArc_setStartAngle (`  
`EmbArc * arc,`  
`float angle )`
- 18.73.1.21** `embArc_startAngle()` `float embArc_startAngle (`  
`EmbArc arc )`
- 18.73.1.22** `embArc_updatePath()` `void embArc_updatePath (`  
`EmbArc arc )`
- 18.73.1.23** `embArc_updateRubber()` `void embArc_updateRubber (`  
`EmbArc arc,`  
`int pattern,`  
`int layer,`  
`int index )`
- 18.73.1.24** `embBase_setColorRGB()` `void embBase_setColorRGB (`  
`EmbGeometry * obj,`  
`unsigned int rgb )`
- 18.73.1.25** `embCircle_prompt()` `void embCircle_prompt (`  
`const char * str )`

**18.73.1.26 embCircle\_setArea()** void embCircle\_setArea (   
    EmbCircle \* circle,   
    float area )

**18.73.1.27 embCircle\_setCircumference()** void embCircle\_setCircumference (   
    EmbCircle \* circle,   
    float circumference )

**18.73.1.28 embEllipse\_click()** void embEllipse\_click (   
    float x,   
    float y )

**18.73.1.29 embEllipse\_main()** void embEllipse\_main ( )

**18.73.1.30 embRect\_bottomLeft()** EmbVector embRect\_bottomLeft (   
    EmbRect rect )

**18.73.1.31 embRect\_bottomRight()** EmbVector embRect\_bottomRight (   
    EmbRect rect )

**18.73.1.32 getArcCenter()** void getArcCenter (   
    EmbArc arc,   
    EmbVector \* arcCenter )

**18.73.1.33 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 )

**18.73.1.34 set\_object\_color()** void set\_object\_color (   
    EmbGeometry \* obj,   
    EmbColor color )

## 18.74 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)

### 18.74.1 Function Documentation

**18.74.1.1 [embCircle\\_area](#)()** [EmbReal](#) [embCircle\\_area](#) (  
[EmbCircle](#) circle )

**18.74.1.2 [embCircle\\_circumference](#)()** [EmbReal](#) [embCircle\\_circumference](#) (  
[EmbCircle](#) circle )

**18.74.1.3 [embCircle\\_init](#)()** [EmbCircle](#) [embCircle\\_init](#) (  
void )

**18.74.1.4 [getCircleCircleIntersections](#)()** int [getCircleCircleIntersections](#) (  
[EmbCircle](#) c0,  
[EmbCircle](#) c1,  
[EmbVector](#) \* p0,  
[EmbVector](#) \* p1 )

**18.74.1.5 [getCircleTangentPoints](#)()** int [getCircleTangentPoints](#) (  
[EmbCircle](#) c,  
[EmbVector](#) point,  
[EmbVector](#) \* t0,  
[EmbVector](#) \* t1 )

## 18.75 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` ()

### 18.75.1 Function Documentation

**18.75.1.1 `ellipse_objectQuadrant0()`** `EmbVector` `ellipse_objectQuadrant0` (  
`EmbEllipse` \* `ellipse` )

**18.75.1.2 `ellipse_objectQuadrant180()`** `EmbVector` `ellipse_objectQuadrant180` (  
`EmbEllipse` \* `ellipse` )

**18.75.1.3 `ellipse_objectQuadrant270()`** `EmbVector` `ellipse_objectQuadrant270` (  
`EmbEllipse` \* `ellipse` )

**18.75.1.4 `ellipse_objectQuadrant90()`** `EmbVector` `ellipse_objectQuadrant90` (  
`EmbEllipse` \* `ellipse` )

**18.75.1.5 `embEllipse_area()`** `EmbReal` `embEllipse_area` (  
`EmbEllipse` `ellipse` )

**18.75.1.6 `embEllipse_diameterX()`** `EmbReal` `embEllipse_diameterX` (  
`EmbEllipse` `ellipse` )

**18.75.1.7 `embEllipse_diameterY()`** `EmbReal` `embEllipse_diameterY` (  
`EmbEllipse` `ellipse` )

**18.75.1.8 `embEllipse_height()`** `EmbReal` `embEllipse_height` (  
`EmbEllipse` `ellipse` )

**18.75.1.9 `embEllipse_init()`** `EmbEllipse` `embEllipse_init` (  
void )

**18.75.1.10 `embEllipse_perimeter()`** `EmbReal` `embEllipse_perimeter` (  
`EmbEllipse` `ellipse` )

**18.75.1.11 embEllipse\_setDiameterMajor()** void embEllipse\_setDiameterMajor (   
     EmbEllipse \* ellipse,   
     float diameter )

**18.75.1.12 embEllipse\_setDiameterMinor()** void embEllipse\_setDiameterMinor (   
     EmbEllipse \* ellipse,   
     float diameter )

**18.75.1.13 embEllipse\_setRadiusMajor()** void embEllipse\_setRadiusMajor (   
     float radius )

**18.75.1.14 embEllipse\_setRadiusMinor()** void embEllipse\_setRadiusMinor (   
     float radius )

**18.75.1.15 embEllipse\_setSize()** void embEllipse\_setSize (   
     float width,   
     float height )

**18.75.1.16 embEllipse\_updatePath()** void embEllipse\_updatePath ( )

**18.75.1.17 embEllipse\_width()** EmbReal embEllipse\_width (   
     EmbEllipse ellipse )

## 18.76 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)

### 18.76.1 Function Documentation

**18.76.1.1 degrees()** EmbReal degrees (   
     EmbReal radian )

**18.76.1.2 emb\_round()** int emb\_round (   
     EmbReal x )

**18.76.1.3 radians()** `EmbReal radians (`  
`EmbReal degree )`

## 18.77 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)

### 18.77.1 Function Documentation

**18.77.1.1 embLine\_intersectionPoint()** `EmbVector embLine_intersectionPoint (`  
`EmbLine line1,`  
`EmbLine line2 )`

**18.77.1.2 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)

**18.77.1.3 embLine\_toVector()** `EmbVector embLine_toVector (`  
`EmbLine line )`

## 18.78 extern/libembroidery/src/geometry/path.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

## 18.79 extern/libembroidery/src/geometry/polygon.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

## 18.80 extern/libembroidery/src/geometry/polyline.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```



## 18.81 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)

#### 18.81.1 Function Documentation

**18.81.1.1 [embRect\\_area](#)()** [EmbReal](#) embRect\_area (  
    [EmbRect](#) rect )

**18.81.1.2 [embRect\\_init](#)()** [EmbRect](#) embRect\_init (  
    void )

## 18.82 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)

#### 18.82.1 Function Documentation

**18.82.1.1 [textSingle\\_gripEdit](#)()** void textSingle\_gripEdit (  
    [EmbVector](#) before,  
    [EmbVector](#) after )

**18.82.1.2 textSingle\_mouseSnapPoint()** `EmbVector textSingle_mouseSnapPoint ( EmbVector mousePoint )`

**18.82.1.3 textSingle\_paint()** `void textSingle_paint ( )`

**18.82.1.4 textSingle\_setJustify()** `void textSingle_setJustify ( const char * justify )`

**18.82.1.5 textSingle\_setTextBackward()** `void textSingle_setTextBackward ( char val )`

**18.82.1.6 textSingle\_setTextBold()** `void textSingle_setTextBold ( char val )`

**18.82.1.7 textSingle\_setTextFont()** `void textSingle_setTextFont ( const char * font )`

**18.82.1.8 textSingle\_setTextItalic()** `void textSingle_setTextItalic ( char val )`

**18.82.1.9 textSingle\_setTextOverline()** `void textSingle_setTextOverline ( char val )`

**18.82.1.10 textSingle\_setTextSize()** `void textSingle_setTextSize ( float size )`

**18.82.1.11 textSingle\_setTextStrikeOut()** `void textSingle_setTextStrikeOut ( char val )`

**18.82.1.12 textSingle\_setTextStyle()** `void textSingle_setTextStyle ( char bold, char italic, char under, char strike, char over )`

**18.82.1.13 textSingle\_setTextUnderline()** `void textSingle_setTextUnderline ( char val )`

**18.82.1.14 textSingle\_setTextUpsideDown()** `void textSingle_setTextUpsideDown ( char val )`

**18.82.1.15 textSingle\_updateRubber()** void textSingle\_updateRubber ( )

## 18.83 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)

### 18.83.1 Function Documentation

**18.83.1.1 embVector\_add()** EmbVector embVector\_add (

```
EmbVector a,
EmbVector b)
```

The sum of vectors *a* and *b* returned as a vector.

Equivalent to:

$$\mathbf{c} = \mathbf{a} + \mathbf{b} = \begin{pmatrix} a_x + b_x \\ a_y + b_y \end{pmatrix}$$

**18.83.1.2 embVector\_angle()** EmbReal embVector\_angle (

```
EmbVector v)
```

The angle, measured anti-clockwise from the x-axis, of a vector *v*.

**18.83.1.3 embVector\_average()** EmbVector embVector\_average (

```
EmbVector a,
EmbVector b)
```

The average of vectors *v1* and *v2* returned as a vector.

Equivalent to:

$$\mathbf{c} = \frac{\mathbf{a} + \mathbf{b}}{2} = \begin{pmatrix} \frac{a_x + b_x}{2} \\ \frac{a_y + b_y}{2} \end{pmatrix}$$

**18.83.1.4 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:

$$|c| = \left| \begin{pmatrix} a_x \\ a_y \\ 0 \end{pmatrix} \times \begin{pmatrix} b_x \\ b_y \\ 0 \end{pmatrix} \right| = \left| \begin{pmatrix} 0 \\ 0 \\ a_x b_y - a_y b_x \end{pmatrix} \right| = a_x b_y - a_y b_x$$

**18.83.1.5 embVector\_distance()** `EmbReal embVector_distance (`  
`EmbVector a,`  
`EmbVector b )`

The distance between *a* and *b* returned as a real value.

$$d = |\mathbf{a} - \mathbf{b}| = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}$$

**18.83.1.6 embVector\_dot()** `EmbReal embVector_dot (`  
`EmbVector a,`  
`EmbVector b )`

The dot product as vectors *v1* and *v2* returned as a `EmbReal`.

Equivalent to:

$$c = \mathbf{a} \cdot \mathbf{b} = a_x b_x + a_y b_y$$

**18.83.1.7 embVector\_length()** `EmbReal embVector_length (`  
`EmbVector vector )`

The length or absolute value of the vector *vector*.

Equivalent to:

$$|v| = \sqrt{v_x^2 + v_y^2}$$

**18.83.1.8 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.

**18.83.1.9 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.

**18.83.1.10** `embVector_relativeX()` `EmbReal` `embVector_relativeX` (  
`EmbVector` *a1*,  
`EmbVector` *a2*,  
`EmbVector` *a3* )

The x-component of the vector

**18.83.1.11** `embVector_relativeY()` `EmbReal` `embVector_relativeY` (  
`EmbVector` *a1*,  
`EmbVector` *a2*,  
`EmbVector` *a3* )

The y-component of the vector

**18.83.1.12** `embVector_subtract()` `EmbVector` `embVector_subtract` (  
`EmbVector` *v1*,  
`EmbVector` *v2* )

The difference between vectors *v1* and *v2* returned as *result*.

Equivalent to:

$$\mathbf{c} = \mathbf{a} - \mathbf{b} = \begin{pmatrix} a_x - b_x \\ a_y - b_y \end{pmatrix}$$

**18.83.1.13** `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  $\begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} xa & xb \end{pmatrix}$

**18.83.1.14** `embVector_unit()` `EmbVector` `embVector_unit` (  
`EmbReal` *alpha* )

The unit vector in the direction *angle*.

$$\mathbf{a}_\alpha = \begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \end{pmatrix}$$

## 18.84 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)

### 18.84.1 Detailed Description

This backends to the stb libraries and nanosvg library.  
 Use Python PEP7 for coding style.

### 18.84.2 Function Documentation

**18.84.2.1 image\_diff()** float image\_diff (

```

 unsigned char * a,
 unsigned char * b,
 int size)

```

The distance between the arrays *a* and *b* of length *size*.

**18.84.2.2 writelImage()** void writeImage (

```

 FILE * file,
 unsigned char image[][48])

```

Write a PES embedded *image* to the given *file* pointer.

## 18.85 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"

### 18.85.1 Macro Definition Documentation



**18.85.1.1 FLAG\_CIRCLE** `#define FLAG_CIRCLE 12`

**18.85.1.2 FLAG\_CIRCLE\_SHORT** `#define FLAG_CIRCLE_SHORT 13`

**18.85.1.3 FLAG\_COMBINE** `#define FLAG_COMBINE 35`

**18.85.1.4 FLAG\_CROSS\_STITCH** `#define FLAG_CROSS_STITCH 36`

**18.85.1.5 FLAG\_ELLIPSE** `#define FLAG_ELLIPSE 14`

**18.85.1.6 FLAG\_ELLIPSE\_SHORT** `#define FLAG_ELLIPSE_SHORT 15`

**18.85.1.7 FLAG\_FILL** `#define FLAG_FILL 32`

**18.85.1.8 FLAG\_FILL\_SHORT** `#define FLAG_FILL_SHORT 33`

**18.85.1.9 FLAG\_FORMATS** `#define FLAG_FORMATS 4`

**18.85.1.10 FLAG\_FORMATS\_SHORT** `#define FLAG_FORMATS_SHORT 5`

**18.85.1.11 FLAG\_FULL\_TEST\_SUITE** `#define FLAG_FULL_TEST_SUITE 29`

**18.85.1.12 FLAG\_HELP** `#define FLAG_HELP 2`

**18.85.1.13 FLAG\_HELP\_SHORT** `#define FLAG_HELP_SHORT 3`

**18.85.1.14 FLAG\_HILBERT\_CURVE** `#define FLAG_HILBERT_CURVE 30`

**18.85.1.15 FLAG\_LINE** `#define FLAG_LINE 16`

**18.85.1.16 FLAG\_LINE\_SHORT** `#define FLAG_LINE_SHORT 17`

**18.85.1.17 FLAG\_POLYGON** `#define FLAG_POLYGON 18`

**18.85.1.18 FLAG\_POLYGON\_SHORT** `#define FLAG_POLYGON_SHORT 19`

**18.85.1.19 FLAG\_POLYLINE** `#define FLAG_POLYLINE 20`

**18.85.1.20 FLAG\_POLYLINE\_SHORT** `#define FLAG_POLYLINE_SHORT 21`

**18.85.1.21 FLAG\_QUIET** `#define FLAG_QUIET 6`

**18.85.1.22 FLAG\_QUIET\_SHORT** `#define FLAG_QUIET_SHORT 7`

**18.85.1.23 FLAG\_RENDER** `#define FLAG_RENDER 22`

**18.85.1.24 FLAG\_RENDER\_SHORT** `#define FLAG_RENDER_SHORT 23`

**18.85.1.25 FLAG\_SATIN** `#define FLAG_SATIN 24`

**18.85.1.26 FLAG\_SATIN\_SHORT** `#define FLAG_SATIN_SHORT 25`

**18.85.1.27 FLAG\_SIERPINSKI\_TRIANGLE** `#define FLAG_SIERPINSKI_TRIANGLE 31`

**18.85.1.28 FLAG\_SIMULATE** `#define FLAG_SIMULATE 34`

**18.85.1.29 FLAG\_STITCH** `#define FLAG_STITCH 26`

**18.85.1.30 FLAG\_STITCH\_SHORT** `#define FLAG_STITCH_SHORT 27`

**18.85.1.31 FLAG\_TEST** `#define FLAG_TEST 28`

**18.85.1.32 FLAG\_TO** `#define FLAG_TO 0`

**18.85.1.33 FLAG\_TO\_SHORT** `#define FLAG_TO_SHORT 1`

**18.85.1.34 FLAG\_VERBOSE** `#define FLAG_VERBOSE 8`

**18.85.1.35 FLAG\_VERBOSE\_SHORT** `#define FLAG_VERBOSE_SHORT 9`

**18.85.1.36 FLAG\_VERSION** `#define FLAG_VERSION 10`

**18.85.1.37 FLAG\_VERSION\_SHORT** `#define FLAG_VERSION_SHORT 11`

**18.85.1.38 NUM\_FLAGS** `#define NUM_FLAGS 37`

## 18.85.2 Function Documentation

**18.85.2.1 bcf\_diffat\_create()** `bcf_file_difat * bcf_difat_create (`  
    `FILE * file,`  
    `unsigned int fatSectors,`  
    `const unsigned int sectorSize )`  
*file fatSectors sectorSize*

Returns

`bcf_file_difat*`

**18.85.2.2 bcf\_directory\_free()** `void bcf_directory_free (`  
    `bcf_directory ** dir )`  
*dir*

**18.85.2.3 bcf\_file\_free()** `void bcf_file_free (`  
    `bcf_file * bcfFile )`  
*bcfFile*

**18.85.2.4 bcfFile\_read()** `int bcfFile_read (`  
    `FILE * file,`  
    `bcf_file * bcfFile )`  
*file bcfFile*

Returns

`int`

**18.85.2.5 bcfFileFat\_create()** `bcf_file_fat * bcfFileFat_create (`  
    `const unsigned int sectorSize )`  
*sectorSize*

Returns

`bcf_file_fat*`

**18.85.2.6 bcfFileHeader\_read()** `bcf_file_header bcfFileHeader_read (`  
    `FILE * file )`  
*file*

Returns

`bcf_file_header`

**18.85.2.7 binaryReadString()** void binaryReadString (   
FILE \* *file*,  
char \* *buffer*,  
int *maxLength* )  
*file buffer maxLength*

**18.85.2.8 binaryReadUnicodeString()** void binaryReadUnicodeString (   
FILE \* *file*,  
char \* *buffer*,  
const int *stringLength* )  
*file buffer stringLength*

**18.85.2.9 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.

**18.85.2.10 CompoundFileDirectory()** [bcf\\_directory](#) \* CompoundFileDirectory (   
const unsigned int *maxNumberOfDirectoryEntries* )  
*maxNumberOfDirectoryEntries*

**Returns**

[bcf\\_directory](#)\*

**18.85.2.11 CompoundFileDirectoryEntry()** [bcf\\_directory\\_entry](#) \* CompoundFileDirectoryEntry (   
FILE \* *file* )  
*file*

**Returns**

[bcf\\_directory\\_entry](#)\*

**18.85.2.12 copy\_trim()** char \* copy\_trim (   
char const \* *s* )

*s*

**Returns**

char\*

**Todo** decription

**18.85.2.13 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\*

**18.85.2.14 emb\_readline()** int emb\_readline (   
     FILE \* file,   
     char \* line,   
     int maxLength )

*file line maxLength*

Returns

int

**18.85.2.15 embArc\_print()** void embArc\_print (   
     EmbArc arc )

*arc*

**Todo** move to [arc.c](#)

**18.85.2.16 embColor\_distance()** int embColor\_distance (   
     EmbColor a,   
     EmbColor b )

*a b*

Returns

int

**18.85.2.17 embColor\_read()** void embColor\_read (   
     FILE \* f,   
     EmbColor \* c,   
     int toRead )

*f c toRead*

**18.85.2.18 embColor\_write()** void embColor\_write (   
     FILE \* f,   
     EmbColor c,   
     int toWrite )

*f c toWrite*

**18.85.2.19 embSatinOutline\_generateSatinOutline()** void embSatinOutline\_generateSatinOutline (   
     EmbArray \* lines,   
     EmbReal thickness,   
     EmbSatinOutline \* result )

*lines thickness result*

**18.85.2.20 embSatinOutline\_renderStitches()** `EmbArray * embSatinOutline_renderStitches (`  
`EmbSatinOutline * result,`  
`EmbReal density )`

*result density*

Returns

EmbArray\*

**18.85.2.21 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.

**18.85.2.22 embThread\_findNearestThread()** `int embThread_findNearestThread (`  
`EmbColor color,`  
`EmbThread * thread_list,`  
`int n_threads )`

*color thread\_list n\_threads*

Returns

int

**18.85.2.23 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.

**18.85.2.24 embTime\_initNow()** `void embTime_initNow (`  
`EmbTime * t )`

*t*

**18.85.2.25 embTime\_time()** `EmbTime embTime_time (`  
`EmbTime * t )`

*t*

Returns

EmbTime

**18.85.2.26 embVector\_print()** void embVector\_print (   
     EmbVector v,   
     char \* label )

*v label*

move to [vector.c](#)

**18.85.2.27 entriesInDifatSector()** unsigned int entriesInDifatSector (   
     bcf\_file\_difat \* fat )

*fat*

Returns

unsigned int

**18.85.2.28 get\_trim\_bounds()** void get\_trim\_bounds (   
     char const \* s,   
     char const \*\* firstWord,   
     char const \*\* trailingSpace )

Get the trim bounds object.

*s firstWord trailingSpace*

**18.85.2.29 GetFile()** FILE \* GetFile (   
     bcf\_file \* bcfFile,   
     FILE \* file,   
     char \* fileToFind )

Get the File object.

*bcfFile file fileToFind*

Returns

FILE\*

**18.85.2.30 haveExtraDIFATSectors()** int haveExtraDIFATSectors (   
     bcf\_file \* file )

*file*

Returns

int

**18.85.2.31 loadFatFromSector()** void loadFatFromSector (   
     bcf\_file\_fat \* fat,   
     FILE \* file )

*fat file*

**18.85.2.32 parseDIFATSectors()** void parseDIFATSectors (   
     FILE \* file,   
     bcf\_file \* bcfFile )

*file bcfFile*

**18.85.2.33 parseDirectoryEntryName()** void parseDirectoryEntryName (   
     FILE \* file,   
     bcf\_directory\_entry \* dir )

*file dir*

**18.85.2.34 parseTime()** `EmbTime parseTime (`  
`FILE * file )`

*file*

Returns

EmbTime

**18.85.2.35 readFullSector()** `unsigned int readFullSector (`  
`FILE * file,`  
`bcf_file_difat * bcfFile,`  
`unsigned int * difatEntriesToRead )`

*file bcfFile difatEntriesToRead*

Returns

unsigned int

**18.85.2.36 readNextSector()** `void readNextSector (`  
`FILE * file,`  
`bcf_directory * dir )`

*file dir*

**18.85.2.37 sectorSize()** `unsigned int sectorSize (`  
`bcf_file * bcfFile )`

*bcfFile*

Returns

unsigned int

**18.85.2.38 seekToSector()** `int seekToSector (`  
`bcf_file * bcfFile,`  
`FILE * file,`  
`const unsigned int sector )`

*bcfFile file sector*

Returns

int

**18.85.2.39 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.



**18.85.2.40 write\_24bit()** void write\_24bit (   
     FILE \* *file*,   
     int *x* )   
*file x*

### 18.85.3 Variable Documentation

**18.85.3.1 black\_thread** [EmbThread](#) black\_thread = { { 0, 0, 0 }, "Black", "Black" }

**18.85.3.2 difatEntriesInHeader** const unsigned int difatEntriesInHeader = 109

**18.85.3.3 emb\_error** int emb\_error = 0   
 Error code storage for optional control flow blocking.

**18.85.3.4 emb\_verbose** int emb\_verbose = 0   
 Verbosity level.

**18.85.3.5 embConstantPi** const [EmbReal](#) embConstantPi = 3.1415926535

**18.85.3.6 sizeOfChainingEntryAtEndOfDifatSector** const unsigned int sizeOfChainingEntryAtEndOf↔   
 DifatSector = 4

**18.85.3.7 sizeOfDifatEntry** const unsigned int sizeOfDifatEntry = 4

**18.85.3.8 sizeOfDirectoryEntry** const unsigned int sizeOfDirectoryEntry = 128

**18.85.3.9 sizeOfFatEntry** const unsigned int sizeOfFatEntry = sizeof(unsigned int)

**18.85.3.10 WHITESPACE** char const WHITESPACE[] = " \t\n\r"

## 18.86 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`)

### 18.86.1 Detailed Description

The file is for the management of the main struct: `EmbPattern`.

### 18.86.2 Function Documentation

**18.86.2.1 `convert()`** `int convert (`  
`const char * inf,`  
`const char * outf )`

**18.86.2.2 `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.

**18.86.2.3 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.

**18.86.2.4 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.

**18.86.2.5 embPattern\_addPathAbs()** void embPattern\_addPathAbs (  
 EmbPattern \* p,  
 EmbPath obj )

**18.86.2.6 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.

**18.86.2.7 embPattern\_addPolygonAbs()** void embPattern\_addPolygonAbs (  
 EmbPattern \* p,  
 EmbPolygon obj )

**18.86.2.8 embPattern\_addPolylineObjectAbs()** void embPattern\_addPolylineObjectAbs (  
 EmbPattern \* p,  
 EmbPolyline obj )

**18.86.2.9 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.

**18.86.2.10 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.

**18.86.2.11 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.

**18.86.2.12 embPattern\_addThread()** int embPattern\_addThread (

```

 EmbPattern * pattern,
 EmbThread thread)

```

*pattern thread*

Returns

int

**18.86.2.13 embPattern\_calcBoundingBox()** EmbRect embPattern\_calcBoundingBox (

```

 EmbPattern * p)

```

Returns an EmbRect that encapsulates all stitches and objects in the pattern (*p*).

**18.86.2.14 embPattern\_center()** void embPattern\_center (

```

 EmbPattern * p)

```

Center the pattern *p*.

**18.86.2.15 embPattern\_changeColor()** void embPattern\_changeColor (

```

 EmbPattern * p,
 int index)

```

Change the currentColorIndex of pattern *p* to *index*.

**18.86.2.16 embPattern\_color\_count()** int embPattern\_color\_count (

```

 EmbPattern * pattern,
 EmbColor startColor)

```

**18.86.2.17 embPattern\_combineJumpStitches()** void embPattern\_combineJumpStitches (

```

 EmbPattern * p)

```

*p*

**18.86.2.18 embPattern\_copyPolylinesToStitch\_list()** void embPattern\_copyPolylinesToStitch\_list (

```

 EmbPattern * p)

```

Copies all of the EmbPolylineObjectList data to Embstitch\_list data for pattern (*p*).

**18.86.2.19 embPattern\_copystitch\_listToPolylines()** void embPattern\_copystitch\_listToPolylines (

```

 EmbPattern * p)

```

Copies all of the Embstitch\_list data to EmbPolylineObjectList data for pattern (*p*).

**18.86.2.20 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.

**18.86.2.21 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\*

**18.86.2.22 embPattern\_designDetails()** void embPattern\_designDetails (   
     EmbPattern \* pattern )

**18.86.2.23 embPattern\_end()** void embPattern\_end (   
     EmbPattern \* p )

**18.86.2.24 embPattern\_fixColorCount()** void embPattern\_fixColorCount (   
     EmbPattern \* p )   
*p*

**18.86.2.25 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.

**18.86.2.26 embPattern\_flipHorizontal()** void embPattern\_flipHorizontal (   
     EmbPattern \* p )

Flips the entire pattern (*p*) horizontally about the y-axis.

**18.86.2.27 embPattern\_flipVertical()** void embPattern\_flipVertical (   
     EmbPattern \* p )

Flips the entire pattern (*p*) vertically about the x-axis.

**18.86.2.28 embPattern\_free()** void embPattern\_free (   
     EmbPattern \* p )

Frees all memory allocated in the pattern (*p*).

**18.86.2.29 embPattern\_hideStitchesOverLength()** void embPattern\_hideStitchesOverLength (   
     EmbPattern \* p,   
     int length )

*p length*

**18.86.2.30 embPattern\_jumpStitches()** int embPattern\_jumpStitches (   
     EmbPattern \* pattern )

**18.86.2.31 embPattern\_lengthHistogram()** void embPattern\_lengthHistogram (   
     EmbPattern \* pattern,   
     int \* bin,   
     int NUMBINS )

**18.86.2.32 embPattern\_loadExternalColorFile()** void embPattern\_loadExternalColorFile (   
     EmbPattern \* p,   
     const char \* fileName )

TODO: Description needed.

**18.86.2.33 embPattern\_maximumStitchLength()** float embPattern\_maximumStitchLength (   
     EmbPattern \* pattern )

**18.86.2.34** `embPattern_minimumStitchLength()` `float embPattern_minimumStitchLength (`  
`EmbPattern * pattern )`

**18.86.2.35** `embPattern_movePolylinesTostitch_list()` `void embPattern_movePolylinesTostitch_list (`  
`EmbPattern * p )`

Moves all of the EmbPolylineObjectList data to Embstitch\_list data for pattern (*p*).

**18.86.2.36** `embPattern_movestitch_listToPolylines()` `void embPattern_movestitch_listToPolylines (`  
`EmbPattern * p )`

Moves all of the Embstitch\_list data to EmbPolylineObjectList data for pattern (*p*).

**18.86.2.37** `embPattern_realStitches()` `int embPattern_realStitches (`  
`EmbPattern * pattern )`

**18.86.2.38** `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.

**18.86.2.39** `embPattern_totalStitchLength()` `float embPattern_totalStitchLength (`  
`EmbPattern * pattern )`

*pattern*

Returns

float

**18.86.2.40** `embPattern_trimStitches()` `int embPattern_trimStitches (`  
`EmbPattern * pattern )`

## 18.87 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` []

### 18.87.1 Function Documentation

**18.87.1.1 threadColor()** `int threadColor (`  
     `const char * name,`  
     `int brand )`

**18.87.1.2 threadColorName()** `const char * threadColorName (`  
     `unsigned int color,`  
     `int brand )`

**18.87.1.3 threadColorNum()** `int threadColorNum (`  
     `unsigned int color,`  
     `int brand )`

### 18.87.2 Variable Documentation

**18.87.2.1 \_dxfColorTable** `const unsigned char _dxfColorTable[][3] = {{ 0, 0, 0 }}`

**18.87.2.2 brand\_codes** `thread_color* brand_codes[]`

**18.87.2.3 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"
}
```

**18.87.2.4 husThreads** `const EmbThread husThreads[] = {{{ 0, 0, 0 }, "END", "END"}}`

**18.87.2.5 jefThreads** `const EmbThread jefThreads[] = {{{ 0, 0, 0 }, "END", "END"}}`

**18.87.2.6 pcmThreads** `const EmbThread pcmThreads[] = {{{ 0, 0, 0 }, "END", "END"}}`

**18.87.2.7 pecThreadCount** `const int pecThreadCount = 65`

**18.87.2.8 pecThreads** `const EmbThread pecThreads[] = {{{ 0, 0, 0 }, "END", "END"}}`

**18.87.2.9 shvThreadCount** `const int shvThreadCount = 42`

**18.87.2.10 shvThreads** `const EmbThread shvThreads[] = {{{ 0, 0, 0 }, "END", "END"}}`

## 18.88 privacy\_policy.md File Reference

## 18.89 src/cmdprompt.cpp File Reference

```
#include "embroidermodder.h"
```

### 18.89.1 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/>

---

## 18.90 src/em2\_dev\_script.py File Reference

### Namespaces

- namespace `em2_dev_script`

### Variables

- string `header`
- dictionary `d = {}`
- `s = f.read()`

## 18.91 src/embdetails-dialog.cpp File Reference

```
#include "embroidermodder.h"
```

## 18.92 src/embroidermodder.cpp File Reference

```
#include "embroidermodder.h"
```

### Functions

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

### Variables

- static const char \* `_appVer_` = "v2.0.0-alpha3"
- static bool `exitApp` = false
- const char \* `usage_msg`



### 18.92.1 Function Documentation

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

qMain

## Parameters

|             |  |
|-------------|--|
| <i>argc</i> |  |
| <i>argv</i> |  |

## Returns

### 18.92.2 Variable Documentation

```
18.92.2.1 _appVer_ const char* _appVer_ = "v2.0.0-alpha3" [static]
```

### 18.92.2.2 exitApp

```
bool exitApp = false [static]
```

### 18.92.2.3 usage\_msg const char\* usage\_msg

**Initial value:**

[illegible]

### 18.93 src/embroidermodder.h File Reference

```
#include <cstdio>
#include <cmath>
#include <ctime>
#include <cstdint>
#include <cstdint>
#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\\_INFITELINE](#) = 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)  
[add\\_polyline](#)
- [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](#)
- QPainterPath [add\\_to\\_path](#) (QPainterPath path, [EmbVector](#) scale, [String](#) s)
- [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](#), [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](#)
- [MainWindow](#) \* [\\_mainWin](#)
- [CmdPrompt](#) \* [prompt](#)
- [PropertyEditor](#) \* [dockPropEdit](#)
- [UndoEditor](#) \* [dockUndoEdit](#)
- [StatusBar](#) \* [statusbar](#)

### 18.93.1 Detailed Description

The only header for the GUI part: a good overview of this source code.

**18.93.1.1 Stuff for 2.0 alpha1** WIP - Statistics from 1.0, needs histogram WIP - Saving DST/PES/JEF (varga)  
WIP - Saving CSV/SVG (rt) + CSV read/write UNKNOWN interpreted as COLOR bug

#### 18.93.1.2 Stuff for 2.0 alpha2

**Todo** Notify user of data loss if not saving to an object format.

Import Raster Image

SNAP/ORTHO/POLAR

Layer Manager + LayerSwitcher DockWidget

Reading DXF

#### 18.93.1.3 Stuff for 2.0 alpha3

**Todo** Writing DXF DONE - Up and Down keys cycle thru commands in the command prompt

Amount of Thread & Machine Time Estimation (also allow customizable times for setup, color changes, manually trimming jump threads, etc...that way a realistic total time can be estimated)

Otto Theme Icons - whatsthis icon doesn't scale well, needs redone

embroidermodder2.ico 16 x 16 looks horrible

#### 18.93.1.4 Stuff for 2.0 alpha4

 WIP - CAD Command: Arc (rt)

**Todo** Load/Save Menu/Toolbars configurations into settings.ini

automate changelog and write to a javascript file for the docs: `git log --pretty=tformat:' s'`

### 18.93.1.5 Stuff for 2.0 beta1

**Todo** Custom Filter Bug - doesn't save changes in some cases

Cannot open file with # in name when opening multiple files (works fine when opening the single file)

Closing Settings Dialog with the X in the window saves settings rather than discards them WIP - Advanced Printing

Filling Algorithms (varga)

Otto Theme Icons - beta (rt) - Units, Render, Selectors

### 18.93.1.6 Stuff for 2.0 rc1

**Todo** QDoc Comments

Review KDE4 Thumbnailer

Documentation for libembroidery & formats

HTML Help files

Update language translations

CAD Command review: line

CAD Command review: circle

CAD Command review: rectangle

CAD Command review: polygon

CAD Command review: polyline

CAD Command review: point

CAD Command review: ellipse

CAD Command review: arc

CAD Command review: distance

CAD Command review: locatepoint

CAD Command review: move

CAD Command review: rgb

CAD Command review: rotate

CAD Command review: scale

CAD Command review: singlelinetext

CAD Command review: star

Clean up all compiler warning messages, right now theres plenty :P

### 18.93.1.7 Stuff for 2.0 release

**Todo** tar.gz archive

zip archive

Debian Package (rt)

NSIS Installer (rt)

Mac Bundle?

press release

Stuff for 2.x/Ideas:

**Todo** libembroidery.mk for MXE project (refer to qt submodule packages for qmake based building. Also refer to plibc.mk for example of how write an update macro for github.)

libembroidery safeguard for all writers - check if the last stitch is an END stitch. If not, add an end stitch in the writer and modify the header data if necessary.

Cut/Copy - Allow Post-selection

CAD Command: Array

CAD Command: Offset

CAD Command: Extend

CAD Command: Trim

CAD Command: BreakAtPoint

CAD Command: Break2Points

CAD Command: Fillet

CAD Command: Chamfer

CAD Command: Split

CAD Command: Area

CAD Command: Time

CAD Command: PickAdd

CAD Command: Product

CAD Command: Program

CAD Command: ZoomFactor

CAD Command: GripHot

CAD Command: GripColor & GripCool

CAD Command: GripSize

CAD Command: Highlight

CAD Command: Units

CAD Command: Grid

CAD Command: Find

CAD Command: Divide

CAD Command: ZoomWindow (Move out of [view.cpp](#))

Command: Web (Generates Spiderweb patterns)

Command: Guilloche (Generates Guilloche patterns)

Command: Celtic Knots

Command: Knotted Wreath

Lego Mindstorms NXT/EV3 ports and/or commands.

native function that flashes the command prompt to get users attention when using the prompt is required for a command.

libembroidery-composer like app that combines multiple files into one.

Settings Dialog, it would be nice to have it notify you when switching tabs that a setting has been changed.

Adding an Apply button is what would make sense for this to happen.

Keyboard Zooming/Panning

G-Code format?

3D Raised Embroidery

Gradient Filling Algorithms

Stitching Simulation

RPM packages?

Reports?

Record and Playback Commands

Settings option for reversing zoom scrolling direction

Qt GUI for libembroidery-convert

EPS format? Look at using Ghostscript as an optional add-on to libembroidery...

optional compile option for including LGPL/GPL libs etc... with warning to user about license requirements.

Realistic Visualization - Bump Mapping/OpenGL/Gradients?

Stippling Fill

User Designed Custom Fill

Honeycomb Fill

Hilbert Curve Fill

Sierpinski Triangle fill

Circle Grid Fill

Spiral Fill

Offset Fill

Brick Fill

Trim jumps over a certain length.

FAQ about setting high number of jumps for more controlled trimming.

Minimum stitch length option. (Many machines also have this option too)

Add 'Design Details' functionality to libembroidery-convert

Add 'Batch convert many to one format' functionality to libembroidery-convert

EmbroideryFLOSS - Color picker that displays catalog numbers and names.

emscripten/javascript port of libembroidery

#### **18.93.1.8 Stuff for Arduino:**

**Todo** Fix emb-outline files

Fix thread-color files

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

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

Smoothieboard experiments

#### **18.93.1.9 libembroidery-tests**

**Todo** looping test that reads 10 times while running valgrind. See [embPattern\\_loadExternalColorFile\(\)](#) Arduino leak note for more info.

#### **18.93.2 Macro Definition Documentation**

**18.93.2.1 BOOL\_TYPE** `#define BOOL_TYPE 4`

**18.93.2.2 FUNCTION\_TYPE** `#define FUNCTION_TYPE 5`

**18.93.2.3 INT\_TYPE** `#define INT_TYPE 3`

**18.93.2.4 REAL\_TYPE** `#define REAL_TYPE 2`



**18.93.2.5** **STRING\_LIST\_TYPE** `#define STRING_LIST_TYPE 1`

**18.93.2.6** **STRING\_TYPE** `#define STRING_TYPE 0`

**18.93.2.7** **UNKNOWN\_TYPE** `#define UNKNOWN_TYPE 7`

**18.93.2.8** **VECTOR\_TYPE** `#define VECTOR_TYPE 6`

### 18.93.3 Typedef Documentation

**18.93.3.1** **Command** `typedef String(* Command) (String)`

**18.93.3.2** **Dictionary** `typedef std::unordered_map<String, Node> Dictionary`

**18.93.3.3** **Node** `typedef struct Node_ Node`

**18.93.3.4** **NodeList** `typedef std::vector<Node> NodeList`

**18.93.3.5** **String** `typedef std::string String`

**18.93.3.6** **StringList** `typedef std::vector<String> StringList`

### 18.93.4 Enumeration Type Documentation

**18.93.4.1** **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<br><b>Todo</b> Use color chart in formats/format-dxf.h for this |
| OBJ_LWT    |                                                                                         |
| OBJ_RUBBER |                                                                                         |

#### 18.93.4.2 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    |  |
| 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      |  |

#### 18.93.5 Function Documentation

**18.93.5.1 activeScene()** QGraphicsScene \* activeScene ( )  
MainWindow::activeScene.

Returns

**18.93.5.2 activeView()** View \* activeView ( )  
void )

activeView

Returns

**18.93.5.3 actuator()** `String actuator (`  
`String line )`

MainWindow::actuator.

Parameters

|                |  |
|----------------|--|
| <i>command</i> |  |
|----------------|--|

**18.93.5.4 RUN COMMAND** `QAction* act = qobject_cast<QAction*>(sender()); if (act) { prompt->endCommand(); prompt->setCurrentText(act->objectName()); prompt->processInput(); }`

**18.93.5.5 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);`

**18.93.5.6 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.toUpperCase() + ")", fileName); }`

**18.93.5.7 add\_polyline()** `void add_polyline (`  
`QPainterPath p,`  
`String rubberMode )`

add\_polyline

Parameters

|                   |  |
|-------------------|--|
| <i>p</i>          |  |
| <i>rubberMode</i> |  |

**18.93.5.8 add\_to\_path()** `QPainterPath add_to_path (`  
`QPainterPath path,`  
`EmbVector scale,`  
`String command )`

**18.93.5.9 construct\_command()** `String construct_command (`  
`String command,`  
`const char * fmt,`  
`... )`

construct\_command

Parameters

|                |  |
|----------------|--|
| <i>command</i> |  |
| <i>fmt</i>     |  |

## Returns

**18.93.5.10 contains()** `bool contains (`  
    `StringList list,`  
    `String entry )`

**18.93.5.11 convert\_args\_to\_type()** `String convert_args_to_type (`  
    `String label,`  
    `StringList args,`  
    `const char * args_template,`  
    `NodeList a )`

**18.93.5.12 create\_menu()** `void create_menu (`  
    `std::string menu,`  
    `StringList def,`  
    `bool topLevel )`

`create_menu`

## Parameters

|                 |  |
|-----------------|--|
| <i>menu</i>     |  |
| <i>def</i>      |  |
| <i>topLevel</i> |  |

**18.93.5.13 debug\_message()** `void debug_message (`  
    `std::string msg )`

`debug_message`

## Parameters

|            |  |
|------------|--|
| <i>msg</i> |  |
|------------|--|

**18.93.5.14 degrees\_\_()** `EmbReal degrees__ (`  
    `EmbReal radian )`

`degrees__`

## Parameters

|               |  |
|---------------|--|
| <i>radian</i> |  |
|---------------|--|

## Returns

**18.93.5.15 fileExtension()** `QString fileExtension (`  
    `String fileName )`

MdiWindow::fileExtension.

#### Parameters

|                 |  |
|-----------------|--|
| <i>fileName</i> |  |
|-----------------|--|

#### Returns

**18.93.5.16** **get\_bool()** bool get\_bool (  
Dictionary *d*,  
String *key* )

**18.93.5.17** **get\_int()** int32\_t get\_int (  
Dictionary *d*,  
String *key* )

**18.93.5.18** **get\_qstr()** QString get\_qstr (  
Dictionary *d*,  
String *key* )

**18.93.5.19** **get\_real()** EmbReal get\_real (  
Dictionary *d*,  
String *key* )

**18.93.5.20** **get\_str()** String get\_str (  
Dictionary *d*,  
String *key* )

**18.93.5.21** **get\_str\_list()** StringList get\_str\_list (  
Dictionary *d*,  
String *key* )

**18.93.5.22** **get\_uint()** uint32\_t get\_uint (  
Dictionary *d*,  
String *key* )

**18.93.5.23** **make\_checkbox()** QCheckBox \* make\_checkbox (  
QGroupBox \* *gb*,  
String *dictionary*,  
const char \* *label*,  
const char \* *icon*,  
String *key* )

**18.93.5.24 make\_spinbox()** `QDoubleSpinBox * make_spinbox (`  
    `QGroupBox * gb,`  
    `String dictionary,`  
    `QString object_name,`  
    `EmbReal single_step,`  
    `EmbReal lower,`  
    `EmbReal upper,`  
    `String key )`

**18.93.5.25 make\_ui\_element()** `void make_ui_element (`  
    `String description )`

**18.93.5.26 node\_bool()** `Node node_bool (`  
    `bool value )`

set\_node

Parameters

|              |  |
|--------------|--|
| <i>node</i>  |  |
| <i>value</i> |  |

**18.93.5.27 node\_int()** `Node node_int (`  
    `int32_t value )`

create\_node

Parameters

|             |  |
|-------------|--|
| <i>mode</i> |  |
|-------------|--|

Returns

**18.93.5.28 node\_qstr()** `Node node_qstr (`  
    `QString value )`

set\_node

Parameters

|              |  |
|--------------|--|
| <i>node</i>  |  |
| <i>value</i> |  |

**18.93.5.29 node\_real()** `Node node_real (`  
    `EmbReal value )`

set\_node

Parameters

|             |  |
|-------------|--|
| <i>node</i> |  |
|-------------|--|

## Parameters

|              |  |
|--------------|--|
| <i>value</i> |  |
|--------------|--|

**18.93.5.30 node\_str()** `Node node_str (`  
    `String value )`

set\_node

## Parameters

|              |  |
|--------------|--|
| <i>node</i>  |  |
| <i>value</i> |  |

**18.93.5.31 node\_str\_list()** `Node node_str_list (`  
    `StringList value )`

set\_node

## Parameters

|              |  |
|--------------|--|
| <i>node</i>  |  |
| <i>value</i> |  |

**18.93.5.32 node\_uint()** `Node node_uint (`  
    `uint32_t value )`

create\_node

## Parameters

|             |  |
|-------------|--|
| <i>mode</i> |  |
|-------------|--|

## Returns

**18.93.5.33 operator\*()** `EmbVector operator* (`  
    `EmbVector v,`  
    `EmbReal s )`

operator \*

## Parameters

|          |  |
|----------|--|
| <i>v</i> |  |
| <i>s</i> |  |

## Returns

**18.93.5.34 operator+()** `EmbVector` operator+ (   
     `EmbVector` *a*,   
     `EmbVector` *b* )

operator + Wrapper for `embVector_add` to use the syntax *a + b*.

**18.93.5.35 operator-()** `EmbVector` operator- (   
     `EmbVector` *a*,   
     `EmbVector` *b* )

operator - Wrapper for `embVector_subtract` to use the syntax *a - b*.

**18.93.5.36 radians\_\_()** `EmbReal` radians\_\_ (   
     `EmbReal` *degrees* )

radians\_\_

Parameters

|                |  |
|----------------|--|
| <i>degrees</i> |  |
|----------------|--|

Returns

**18.93.5.37 read\_configuration()** `int` read\_configuration (   
     const char \* *file* )

**18.93.5.38 read\_settings()** `void` read\_settings (   
     void )

read\_settings

This file needs to be read from the users home directory to ensure it is writable.

**18.93.5.39 read\_string\_setting()** `String` read\_string\_setting (   
     toml\_table\_t \* *table*,   
     const char \* *key* )

**18.93.5.40 rotate\_vector()** `EmbVector` rotate\_vector (   
     `EmbVector` *v*,   
     `EmbReal` *alpha* )

Returns

**18.93.5.41 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

```

**18.93.5.42 run\_script\_file()** `String run_script_file (`  
     `String fname )`

MainWindow::run\_script\_file.

#### Parameters

|              |                                |
|--------------|--------------------------------|
| <i>fname</i> | The path of the script to run. |
|--------------|--------------------------------|

**18.93.5.43 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.

**18.93.5.44 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.

**18.93.5.45 to\_EmbVector()** `EmbVector to_EmbVector (`  
`QPointF a )`

**18.93.5.46 to\_qlist()** `QList< QGraphicsItem * > to_qlist (`  
`std::vector< QGraphicsItem * > list )`

`to_qlist`

Parameters

|             |  |
|-------------|--|
| <i>list</i> |  |
|-------------|--|

Returns

**18.93.5.47 to\_QPointF()** `QPointF to_QPointF (`  
`EmbVector a )`

**18.93.5.48 to\_string\_vector()** `StringList to_string_vector (`  
`QStringList list )`

`to_string_vector`

Parameters

|             |  |
|-------------|--|
| <i>list</i> |  |
|-------------|--|

Returns

**18.93.5.49 to\_vector()** `std::vector< QGraphicsItem * > to_vector (`  
`QList< QGraphicsItem * > list )`

`to_vector`

Parameters

|             |  |
|-------------|--|
| <i>list</i> |  |
|-------------|--|

Returns

**18.93.5.50 tokenize()** `StringList tokenize (`  
`String str,`  
`const char delim )`

`tokenize`

Parameters

|              |  |
|--------------|--|
| <i>str</i>   |  |
| <i>delim</i> |  |

## Returns

**18.93.5.51 translate\_str()** `QString translate_str (`  
`const char * str )`

**18.93.5.52 validFileFormat()** `bool validFileFormat (`  
`String fileName )`  
 MainWindow::validFileFormat.

## Parameters

|                 |  |
|-----------------|--|
| <i>fileName</i> |  |
|-----------------|--|

## Returns

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

**18.93.5.53 write\_settings()** `void write_settings (`  
`void )`  
 MainWindow::writeSettings.  
 This file needs to be read from the users home directory to ensure it is writable

**18.93.6 Variable Documentation**

**18.93.6.1 \_mainWin** `MainWindow* _mainWin [extern]`

**18.93.6.2 actionHash** `std::unordered_map<String, QAction*> actionHash [extern]`

**18.93.6.3 checkBoxes** `std::unordered_map<String, QCheckBox *> checkBoxes [extern]`

**18.93.6.4 comboBoxes** `std::unordered_map<String, QComboBox *> comboBoxes [extern]`

**18.93.6.5 config** `Dictionary config`

**18.93.6.6 config\_tables** `std::unordered_map<String, Dictionary> config_tables [extern]`

**18.93.6.7 dialog** `Dictionary dialog`

**18.93.6.8 dockPropEdit** `PropertyEditor* dockPropEdit [extern]`

**18.93.6.9 dockUndoEdit** `UndoEditor*` dockUndoEdit [extern]

**18.93.6.10 doubleSpinBoxes** `std::unordered_map<String, QDoubleSpinBox *>` doubleSpinBoxes [extern]

**18.93.6.11 emb\_constant\_pi** `const EmbReal` emb\_constant\_pi = 3.14159265358979323846 [static]

**18.93.6.12 groupBoxes** `std::unordered_map<String, QGroupBox *>` groupBoxes [extern]

**18.93.6.13 labels** `std::unordered_map<String, QLabel *>` labels [extern]

**18.93.6.14 lineEdits** `std::unordered_map<String, QLineEdit *>` lineEdits [extern]

**18.93.6.15 mdiArea** `MdiArea*` mdiArea [extern]

**18.93.6.16 menuHash** `std::unordered_map<String, QMenu*>` menuHash [extern]

**18.93.6.17 prompt** `CmdPrompt*` prompt [extern]

**18.93.6.18 scripts** `std::unordered_map<String, StringList>` scripts [extern]

**18.93.6.19 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.

**18.93.6.20 spinBoxes** `std::unordered_map<String, QSpinBox *>` spinBoxes [extern]

**18.93.6.21 statusbar** `StatusBar*` statusbar [extern]

**18.93.6.22 subMenuHash** `std::unordered_map<String, QMenu*>` subMenuHash [extern]

**18.93.6.23 toolbarHash** `std::unordered_map<String, QToolBar*>` toolbarHash [extern]

**18.93.6.24 toolButtons** `std::unordered_map<String, QToolButton *>` toolButtons [extern]

## 18.94 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
173 #ifndef __EMBROIDERMODDER_UTILITY_H__
174 #define __EMBROIDERMODDER_UTILITY_H__
175
176 /*
177 * C/C++ Standard Libraries.
178 */
179 #include <cstdio>
180 #include <cmath>
181 #include <ctime>
182 #include <cinttypes>
183 #include <cstdarg>
184 #include <vector>
185 #include <unordered_map>
186 #include <string>
187 #include <filesystem>
188
189 /*
190 * Libraries included in "extern/".
191 */
192 #include "embroidery.h"
193 #include "toml.h"
194
195 /*
196 * Qt 6.0+ libraries.
197 */
198 #include <QAction>
199 #include <QApplication>
200
201 #include <QtPrintSupport>
202
203 #define STRING_TYPE 0
204 #define STRING_LIST_TYPE 1
205 #define REAL_TYPE 2
206 #define INT_TYPE 3
207 #define BOOL_TYPE 4
208 #define FUNCTION_TYPE 5
209 #define VECTOR_TYPE 6
210 #define UNKNOWN_TYPE 7
211
212 class ImageWidget;
213 class MdiArea;
214 class MdiWindow;
215 class View;
216 class StatusBar;
217 class CmdPrompt;
218 class PropertyEditor;
219 class UndoEditor;
220 class MainWindow;
221 class Geometry;
222
223 typedef std::string String;
224 typedef std::vector<String> StringList;
225
226 typedef struct Node_ {
227 String s;
228 EmbReal r;
229 int32_t i;
230 bool b;
231 StringList sl;
232 int type;
233 } Node;
234
235 typedef String (*Command) (String);
236 typedef std::vector<Node> NodeList;
237 typedef std::unordered_map<String, Node> Dictionary;
238
239 //Values
240 enum OBJ_TYPE_VALUES {
241 OBJ_TYPE_NULL = 0,

```

```

242 /*< NOTE: Allow this enum to evaluate false */
243 OBJ_TYPE_BASE = 100000,
244 /*< NOTE: Values >= 65536 ensure compatibility with qgraphicsitem_cast() */
245 OBJ_TYPE_ARC = 100001,
246 OBJ_TYPE_BLOCK = 100002,
247 /*< For the block type, that has to exist for SVG. */
248 OBJ_TYPE_CIRCLE = 100003,
249 OBJ_TYPE_DIMALIGNED = 100004,
250 /*< For the Aligned Dimension, that has to exist for DXF drawings. */
251 OBJ_TYPE_DIMANGULAR = 100005,
252 /*< For the Angular Dimension, that has to exist for DXF drawings. */
253 OBJ_TYPE_DIMARCLENGTH = 100006,
254 /*< For the Arc Length Dimension, that has to exist for DXF drawings. */
255 OBJ_TYPE_DIMDIAMETER = 100007,
256 OBJ_TYPE_DIMLEADER = 100008,
257 OBJ_TYPE_DIMLINEAR = 100009,
258 /*< For the Linear Dimension, that has to exist for DXF drawings. */
259 OBJ_TYPE_DIMORDINATE = 100010,
260 /*< For the Ordinate Dimension, that has to exist for DXF drawings. */
261 OBJ_TYPE_DIMRADIUS = 100011,
262 /*< For the Radial Dimension, that has to exist for DXF drawings. */
263 OBJ_TYPE_ELLIPSE = 100012,
264 OBJ_TYPE_ELLIPSEARC = 100013,
265 OBJ_TYPE_RUBBER = 100014,
266 OBJ_TYPE_GRID = 100015,
267 OBJ_TYPE_HATCH = 100016,
268 OBJ_TYPE_IMAGE = 100017,
269 OBJ_TYPE_INFINITELINE = 100018,
270 /*< For the Infinite Line object. Which should be removed from output as it exists
271 for drafting reasons. */
272 OBJ_TYPE_LINE = 100019,
273 OBJ_TYPE_PATH = 100020,
274 OBJ_TYPE_POINT = 100021,
275 OBJ_TYPE_POLYGON = 100022,
276 OBJ_TYPE_POLYLINE = 100023,
277 OBJ_TYPE_RAY = 100024,
278 /*< For the Ray object. */
279 OBJ_TYPE_RECTANGLE = 100025,
280 OBJ_TYPE_SLOT = 100026,
281 OBJ_TYPE_SPLINE = 100027,
282 OBJ_TYPE_TEXTMULTI = 100028,
283 OBJ_TYPE_TEXTSINGLE = 100029,
284 OBJ_TYPE_UNKNOWN = 100030
285 };
286
287 enum OBJ_KEYS {
288 OBJ_TYPE = 0,
289 /*< value type - int: See OBJ_TYPE_VALUES */
290 OBJ_NAME = 1,
291 /*< value type - str: See OBJ_NAME_VALUES */
292 OBJ_LAYER = 2,
293 /*< value type - str: "USER", "DEFINED", "STRINGS", etc... */
294 OBJ_COLOR = 3,
295 OBJ_LTYPE = 4,
296 /*< value type - int: See OBJ_LTYPE_VALUES */
297 OBJ_LWT = 5, //value type - int: 0-27
298 OBJ_RUBBER = 6 //value type - int: See OBJ_RUBBER_VALUES
299 };
300
301 static const EmbReal emb_constant_pi = 3.14159265358979323846;
302
303 /* Global variables
304 * -----
305 */
306 extern MdiArea* mdiArea;
307
308 extern Dictionary settings, dialog, config;
309 extern std::unordered_map<String, QStringList> scripts;
310 extern std::unordered_map<String, QGroupBox*> groupBoxes;
311 extern std::unordered_map<String, QCheckBox*> checkBoxes;
312 extern std::unordered_map<String, QSpinBox*> spinBoxes;
313 extern std::unordered_map<String, QDoubleSpinBox*> doubleSpinBoxes;
314 extern std::unordered_map<String, QLabel*> labels;
315 extern std::unordered_map<String, QComboBox*> comboBoxes;
316 extern std::unordered_map<String, QLineEdit*> lineEdits;
317 extern std::unordered_map<String, QToolButton*> toolButtons;
318 extern std::unordered_map<String, Dictionary> config_tables;
319 extern std::unordered_map<String, QAction*> actionHash;
320 extern std::unordered_map<String, QToolBar*> toolbarHash;
321 extern std::unordered_map<String, QMenu*> menuHash;
322 extern std::unordered_map<String, QMenu*> subMenuHash;
323
324 extern MainWindow* _mainWin;
325 extern CmdPrompt* prompt;
326 extern PropertyEditor* dockPropEdit;
327 extern UndoEditor* dockUndoEdit;
328 extern StatusBar* statusBar;

```

```

352
353 /* Functions in the global namespace
354 * -----
355 */
356 int read_configuration(const char *file);
357 void read_settings(void);
358 void write_settings(void);
359 EmbVector rotate_vector(EmbVector v, EmbReal alpha);
360
361 QString translate_str(const char *str);
362 bool contains(StringList, String);
363 bool validFileFormat(String fileName);
364 QString fileExtension(String fileName);
365
366 void add_polyline(QPainterPath p, String rubberMode);
367
368 String read_string_setting(toml_table_t *table, const char *key);
369 StringList tokenize(String str, const char delim);
370 String convert_args_to_type(String label, StringList args,
371 const char *args_template, NodeList a);
372
373 View *activeView(void);
374 QGraphicsScene* activeScene();
375
376 void debug_message(String msg);
377 void set_enabled(QObject *parent, const char *key, bool enabled);
378 void set_visibility(QObject *parent, const char *name, bool visibility);
379 QPainterPath add_to_path(QPainterPath path, EmbVector scale, String s);
380
381 String actuator(String line);
382 String run_script_file(String fname);
383 String run_script(StringList script);
384 String construct_command(String command, const char *fmt, ...);
385
386 void create_menu(String menu, StringList def, bool topLevel);
387
388 QPointF to_QPointF(EmbVector a);
389 EmbVector to_EmbVector(QPointF a);
390 EmbVector operator+(EmbVector a, EmbVector b);
391 EmbVector operator-(EmbVector a, EmbVector b);
392 EmbVector operator*(EmbVector v, EmbReal s);
393 EmbReal radians__(EmbReal degrees);
394 EmbReal degrees__(EmbReal radian);
395
396 std::vector<QGraphicsItem*> to_vector(QList<QGraphicsItem*> list);
397 QList<QGraphicsItem*> to_qlist(std::vector<QGraphicsItem*> list);
398
399 StringList to_string_vector(QStringList list);
400
401 /* Interface creation functions.
402 */
403 void make_ui_element(String description);
404 QDoubleSpinBox *make_spinbox(QGroupBox *gb, String d,
405 QString object_name, EmbReal single_step, EmbReal lower, EmbReal upper, String key);
406 QCheckBox *make_checkbox(QGroupBox *gb, String d,
407 const char *label, const char *icon, String key);
408
409 /* Dictionary management functions.
410 */
411 Node node_bool(bool value);
412 Node node_int(int32_t value);
413 Node node_uint(uint32_t value);
414 Node node_real(EmbReal value);
415 Node node_str(String value);
416 Node node_qstr(QString value);
417 Node node_str_list(StringList value);
418
419 bool get_bool(Dictionary d, String key);
420 int32_t get_int(Dictionary d, String key);
421 uint32_t get_uint(Dictionary d, String key);
422 EmbReal get_real(Dictionary d, String key);
423 String get_str(Dictionary d, String key);
424 QString get_qstr(Dictionary d, String key);
425 StringList get_str_list(Dictionary d, String key);
426
427 class Geometry : public QGraphicsPathItem
428 {
429 public:
430 enum ArrowStyle {
431 NoArrow, //NOTE: Allow this enum to evaluate false
432 Open,
433 Closed,
434 Dot,
435 Box,
436 Tick
437 };
438 };
439
440
441
442
443
444
445

```

```

446 enum lineStyle {
447 NoLine, //NOTE: Allow this enum to evaluate false
448 Flared,
449 Fletching
450 };
451
452 Dictionary properties;
453
454 QPen objPen;
455 QPen lwtPen;
456 QLineF objLine;
457 String objRubberMode = "OBJ_RUBBER_OFF";
458 QHash<QString, QPointF> objRubberPoints;
459 QHash<QString, QString> objRubberTexts;
460 int64_t objID;
461
462 QPointF arcStartPoint;
463 QPointF arcMidPoint;
464 QPointF arcEndPoint;
465
466 bool curved;
467 bool filled;
468 QPainterPath lineStylePath;
469 QPainterPath arrowStylePath;
470 EmbReal arrowStyleAngle;
471 EmbReal arrowStyleLength;
472 EmbReal lineStyleAngle;
473 EmbReal lineStyleLength;
474
475 QPainterPath normalPath;
476
477 QString objText;
478 QString objTextFont;
479 QString objTextJustify;
480 bool objTextBackward;
481 bool objTextUpsideDown;
482 QPainterPath objTextPath;
483
484 std::vector<EmbReal> x_values;
485 std::vector<EmbReal> y_values;
486
487 int gripIndex;
488
489 int Type = OBJ_TYPE_BASE;
490 virtual int type(){ return Type; }
491
492 Geometry(int object_type = OBJ_TYPE_BASE, QGraphicsItem* parent = 0);
493 Geometry(Geometry *obj, QGraphicsItem* parent = 0);
494 Geometry(EmbArc arc, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
495 Geometry(EmbCircle circle, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
496 Geometry(EmbLine line, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
497 Geometry(EmbEllipse ellipse, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
498 Geometry(EmbRect rect, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
499 Geometry(QString str, EmbVector position, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent =
500 0);
501 Geometry(EmbLine line, int Type_, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent);
502 Geometry(QPainterPath p, int type_, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
503 Geometry(EmbVector pos, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
504
505 void init_arc(EmbArc arc, QRgb rgb, Qt::PenStyle lineType);
506 void init_circle(EmbCircle circle, QRgb rgb, Qt::PenStyle lineType);
507 void init_line(EmbLine line, QRgb rgb, Qt::PenStyle lineType);
508 void init_ellipse(EmbEllipse ellipse, QRgb rgb, Qt::PenStyle lineType);
509 void init_rect(EmbRect rect, QRgb rgb, Qt::PenStyle lineType);
510 void init_text_single(QString str, EmbVector position, QRgb rgb, Qt::PenStyle lineType);
511 void init_path(QPainterPath p, QRgb rgb, Qt::PenStyle lineType);
512 void init_point(EmbVector pos, QRgb rgb, Qt::PenStyle lineType);
513
514 void init(void);
515
516 ~Geometry();
517
518 /* Getters */
519 Qt::PenStyle objectLineType() { return objPen.style(); }
520 EmbReal objectLineWeight() { return lwtPen.widthF(); }
521 QPointF objectRubberPoint(QString key);
522 QString objectRubberText(QString key);
523
524 QPointF objectCenter() { return scenePos(); }
525 QPointF objectPos() { return scenePos(); }
526 EmbReal objectX(){ return scenePos().x(); }
527 EmbReal objectY(){ return scenePos().y(); }
528
529 QPointF objectTopLeft();
530 QPointF objectTopRight();
531 QPointF objectBottomLeft();
532 QPointF objectBottomRight();

```



```

532 EmbReal objectArea();
533 QPointF objectStartPoint();
534 QPointF objectMidPoint();
535 QPointF objectEndPoint();
536
537 QRectF rect();
538 void circle_click(Dictionary global, EmbVector v);
539 EmbReal objectWidth();
540 EmbReal objectHeight();
541 EmbReal objectRadiusMajor();
542 EmbReal objectRadiusMinor();
543 EmbReal objectDiameterMajor();
544 EmbReal objectDiameterMinor();
545 QPointF objectEndPoint1();
546 QPointF objectEndPoint2();
547 EmbReal objectStartAngle();
548 EmbReal objectEndAngle();
549 EmbReal objectArcLength();
550 EmbReal objectChord();
551 EmbReal objectIncludedAngle();
552 bool objectClockwise();
553 EmbReal objectX1() { return objectEndPoint1().x(); }
554 EmbReal objectY1() { return objectEndPoint1().y(); }
555 EmbReal objectX2() { return objectEndPoint2().x(); }
556 EmbReal objectY2() { return objectEndPoint2().y(); }
557 EmbReal objectAngle();
558 QPointF objectDelta() { return objectEndPoint2() - objectEndPoint1(); }
559 EmbReal objectLength() { return objLine.length()*scale(); }
560 EmbReal objectRadius();
561 EmbReal objectDiameter();
562 EmbReal objectCircumference();
563 QPointF objectQuadrant0();
564 QPointF objectQuadrant90();
565 QPointF objectQuadrant180();
566 QPointF objectQuadrant270();
567 QPainterPath objectCopyPath();
568 QPainterPath objectSavePath();
569
570 std::vector<QPainterPath> objectSavePathList() { return subPathList(); }
571 std::vector<QPainterPath> subPathList();
572
573 int findIndex(const QPointF& point);
574
575 void setObjectEndPoint1(EmbVector endPt1);
576 void setObjectEndPoint2(EmbVector endPt2);
577
578 void updatePath();
579 void updatePath(const QPainterPath& p);
580 void updateLeader(void);
581
582 virtual QRectF boundingRect();
583
584 void drawRubberLine(const QLineF& rubLine, QPainter* painter = 0, const char* colorFromScene = 0);
585
586 void updateRubber(QPainter* painter = 0);
587 void vulcanize(void);
588 QPointF mouseSnapPoint(const QPointF& mousePoint);
589 std::vector<QPointF> allGripPoints();
590 void gripEdit(const QPointF& before, const QPointF& after);
591
592 void realRender(QPainter* painter, const QPainterPath& renderPath);
593 void paint(QPainter*, const QStyleOptionGraphicsItem*, QWidget*);
594
595 /* Updaters, todo: combine */
596 void calculateArcData(EmbArc arc);
597 void updateArcRect(EmbReal radius);
598
599 /* Setters */
600 void setObjectPos(const QPointF& point) { setPos(point.x(), point.y()); }
601 void setObjectX(EmbReal x) { setPos(x, objectY()); }
602 void setObjectY(EmbReal y) { setPos(objectX(), y); }
603 void setObjectCenter(EmbVector center);
604 void setObjectCenterX(EmbReal centerX);
605 void setObjectCenterY(EmbReal centerY);
606 void setObjectSize(EmbReal width, EmbReal height);
607 void setObjectRect(EmbReal x, EmbReal y, EmbReal w, EmbReal h);
608 void setRect(const QRectF& r);
609 void setRect(EmbReal x, EmbReal y, EmbReal w, EmbReal h);
610 void setLine(const QLineF& li);
611 void setLine(EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2);
612 void setObjectLineWeight(String lineWeight);
613 void setObjectRadius(EmbReal radius);
614 void setObjectStartAngle(EmbReal angle);
615 void setObjectEndAngle(EmbReal angle);
616 void setObjectStartPoint(EmbVector point);
617 void setObjectMidPoint(EmbVector point);
618 void setObjectEndPoint(EmbVector point);

```

```

619 void setObjectDiameter(EmbReal diameter);
620 void setObjectArea(EmbReal area);
621 void setObjectCircumference(EmbReal circumference);
622 void setObjectPos(EmbReal x, EmbReal y) { setPos(x, y); }
623 void setObjectText(QString str);
624 void setObjectTextFont(QString font);
625 void setObjectTextJustify(QString justify);
626 void setObjectTextSize(EmbReal size);
627 void setObjectTextStyle(bool bold, bool italic, bool under, bool strike, bool over);
628 void setObjectTextBold(bool val);
629 void setObjectTextItalic(bool val);
630 void setObjectTextUnderline(bool val);
631 void setObjectTextStrikeOut(bool val);
632 void setObjectTextOverline(bool val);
633 void setObjectTextBackward(bool val);
634 void setObjectTextUpsideDown(bool val);
635 void setObjectRadiusMajor(EmbReal radius);
636 void setObjectRadiusMinor(EmbReal radius);
637 void setObjectDiameterMajor(EmbReal diameter);
638 void setObjectDiameterMinor(EmbReal diameter);
639
640 /* Scripted commands, uses the script string in */
641 void script_main(void);
642 void script_click(EmbVector v);
643 void script_context(QString str);
644 void script_prompt(QString str);
645 };
646
647 class SaveObject : public QObject
648 {
649 Q_OBJECT
650 public:
651 SaveObject(QGraphicsScene* theScene, QObject* parent = 0);
652 ~SaveObject();
653
654 bool save(QString fileName);
655
656 void addArc(EmbPattern* pattern, QGraphicsItem* item);
657 void addBlock(EmbPattern* pattern, QGraphicsItem* item);
658 void addCircle(EmbPattern* pattern, QGraphicsItem* item);
659 void addDimAligned(EmbPattern* pattern, QGraphicsItem* item);
660 void addDimAngular(EmbPattern* pattern, QGraphicsItem* item);
661 void addDimArcLength(EmbPattern* pattern, QGraphicsItem* item);
662 void addDimDiameter(EmbPattern* pattern, QGraphicsItem* item);
663 void addDimLeader(EmbPattern* pattern, QGraphicsItem* item);
664 void addDimLinear(EmbPattern* pattern, QGraphicsItem* item);
665 void addDimOrdinate(EmbPattern* pattern, QGraphicsItem* item);
666 void addDimRadius(EmbPattern* pattern, QGraphicsItem* item);
667 void addEllipse(EmbPattern* pattern, QGraphicsItem* item);
668 void addEllipseArc(EmbPattern* pattern, QGraphicsItem* item);
669 void addGrid(EmbPattern* pattern, QGraphicsItem* item);
670 void addHatch(EmbPattern* pattern, QGraphicsItem* item);
671 void addImage(EmbPattern* pattern, QGraphicsItem* item);
672 void addInfiniteLine(EmbPattern* pattern, QGraphicsItem* item);
673 void addLine(EmbPattern* pattern, QGraphicsItem* item);
674 void addPath(EmbPattern* pattern, QGraphicsItem* item);
675 void addPoint(EmbPattern* pattern, QGraphicsItem* item);
676 void addPolygon(EmbPattern* pattern, QGraphicsItem* item);
677 void addPolyline(EmbPattern* pattern, QGraphicsItem* item);
678 void addRay(EmbPattern* pattern, QGraphicsItem* item);
679 void addRectangle(EmbPattern* pattern, QGraphicsItem* item);
680 void addSlot(EmbPattern* pattern, QGraphicsItem* item);
681 void addSpline(EmbPattern* pattern, QGraphicsItem* item);
682 void addTextMulti(EmbPattern* pattern, QGraphicsItem* item);
683 void addTextSingle(EmbPattern* pattern, QGraphicsItem* item);
684
685 QGraphicsScene* gscene;
686 int formatType;
687
688 void toPolyline(EmbPattern* pattern, const QPointF& objPos, const QPainterPath& objPath, QString
 layer, const QColor& color, QString lineType, QString lineWeight);
689 };
690
691 class Application : public QApplication
692 {
693 Q_OBJECT
694 public:
695 Application(int argc, char **argv);
696 void setMainWin(MainWindow* mainWin) { __mainWin = __mainWin; }
697 MainWindow* __mainWin;
698 protected:
699 virtual bool event(QEvent *e);
700 };
701
702 class CmdPromptInput : public QLineEdit

```

```

716 {
717 Q_OBJECT
718
719 public:
720 CmdPromptInput(QWidget* parent = 0);
721 ~CmdPromptInput() {}
722
723 QString curText;
724 QString defaultPrefix;
725 QString prefix;
726
727 QString lastCmd;
728 QString curCmd;
729 bool cmdActive;
730
731 bool rapidFireEnabled;
732 bool isBlinking;
733
734 void changeFormatting(std::vector<QTextLayout::FormatRange> formats);
735 void clearFormatting();
736 void applyFormatting();
737
738 protected:
739 void contextMenuEvent(QContextMenuEvent *event);
740 bool eventFilter(QObject *obj, QEvent *event);
741
742 signals:
743 void appendHistory(QString txt, int prefixLength);
744
745 //These connect to the CmdPrompt signals
746 void startCommand(QString cmd);
747 void runCommand(QString cmd, QString cmdtxt);
748 void deletePressed();
749 void tabPressed();
750 void escapePressed();
751 void upPressed();
752 void downPressed();
753 void F1Pressed();
754 void F2Pressed();
755 void F3Pressed();
756 void F4Pressed();
757 void F5Pressed();
758 void F6Pressed();
759 void F7Pressed();
760 void F8Pressed();
761 void F9Pressed();
762 void F10Pressed();
763 void F11Pressed();
764 void F12Pressed();
765 void cutPressed();
766 void copyPressed();
767 void pastePressed();
768 void selectAllPressed();
769 void undoPressed();
770 void redoPressed();
771
772 void shiftPressed();
773 void shiftReleased();
774
775 void showSettings();
776
777 void stopBlinking();
778
779 public slots:
780 void endCommand();
781 void processInput(void);
782 void checkSelection();
783 void updateCurrentText(QString txt);
784 void checkEditedText(QString txt);
785 void checkChangedText(QString txt);
786 void checkCursorPosition(int oldpos, int newpos);
787 private slots:
788 void copyClip();
789 void pasteClip();
790 };
791
792 class CmdPromptHistory : public QTextBrowser
793 {
794 Q_OBJECT
795
796 public:
797 CmdPromptHistory(QWidget* parent = 0);
798 ~CmdPromptHistory();
799
800 int tmpHeight;
801 QString applyFormatting(QString txt, int prefixLength);
802
803

```

```

806 protected:
807 void contextMenuEvent(QContextMenuEvent* event);
808
809 public slots:
810 void appendHistory(QString txt, int prefixLength);
811 void startResizeHistory(int y);
812 void stopResizeHistory(int y);
813 void resizeHistory(int y);
814
815 signals:
816 void historyAppended(QString txt);
817 };
818
822 class CmdPromptSplitter : public QSplitter
823 {
824 Q_OBJECT
825
826 public:
827 CmdPromptSplitter(QWidget* parent = 0);
828 ~CmdPromptSplitter();
829
830 protected:
831 QSplitterHandle* createHandle();
832
833 signals:
834 void pressResizeHistory(int y);
835 void releaseResizeHistory(int y);
836 void moveResizeHistory(int y);
837 };
838
842 class CmdPromptHandle : public QSplitterHandle
843 {
844 Q_OBJECT
845
846 public:
847 CmdPromptHandle(Qt::Orientation orientation, QSplitter* parent);
848 ~CmdPromptHandle();
849
850 int pressY;
851 int releaseY;
852 int moveY;
853
854 protected:
855 void mousePressEvent(QMouseEvent* e);
856 void mouseReleaseEvent(QMouseEvent* e);
857 void mouseMoveEvent(QMouseEvent* e);
858
859 signals:
860 void handlePressed(int y);
861 void handleReleased(int y);
862 void handleMoved(int y);
863 };
864
868 class CmdPrompt : public QWidget
869 {
870 Q_OBJECT
871
872 public:
873 CmdPrompt(QWidget* parent = 0);
874 ~CmdPrompt();
875
876 CmdPromptInput* promptInput;
877 CmdPromptHistory* promptHistory;
878 QVBoxLayout* promptVBoxLayout;
879 QFrame* promptDivider;
880
881 CmdPromptSplitter* promptSplitter;
882
883 QHash<QString, QString>* styleHash;
884 void updateStyle();
885 QTimer* blinkTimer;
886 bool blinkState;
887
888 public slots:
889 void setCurrentText(QString txt) {
890 promptInput->curText = promptInput->prefix + txt;
891 promptInput->setText(promptInput->curText);
892 }
893 void setHistory(QString txt) {
894 promptHistory->setHtml(txt);
895 promptHistory->moveCursor(QTextCursor::End, QTextCursor::MoveAnchor);
896 }
897 void setPrefix(QString txt);
898 void appendHistory(QString txt);
899
900 void alert(QString txt);
901

```

```

902 void startBlinking();
903 void stopBlinking();
904 void blink();
905
906 void setPromptTextColor(const QColor&);
907 void setPromptBackgroundColor(const QColor&);
908 void setPromptFontFamily(QString);
909 void setPromptFontStyle(QString);
910 void setPromptFontSize(int);
911
912 void floatingChanged(bool);
913
914 void saveHistory(QString fileName, bool html);
915
916 signals:
917 void appendTheHistory(QString txt, int prefixLength);
918
919 //For connecting outside of command prompt
920 void startCommand(QString cmd);
921 void runCommand(QString cmd, QString cmdtxt);
922 void deletePressed();
923 void tabPressed();
924 void escapePressed();
925 void upPressed();
926 void downPressed();
927 void F1Pressed();
928 void F2Pressed();
929 void F3Pressed();
930 void F4Pressed();
931 void F5Pressed();
932 void F6Pressed();
933 void F7Pressed();
934 void F8Pressed();
935 void F9Pressed();
936 void F10Pressed();
937 void F11Pressed();
938 void F12Pressed();
939 void cutPressed();
940 void copyPressed();
941 void pastePressed();
942 void selectAllPressed();
943 void undoPressed();
944 void redoPressed();
945
946 void shiftPressed();
947 void shiftReleased();
948
949 void showSettings();
950
951 void historyAppended(QString txt);
952 };
953
954 class EmbDetailsDialog : public QDialog
955 {
956 Q_OBJECT
957
958 public:
959 EmbDetailsDialog(QGraphicsScene* theScene, QWidget *parent = 0);
960 ~EmbDetailsDialog();
961
962 QWidget* mainWidget;
963
964 void getInfo();
965 QWidget* createMainWidget();
966 QWidget* createHistogram();
967
968 QDialogButtonBox* buttonBox;
969
970 uint32_t stitchesTotal;
971 uint32_t stitchesReal;
972 uint32_t stitchesJump;
973 uint32_t stitchesTrim;
974 uint32_t colorTotal;
975 uint32_t colorChanges;
976
977 QRectF boundingRect;
978 };
979
980 class ImageWidget : public QWidget
981 {
982 Q_OBJECT
983
984 public:
985 QImage img;
986 ImageWidget(QString filename, QWidget* parent = 0);
987 ~ImageWidget();
988 };

```

```

995 bool load(QString fileName);
996 bool save(QString fileName);
997
998 protected:
999 void paintEvent(QPaintEvent* event);
1000 };
1001
1002 class LayerManager : public QDialog
1003 {
1004 Q_OBJECT
1005 public:
1006 QStandardItemModel* layerModel;
1007 QSortFilterProxyModel* layerModelSorted;
1008 QTreeView* treeView;
1009
1010 LayerManager(QWidget *parent = 0);
1011 ~LayerManager();
1012
1013 void addLayer(QString name, const bool visible, const bool frozen,
1014 const EmbReal zValue, const QColor color, QString lineType,
1015 QString lineWeight, const bool print);
1016 };
1017
1018 class MainWindow: public QMainWindow
1019 {
1020 Q_OBJECT
1021 public:
1022 MainWindow();
1023 ~MainWindow();
1024
1025 MdiWindow* activeMdiWindow();
1026 QUndoStack* activeUndoStack();
1027
1028 void setUndoCleanIcon(bool opened);
1029
1030 virtual void updateMenuToolBarStatusBar();
1031
1032 std::vector<QGraphicsItem*> cutCopyObjectList;
1033
1034 QString formatFilterOpen;
1035 QString formatFilterSave;
1036
1037 bool isCommandActive() { return prompt->promptInput->cmdActive; }
1038 QString activeCommand() { return prompt->promptInput->curCmd; }
1039 QIcon create_icon(QString stub);
1040 void create_toolbar(String toolbar, String label, QStringList entries);
1041
1042 QString platformString();
1043
1044 public slots:
1045
1046 void onCloseWindow();
1047 virtual void onCloseMdiWin(MdiWindow*);
1048
1049 void recentMenuAboutToShow();
1050
1051 void onWindowActivated(QMdiSubWindow* w);
1052 void windowMenuAboutToShow();
1053 void windowMenuActivated(bool checked/*int id*/);
1054
1055 void updateAllViewScrollBars(bool val);
1056 void updateAllViewCrossHairColors(QColor color);
1057 void updateAllViewBackgroundColors(QColor color);
1058 void updateAllViewSelectBoxColors(QColor colorL, QColor fillL, QColor colorR, QColor fillR, int alpha);
1059 void updateAllViewGridColors(QColor color);
1060 void updateAllViewRulerColors(QColor color);
1061
1062 void updatePickAddMode(bool val);
1063 void pickAddModeToggled();
1064
1065 void settingsPrompt();
1066
1067 protected:
1068 virtual void resizeEvent(QResizeEvent*);
1069 void closeEvent(QCloseEvent *event);
1070 QAction* getFileSeparator();
1071 void loadFormats();
1072
1073 bool shiftKeyPressedState;
1074
1075 QByteArray layoutState;
1076
1077 int numOfDocs;
1078 int docIndex;
1079
1080

```

```

1088 std::vector<MdiWindow*> listMdiWin;
1089 QMdiSubWindow* findMdiWindow(String fileName);
1090
1091 QAction* myFileSeparator;
1092
1093 void createAllActions();
1094 void createAllMenus();
1095 void createAllToolbars();
1096
1097 // Selectors
1098 QComboBox* layerSelector;
1099 QComboBox* colorSelector;
1100 QComboBox* linetypeSelector;
1101 QComboBox* linewidthSelector;
1102 QFontComboBox* textFontSelector;
1103 QComboBox* textSizeSelector;
1104
1105 private slots:
1106 void hideUnimplemented();
1107
1108 public slots:
1109 void stub_testing();
1110
1111 void promptHistoryAppended(QString txt);
1112 void logPromptInput(QString txt);
1113 void promptInputPrevious();
1114 void promptInputNext();
1115
1116 void about(void);
1117 void tipOfTheDay(void);
1118
1119 void newFile();
1120 void openFile(bool recent = false, String recentFile = "");
1121 void openFilesSelected(StringList files);
1122 void openrecentfile();
1123 void savefile();
1124 void saveasfile();
1125 void quit();
1126 void checkForUpdates();
1127 // Help Menu
1128 void buttonTipOfTheDayClicked(int);
1129
1130 void closeToolBar(QAction*);
1131 void floatingChangedToolBar(bool);
1132
1133 void toggleGrid();
1134 void toggleRuler();
1135 void toggleLwt();
1136
1137 // Icons
1138 void iconResize(int iconSize);
1139
1140 //Selectors
1141 void layerSelectorIndexChanged(int index);
1142 void colorSelectorIndexChanged(int index);
1143 void linetypeSelectorIndexChanged(int index);
1144 void linewidthSelectorIndexChanged(int index);
1145 void textFontSelectorCurrentFontChanged(const QFont& font);
1146 void textSizeSelectorIndexChanged(int index);
1147
1148 void setTextFont(QString str);
1149 void setTextSize(EmbReal num);
1150
1151 QString getCurrentLayer();
1152 QColor getCurrentColor();
1153 QString getCurrentLineType();
1154 QString getCurrentLineWeight();
1155
1156 bool isShiftPressed();
1157 void setShiftPressed();
1158 void setShiftReleased();
1159
1160 void deletePressed();
1161 void escapePressed();
1162 };
1163
1164 class MdiWindow: public QMdiSubWindow
1165 {
1166 Q_OBJECT
1167
1168 public:
1169 MdiWindow(const int theIndex, QMdiArea* parent, Qt::WindowFlags wflags);
1170 ~MdiWindow();
1171
1172 QMdiArea* mdiArea;
1173 QGraphicsScene* gscene;
1174 View* gview;

```

```

1175
1176 bool fileWasLoaded;
1177
1178 QString promptHistory;
1179 std::vector<QString> promptInputList;
1180 int promptInputNum;
1181
1182 QPrinter printer;
1183
1184 QString curFile;
1185 void setCurrentFile(QString fileName);
1186
1187 int myIndex;
1188
1189 QString curLayer;
1190 QRgb curColor;
1191 QString curLineType;
1192 QString curLineWeight;
1193
1194 void promptInputPrevNext(bool prev);
1195
1196 virtual QSize sizeHint();
1197 QString getShortCurrentFile();
1198 void designDetails();
1199 bool loadFile(QString fileName);
1200 bool saveFile(QString fileName);
1201 signals:
1202 void sendCloseMdiWin(MdiWindow*);
1203
1204 public slots:
1205 void closeEvent(QCloseEvent* e);
1206 void onWindowActivated();
1207
1208 void currentLayerChanged(QString layer);
1209 void currentColorChanged(const QRgb& color);
1210 void currentLinetypeChanged(QString type);
1211 void currentLineweightChanged(QString weight);
1212
1213 void updateColorLinetypeLineweight();
1214 void deletePressed();
1215 void escapePressed();
1216
1217 void showViewScrollBars(bool val);
1218 void setViewCrossHairColor(QRgb color);
1219 void setViewBackgroundColor(QRgb color);
1220 void setViewSelectBoxColors(QRgb colorL, QRgb fillL, QRgb colorR, QRgb fillR, int alpha);
1221 void setViewGridColor(QRgb color);
1222 void setViewRulerColor(QRgb color);
1223
1224 void print();
1225 void saveBMC();
1226
1227 void promptHistoryAppended(QString txt);
1228 void logPromptInput(QString txt);
1229 void promptInputPrevious();
1230 void promptInputNext();
1231 };
1232
1233 class MdiArea : public QMdiArea
1234 {
1235 Q_OBJECT
1236
1237 public:
1238 bool useLogo;
1239 bool useTexture;
1240 bool useColor;
1241
1242 QPixmap bgLogo;
1243 QPixmap bgTexture;
1244 QColor bgColor;
1245
1246 void zoomExtentsAllSubWindows();
1247 void forceRepaint();
1248
1249 MdiArea(QWidget* parent = 0);
1250 ~MdiArea();
1251
1252 void useBackgroundLogo(bool use);
1253 void useBackgroundTexture(bool use);
1254 void useBackgroundColor(bool use);
1255
1256 void setBackgroundLogo(QString fileName);
1257 void setBackgroundTexture(QString fileName);
1258 void setBackgroundColor(const QColor& color);
1259
1260 public slots:
1261 void cascade();

```



```

1265 void tile();
1266 protected:
1267 virtual void mouseDoubleClickEvent(QMouseEvent* e);
1268 virtual void paintEvent(QPaintEvent* e);
1269 };
1270
1271 class PreviewDialog : public QFileDialog
1272 {
1273 Q_OBJECT
1274 public:
1275 PreviewDialog(QWidget* parent = 0,
1276 QString caption = QString(),
1277 QString directory = QString(),
1278 QString filter = QString());
1279 ~PreviewDialog();
1280 ImageWidget* imgWidget;
1281 };
1282
1283 class PropertyEditor : public QDockWidget
1284 {
1285 Q_OBJECT
1286 public:
1287 PropertyEditor(QString iconDirectory = QString(), bool pickAddMode = true, QWidget* widgetToFocus
1288 = 0, QWidget* parent = 0); //, Qt::WindowFlags flags = 0);
1289 ~PropertyEditor();
1290 QWidget* focusWidget;
1291 QString iconDir;
1292 int iconSize;
1293 Qt::ToolButtonStyle propertyEditorButtonStyle;
1294 bool pickAdd;
1295 std::vector<QGraphicsItem*> selectedItemList;
1296 QToolButton* createToolButton(QString iconName, QString txt);
1297 QLineEdit* createLineEdit(QString validatorType = QString(), bool readOnly = false);
1298 int precisionAngle;
1299 int precisionLength;
1300 void updateLineEditStrIfVaries(QLineEdit* lineEdit, QString str);
1301 void updateLineEditNumIfVaries(QLineEdit* lineEdit, EmbReal num, bool useAnglePrecision);
1302 void updateFontComboBoxStrIfVaries(QFontComboBox* fontComboBox, QString str);
1303 void updateComboBoxStrIfVaries(QComboBox* comboBox, QString str, QStringList strList);
1304 void updateComboBoxBoolIfVaries(QComboBox* comboBox, bool val, bool yesOrNoText);
1305 QSignalMapper* signalMapper;
1306 void mapSignal(QObject* fieldObj, QString name, QVariant value);
1307 // Selection
1308 // =====
1309 QComboBox* createComboBoxSelected();
1310 QToolButton* createToolButtonQSelect();
1311 QToolButton* createToolButtonPickAdd();
1312 QComboBox* comboBoxSelected;
1313 QToolButton* toolButtonQSelect;
1314 QToolButton* toolButtonPickAdd;
1315 //TODO: Alphabetic/Categorized TabWidget
1316 void createGroupBox(QString group_box_key, const char *title);
1317 protected:
1318 bool eventFilter(QObject *obj, QEvent *event);
1319 signals:
1320 void pickAddModeToggled();
1321 public slots:
1322 void setSelectedItems(std::vector<QGraphicsItem*> itemList);
1323 void updatePickAddModeButton(bool pickAddMode);
1324 private slots:
1325 void fieldEdited(QObject* fieldObj);
1326 void showGroups(int objType);
1327 void showOneType(int index);
1328 void hideAllGroups();
1329 void clearAllFields();
1330 void togglePickAddMode();
1331 };

```

```

1354
1355
1356 class SelectBox : public QRubberBand
1357 {
1358 Q_OBJECT
1359
1360 public:
1361 SelectBox(Shape s, QWidget* parent = 0);
1362
1363 QColor leftBrushColor;
1364 QColor rightBrushColor;
1365 QColor leftPenColor;
1366 QColor rightPenColor;
1367 uint8_t alpha;
1368
1369 QBrush dirBrush;
1370 QBrush leftBrush;
1371 QBrush rightBrush;
1372
1373 QPen dirPen;
1374 QPen leftPen;
1375 QPen rightPen;
1376
1377 bool boxDir;
1378
1379 void forceRepaint();
1380
1381 public slots:
1382 void setDirection(int dir);
1383 void setColors(const QColor& colorL, const QColor& fillL, const QColor& colorR, const QColor&
 fillR, int newAlpha);
1384
1385 protected:
1386 void paintEvent(QPaintEvent*);
1387 };
1388
1389 class Settings_Dialog : public QDialog
1390 {
1391 Q_OBJECT
1392
1393 public:
1394 Settings_Dialog(QString showTab = QString(), QWidget *parent = 0);
1395 ~Settings_Dialog();
1396
1397 QTabWidget* tabWidget;
1398
1399 QWidget* createTabGeneral();
1400 QWidget* createTabFilesPaths();
1401 QWidget* createTabDisplay();
1402 QWidget* createTabPrompt();
1403 QWidget* createTabOpenSave();
1404 QWidget* createTabPrinting();
1405 QWidget* createTabSnap();
1406 QWidget* createTabGridRuler();
1407 QWidget* createTabOrthoPolar();
1408 QWidget* createTabQuickSnap();
1409 QWidget* createTabQuickTrack();
1410 QWidget* createTabLineWeight();
1411 QWidget* createTabSelection();
1412
1413 QDialogButtonBox* buttonBox;
1414
1415 void addColorsToComboBox(QComboBox* comboBox);
1416
1417 void create_float_spinbox(
1418 QGroupBox *gb,
1419 QGridLayout* gridLayout,
1420 const char *label_in,
1421 EmbReal single_step,
1422 EmbReal lower,
1423 EmbReal upper,
1424 String,
1425 int row);
1426 QCheckBox* create_checkbox(QGroupBox *groupbox, String label);
1427
1428 private slots:
1429 void comboBoxIconSizeCurrentIndexChanged(int);
1430 void checkBoxGeneralMdiBGUseLogoStateChanged(int);
1431 void chooseGeneralMdiBackgroundLogo();
1432 void checkBoxGeneralMdiBGUseTextureStateChanged(int);
1433 void chooseGeneralMdiBackgroundTexture();
1434 void checkBoxGeneralMdiBGUseColorStateChanged(int);
1435 void chooseGeneralMdiBackgroundColor();
1436 void currentGeneralMdiBackgroundColorChanged(const QColor&);
1437 void checkBoxShowScrollBarsStateChanged(int);
1438 void comboBoxScrollBarWidgetCurrentIndexChanged(int);
1439 void chooseDisplayCrossHairColor();

```

```

1443 void currentDisplayCrossHairColorChanged(const QColor&);
1444 void chooseDisplayBackgroundColor();
1445 void currentDisplayBackgroundColorChanged(const QColor&);
1446 void chooseDisplaySelectBoxLeftColor();
1447 void currentDisplaySelectBoxLeftColorChanged(const QColor&);
1448 void chooseDisplaySelectBoxLeftFill();
1449 void currentDisplaySelectBoxLeftFillChanged(const QColor&);
1450 void chooseDisplaySelectBoxRightColor();
1451 void currentDisplaySelectBoxRightColorChanged(const QColor&);
1452 void chooseDisplaySelectBoxRightFill();
1453 void currentDisplaySelectBoxRightFillChanged(const QColor&);
1454 void comboBoxSelectionCoolGripColorCurrentIndexChanged(int index);
1455 void comboBoxSelectionHotGripColorCurrentIndexChanged(int index);
1456 void spinBoxDisplaySelectBoxAlphaValueChanged(int);
1457 void choosePromptTextColor();
1458 void currentPromptTextColorChanged(const QColor&);
1459 void choosePromptBackgroundColor();
1460 void currentPromptBackgroundColorChanged(const QColor&);
1461 void comboBoxPromptFontFamilyCurrentIndexChanged(QString);
1462 void comboBoxPromptFontStyleCurrentIndexChanged(QString);
1463 void spinBoxPromptFontSizeValueChanged(int);
1464 void checkBoxPromptSaveHistoryAsHtmlStateChanged(int);
1465 void checkBoxCustomFilterStateChanged(int);
1466 void buttonCustomFilterSelectAllClicked();
1467 void buttonCustomFilterClearAllClicked();
1468 void checkBoxGridColorMatchCrossHairStateChanged(int);
1469 void chooseGridColor();
1470 void currentGridColorChanged(const QColor&);
1471 void checkBoxGridLoadFromFileStateChanged(int);
1472 void comboBoxGridTypeCurrentIndexChanged(QString);
1473 void checkBoxGridCenterOnOriginStateChanged(int);
1474 void checkBoxRulerShowOnLoadStateChanged(int);
1475 void comboBoxRulerMetricCurrentIndexChanged(int);
1476 void chooseRulerColor();
1477 void currentRulerColorChanged(const QColor&);
1478 void spinBoxRulerPixelSizeValueChanged(double);
1479 void buttonQSnapSelectAllClicked();
1480 void buttonQSnapClearAllClicked();
1481 void comboBoxQSnapLocatorColorCurrentIndexChanged(int);
1482 void checkBoxLwtShowLwtStateChanged(int);
1483 void checkBoxLwtRealRenderStateChanged(int);
1484
1485 void acceptChanges();
1486 void rejectChanges();
1487
1488 signals:
1489 void buttonCustomFilterSelectAll(bool);
1490 void buttonCustomFilterClearAll(bool);
1491 void buttonQSnapSelectAll(bool);
1492 void buttonQSnapClearAll(bool);
1493 };
1494
1495 class StatusBar : public QStatusBar
1496 {
1497 Q_OBJECT
1498 public:
1499 StatusBar(QWidget* parent = 0);
1500 std::unordered_map<String, QToolButton*> buttons;
1501 QLabel* statusBarMouseCoord;
1502 void setMouseCoord(EmbReal x, EmbReal y);
1503 void context_menu_action(QToolButton *button, const char *icon, const char *label, QMenu *menu,
1504 String setting_page);
1505 void toggle(String key, bool on);
1506 void context_menu_event(QContextMenuEvent *event, QToolButton *button);
1507 };
1508
1509 class UndoEditor : public QDockWidget
1510 {
1511 Q_OBJECT
1512 public:
1513 UndoEditor(QString iconDirectory = QString(), QWidget* widgetToFocus = 0, QWidget* parent = 0);
1514 //, Qt::WindowFlags flags = 0);
1515 ~UndoEditor();
1516
1517 void addStack(QUndoStack* stack);
1518
1519 bool canUndo();
1520 bool canRedo();
1521
1522 QWidget* focusWidget;
1523
1524 QString iconDir;
1525 int iconSize;
1526
1527 QUndoGroup* undoGroup;

```

```

1534 QUndoView* undoView;
1535
1536 QString undoText();
1537 QString redoText();
1538 protected:
1539
1540 public slots:
1541 void undo();
1542 void redo();
1543
1544 void updateCleanIcon(bool opened);
1545 };
1546
1547 class UndoableCommand : public QUndoCommand
1548 {
1549 public:
1550 UndoableCommand(String command, QString text, Geometry* obj, View* v, QUndoCommand* parent = 0);
1551 UndoableCommand(EmbVector d, QString text, Geometry* obj, View* v, QUndoCommand* parent = 0);
1552 UndoableCommand(String command, EmbVector pivot, EmbReal angle, QString text, Geometry* obj, View*
1553 v, QUndoCommand* parent = 0);
1554 UndoableCommand(QString type, View* v, QUndoCommand* parent = 0);
1555 UndoableCommand(const QPointF beforePoint, const QPointF afterPoint, QString text, Geometry* obj,
1556 View* v, QUndoCommand* parent = 0);
1557 UndoableCommand(EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, QString text, Geometry* obj, View*
1558 v, QUndoCommand* parent = 0);
1559
1560 int id(){ return 1234; }
1561 bool mergeWith(const QUndoCommand* command);
1562 void undo();
1563 void redo();
1564 void mirror();
1565 void rotate(EmbVector pivot, EmbReal rot);
1566
1567 Geometry* object;
1568 View* gview;
1569 String command;
1570 EmbVector delta;
1571 EmbVector pivot;
1572 QPointF before;
1573 QPointF after;
1574 EmbReal angle;
1575 EmbReal factor;
1576 QString navType;
1577 QTransform fromTransform;
1578 QTransform toTransform;
1579 QPointF fromCenter;
1580 QPointF toCenter;
1581 QLineF mirrorLine;
1582 bool done;
1583 };
1584
1585 class View : public QGraphicsView
1586 {
1587 Q_OBJECT
1588
1589 public:
1590 View(QGraphicsScene* theScene, QWidget* parent);
1591 ~View();
1592
1593 Dictionary state;
1594
1595 std::vector<QGraphicsItem*> selected_items();
1596
1597 bool allowZoomIn();
1598 bool allowZoomOut();
1599
1600 QColor gridColor;
1601 QPainterPath gridPath;
1602 QPainterPath originPath;
1603 bool rulerMetric;
1604 QColor rulerColor;
1605 uint8_t rulerPixelSize;
1606
1607 bool grippingActive;
1608 bool rapidMoveActive;
1609 bool previewActive;
1610 bool pastingActive;
1611 bool movingActive;
1612 bool selectingActive;
1613 bool zoomWindowActive;
1614 bool panningRealTimeActive;
1615 bool panningPointActive;
1616 bool panningActive;
1617 bool qSnapActive;
1618 bool qSnapToggle;
1619
1620 Geometry* gripBaseObj;

```

```

1624 Geometry* tempBaseObj;
1625
1626 QGraphicsScene* gscene;
1627 QUndoStack* undoStack;
1628
1629 SelectBox* selectBox;
1630 QPointF scenePressPoint;
1631 QPoint pressPoint;
1632 QPointF sceneMovePoint;
1633 QPoint movePoint;
1634 QPointF sceneReleasePoint;
1635 QPoint releasePoint;
1636 QPointF sceneGripPoint;
1637
1638 void updateMouseCoords(int x, int y);
1639 QPoint viewMousePoint;
1640 QPointF sceneMousePoint;
1641 QRgb qsnapLocatorColor;
1642 uint8_t qsnapLocatorSize;
1643 uint8_t qsnapApertureSize;
1644 QRgb gripColorCool;
1645 QRgb gripColorHot;
1646 uint8_t gripSize;
1647 uint8_t pickBoxSize;
1648 QRgb crosshairColor;
1649 uint32_t crosshairSize;
1650
1651 void recalculateLimits();
1652 void zoomToPoint(const QPoint& mousePoint, int zoomDir);
1653 void centerAt(const QPointF& centerPoint);
1654 QPointF center() { return mapToScene(rect().center()); }
1655
1656 QUndoStack* getUndoStack() { return undoStack; }
1657 void addObject(Geometry* obj);
1658 void deleteObject(Geometry* obj);
1659 void vulcanizeObject(Geometry* obj);
1660
1661 public slots:
1662 void zoomIn();
1663 void zoomOut();
1664 void zoomWindow();
1665 void zoomSelected();
1666 void zoomExtents();
1667 void panRealTime();
1668 void panPoint();
1669 void panLeft();
1670 void panRight();
1671 void panUp();
1672 void panDown();
1673 void selectAll();
1674 void selectionChanged();
1675 void clearSelection();
1676 void deleteSelected();
1677 void moveSelected(EmbReal dx, EmbReal dy);
1678 void cut();
1679 void copy();
1680 void paste();
1681 void repeatAction();
1682 void moveAction();
1683 void scaleAction();
1684 void scaleSelected(EmbReal x, EmbReal y, EmbReal factor);
1685 void rotateAction();
1686 void rotateSelected(EmbReal x, EmbReal y, EmbReal rot);
1687 void mirrorSelected(EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2);
1688 int numSelected();
1689
1690 void deletePressed();
1691 void escapePressed();
1692
1693 void cornerButtonClicked();
1694
1695 void showScrollBars(bool val);
1696 void setCornerButton();
1697 void setCrossHairColor(QRgb color);
1698 void setCrossHairSize(uint8_t percent);
1699 void setBackgroundColor(QRgb color);
1700 void setSelectBoxColors(QRgb colorL, QRgb fillL, QRgb colorR, QRgb fillR, int alpha);
1701 void toggleSnap(bool on);
1702 void toggleGrid(bool on);
1703 void toggleRuler(bool on);
1704 void toggleOrtho(bool on);
1705 void togglePolar(bool on);
1706 void toggleQSnap(bool on);
1707 void toggleQTrack(bool on);
1708 void toggleLwt(bool on);
1709 void toggleReal(bool on);
1710 bool isLwtEnabled();

```

```

1711 bool isRealEnabled();
1712
1713 void setGridColor(QRgb color);
1714 void createGrid(QString gridType);
1715 void setRulerColor(QRgb color);
1716
1717 void previewOn(String clone, String mode, EmbReal x, EmbReal y, EmbReal data);
1718 void previewOff();
1719
1720 bool allowRubber();
1721 void addToRubberRoom(QGraphicsItem* item);
1722 void vulcanizeRubberRoom();
1723 void clearRubberRoom();
1724 void spareRubber(int64_t id);
1725 void setRubberMode(String mode);
1726 void setRubberPoint(QString key, const QPointF& point);
1727 void setRubberText(QString key, QString txt);
1728
1729 protected:
1730 void mouseDoubleClickEvent(QMouseEvent* event);
1731 void mousePressEvent(QMouseEvent* event);
1732 void mouseMoveEvent(QMouseEvent* event);
1733 void mouseReleaseEvent(QMouseEvent* event);
1734 void wheelEvent(QWheelEvent* event);
1735 void contextMenuEvent(QContextMenuEvent* event);
1736 void drawBackground(QPainter* painter, const QRectF& rect);
1737 void drawForeground(QPainter* painter, const QRectF& rect);
1738 void enterEvent(QEvent* event);
1739
1740 private:
1741 QHash<int64_t, QGraphicsItem*> hashDeletedObjects;
1742
1743 QStringList spareRubberList;
1744
1745 void createGridRect();
1746 void createGridPolar();
1747 void createGridIso();
1748 void createOrigin();
1749
1750 void loadRulerSettings();
1751
1752 bool willUnderflowInt32(int64_t a, int64_t b);
1753 bool willOverflowInt32(int64_t a, int64_t b);
1754 int roundToMultiple(bool roundUp, int numToRound, int multiple);
1755 QPainterPath createRulerTextPath(EmbVector position, QString str, EmbReal height);
1756
1757 QList<QGraphicsItem*> previewObjectList;
1758 QGraphicsItemGroup* previewObjectItemGroup;
1759 QPointF previewPoint;
1760 EmbReal previewData;
1761 String previewMode;
1762
1763 std::vector<QGraphicsItem*> createObjectList(std::vector<QGraphicsItem*> list);
1764 QPointF cutCopyMousePoint;
1765 QGraphicsItemGroup* pasteObjectItemGroup;
1766 QPointF pasteDelta;
1767
1768 std::vector<QGraphicsItem*> rubberRoomList;
1769
1770 void copySelected();
1771
1772 void startGripping(Geometry* obj);
1773 void stopGripping(bool accept = false);
1774
1775 void panStart(const QPoint& point);
1776 int panDistance;
1777 int panStartX;
1778 int panStartY;
1779
1780 void alignScenePointWithViewPoint(const QPointF& scenePoint, const QPoint& viewPoint);
1781 };
1782
1783 #endif

```

## 18.95 src/imagewidget.cpp File Reference

```
#include "embroidermodder.h"
```

## 18.96 src/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)
- QPainterPath [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)

### 18.96.1 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.

### 18.96.2 Function Documentation

**18.96.2.1 `add_to_path()`** QPainterPath add\_to\_path ( QPainterPath path, EmbVector scale, String command )

**18.96.2.2 `debug_message()`** void debug\_message ( std::string msg )

`debug_message`

#### Parameters

|            |  |
|------------|--|
| <i>msg</i> |  |
|------------|--|

**18.96.2.3 `degrees__()`** EmbReal degrees\_\_ ( EmbReal radian )

`degrees__`

#### Parameters

|               |  |
|---------------|--|
| <i>radian</i> |  |
|---------------|--|

#### Returns

**18.96.2.4 `get_bool()`** bool get\_bool ( Dictionary d, String key )

**18.96.2.5 `get_int()`** int get\_int (



```
Dictionary d,
String key)
```

**18.96.2.6 get\_n\_reals()** `std::vector< float > get_n_reals (`  
    `StringList list,`  
    `int n,`  
    `int * offset )`

Utility function for add\_to\_path.

**18.96.2.7 get\_qstr()** `QString get_qstr (`  
    `Dictionary d,`  
    `String key )`

**18.96.2.8 get\_real()** `EmbReal get_real (`  
    `Dictionary d,`  
    `String key )`

**18.96.2.9 get\_str()** `String get_str (`  
    `Dictionary d,`  
    `String key )`

**18.96.2.10 get\_str\_list()** `StringList get_str_list (`  
    `Dictionary d,`  
    `String key )`

**18.96.2.11 get\_uint()** `uint32_t get_uint (`  
    `Dictionary d,`  
    `String key )`

**18.96.2.12 make\_checkbox()** `QCheckBox * make_checkbox (`  
    `QGroupBox * gb,`  
    `String dictionary,`  
    `const char * label,`  
    `const char * icon,`  
    `String key )`

**18.96.2.13 make\_spinbox()** `QDoubleSpinBox * make_spinbox (`  
    `QGroupBox * gb,`  
    `String dictionary,`  
    `QString object_name,`  
    `EmbReal single_step,`  
    `EmbReal lower,`  
    `EmbReal upper,`  
    `String key )`

**18.96.2.14 make\_ui\_element()** `void make_ui_element (`  
    `Dictionary description )`

**18.96.2.15 node\_bool()** `Node node_bool (`  
    `bool value )`  
`set_node`

**Parameters**

|              |  |
|--------------|--|
| <i>node</i>  |  |
| <i>value</i> |  |

**18.96.2.16 node\_int()** `Node node_int (`  
    `int32_t value )`  
`create_node`

**Parameters**

|             |  |
|-------------|--|
| <i>node</i> |  |
|-------------|--|

**Returns**

**18.96.2.17 node\_qstr()** `Node node_qstr (`  
    `QString value )`  
`set_node`

**Parameters**

|              |  |
|--------------|--|
| <i>node</i>  |  |
| <i>value</i> |  |

**18.96.2.18 node\_real()** `Node node_real (`  
    `EmbReal value )`  
`set_node`

**Parameters**

|              |  |
|--------------|--|
| <i>node</i>  |  |
| <i>value</i> |  |

**18.96.2.19 node\_str()** `Node node_str (`  
    `String value )`  
`set_node`

**Parameters**

|              |  |
|--------------|--|
| <i>node</i>  |  |
| <i>value</i> |  |

**18.96.2.20 node\_str\_list()** `Node node_str_list (`  
`StringList value )`

set\_node

Parameters

|              |  |
|--------------|--|
| <i>node</i>  |  |
| <i>value</i> |  |

**18.96.2.21 node\_uint()** `Node node_uint (`  
`uint32_t value )`

create\_node

Parameters

|             |  |
|-------------|--|
| <i>mode</i> |  |
|-------------|--|

Returns

**18.96.2.22 operator\*()** `EmbVector operator* (`  
`EmbVector v,`  
`EmbReal s )`

operator \*

Parameters

|          |  |
|----------|--|
| <i>v</i> |  |
| <i>s</i> |  |

Returns

**18.96.2.23 operator+()** `EmbVector operator+ (`  
`EmbVector a,`  
`EmbVector b )`

operator + Wrapper for embVector\_add to use the syntax *a + b*.

**18.96.2.24 operator-()** `EmbVector operator- (`  
`EmbVector a,`  
`EmbVector b )`

operator - Wrapper for embVector\_subtract to use the syntax *a - b*.

**18.96.2.25 radians\_\_()** `EmbReal radians__ (`  
`EmbReal degrees )`

radians\_\_

## Parameters

|                |  |
|----------------|--|
| <i>degrees</i> |  |
|----------------|--|

## Returns

**18.96.2.26 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.

**18.96.2.27 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.

**18.96.2.28 to\_EmbVector()** EmbVector to\_EmbVector (  
    QPointF *a* )

**18.96.2.29 to\_qlist()** QList< QGraphicsItem \* > to\_qlist (  
    std::vector< QGraphicsItem \* > *list* )

to\_qlist

## Parameters

|             |  |
|-------------|--|
| <i>list</i> |  |
|-------------|--|

Returns

**18.96.2.30 to\_QPointF()** QPointF to\_QPointF (   
 EmbVector a )

**18.96.2.31 to\_string\_vector()** StringList to\_string\_vector (   
 QStringList list )   
 to\_string\_vector

Parameters

|             |  |
|-------------|--|
| <i>list</i> |  |
|-------------|--|

Returns

**18.96.2.32 to\_vector()** std::vector< QGraphicsItem \* > to\_vector (   
 QList< QGraphicsItem \* > list )   
 to\_vector

Parameters

|             |  |
|-------------|--|
| <i>list</i> |  |
|-------------|--|

Returns

**18.96.2.33 tokenize()** StringList tokenize (   
 String str,   
 const char delim )   
 tokenize

Parameters

|              |  |
|--------------|--|
| <i>str</i>   |  |
| <i>delim</i> |  |

Returns

**18.96.2.34 translate\_str()** QString translate\_str (   
 const char \* str )

## 18.97 src/layer-manager.cpp File Reference

```
#include "embroidermodder.h"
```

### 18.97.1 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/>

## 18.98 src/mainwindow-menus.cpp File Reference

```
#include "embroidermodder.h"
```

### Functions

- void [create\\_menu](#) (std::string menu, [StringList](#) def, bool topLevel)  
*create\_menu*

### 18.98.1 Function Documentation

**18.98.1.1 create\_menu()** void create\_menu (  
std::string menu,  
[StringList](#) def,  
bool topLevel )

create\_menu

#### Parameters

|                 |  |
|-----------------|--|
| <i>menu</i>     |  |
| <i>def</i>      |  |
| <i>topLevel</i> |  |

## 18.99 src/mainwindow-toolbars.cpp File Reference

```
#include "embroidermodder.h"
```

## 18.100 src/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](#)

### 18.100.1 Enumeration Type Documentation

#### 18.100.1.1 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 |  |

#### 18.100.1.2 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      |  |

## Enumerator

|            |  |
|------------|--|
| OBJ_LWT_22 |  |
| OBJ_LWT_23 |  |
| OBJ_LWT_24 |  |

**18.100.1.3 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        |  |

**18.100.2 Function Documentation**

**18.100.2.1 about\_action()** [String](#) about\_action (   
[String](#) args ) [static]

ACTIONS

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

**18.100.2.2 activeScene()** [QGraphicsScene](#) \* activeScene ( )   
 MainWindow::activeScene.

Returns

**18.100.2.3 activeView()** [View](#) \* activeView (   
 void )

activeView

Returns

**18.100.2.4 actuator()** `String` actuator (  
`String line` )

MainWindow::actuator.

Parameters

|                      |  |
|----------------------|--|
| <code>command</code> |  |
|----------------------|--|

**18.100.2.5 RUN COMMAND** `QAction*` act = qobject\_cast<`QAction*`>(sender()); if (act) { prompt->endCommand(); prompt->setCurrentText(act->objectName()); prompt->processInput(); }

**18.100.2.6 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);

**18.100.2.7 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.toUpperCase() + "\")", fileName); }

**18.100.2.8 add\_arc\_action()** `static String` add\_arc\_action (  
`String args` ) [static]  
 add\_arc\_action

Parameters

|                   |  |
|-------------------|--|
| <code>args</code> |  |
|-------------------|--|

Returns

`EmbReal` startX, `EmbReal` startY, `EmbReal` midX, `EmbReal` midY, `EmbReal` endX, `EmbReal` endY, `String` rubberMode

**18.100.2.9 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

**18.100.2.10 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

**18.100.2.11 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

**18.100.2.12 add\_geometry\_action()** static `String` add\_geometry\_action (   
     `String` args ) [static]   
 add\_geometry\_action

## Parameters

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|

## Returns

**18.100.2.13 add\_horizontal\_dimension\_action()** static `String` add\_horizontal\_dimension\_action (   
     `String` args ) [static]   
 EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, EmbReal legHeight

**18.100.2.14 add\_image\_action()** static `String` add\_image\_action (   
     `String` args ) [static]   
 QString img, EmbReal x, EmbReal y, EmbReal w, EmbReal h, EmbReal rot

**18.100.2.15 add\_infinite\_line\_action()** static `String` add\_infinite\_line\_action (   
     `String` args ) [static]   
 EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, EmbReal rot

**18.100.2.16 add\_line\_action()** static `String` add\_line\_action (   
     `String` args ) [static]   
 EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, EmbReal rot, String rubberMode

**18.100.2.17 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

**18.100.2.18 add\_point\_action()** [1/2] static `String` add\_point\_action (   
     `EmbReal` x,   
     `EmbReal` y ) [static]

AddPoint.

## Parameters

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|

## Returns



**18.100.2.19 add\_point\_action()** [2/2] static `String` add\_point\_action (   
`String args` ) [static]   
 add\_point\_action

#### Parameters

|                   |  |
|-------------------|--|
| <code>args</code> |  |
|-------------------|--|

#### Returns

**18.100.2.20 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

**18.100.2.21 add\_polyline\_action()** static `String` add\_polyline\_action (   
`String args` ) [static]

**18.100.2.22 add\_ray\_action()** static `String` add\_ray\_action (   
`String args` ) [static]   
 EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, EmbReal rot

**18.100.2.23 add\_rectangle\_action()** static `String` add\_rectangle\_action (   
`String args` ) [static]   
 add\_rectangle\_action

#### Parameters

|                   |  |
|-------------------|--|
| <code>args</code> |  |
|-------------------|--|

#### Returns

**18.100.2.24 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

**18.100.2.25 add\_rounded\_rectangle\_action()** static `String` add\_rounded\_rectangle\_action (   
 `String args` ) [static]  
add\_rounded\_rectangle\_action

**Parameters**

|                   |  |
|-------------------|--|
| <code>args</code> |  |
|-------------------|--|

**Returns**

EmbReal x, EmbReal y, EmbReal w, EmbReal h, EmbReal rad, EmbReal rot, bool fill

**18.100.2.26 add\_rubber\_action()** `String` add\_rubber\_action (   
 `String args` ) [static]  
add\_rubber\_action

**Parameters**

|                   |  |
|-------------------|--|
| <code>args</code> |  |
|-------------------|--|

**Returns**

**18.100.2.27 add\_slot\_action()** static `String` add\_slot\_action (   
 `String args` ) [static]  
add\_slot\_action

**Parameters**

|                   |  |
|-------------------|--|
| <code>args</code> |  |
|-------------------|--|

**Returns**

EmbReal centerX, EmbReal centerY, EmbReal diameter, EmbReal length, EmbReal rot, bool fill, String rubberMode

**18.100.2.28 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

**18.100.2.29 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

**18.100.2.30 add\_to\_selection\_action()** static `String` add\_to\_selection\_action (   
 `String args` ) [static]  
add\_to\_selection\_action

#### Parameters

|                   |  |
|-------------------|--|
| <code>args</code> |  |
|-------------------|--|

#### Returns

**18.100.2.31 add\_triangle\_action()** static `String` add\_triangle\_action (   
 `String args` ) [static]  
add\_triangle\_action

#### Parameters

|                   |  |
|-------------------|--|
| <code>args</code> |  |
|-------------------|--|

#### Returns

**18.100.2.32 add\_vertical\_dimension\_action()** static `String` add\_vertical\_dimension\_action (   
 `String args` ) [static]  
EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, EmbReal legHeight

**18.100.2.33 alert\_action()** static `String` alert\_action (   
 `String args` ) [static]  
alert\_action

#### Parameters

|                   |  |
|-------------------|--|
| <code>args</code> |  |
|-------------------|--|

#### Returns

**18.100.2.34 allow\_rubber\_action()** static `String` allow\_rubber\_action (   
 `String args` ) [static]  
AllowRubber.

#### Returns

**18.100.2.35 append\_history\_action()** static `String` append\_history\_action (   
 `String args` ) [static]  
append\_history\_action

**Parameters**

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|

**Returns**

**18.100.2.36 append\_prompt\_history\_action()** `String` append\_prompt\_history\_action (   
 `String` *args* ) [static]  
AppendPromptHistory.

**Parameters**

|          |  |
|----------|--|
| <i>a</i> |  |
|----------|--|

**Returns**

**18.100.2.37 blink\_prompt\_action()** `static String` blink\_prompt\_action (   
 `String` *args* ) [static]

**18.100.2.38 calculate\_angle\_action()** `static String` calculate\_angle\_action (   
 `String` *args* ) [static]  
calculate\_angle\_action

**Parameters**

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|

**Returns**

**18.100.2.39 calculate\_distance\_action()** `static String` calculate\_distance\_action (   
 `String` *args* ) [static]  
calculate\_distance

**Parameters**

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|

**Returns**

**18.100.2.40 changelog\_action()** `static String` changelog\_action (   
 `String` *args* ) [static]

changelog\_action

#### Parameters

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|

#### Returns

**18.100.2.41 clear\_rubber\_action()** static `String` clear\_rubber\_action (   
 `String args` ) [static]

ClearRubber.

**18.100.2.42 clear\_selection\_action()** static `String` clear\_selection\_action (   
 `String args` ) [static]

**18.100.2.43 construct\_command()** `String` construct\_command (   
 `String command`,   
 `const char * fmt`,   
 ... )

construct\_command

#### Parameters

|                |  |
|----------------|--|
| <i>command</i> |  |
| <i>fmt</i>     |  |

#### Returns

**18.100.2.44 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.

**18.100.2.45** **copy\_action()** static `String` copy\_action (   
 `String` args ) [static]   
 copy\_action

Parameters

|      |  |
|------|--|
| args |  |
|------|--|

Returns

**18.100.2.46** **copy\_selected\_action()** static `String` copy\_selected\_action (   
 `String` args ) [static]   
 CopySelected x y.

**18.100.2.47** **cut\_action()** static `String` cut\_action (   
 `String` args ) [static]   
 cut\_action

Parameters

|      |  |
|------|--|
| args |  |
|------|--|

Returns

**18.100.2.48** **cut\_selected\_action()** static `String` cut\_selected\_action (   
 `String` args ) [static]   
 CutSelected x y.

**18.100.2.49** **day\_vision\_action()** `String` day\_vision\_action (   
 `String` args ) [static]   
 MainWindow::dayVision.

**Todo** Make day vision color settings.

**18.100.2.50** **debug\_action()** static `String` debug\_action (   
 `String` args ) [static]

**18.100.2.51** **delete\_selected\_action()** static `String` delete\_selected\_action (   
 `String` args ) [static]   
 DeleteSelected.

**18.100.2.52** **design\_details\_action()** `String` design\_details\_action (   
 `String` args ) [static]

**18.100.2.53 disable\_action()** `String disable_action (`  
    `String variable )`  
`disable_action`

#### Parameters

|                 |  |
|-----------------|--|
| <i>variable</i> |  |
|-----------------|--|

#### Returns

**18.100.2.54 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.

**18.100.2.55 end\_action()** `static String end_action (`  
    `String args ) [static]`  
`end_action`

#### Parameters

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|

#### Returns

**18.100.2.56 error\_action()** `String error_action (`  
    `String args ) [static]`  
`Error.`

#### Parameters

|          |  |
|----------|--|
| <i>a</i> |  |
|----------|--|

#### Returns

**18.100.2.57 help\_action()** `static String help_action (`  
    `String args ) [static]`

help\_action

Parameters

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|

Returns

**18.100.2.58 icon\_action()** static `String` icon\_action (   
 `String` *command* ) [static]

icon\_action

Parameters

|                |  |
|----------------|--|
| <i>command</i> |  |
|----------------|--|

Returns

**18.100.2.59 include\_action()** `String` include\_action (   
 `NodeList` *a* )

Include.

Parameters

|          |  |
|----------|--|
| <i>a</i> |  |
|----------|--|

Returns

**18.100.2.60 init\_action()** static `String` init\_action (   
 `String` *args* ) [static]

init\_action

Parameters

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|

Returns

**18.100.2.61 is\_int\_action()** `String` is\_int\_action (   
 `String` *args* )

argument string "i"



**18.100.2.62 layer\_manager\_action()** `String layer_manager_action (`  
    `String args )`  
`layer_manager_action`

#### Parameters

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|

#### Returns

**18.100.2.63 layer\_previous\_action()** `String layer_previous_action (`  
    `String args )`  
`layer_previous_action`

#### Parameters

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|

#### Returns

**18.100.2.64 make\_layer\_active\_action()** `String make_layer_active_action (`  
    `String args )` [static]  
`MainWindow::makeLayerActive.`

#### Returns

**18.100.2.65 messagebox\_action()** `static String messagebox_action (`  
    `String args )` [static]  
`MessageBox type title text.`

**18.100.2.66 mirror\_selected\_action()** `static String mirror_selected_action (`  
    `String args )` [static]  
`MirrorSelected x1 y1 x2 y2.`

**18.100.2.67 mouse\_x\_action()** `static String mouse_x_action (`  
    `String args )` [static]  
`MouseX.`

#### Returns

**18.100.2.68 mouse\_y\_action()** static `String` mouse\_y\_action (   
 `String` args ) [static]

MouseY.

Returns

**18.100.2.69 move\_selected\_action()** static `String` move\_selected\_action (   
 `String` args ) [static]

MoveSelected dx dy.

**18.100.2.70 new\_action()** static `String` new\_action (   
 `String` args ) [static]

new\_action

Parameters

|      |  |
|------|--|
| args |  |
|------|--|

Returns

**18.100.2.71 night\_vision\_action()** `String` night\_vision\_action (   
 `String` args ) [static]

MainWindow::nightVision.

**Todo** Make night vision color settings.

**18.100.2.72 no\_argument\_debug()** void no\_argument\_debug (   
 `String` function\_name,   
 `String` args )

no\_argument\_debug

Parameters

|               |  |
|---------------|--|
| function_name |  |
| args          |  |

**18.100.2.73 num\_selected\_action()** static `String` num\_selected\_action (   
 `String` args ) [static]

NumSelected.

Parameters

|      |  |
|------|--|
| args |  |
|------|--|

Returns

**18.100.2.74 open\_action()** static `String` open\_action (  
    `String` args ) [static]

open\_action

Parameters

|      |  |
|------|--|
| args |  |
|------|--|

Returns

**18.100.2.75 pan\_action()** `String` pan\_action (  
    `String` mode ) [static]

pan\_action

Parameters

|      |  |
|------|--|
| mode |  |
|------|--|

Returns

**18.100.2.76 paste\_action()** static `String` paste\_action (  
    `String` args ) [static]

paste\_action

Parameters

|      |  |
|------|--|
| args |  |
|------|--|

Returns

**18.100.2.77 paste\_selected\_action()** static `String` paste\_selected\_action (  
    `String` args ) [static]

PasteSelected x y.

**18.100.2.78 perpendicular\_distance\_action()** static `String` perpendicular\_distance\_action (  
    `String` args ) [static]

**18.100.2.79 platform\_action()** static `String` platform\_action (   
 `String` args ) [static]   
 platform\_action

**Parameters**

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|

**Returns**

**18.100.2.80 platformString()** `String` platformString (   
 void )   
 platformString

**Returns**

**18.100.2.81 preview\_off\_action()** static `String` preview\_off\_action (   
 `String` args ) [static]   
 PreviewOff.

**18.100.2.82 preview\_on\_action()** [1/2] `String` preview\_on\_action (   
 `String` args ) [static]   
 preview\_on\_action

**Parameters**

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|

**Returns**

**18.100.2.83 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>  |  |

**18.100.2.84 print\_action()** `String print_action (`  
`String args ) [static]`

print\_action

Parameters

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|

Returns

**18.100.2.85 print\_area\_action()** `static String print_area_action (`  
`String args ) [static]`

PrintArea x y w h.

EmbReal x, EmbReal y, EmbReal w, EmbReal h

**18.100.2.86 qsnap\_x\_action()** `static String qsnap_x_action (`  
`String args ) [static]`

QSnapX.

Returns

**18.100.2.87 qsnap\_y\_action()** `static String qsnap_y_action (`  
`String args ) [static]`

QSnapY.

Returns

**18.100.2.88 quit\_action()** `static String quit_action (`  
`String args ) [static]`

quit\_action

Parameters

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|

Returns

**18.100.2.89 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"}.

**18.100.2.90 read\_string\_list\_setting()** `std::vector< String > read_string_list_setting (`  
`toml_table_t * table,`  
`const char * key )`

**18.100.2.91 read\_string\_setting()** `String read_string_setting (`  
`toml_table_t * table,`  
`const char * key )`

**18.100.2.92 redo\_action()** `static String redo_action (`  
`String args ) [static]`  
`redo_action`

#### Parameters

|                   |  |
|-------------------|--|
| <code>args</code> |  |
|-------------------|--|

#### Returns

**18.100.2.93 rotate\_selected\_action()** `static String rotate_selected_action (`  
`String args ) [static]`  
`RotateSelected x y rot.`

**18.100.2.94 rubber\_action()** `static String rubber_action (`  
`String command ) [static]`

**18.100.2.95 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 ';' separates 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

```

**18.100.2.96 run\_script\_file()** `String run_script_file (`  
`String fname )`  
 MainWindow::run\_script\_file.

## Parameters

|              |                                |
|--------------|--------------------------------|
| <i>fname</i> | The path of the script to run. |
|--------------|--------------------------------|

**18.100.2.97 scale\_selected\_action()** `static String scale_selected_action (`  
`String args ) [static]`  
 ScaleSelected *x y factor*.

**18.100.2.98 select\_all\_action()** `static String select_all_action (`  
`String args ) [static]`  
 select\_all\_action

## Parameters

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|

## Returns

**18.100.2.99 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

**18.100.2.100 set\_crosshair\_color\_action() [1/2]** `static String set_crosshair_color_action (`  
`String args ) [static]`  
 argument string "iii"

**18.100.2.101 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> |  |

**18.100.2.102 set\_cursor\_shape\_action()** static `String` set\_cursor\_shape\_action ( `String` *str* ) [static]

**18.100.2.103 set\_grid\_color\_action()** [1/2] `String` set\_grid\_color\_action ( `String` *args* ) [static]  
argument string "iii"

**18.100.2.104 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> |  |
| <i>g</i> |  |
| <i>b</i> |  |

**18.100.2.105 set\_prompt\_prefix\_action()** static `String` set\_prompt\_prefix\_action ( `String` *args* ) [static]  
set\_prompt\_prefix\_action

Parameters

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|

Returns

**18.100.2.106 set\_rubber\_filter\_action()** static `String` set\_rubber\_filter\_action ( `String` *args* ) [static]

**18.100.2.107 set\_rubber\_mode\_action()** static `String` set\_rubber\_mode\_action ( `String` *args* ) [static]

**18.100.2.108 set\_rubber\_point\_action()** static `String` set\_rubber\_point\_action ( `String` *args* ) [static]  
QString key, EmbReal x, EmbReal y

**18.100.2.109 set\_rubber\_text\_action()** `String` set\_rubber\_text\_action ( `String` *args* ) [static]  
set\_rubber\_text\_action

Parameters

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|



## Returns

**18.100.2.110 SetRubberText()** static `String` SetRubberText (   
 `QString key`,   
 `QString txt` ) [static]

**18.100.2.111 SetTextAngle\_action()** `String` SetTextAngle\_action (   
 `String args` )

**18.100.2.112 settings\_dialog\_action()** `String` settings\_dialog\_action (   
 `String showTab` ) [static]   
 settings\_dialog

## Parameters

|                      |  |
|----------------------|--|
| <code>showTab</code> |  |
|----------------------|--|

**18.100.2.113 spare\_rubber\_action()** static `String` spare\_rubber\_action (   
 `String args` ) [static]   
 SpareRubber.

## Parameters

|                     |                 |
|---------------------|-----------------|
| <code>qint64</code> | <code>id</code> |
|---------------------|-----------------|

**18.100.2.114 tip\_of\_the\_day\_action()** `String` tip\_of\_the\_day\_action (   
 `String args` ) [static]   
 tip\_of\_the\_day\_action

## Parameters

|                   |  |
|-------------------|--|
| <code>args</code> |  |
|-------------------|--|

## Returns

**18.100.2.115 todo\_action()** `String` todo\_action (   
 `String args` ) [static]   
 Todo.

## Parameters

|                |  |
|----------------|--|
| <code>a</code> |  |
|----------------|--|

#### Returns

**18.100.2.116 undo\_action()** static `String` undo\_action ( `String` args ) [static]

undo\_action

#### Parameters

|      |  |
|------|--|
| args |  |
|------|--|

#### Returns

**18.100.2.117 validFileFormat()** bool validFileFormat ( `String` fileName )

MainWindow::validFileFormat.

#### Parameters

|          |  |
|----------|--|
| fileName |  |
|----------|--|

#### Returns

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

**18.100.2.118 validRGB()** bool validRGB (   
int r,   
int g,   
int b )

**18.100.2.119 version\_action()** static `String` version\_action ( `String` args ) [static]

version\_action

#### Parameters

|      |  |
|------|--|
| args |  |
|------|--|

#### Returns

**18.100.2.120 vulcanize\_action()** static `String` vulcanize\_action ( `String` args ) [static]

**18.100.2.121** **whats\_this\_action()** `String` whats\_this\_action (  
    `String` args ) [static]  
whats\_this\_action

#### Parameters

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|

#### Returns

**18.100.2.122** **window\_action()** `static String` window\_action (  
    `String` args ) [static]  
window\_action

#### Parameters

|             |  |
|-------------|--|
| <i>args</i> |  |
|-------------|--|

#### Returns

**18.100.2.123** **zoom\_action()** `String` zoom\_action (  
    `String` mode ) [static]  
zoom\_action

#### Parameters

|             |  |
|-------------|--|
| <i>mode</i> |  |
|-------------|--|

#### Returns

### 18.100.3 Variable Documentation

**18.100.3.1** **\_mainWin** `MainWindow*` \_mainWin = 0

**18.100.3.2** **actionHash** `std::unordered_map<String, QAction*>` actionHash

**18.100.3.3** **checkBoxes** `std::unordered_map<String, QCheckBox *>` checkBoxes

**18.100.3.4** **checkBoxTipOfTheDay** `QCheckBox*` checkBoxTipOfTheDay

**18.100.3.5 comboBoxes** `std::unordered_map<String, QComboBox *> comboBoxes`

**18.100.3.6 command\_map** `std::unordered_map<String, Command> command_map`

**18.100.3.7 config** `Dictionary config`

**18.100.3.8 config\_tables** `std::unordered_map<String, Dictionary> config_tables`

**18.100.3.9 dialog** `Dictionary dialog`

**18.100.3.10 dockPropEdit** `PropertyEditor* dockPropEdit = 0`

**18.100.3.11 dockUndoEdit** `UndoEditor* dockUndoEdit = 0`

**18.100.3.12 doubleSpinBoxes** `std::unordered_map<String, QDoubleSpinBox *> doubleSpinBoxes`

**18.100.3.13 groupBoxes** `std::unordered_map<String, QGroupBox *> groupBoxes`

**18.100.3.14 labels** `std::unordered_map<String, QLabel *> labels`

**18.100.3.15 labelTipOfTheDay** `QLabel* labelTipOfTheDay`

**18.100.3.16 lineEdits** `std::unordered_map<String, QLineEdit *> lineEdits`

**18.100.3.17 mdiArea** `MdiArea* mdiArea = 0`

**18.100.3.18 menuHash** `std::unordered_map<String, QMenu*> menuHash`

**18.100.3.19 prompt** `CmdPrompt* prompt = 0`

**18.100.3.20 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"
}

```

**18.100.3.21 scripts** `std::unordered_map<String, StringList> scripts`

**18.100.3.22 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.

**18.100.3.23 spinBoxes** `std::unordered_map<String, QSpinBox *> spinBoxes`

**18.100.3.24 statusBar** `StatusBar* statusbar = 0`

**18.100.3.25 subMenuHash** `std::unordered_map<String, QMenu*> subMenuHash`

**18.100.3.26 toolbarHash** `std::unordered_map<String, QToolBar*> toolbarHash`

**18.100.3.27 toolButtons** `std::unordered_map<String, QToolButton *> toolButtons`

**18.100.3.28 wizardTipOfTheDay** `QWizard* wizardTipOfTheDay`

## 18.101 src/mdiarea.cpp File Reference

```
#include "embroidermodder.h"
```

## 18.102 src/mdiwindow.cpp File Reference

```
#include "embroidermodder.h"
```

### Functions

- `QString fileExtension (String fileName)`  
*MdiWindow::fileExtension.*

### 18.102.1 Function Documentation

**18.102.1.1 fileExtension()** `QString fileExtension (String fileName)`

*MdiWindow::fileExtension.*

## Parameters

|                 |  |
|-----------------|--|
| <i>fileName</i> |  |
|-----------------|--|

## Returns

## 18.103 src/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)  
*fourier\_series*
- void [add\\_polyline](#) (QPainterPath p, [String](#) rubberMode)  
*add\_polyline*
- [EmbVector](#) [rotate\\_vector](#) ([EmbVector](#) v, [EmbReal](#) alpha)

## 18.103.1 Function Documentation

**18.103.1.1 add\_polyline()** void add\_polyline (  
    QPainterPath *p*,  
    [String](#) *rubberMode* )  
add\_polyline

## Parameters

|                   |  |
|-------------------|--|
| <i>p</i>          |  |
| <i>rubberMode</i> |  |

**18.103.1.2 closest\_point()** QPointF closest\_point (  
    QPointF *position*,  
    std::vector< QPointF > *points* )  
mouse\_snap\_point

## Parameters

|               |  |
|---------------|--|
| <i>points</i> |  |
|---------------|--|

## Returns

**18.103.1.3 fourier\_series()** [EmbReal](#) fourier\_series (  
    [EmbReal](#) *arg*,  
    std::vector< [EmbReal](#) > *terms* )

fourier\_series

#### Parameters

|              |  |
|--------------|--|
| <i>arg</i>   |  |
| <i>terms</i> |  |

**18.103.1.4 rotate\_vector()** `EmbVector rotate_vector (`  
     `EmbVector v,`  
     `EmbReal alpha )`

Returns

## 18.104 src/preview-dialog.cpp File Reference

```
#include "embroidermodder.h"
```

## 18.105 src/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`

### 18.105.1 Function Documentation

**18.105.1.1 load\_group\_box\_data\_from\_table()** `std::vector< Dictionary > load_group_box_data_↵`  
`from_table (`  
     `String key )`

### 18.105.2 Variable Documentation

**18.105.2.1 comboBoxTextSingleFont** `QFontComboBox* comboBoxTextSingleFont`

**18.105.2.2 fieldNewText** `QString fieldNewText`

**18.105.2.3 fieldNoText** `QString fieldNoText`

**18.105.2.4 fieldOffText** `QString fieldOffText`

**18.105.2.5 fieldOldText** `QString fieldOldText`

**18.105.2.6 fieldOnText** `QString fieldOnText`

**18.105.2.7 fieldVariesText** `QString fieldVariesText`

**18.105.2.8 fieldYesText** `QString fieldYesText`

**18.105.2.9 group\_box\_data** `std::unordered_map<String, Dictionary> group_box_data`

**18.105.2.10 group\_box\_types** `std::vector<std::pair<String, int> > group_box_types`

**18.105.2.11 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"
}
```

**18.106 src/README.md File Reference**

**18.107 src/selectbox.cpp File Reference**

```
#include "embroidermodder.h"
```



## 18.108 src/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

### 18.108.1 Function Documentation

**18.108.1.1 [make\\_editing\\_copy\(\)](#)** void [make\\_editing\\_copy](#) (  
[StringList](#) props )

**18.108.1.2 [read\\_settings\(\)](#)** void [read\\_settings](#) (  
void )

[read\\_settings](#)

This file needs to be read from the users home directory to ensure it is writable.

**18.108.1.3 [write\\_settings\(\)](#)** void [write\\_settings](#) (  
void )

[MainWindow::writeSettings.](#)

This file needs to be read from the users home directory to ensure it is writable

### 18.108.2 Variable Documentation

**18.108.2.1 [accept\\_](#)** [Dictionary](#) [accept\\_](#)

**18.108.2.2 [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"
}

```

### 18.108.2.3 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",
 "EXY", "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"
}

```

### 18.108.2.4 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"
}

```

### 18.108.2.5 opensave\_props `StringList` opensave\_props

Initial value:

```

= {
 "opensave_custom_filter"
}

```

### 18.108.2.6 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.

### 18.108.2.7 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"
}

```

**18.108.2.8 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"
}

```

**18.109 src/statusbar.cpp File Reference**

```
#include "embroidermodder.h"
```

**18.110 src/undo-commands.cpp File Reference**

```
#include "embroidermodder.h"
```

**18.111 src/undo-editor.cpp File Reference**

```
#include "embroidermodder.h"
```

**18.111.1 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/>

**18.112 src/view.cpp File Reference**

```

#include "embroidermodder.h"
#include <cassert>
#include <QtOpenGL>

```

**Functions**

- bool `contains` (`StringList` list, `String` entry)

**18.112.1 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/>

**18.112.2 Function Documentation**

**18.112.2.1 contains()** `bool contains (`  
    `StringList list,`  
    `String entry )`

## References

- [1] acatina. Technical info. [343](#)
- [2] KDE Community. Projects/liberty/file formats/tajima ternary - kde community wiki. [343](#)
- [3] G. van Rossum and B. Warsaw. Python pep 7. [12](#)



## Index

- `__mainWin`
    - Application, [53](#)
- `_appVer_`
  - embroidermodder.cpp, [411](#)
- `_bcf_directory,` [43](#)
  - dirEntries, [43](#)
  - maxNumberOfDirectoryEntries, [43](#)
- `_bcf_directory_entry,` [43](#)
  - childId, [44](#)
  - CLSID, [44](#)
  - colorFlag, [44](#)
  - creationTime, [44](#)
  - directoryEntryName, [44](#)
  - directoryEntryNameLength, [44](#)
  - leftSiblingId, [44](#)
  - modifiedTime, [44](#)
  - next, [44](#)
  - objectType, [44](#)
  - rightSiblingId, [44](#)
  - startingSectorLocation, [45](#)
  - stateBits, [45](#)
  - streamSize, [45](#)
  - streamSizeHigh, [45](#)
- `_bcf_file,` [45](#)
  - difat, [45](#)
  - directory, [45](#)
  - fat, [45](#)
  - header, [46](#)
- `_bcf_file_difat,` [46](#)
  - fatSectorCount, [46](#)
  - fatSectorEntries, [46](#)
  - sectorSize, [46](#)
- `_bcf_file_fat,` [46](#)
  - fatEntries, [47](#)
  - fatEntryCount, [47](#)
  - numberOfEntriesInFatSector, [47](#)
- `_bcf_file_header,` [47](#)
  - byteOrder, [48](#)
  - CLSID, [48](#)
  - firstDifatSectorLocation, [48](#)
  - firstDirectorySectorLocation, [48](#)
  - firstMiniFATSectorLocation, [48](#)
  - majorVersion, [48](#)
  - miniSectorShift, [48](#)
  - miniStreamCutoffSize, [48](#)
  - minorVersion, [48](#)
  - numberOfDifatSectors, [48](#)
  - numberOfDirectorySectors, [49](#)
  - numberOfFAT Sectors, [49](#)
  - numberOfMiniFatSectors, [49](#)
  - reserved1, [49](#)
  - reserved2, [49](#)
  - sectorShift, [49](#)
  - signature, [49](#)
  - transactionSignatureNumber, [49](#)
- `_dxfColorTable`
  - embroidery.h, [268](#)
  - thread-color.c, [409](#)
- `_mainWin`
  - embroidermodder.h, [429](#)
  - mainwindow.cpp, [485](#)
- `_subMask`
  - format\_csd.c, [340](#)
- `_vp3Hoop,` [49](#)
  - bottom, [50](#)
  - bottom2, [50](#)
  - byte1, [50](#)
  - byte2, [50](#)
  - byte3, [50](#)
  - height, [50](#)
  - left, [51](#)
  - left2, [51](#)
  - numberOfBytesRemaining, [51](#)
  - numberOfColors, [51](#)
  - right, [51](#)
  - right2, [51](#)
  - threadLength, [51](#)
  - top, [51](#)
  - top2, [51](#)
  - unknown2, [51](#)
  - unknown3, [51](#)
  - unknown4, [52](#)
  - width, [52](#)
  - xOffset, [52](#)
  - yOffset, [52](#)
- `_xorMask`
  - format\_csd.c, [340](#)
- `~CmdPrompt`
  - CmdPrompt, [55](#)
- `~CmdPromptHandle`
  - CmdPromptHandle, [63](#)
- `~CmdPromptHistory`
  - CmdPromptHistory, [66](#)
- `~CmdPromptInput`
  - CmdPromptInput, [70](#)
- `~CmdPromptSplitter`
  - CmdPromptSplitter, [77](#)
- `~EmbDetailsDialog`
  - EmbDetailsDialog, [85](#)
- `~Geometry`
  - Geometry, [114](#)
- `~ImageWidget`
  - ImageWidget, [143](#)
- `~LayerManager`
  - LayerManager, [144](#)
- `~MainWindow`
  - MainWindow, [150](#)
- `~MdiArea`
  - MdiArea, [166](#)
- `~MdiWindow`

- MdiWindow, 171
- ~PreviewDialog
  - PreviewDialog, 180
- ~PropertyEditor
  - PropertyEditor, 182
- ~SaveObject
  - SaveObject, 188
- ~Settings\_Dialog
  - Settings\_Dialog, 201
- ~UndoEditor
  - UndoEditor, 216
- ~View
  - View, 220
- 10o, 9, 335
- 100, 9, 335
- about
  - MainWindow, 150
- about\_action
  - mainwindow.cpp, 464
- accept\_
  - settings-dialog.cpp, 491
- acceptChanges
  - Settings\_Dialog, 202
- actionHash
  - embroidermodder.h, 429
  - mainwindow.cpp, 485
- activeCommand
  - MainWindow, 151
- activeMdiWindow
  - MainWindow, 151
- activeScene
  - embroidermodder.h, 420
  - mainwindow.cpp, 464
- activeUndoStack
  - MainWindow, 151
- activeView
  - embroidermodder.h, 420
  - mainwindow.cpp, 464
- actuator
  - embroidermodder.h, 421
  - mainwindow.cpp, 464
- add\_arc\_action
  - mainwindow.cpp, 465
- add\_circle\_action
  - mainwindow.cpp, 465
- add\_dim\_leader\_action
  - mainwindow.cpp, 465
- add\_ellipse\_action
  - mainwindow.cpp, 465
- add\_geometry\_action
  - mainwindow.cpp, 466
- add\_horizontal\_dimension\_action
  - mainwindow.cpp, 466
- add\_image\_action
  - mainwindow.cpp, 466
- add\_infinite\_line\_action
  - mainwindow.cpp, 466
- add\_line\_action
  - mainwindow.cpp, 466
- add\_path\_action
  - mainwindow.cpp, 466
- add\_point\_action
  - mainwindow.cpp, 466
- add\_polygon\_action
  - mainwindow.cpp, 467
- add\_polyline
  - embroidermodder.h, 421
  - objects.cpp, 488
- add\_polyline\_action
  - mainwindow.cpp, 467
- add\_ray\_action
  - mainwindow.cpp, 467
- add\_rectangle\_action
  - mainwindow.cpp, 467
- add\_regular\_polygon\_action
  - mainwindow.cpp, 467
- add\_rounded\_rectangle\_action
  - mainwindow.cpp, 467
- add\_rubber\_action
  - mainwindow.cpp, 468
- add\_slot\_action
  - mainwindow.cpp, 468
- add\_text\_multi\_action
  - mainwindow.cpp, 468
- add\_text\_single\_action
  - mainwindow.cpp, 468
- add\_to\_path
  - embroidermodder.h, 421
  - interface.cpp, 450
- add\_to\_selection\_action
  - mainwindow.cpp, 468
- add\_triangle\_action
  - mainwindow.cpp, 469
- add\_vertical\_dimension\_action
  - mainwindow.cpp, 469
- addArc
  - SaveObject, 189
- addBlock
  - SaveObject, 189
- addCircle
  - SaveObject, 189
- addColorsToComboBox
  - Settings\_Dialog, 202
- addDimAligned
  - SaveObject, 189
- addDimAngular
  - SaveObject, 190
- addDimArcLength
  - SaveObject, 190
- addDimDiameter
  - SaveObject, 190
- addDimLeader
  - SaveObject, 190
- addDimLinear
  - SaveObject, 191



- addDimOrdinate
  - SaveObject, [191](#)
- addDimRadius
  - SaveObject, [191](#)
- addEllipse
  - SaveObject, [192](#)
- addEllipseArc
  - SaveObject, [192](#)
- addGrid
  - SaveObject, [192](#)
- addHatch
  - SaveObject, [192](#)
- addImage
  - SaveObject, [193](#)
- addInfiniteLine
  - SaveObject, [193](#)
- addLayer
  - LayerManager, [145](#)
- addLine
  - SaveObject, [193](#)
- addObject
  - View, [220](#)
- addPath
  - SaveObject, [193](#)
- addPoint
  - SaveObject, [194](#)
- addPolygon
  - SaveObject, [194](#)
- addPolyline
  - SaveObject, [194](#)
- addRay
  - SaveObject, [195](#)
- addRectangle
  - SaveObject, [195](#)
- addSlot
  - SaveObject, [195](#)
- addSpline
  - SaveObject, [195](#)
- addStack
  - UndoEditor, [216](#)
- addTextMulti
  - SaveObject, [196](#)
- addTextSingle
  - SaveObject, [196](#)
- addToRubberRoom
  - View, [220](#)
- after
  - UndoableCommand, [214](#)
- alert
  - CmdPrompt, [56](#)
- alert\_action
  - mainwindow.cpp, [469](#)
- alignScenePointWithViewPoint
  - View, [220](#)
- allGripPoints
  - Geometry, [114](#)
- allow\_rubber\_action
  - mainwindow.cpp, [469](#)
- allowRubber
  - View, [221](#)
- allowZoomIn
  - View, [221](#)
- allowZoomOut
  - View, [221](#)
- alpha
  - SelectBox, [199](#)
- alphabet
  - LSYSTEM, [146](#)
- Ameco, [305](#), [356](#)
- angle
  - UndoableCommand, [214](#)
- append\_history\_action
  - mainwindow.cpp, [469](#)
- append\_prompt\_history\_action
  - mainwindow.cpp, [470](#)
- appendHistory
  - CmdPrompt, [56](#)
  - CmdPromptHistory, [66](#)
  - CmdPromptInput, [70](#)
- appendTheHistory
  - CmdPrompt, [56](#)
- Application, [52](#)
  - \_\_mainWin, [53](#)
  - Application, [52](#)
  - event, [53](#)
  - setMainWin, [53](#)
- applyFormatting
  - CmdPromptHistory, [67](#)
  - CmdPromptInput, [70](#)
- arc
  - EmbGeometry\_, [90](#)
- arc.c
  - Arc\_clockwise, [379](#)
  - Base\_objectRubberPoint, [379](#)
  - Base\_objectRubberText, [380](#)
  - Base\_setLineType, [380](#)
  - Base\_setLineWeight, [380](#)
  - clockwise, [380](#)
  - embArc\_arcLength, [380](#)
  - embArc\_area, [380](#)
  - embArc\_chord, [380](#)
  - embArc\_clockwise, [380](#)
  - embArc\_endAngle, [380](#)
  - embArc\_gripEdit, [380](#)
  - embArc\_includedAngle, [380](#)
  - embArc\_init, [380](#)
  - embArc\_mouseSnapPoint, [381](#)
  - embArc\_paint, [381](#)
  - embArc\_setCenter, [381](#)
  - embArc\_setEndAngle, [381](#)
  - embArc\_setRadius, [381](#)
  - embArc\_setStartAngle, [381](#)
  - embArc\_startAngle, [381](#)
  - embArc\_updatePath, [381](#)
  - embArc\_updateRubber, [381](#)
  - embBase\_setColorRGB, [381](#)

- embCircle\_prompt, 381
- embCircle\_setArea, 381
- embCircle\_setCircumference, 382
- embEllipse\_click, 382
- embEllipse\_main, 382
- embRect\_bottomLeft, 382
- embRect\_bottomRight, 382
- getArcCenter, 382
- getArcDataFromBulge, 382
- set\_object\_color, 382
- Arc\_clockwise
  - arc.c, 379
- Arc\_Polyester
  - embroidery.h, 243
- Arc\_Rayon
  - embroidery.h, 243
- arcEndPoint
  - Geometry, 138
- arcMidPoint
  - Geometry, 138
- arcStartPoint
  - Geometry, 138
- array.c
  - embArray\_addArc, 232
  - embArray\_addCircle, 232
  - embArray\_addEllipse, 232
  - embArray\_addFlag, 232
  - embArray\_addLine, 233
  - embArray\_addPath, 233
  - embArray\_addPoint, 233
  - embArray\_addPolygon, 233
  - embArray\_addPolyline, 233
  - embArray\_addRect, 233
  - embArray\_addStitch, 233
  - embArray\_addVector, 233
  - embArray\_copy, 233
  - embArray\_create, 233
  - embArray\_free, 233
  - embArray\_resize, 234
- ArrowStyle
  - Geometry, 110
- arrowStyleAngle
  - Geometry, 138
- arrowStyleLength
  - Geometry, 138
- arrowStylePath
  - Geometry, 138
- art, 9, 336
- attributeList
  - format\_svg.c, 369
- attributeOffset
  - VipHeader\_, 231
- AutoCAD, 308, 346, 365
- AutoDesk, 346
- auxFormat
  - ThredExtension\_, 211
- axiom
  - LSYSTEM, 146
- b
  - EmbColor\_, 84
  - Node\_, 179
  - Barudan, 309, 342, 373
  - Base\_objectRubberPoint
    - arc.c, 379
  - Base\_objectRubberText
    - arc.c, 380
  - Base\_setLineType
    - arc.c, 380
  - Base\_setLineWeight
    - arc.c, 380
  - bcf\_difat\_create
    - embroidery\_internal.h, 293
    - main.c, 397
  - bcf\_directory
    - embroidery\_internal.h, 292
  - bcf\_directory\_entry
    - embroidery\_internal.h, 292
  - bcf\_directory\_free
    - embroidery\_internal.h, 294
    - main.c, 397
  - bcf\_file
    - embroidery\_internal.h, 292
  - bcf\_file\_difat
    - embroidery\_internal.h, 292
  - bcf\_file\_difat\_free
    - embroidery\_internal.h, 294
  - bcf\_file\_fat
    - embroidery\_internal.h, 292
  - bcf\_file\_fat\_free
    - embroidery\_internal.h, 294
  - bcf\_file\_free
    - embroidery\_internal.h, 294
    - main.c, 397
  - bcf\_file\_header
    - embroidery\_internal.h, 292
  - bcfFile\_read
    - embroidery\_internal.h, 294
    - main.c, 397
  - bcfFileFat\_create
    - embroidery\_internal.h, 294
    - main.c, 397
  - bcfFileHeader\_isValid
    - embroidery\_internal.h, 294
  - bcfFileHeader\_read
    - embroidery\_internal.h, 294
    - main.c, 397
  - before
    - UndoableCommand, 214
  - Bernina, 336
  - beziers
    - EmbSpline\_, 101
  - bgColor
    - MdiArea, 168
  - bgLogo
    - MdiArea, 168
  - bgTexture

- MdiArea, 168
- binaryReadString
  - embroidery\_internal.h, 294
  - main.c, 397
- binaryReadUnicodeString
  - embroidery\_internal.h, 295
  - main.c, 398
- binaryWriteInt
  - embroidery\_internal.h, 295
  - formats.c, 332
- binaryWriteIntBE
  - embroidery\_internal.h, 295
  - formats.c, 332
- binaryWriteShort
  - embroidery\_internal.h, 295
  - formats.c, 332
- binaryWriteUInt
  - embroidery\_internal.h, 295
  - formats.c, 332
- binaryWriteUIntBE
  - embroidery\_internal.h, 295
  - formats.c, 332
- binaryWriteUShort
  - embroidery\_internal.h, 295
  - formats.c, 332
- binaryWriteUShortBE
  - embroidery\_internal.h, 296
  - formats.c, 332
- bit\_position
  - Compress, 78
- Bitmap Cache, 337
- Bits and Volts, 337
- bits\_total
  - Compress, 78
- black\_thread
  - embroidery.h, 268
  - main.c, 403
- blink
  - CmdPrompt, 56
- blink\_prompt\_action
  - mainwindow.cpp, 470
- blinkState
  - CmdPrompt, 62
- blinkTimer
  - CmdPrompt, 62
- block\_elements
  - Compress, 78
- bmc, 337
- BOOL\_TYPE
  - embroidermodder.h, 418
- bottom
  - \_vp3Hoop, 50
  - EmbRect\_, 99
  - hoop\_padding, 141
- bottom2
  - \_vp3Hoop, 50
- boundingRect
  - EmbDetailsDialog, 86
- Geometry, 114
- Box
  - Geometry, 110
- boxDir
  - SelectBox, 199
- brand\_codes
  - thread-color.c, 409
- brand\_codes\_files
  - thread-color.c, 409
- bro, 9, 337
- Brother, 306, 307, 360, 361, 364, 365
- BuildDecryptionTable
  - format\_csd.c, 339
- BULGETOCONTROL
  - embroidery\_internal.h, 284
- BULGETOEND
  - embroidery\_internal.h, 284
- buttonBox
  - EmbDetailsDialog, 86
  - Settings\_Dialog, 207
- buttonCustomFilterClearAll
  - Settings\_Dialog, 202
- buttonCustomFilterClearAllClicked
  - Settings\_Dialog, 202
- buttonCustomFilterSelectAll
  - Settings\_Dialog, 202
- buttonCustomFilterSelectAllClicked
  - Settings\_Dialog, 202
- buttonQSnapClearAll
  - Settings\_Dialog, 202
- buttonQSnapClearAllClicked
  - Settings\_Dialog, 202
- buttonQSnapSelectAll
  - Settings\_Dialog, 202
- buttonQSnapSelectAllClicked
  - Settings\_Dialog, 202
- buttons
  - StatusBar, 209
- buttonTipOfTheDayClicked
  - MainWindow, 151
- byte1
  - \_vp3Hoop, 50
- byte2
  - \_vp3Hoop, 50
- byte3
  - \_vp3Hoop, 50
- byteOrder
  - \_bcf\_file\_header, 48
- calculate\_angle\_action
  - mainwindow.cpp, 470
- calculate\_distance\_action
  - mainwindow.cpp, 470
- calculateArcData
  - Geometry, 115
- canRedo
  - UndoEditor, 216
- canUndo
  - UndoEditor, 216

- cascade
  - MdiArea, 166
- catalogNumber
  - EmbThread\_, 103
- cci
  - format\_dst.c, 344
- center
  - EmbCircle\_, 83
  - EmbEllipse\_, 88
  - View, 221
- centerAt
  - View, 221
- changeFormatting
  - CmdPromptInput, 70
- changelog\_action
  - mainwindow.cpp, 470
- character\_huffman
  - Compress, 78
- character\_length\_huffman
  - Compress, 78
- check\_for\_color\_file
  - EmbFormatList\_, 88
- check\_header\_present
  - embroidery\_internal.h, 296
  - main.c, 398
- checkBoxCustomFilterStateChanged
  - Settings\_Dialog, 202
- checkBoxes
  - embroidermodder.h, 429
  - mainwindow.cpp, 485
- checkBoxGeneralMdiBGUseColorStateChanged
  - Settings\_Dialog, 202
- checkBoxGeneralMdiBGUseLogoStateChanged
  - Settings\_Dialog, 202
- checkBoxGeneralMdiBGUseTextureStateChanged
  - Settings\_Dialog, 202
- checkBoxGridCenterOnOriginStateChanged
  - Settings\_Dialog, 203
- checkBoxGridColorMatchCrossHairStateChanged
  - Settings\_Dialog, 203
- checkBoxGridLoadFromFileStateChanged
  - Settings\_Dialog, 203
- checkBoxLwtRealRenderStateChanged
  - Settings\_Dialog, 203
- checkBoxLwtShowLwtStateChanged
  - Settings\_Dialog, 203
- checkBoxPromptSaveHistoryAsHtmlStateChanged
  - Settings\_Dialog, 203
- checkBoxRulerShowOnLoadStateChanged
  - Settings\_Dialog, 203
- checkBoxShowScrollBarsStateChanged
  - Settings\_Dialog, 203
- checkBoxTipOfTheDay
  - mainwindow.cpp, 485
- checkChangedText
  - CmdPromptInput, 71
- checkCursorPosition
  - CmdPromptInput, 71
- checkEditedText
  - CmdPromptInput, 71
- checkForUpdates
  - MainWindow, 151
- checkSelection
  - CmdPromptInput, 71
- childId
  - \_bcf\_directory\_entry, 44
- chooseDisplayBackgroundColor
  - Settings\_Dialog, 203
- chooseDisplayCrossHairColor
  - Settings\_Dialog, 203
- chooseDisplaySelectBoxLeftColor
  - Settings\_Dialog, 203
- chooseDisplaySelectBoxLeftFill
  - Settings\_Dialog, 203
- chooseDisplaySelectBoxRightColor
  - Settings\_Dialog, 204
- chooseDisplaySelectBoxRightFill
  - Settings\_Dialog, 204
- chooseGeneralMdiBackgroundColor
  - Settings\_Dialog, 204
- chooseGeneralMdiBackgroundLogo
  - Settings\_Dialog, 204
- chooseGeneralMdiBackgroundTexture
  - Settings\_Dialog, 204
- chooseGridColor
  - Settings\_Dialog, 204
- choosePromptBackgroundColor
  - Settings\_Dialog, 204
- choosePromptTextColor
  - Settings\_Dialog, 204
- chooseRulerColor
  - Settings\_Dialog, 204
- CHUNK\_SIZE
  - embroidery.h, 243
- circle
  - EmbGeometry\_, 90
- circle.c
  - embCircle\_area, 383
  - embCircle\_circumference, 383
  - embCircle\_init, 383
  - getCircleCircleIntersections, 383
  - getCircleTangentPoints, 383
- circle\_click
  - Geometry, 115
- clear\_rubber\_action
  - mainwindow.cpp, 471
- clear\_selection\_action
  - mainwindow.cpp, 471
- clearAllFields
  - PropertyEditor, 182
- clearFormatting
  - CmdPromptInput, 71
- clearRubberRoom
  - View, 221
- clearSelection
  - View, 221

- clockwise
  - arc.c, 380
- Closed
  - Geometry, 110
- closeEvent
  - MainWindow, 151
  - MdiWindow, 171
- closest\_point
  - objects.cpp, 488
- closeToolBar
  - MainWindow, 152
- CLSID
  - \_bcf\_directory\_entry, 44
  - \_bcf\_file\_header, 48
- cmdActive
  - CmdPromptInput, 75
- CmdPrompt, 53
  - ~CmdPrompt, 55
  - alert, 56
  - appendHistory, 56
  - appendTheHistory, 56
  - blink, 56
  - blinkState, 62
  - blinkTimer, 62
  - CmdPrompt, 55
  - copyPressed, 56
  - cutPressed, 56
  - deletePressed, 57
  - downPressed, 57
  - escapePressed, 57
  - F10Pressed, 57
  - F11Pressed, 57
  - F12Pressed, 57
  - F1Pressed, 57
  - F2Pressed, 57
  - F3Pressed, 57
  - F4Pressed, 57
  - F5Pressed, 57
  - F6Pressed, 58
  - F7Pressed, 58
  - F8Pressed, 58
  - F9Pressed, 58
  - floatingChanged, 58
  - historyAppended, 58
  - pastePressed, 58
  - promptDivider, 62
  - promptHistory, 62
  - promptInput, 62
  - promptSplitter, 62
  - promptVBoxLayout, 62
  - redoPressed, 58
  - runCommand, 58
  - saveHistory, 59
  - selectAllPressed, 59
  - setCurrentText, 59
  - setHistory, 59
  - setPrefix, 59
  - setPromptBackgroundColor, 59
  - setPromptFontFamily, 60
  - setPromptFontSize, 60
  - setPromptFontStyle, 60
  - setPromptTextColor, 60
  - shiftPressed, 61
  - shiftReleased, 61
  - showSettings, 61
  - startBlinking, 61
  - startCommand, 61
  - stopBlinking, 61
  - styleHash, 62
  - tabPressed, 61
  - undoPressed, 61
  - updateStyle, 61
  - upPressed, 61
- CmdPromptHandle, 62
  - ~CmdPromptHandle, 63
  - CmdPromptHandle, 63
  - handleMoved, 64
  - handlePressed, 64
  - handleReleased, 64
  - mouseMoveEvent, 64
  - mousePressEvent, 64
  - mouseReleaseEvent, 64
  - moveY, 65
  - pressY, 65
  - releaseY, 65
- CmdPromptHistory, 65
  - ~CmdPromptHistory, 66
  - appendHistory, 66
  - applyFormatting, 67
  - CmdPromptHistory, 66
  - contextMenuEvent, 67
  - historyAppended, 67
  - resizeHistory, 67
  - startResizeHistory, 68
  - stopResizeHistory, 68
  - tmpHeight, 68
- CmdPromptInput, 68
  - ~CmdPromptInput, 70
  - appendHistory, 70
  - applyFormatting, 70
  - changeFormatting, 70
  - checkChangedText, 71
  - checkCursorPosition, 71
  - checkEditedText, 71
  - checkSelection, 71
  - clearFormatting, 71
  - cmdActive, 75
  - CmdPromptInput, 70
  - contextMenuEvent, 72
  - copyClip, 72
  - copyPressed, 72
  - curCmd, 75
  - curText, 76
  - cutPressed, 72
  - defaultPrefix, 76
  - deletePressed, 72

- downPressed, [72](#)
- endCommand, [72](#)
- escapePressed, [72](#)
- eventFilter, [72](#)
- F10Pressed, [73](#)
- F11Pressed, [73](#)
- F12Pressed, [73](#)
- F1Pressed, [73](#)
- F2Pressed, [73](#)
- F3Pressed, [73](#)
- F4Pressed, [73](#)
- F5Pressed, [73](#)
- F6Pressed, [73](#)
- F7Pressed, [73](#)
- F8Pressed, [74](#)
- F9Pressed, [74](#)
- isBlinking, [76](#)
- lastCmd, [76](#)
- pasteClip, [74](#)
- pastePressed, [74](#)
- prefix, [76](#)
- processInput, [74](#)
- rapidFireEnabled, [76](#)
- redoPressed, [74](#)
- runCommand, [74](#)
- selectAllPressed, [74](#)
- shiftPressed, [74](#)
- shiftReleased, [74](#)
- showSettings, [75](#)
- startCommand, [75](#)
- stopBlinking, [75](#)
- tabPressed, [75](#)
- undoPressed, [75](#)
- updateCurrentText, [75](#)
- upPressed, [75](#)
- CmdPromptSplitter, [76](#)
  - ~CmdPromptSplitter, [77](#)
  - CmdPromptSplitter, [77](#)
  - createHandle, [77](#)
  - moveResizeHistory, [77](#)
  - pressResizeHistory, [77](#)
  - releaseResizeHistory, [78](#)
- cnd, [9](#), [338](#)
- CoatsAndClark\_Rayon
  - embroidery.h, [243](#)
- CODE\_OF\_CONDUCT.md, [232](#)
- col, [9](#), [338](#)
- color
  - EmbGeometry\_, [90](#)
  - EmbLine\_, [94](#)
  - EmbPath\_, [96](#)
  - EmbPoint\_, [98](#)
  - EmbStitch\_, [101](#)
  - EmbThread\_, [103](#)
- color\_only
  - EmbFormatList\_, [88](#)
- colorChanges
  - EmbDetailsDialog, [86](#)
- colorCode
  - StxThread\_, [209](#)
  - SubDescriptor\_, [210](#)
- colorFlag
  - \_bcf\_directory\_entry, [44](#)
- colorLength
  - VipHeader\_, [231](#)
- colorName
  - StxThread\_, [209](#)
  - SubDescriptor\_, [210](#)
- colorSelector
  - MainWindow, [163](#)
- colorSelectorIndexChanged
  - MainWindow, [152](#)
- colorTotal
  - EmbDetailsDialog, [86](#)
- comboBoxes
  - embroidermodder.h, [429](#)
  - mainwindow.cpp, [485](#)
- comboBoxGridTypeCurrentIndexChanged
  - Settings\_Dialog, [204](#)
- comboBoxIconSizeCurrentIndexChanged
  - Settings\_Dialog, [204](#)
- comboBoxPromptFontFamilyCurrentIndexChanged
  - Settings\_Dialog, [205](#)
- comboBoxPromptFontStyleCurrentIndexChanged
  - Settings\_Dialog, [205](#)
- comboBoxQSnapLocatorColorCurrentIndexChanged
  - Settings\_Dialog, [205](#)
- comboBoxRulerMetricCurrentIndexChanged
  - Settings\_Dialog, [205](#)
- comboBoxScrollBarWidgetCurrentIndexChanged
  - Settings\_Dialog, [205](#)
- comboBoxSelected
  - PropertyEditor, [185](#)
- comboBoxSelectionCoolGripColorCurrentIndexChanged
  - Settings\_Dialog, [205](#)
- comboBoxSelectionHotGripColorCurrentIndexChanged
  - Settings\_Dialog, [205](#)
- comboBoxTextSingleFont
  - property-editor.cpp, [489](#)
- Command
  - embroidermodder.h, [419](#)
- command
  - UndoableCommand, [214](#)
- command\_map
  - mainwindow.cpp, [486](#)
- CompoundFileDirectory
  - embroidery\_internal.h, [296](#)
  - main.c, [398](#)
- CompoundFileDirectoryEntry
  - embroidery\_internal.h, [296](#)
  - main.c, [398](#)
- CompoundFileSector\_DIFAT\_Sector
  - embroidery\_internal.h, [284](#)
- CompoundFileSector\_EndOfChain
  - embroidery\_internal.h, [284](#)
- CompoundFileSector\_FAT\_Sector

- embroidery\_internal.h, 285
- CompoundFileSector\_FreeSector
  - embroidery\_internal.h, 285
- CompoundFileSector\_MaxRegSector
  - embroidery\_internal.h, 285
- CompoundFileStreamId\_MaxRegularStreamId
  - embroidery\_internal.h, 285
- CompoundFileStreamId\_NoStream
  - embroidery\_internal.h, 285
- Compress, 78
  - bit\_position, 78
  - bits\_total, 78
  - block\_elements, 78
  - character\_huffman, 78
  - character\_length\_huffman, 78
  - distance\_huffman, 79
  - input\_data, 79
  - input\_length, 79
- compress
  - embroidery\_internal.h, 292
- compress.c
  - compress\_get\_bits, 234
  - compress\_get\_position, 234
  - compress\_get\_token, 235
  - compress\_init, 235
  - compress\_load\_block, 235
  - compress\_load\_character\_huffman, 235
  - compress\_load\_character\_length\_huffman, 235
  - compress\_load\_distance\_huffman, 235
  - compress\_peek, 235
  - compress\_pop, 235
  - compress\_read\_variable\_length, 235
  - huffman\_build\_table, 235
  - huffman\_lookup, 235
  - huffman\_lookup\_data, 236
  - hus\_compress, 235
  - hus\_decompress, 236
- compress\_get\_bits
  - compress.c, 234
  - embroidery\_internal.h, 296
- compress\_get\_position
  - compress.c, 234
  - embroidery\_internal.h, 296
- compress\_get\_token
  - compress.c, 235
  - embroidery\_internal.h, 296
- compress\_init
  - compress.c, 235
- compress\_load\_block
  - compress.c, 235
  - embroidery\_internal.h, 297
- compress\_load\_character\_huffman
  - compress.c, 235
  - embroidery\_internal.h, 297
- compress\_load\_character\_length\_huffman
  - compress.c, 235
  - embroidery\_internal.h, 297
- compress\_load\_distance\_huffman
  - compress.c, 235
  - embroidery\_internal.h, 297
- compress\_peek
  - compress.c, 235
- compress\_pop
  - compress.c, 235
  - embroidery\_internal.h, 297
- compress\_read\_variable\_length
  - compress.c, 235
  - embroidery\_internal.h, 297
- config
  - embroidermodder.h, 429
  - mainwindow.cpp, 486
- config\_tables
  - embroidermodder.h, 429
  - mainwindow.cpp, 486
- constants
  - LSYSTEM, 146
- construct\_command
  - embroidermodder.h, 421
  - mainwindow.cpp, 471
- contains
  - embroidermodder.h, 422
  - view.cpp, 493
- context\_menu\_action
  - StatusBar, 208
- context\_menu\_event
  - StatusBar, 208
- contextMenuEvent
  - CmdPromptHistory, 67
  - CmdPromptInput, 72
  - View, 221
- control1
  - EmbBezier\_, 82
- control2
  - EmbBezier\_, 82
- convert
  - embroidery.h, 253
  - pattern.c, 404
- convert\_args\_to\_type
  - embroidermodder.h, 422
  - mainwindow.cpp, 471
- copy
  - View, 221
- copy\_action
  - mainwindow.cpp, 471
- copy\_selected\_action
  - mainwindow.cpp, 472
- copy\_trim
  - embroidery\_internal.h, 297
  - main.c, 398
- copyClip
  - CmdPromptInput, 72
- copyPressed
  - CmdPrompt, 56
  - CmdPromptInput, 72
- copySelected
  - View, 221

- cornerButtonClicked
  - View, [221](#)
- count
  - EmbArray\_, [81](#)
- create\_checkbox
  - Settings\_Dialog, [205](#)
- create\_float\_spinbox
  - Settings\_Dialog, [205](#)
- create\_icon
  - MainWindow, [152](#)
- create\_menu
  - embroidermodder.h, [422](#)
  - mainwindow-menus.cpp, [457](#)
- create\_test\_file\_1
  - embroidery\_internal.h, [297](#)
- create\_test\_file\_2
  - embroidery\_internal.h, [297](#)
- create\_test\_file\_3
  - embroidery\_internal.h, [297](#)
- create\_toolbar
  - MainWindow, [152](#)
- createAllActions
  - MainWindow, [153](#)
- createAllMenus
  - MainWindow, [153](#)
- createAllToolbars
  - MainWindow, [153](#)
- createComboBoxSelected
  - PropertyEditor, [182](#)
- createGrid
  - View, [221](#)
- createGridIso
  - View, [221](#)
- createGridPolar
  - View, [221](#)
- createGridRect
  - View, [221](#)
- createGroupBox
  - PropertyEditor, [183](#)
- createHandle
  - CmdPromptSplitter, [77](#)
- createHistogram
  - EmbDetailsDialog, [86](#)
- createLineEdit
  - PropertyEditor, [183](#)
- createMainWidget
  - EmbDetailsDialog, [86](#)
- createObjectList
  - View, [221](#)
- createOrigin
  - View, [222](#)
- createRulerTextPath
  - View, [222](#)
- createTabDisplay
  - Settings\_Dialog, [205](#)
- createTabFilesPaths
  - Settings\_Dialog, [205](#)
- createTabGeneral
  - Settings\_Dialog, [206](#)
- createTabGridRuler
  - Settings\_Dialog, [206](#)
- createTabLineWeight
  - Settings\_Dialog, [206](#)
- createTabOpenSave
  - Settings\_Dialog, [206](#)
- createTabOrthoPolar
  - Settings\_Dialog, [206](#)
- createTabPrinting
  - Settings\_Dialog, [206](#)
- createTabPrompt
  - Settings\_Dialog, [206](#)
- createTabQuickSnap
  - Settings\_Dialog, [206](#)
- createTabQuickTrack
  - Settings\_Dialog, [206](#)
- createTabSelection
  - Settings\_Dialog, [206](#)
- createTabSnap
  - Settings\_Dialog, [206](#)
- createToolButton
  - PropertyEditor, [183](#)
- createToolButtonPickAdd
  - PropertyEditor, [183](#)
- createToolButtonQSelect
  - PropertyEditor, [183](#)
- creationTime
  - \_bcf\_directory\_entry, [44](#)
- creatorName
  - ThredExtension\_, [211](#)
- crosshairColor
  - View, [227](#)
- crosshairSize
  - View, [227](#)
- csd, [9, 339](#)
- csd\_decryptArray
  - format\_csd.c, [340](#)
- CsdSubMaskSize
  - format\_csd.c, [339](#)
- CsdXorMaskSize
  - format\_csd.c, [339](#)
- csv, [341](#)
- CSV\_EXPECT
  - embroidery\_internal.h, [293](#)
- CSV\_EXPECT\_COMMA
  - embroidery\_internal.h, [293](#)
- CSV\_EXPECT\_NULL
  - embroidery\_internal.h, [293](#)
- CSV\_EXPECT\_QUOTE1
  - embroidery\_internal.h, [293](#)
- CSV\_EXPECT\_QUOTE2
  - embroidery\_internal.h, [293](#)
- CSV\_MODE
  - embroidery\_internal.h, [293](#)
- CSV\_MODE\_COMMENT
  - embroidery\_internal.h, [293](#)
- CSV\_MODE\_NULL



- embroidery\_internal.h, [293](#)
- CSV\_MODE\_STITCH
  - embroidery\_internal.h, [293](#)
- CSV\_MODE\_THREAD
  - embroidery\_internal.h, [293](#)
- CSV\_MODE\_VARIABLE
  - embroidery\_internal.h, [293](#)
- csvStitchFlagToStr
  - format\_csv.c, [341](#)
- csvStrToStitchFlag
  - format\_csv.c, [341](#)
- CUBICTOCONTROL1
  - embroidery\_internal.h, [285](#)
- CUBICTOCONTROL2
  - embroidery\_internal.h, [285](#)
- CUBICTOEND
  - embroidery\_internal.h, [285](#)
- curCmd
  - CmdPromptInput, [75](#)
- currentColor
  - MdiWindow, [177](#)
- curFile
  - MdiWindow, [178](#)
- curLayer
  - MdiWindow, [178](#)
- curLineType
  - MdiWindow, [178](#)
- curLineWeight
  - MdiWindow, [178](#)
- current\_element\_id
  - format\_svg.c, [369](#)
- currentAttribute
  - format\_svg.c, [369](#)
- currentColorChanged
  - MdiWindow, [171](#)
- currentColorIndex
  - EmbPattern\_, [97](#)
- currentDisplayBackgroundColorChanged
  - Settings\_Dialog, [206](#)
- currentDisplayCrossHairColorChanged
  - Settings\_Dialog, [206](#)
- currentDisplaySelectBoxLeftColorChanged
  - Settings\_Dialog, [206](#)
- currentDisplaySelectBoxLeftFillColorChanged
  - Settings\_Dialog, [206](#)
- currentDisplaySelectBoxRightColorChanged
  - Settings\_Dialog, [206](#)
- currentDisplaySelectBoxRightFillColorChanged
  - Settings\_Dialog, [207](#)
- currentGeneralMdiBackgroundColorChanged
  - Settings\_Dialog, [207](#)
- currentGridColorChanged
  - Settings\_Dialog, [207](#)
- currentLayerChanged
  - MdiWindow, [171](#)
- currentLinetypeChanged
  - MdiWindow, [172](#)
- currentLineweightChanged
  - MdiWindow, [172](#)
- currentPromptBackgroundColorChanged
  - Settings\_Dialog, [207](#)
- currentPromptTextColorChanged
  - Settings\_Dialog, [207](#)
- currentRulerColorChanged
  - Settings\_Dialog, [207](#)
- currentValue
  - format\_svg.c, [369](#)
- curText
  - CmdPromptInput, [76](#)
- curved
  - Geometry, [138](#)
- cut
  - View, [222](#)
- cut\_action
  - mainwindow.cpp, [472](#)
- cut\_selected\_action
  - mainwindow.cpp, [472](#)
- cutCopyMousePoint
  - View, [227](#)
- cutCopyObjectList
  - MainWindow, [163](#)
- cutPressed
  - CmdPrompt, [56](#)
  - CmdPromptInput, [72](#)
- d
  - em2\_dev\_script, [42](#)
- dat, [9](#)
- data
  - EmblImage\_, [92](#)
- day
  - EmbTime\_, [104](#)
- day\_vision\_action
  - mainwindow.cpp, [472](#)
- debug\_action
  - mainwindow.cpp, [472](#)
- debug\_message
  - embroidermodder.h, [422](#)
  - interface.cpp, [450](#)
- decode\_exy\_flags
  - format\_exy.c, [348](#)
- decode\_record\_flags
  - format\_dst.c, [344](#)
- decode\_t01\_record
  - embroidery\_internal.h, [297](#)
- encoding.c, [324](#)
- decode\_tajima\_ternary
  - embroidery\_internal.h, [298](#)
- encoding.c, [324](#)
- decode\_tap\_record\_flags
  - format\_tap.c, [371](#)
- DecodeCsdByte
  - format\_csd.c, [340](#)
- decodeNewStitch
  - embroidery\_internal.h, [298](#)
- encoding.c, [324](#)
- default\_value

- Huffman, 141
- defaultPrefix
  - CmdPromptInput, 76
- degrees
  - embroidery.h, 253
  - functions.c, 385
- degrees\_\_\_
  - embroidermodder.h, 422
  - interface.cpp, 450
- delete\_selected\_action
  - mainwindow.cpp, 472
- deleteObject
  - View, 222
- deletePressed
  - CmdPrompt, 57
  - CmdPromptInput, 72
  - MainWindow, 153
  - MdiWindow, 172
  - View, 222
- deleteSelected
  - View, 222
- delta
  - UndoableCommand, 215
- dem, 9, 342
- description
  - EmbFormatList\_, 88
  - EmbThread\_, 103
- design\_details\_action
  - mainwindow.cpp, 472
- designDetails
  - MdiWindow, 172
- dialog
  - embroidermodder.h, 429
  - mainwindow.cpp, 486
- Dictionary
  - embroidermodder.h, 419
- difat
  - \_bcf\_file, 45
- difatEntriesInHeader
  - main.c, 403
- dimensions
  - EmblImage\_, 92
- dirBrush
  - SelectBox, 199
- directory
  - \_bcf\_file, 45
- directoryEntryName
  - \_bcf\_directory\_entry, 44
- directoryEntryNameLength
  - \_bcf\_directory\_entry, 44
- dirEntries
  - \_bcf\_directory, 43
- dirPen
  - SelectBox, 199
- disable\_action
  - mainwindow.cpp, 472
- display\_props
  - settings-dialog.cpp, 491
- distance\_huffman
  - Compress, 79
- do\_nothing\_action
  - mainwindow.cpp, 473
- docIndex
  - MainWindow, 163
- dockPropEdit
  - embroidermodder.h, 429
  - mainwindow.cpp, 486
- dockUndoEdit
  - embroidermodder.h, 429
  - mainwindow.cpp, 486
- done
  - UndoableCommand, 215
- Dot
  - Geometry, 110
- doubleSpinBoxes
  - embroidermodder.h, 430
  - mainwindow.cpp, 486
- downPressed
  - CmdPrompt, 57
  - CmdPromptInput, 72
- dragon\_curve
  - fill.c, 327
- drawBackground
  - View, 222
- drawForeground
  - View, 222
- drawRubberLine
  - Geometry, 115
- dsb, 9, 342
- dst, 9, 343
- dstJumpsPerTrim
  - EmbPattern\_, 97
- dsz, 9, 303, 345
- dx, 9, 346
- dx\_color
  - embroidery.h, 243
- DXF\_VERSION\_2000
  - embroidery\_internal.h, 285
- DXF\_VERSION\_2002
  - embroidery\_internal.h, 285
- DXF\_VERSION\_2004
  - embroidery\_internal.h, 285
- DXF\_VERSION\_2006
  - embroidery\_internal.h, 285
- DXF\_VERSION\_2007
  - embroidery\_internal.h, 285
- DXF\_VERSION\_2009
  - embroidery\_internal.h, 285
- DXF\_VERSION\_2010
  - embroidery\_internal.h, 285
- DXF\_VERSION\_2013
  - embroidery\_internal.h, 285
- DXF\_VERSION\_R10
  - embroidery\_internal.h, 285
- DXF\_VERSION\_R11
  - embroidery\_internal.h, 286

DXF\_VERSION\_R12  
    embroidery\_internal.h, 286

DXF\_VERSION\_R13  
    embroidery\_internal.h, 286

DXF\_VERSION\_R14  
    embroidery\_internal.h, 286

DXF\_VERSION\_R15  
    embroidery\_internal.h, 286

DXF\_VERSION\_R18  
    embroidery\_internal.h, 286

DXF\_VERSION\_R21  
    embroidery\_internal.h, 286

DXF\_VERSION\_R24  
    embroidery\_internal.h, 286

DXF\_VERSION\_R27  
    embroidery\_internal.h, 286

edr, 9, 303, 346

ELEMENT\_A  
    embroidery\_internal.h, 286

ELEMENT\_ANIMATE  
    embroidery\_internal.h, 286

ELEMENT\_ANIMATECOLOR  
    embroidery\_internal.h, 286

ELEMENT\_ANIMATEMOTION  
    embroidery\_internal.h, 286

ELEMENT\_ANIMATETRANSFORM  
    embroidery\_internal.h, 286

ELEMENT\_ANIMATION  
    embroidery\_internal.h, 286

ELEMENT\_AUDIO  
    embroidery\_internal.h, 286

ELEMENT\_CIRCLE  
    embroidery\_internal.h, 286

ELEMENT\_DEFS  
    embroidery\_internal.h, 286

ELEMENT\_DESC  
    embroidery\_internal.h, 287

ELEMENT\_DISCARD  
    embroidery\_internal.h, 287

ELEMENT\_ELLIPSE  
    embroidery\_internal.h, 287

ELEMENT\_FONT  
    embroidery\_internal.h, 287

ELEMENT\_FONT\_FACE  
    embroidery\_internal.h, 287

ELEMENT\_FONT\_FACE\_SRC  
    embroidery\_internal.h, 287

ELEMENT\_FONT\_FACE\_URI  
    embroidery\_internal.h, 287

ELEMENT\_FOREIGN\_OBJECT  
    embroidery\_internal.h, 287

ELEMENT\_G  
    embroidery\_internal.h, 287

ELEMENT\_GLYPH  
    embroidery\_internal.h, 287

ELEMENT\_HANDLER  
    embroidery\_internal.h, 287

ELEMENT\_HKERN  
    embroidery\_internal.h, 287

ELEMENT\_IMAGE  
    embroidery\_internal.h, 287

ELEMENT\_LINE  
    embroidery\_internal.h, 287

ELEMENT\_LINEAR\_GRADIENT  
    embroidery\_internal.h, 287

ELEMENT\_LISTENER  
    embroidery\_internal.h, 287

ELEMENT\_METADATA  
    embroidery\_internal.h, 287

ELEMENT\_MISSING\_GLYPH  
    embroidery\_internal.h, 287

ELEMENT\_MPATH  
    embroidery\_internal.h, 288

ELEMENT\_PATH  
    embroidery\_internal.h, 288

ELEMENT\_POLYGON  
    embroidery\_internal.h, 288

ELEMENT\_POLYLINE  
    embroidery\_internal.h, 288

ELEMENT\_PREFETCH  
    embroidery\_internal.h, 288

ELEMENT\_RADIAL\_GRADIENT  
    embroidery\_internal.h, 288

ELEMENT\_RECT  
    embroidery\_internal.h, 288

ELEMENT\_SCRIPT  
    embroidery\_internal.h, 288

ELEMENT\_SET  
    embroidery\_internal.h, 288

ELEMENT\_SOLID\_COLOR  
    embroidery\_internal.h, 288

ELEMENT\_STOP  
    embroidery\_internal.h, 288

ELEMENT\_SVG  
    embroidery\_internal.h, 288

ELEMENT\_SWITCH  
    embroidery\_internal.h, 288

ELEMENT\_TBREAK  
    embroidery\_internal.h, 288

ELEMENT\_TEXT  
    embroidery\_internal.h, 288

ELEMENT\_TEXT\_AREA  
    embroidery\_internal.h, 288

ELEMENT\_TITLE  
    embroidery\_internal.h, 288

ELEMENT\_TSPAN  
    embroidery\_internal.h, 288

ELEMENT\_USE  
    embroidery\_internal.h, 289

ELEMENT\_VIDEO  
    embroidery\_internal.h, 289

ELEMENT\_XML  
    embroidery\_internal.h, 289

ellipse  
    EmbGeometry\_, 90

ellipse.c

- ellipse\_objectQuadrant0, [384](#)
- ellipse\_objectQuadrant180, [384](#)
- ellipse\_objectQuadrant270, [384](#)
- ellipse\_objectQuadrant90, [384](#)
- embEllipse\_area, [384](#)
- embEllipse\_diameterX, [384](#)
- embEllipse\_diameterY, [384](#)
- embEllipse\_height, [384](#)
- embEllipse\_init, [384](#)
- embEllipse\_perimeter, [384](#)
- embEllipse\_setDiameterMajor, [384](#)
- embEllipse\_setDiameterMinor, [385](#)
- embEllipse\_setRadiusMajor, [385](#)
- embEllipse\_setRadiusMinor, [385](#)
- embEllipse\_setSize, [385](#)
- embEllipse\_updatePath, [385](#)
- embEllipse\_width, [385](#)
- ellipse\_objectQuadrant0
  - ellipse.c, [384](#)
- ellipse\_objectQuadrant180
  - ellipse.c, [384](#)
- ellipse\_objectQuadrant270
  - ellipse.c, [384](#)
- ellipse\_objectQuadrant90
  - ellipse.c, [384](#)
- ELLIPSETOEND
  - embroidery\_internal.h, [289](#)
- ELLIPSETORAD
  - embroidery\_internal.h, [289](#)
- Elna, [347](#)
- Eltac, [348](#)
- em2\_dev\_script, [42](#)
  - d, [42](#)
  - header, [42](#)
  - s, [42](#)
- EMB\_ARC
  - embroidery.h, [243](#)
- EMB\_ARRAY
  - embroidery.h, [244](#)
- EMB\_BIG\_ENDIAN
  - embroidery\_internal.h, [289](#)
- EMB\_CIRCLE
  - embroidery.h, [244](#)
- emb\_constant\_pi
  - embroidermodder.h, [430](#)
- EMB\_DIM\_DIAMETER
  - embroidery.h, [244](#)
- EMB\_DIM\_LEADER
  - embroidery.h, [244](#)
- EMB\_ELLIPSE
  - embroidery.h, [244](#)
- emb\_error
  - embroidery.h, [268](#)
  - main.c, [403](#)
- EMB\_FLAG
  - embroidery.h, [244](#)
- EMB\_FORMAT\_100
  - embroidery.h, [244](#)
- EMB\_FORMAT\_ART
  - embroidery.h, [244](#)
- EMB\_FORMAT\_BMC
  - embroidery.h, [244](#)
- EMB\_FORMAT\_BRO
  - embroidery.h, [244](#)
- EMB\_FORMAT\_CND
  - embroidery.h, [244](#)
- EMB\_FORMAT\_COL
  - embroidery.h, [244](#)
- EMB\_FORMAT\_CSD
  - embroidery.h, [244](#)
- EMB\_FORMAT\_CSV
  - embroidery.h, [244](#)
- EMB\_FORMAT\_DAT
  - embroidery.h, [244](#)
- EMB\_FORMAT\_DEM
  - embroidery.h, [244](#)
- EMB\_FORMAT\_DSB
  - embroidery.h, [244](#)
- EMB\_FORMAT\_DST
  - embroidery.h, [245](#)
- EMB\_FORMAT\_DSZ
  - embroidery.h, [245](#)
- EMB\_FORMAT\_DXF
  - embroidery.h, [245](#)
- EMB\_FORMAT\_EDR
  - embroidery.h, [245](#)
- EMB\_FORMAT\_EMD
  - embroidery.h, [245](#)
- EMB\_FORMAT\_EXP
  - embroidery.h, [245](#)
- EMB\_FORMAT\_EXY
  - embroidery.h, [245](#)
- EMB\_FORMAT\_EYS
  - embroidery.h, [245](#)
- EMB\_FORMAT\_FXY
  - embroidery.h, [245](#)
- EMB\_FORMAT\_GC
  - embroidery.h, [245](#)
- EMB\_FORMAT\_GNC
  - embroidery.h, [245](#)
- EMB\_FORMAT\_GT
  - embroidery.h, [245](#)
- EMB\_FORMAT\_HUS
  - embroidery.h, [245](#)
- EMB\_FORMAT\_INB
  - embroidery.h, [245](#)
- EMB\_FORMAT\_INF
  - embroidery.h, [245](#)
- EMB\_FORMAT\_JEF
  - embroidery.h, [245](#)
- EMB\_FORMAT\_KSM
  - embroidery.h, [245](#)
- EMB\_FORMAT\_MAX
  - embroidery.h, [245](#)

- EMB\_FORMAT\_MIT
  - embroidery.h, [246](#)
- EMB\_FORMAT\_NEW
  - embroidery.h, [246](#)
- EMB\_FORMAT\_OFM
  - embroidery.h, [246](#)
- EMB\_FORMAT\_PCD
  - embroidery.h, [246](#)
- EMB\_FORMAT\_PCM
  - embroidery.h, [246](#)
- EMB\_FORMAT\_PCQ
  - embroidery.h, [246](#)
- EMB\_FORMAT\_PCS
  - embroidery.h, [246](#)
- EMB\_FORMAT\_PEC
  - embroidery.h, [246](#)
- EMB\_FORMAT\_PEL
  - embroidery.h, [246](#)
- EMB\_FORMAT\_PEM
  - embroidery.h, [246](#)
- EMB\_FORMAT\_PES
  - embroidery.h, [246](#)
- EMB\_FORMAT\_PHB
  - embroidery.h, [246](#)
- EMB\_FORMAT\_PHC
  - embroidery.h, [246](#)
- EMB\_FORMAT\_PLT
  - embroidery.h, [246](#)
- EMB\_FORMAT\_RGB
  - embroidery.h, [246](#)
- EMB\_FORMAT\_SEW
  - embroidery.h, [246](#)
- EMB\_FORMAT\_SHV
  - embroidery.h, [246](#)
- EMB\_FORMAT\_SST
  - embroidery.h, [246](#)
- EMB\_FORMAT\_STX
  - embroidery.h, [247](#)
- EMB\_FORMAT\_SVG
  - embroidery.h, [247](#)
- EMB\_FORMAT\_T01
  - embroidery.h, [247](#)
- EMB\_FORMAT\_T09
  - embroidery.h, [247](#)
- EMB\_FORMAT\_TAP
  - embroidery.h, [247](#)
- EMB\_FORMAT\_THR
  - embroidery.h, [247](#)
- EMB\_FORMAT\_TXT
  - embroidery.h, [247](#)
- EMB\_FORMAT\_U00
  - embroidery.h, [247](#)
- EMB\_FORMAT\_U01
  - embroidery.h, [247](#)
- EMB\_FORMAT\_VIP
  - embroidery.h, [247](#)
- EMB\_FORMAT\_VP3
  - embroidery.h, [247](#)
- EMB\_FORMAT\_XXX
  - embroidery.h, [247](#)
- EMB\_FORMAT\_ZSK
  - embroidery.h, [247](#)
- emb\_identify\_format
  - embroidery.h, [253](#)
  - formats.c, [333](#)
- EMB\_IMAGE
  - embroidery.h, [247](#)
- EMB\_INT16\_BIG
  - embroidery\_internal.h, [289](#)
- EMB\_INT16\_LITTLE
  - embroidery\_internal.h, [289](#)
- EMB\_INT32\_BIG
  - embroidery\_internal.h, [289](#)
- EMB\_INT32\_LITTLE
  - embroidery\_internal.h, [289](#)
- EMB\_LINE
  - embroidery.h, [247](#)
- EMB\_LITTLE\_ENDIAN
  - embroidery\_internal.h, [289](#)
- EMB\_MAX
  - embroidery\_internal.h, [289](#)
- EMB\_MAX\_LAYERS
  - embroidery.h, [247](#)
- EMB\_MIN
  - embroidery\_internal.h, [289](#)
- emb\_optOut
  - embroidery\_internal.h, [298](#)
  - main.c, [398](#)
- EMB\_PATH
  - embroidery.h, [247](#)
- EMB\_POINT
  - embroidery.h, [247](#)
- EMB\_POLYGON
  - embroidery.h, [248](#)
- EMB\_POLYLINE
  - embroidery.h, [248](#)
- EMB\_PUBLIC
  - embroidery.h, [248](#)
- emb\_readline
  - embroidery\_internal.h, [298](#)
  - main.c, [399](#)
- EMB\_RECT
  - embroidery.h, [248](#)
- emb\_round
  - embroidery.h, [253](#)
  - functions.c, [385](#)
- EMB\_SPLINE
  - embroidery.h, [248](#)
- EMB\_STITCH
  - embroidery.h, [248](#)
- EMB\_TEXT\_MULTI
  - embroidery.h, [248](#)
- EMB\_TEXT\_SINGLE
  - embroidery.h, [248](#)
- EMB\_THREAD
  - embroidery.h, [248](#)

EMB\_VECTOR  
  embroidery.h, 248  
emb\_verbose  
  embroidery.h, 268  
  main.c, 403  
EmbAlignedDim  
  embroidery.h, 250  
EmbAlignedDim\_, 79  
  position, 79  
EmbAngularDim  
  embroidery.h, 251  
EmbAngularDim\_, 79  
  position, 80  
EmbArc  
  embroidery.h, 251  
EmbArc\_, 80  
  end, 80  
  mid, 80  
  start, 80  
embArc\_arcLength  
  arc.c, 380  
embArc\_area  
  arc.c, 380  
embArc\_chord  
  arc.c, 380  
embArc\_clockwise  
  arc.c, 380  
  embroidery.h, 253  
embArc\_endAngle  
  arc.c, 380  
embArc\_gripEdit  
  arc.c, 380  
embArc\_includedAngle  
  arc.c, 380  
embArc\_init  
  arc.c, 380  
  embroidery.h, 253  
embArc\_mouseSnapPoint  
  arc.c, 381  
embArc\_paint  
  arc.c, 381  
embArc\_print  
  main.c, 399  
embArc\_setCenter  
  arc.c, 381  
embArc\_setEndAngle  
  arc.c, 381  
embArc\_setRadius  
  arc.c, 381  
embArc\_setStartAngle  
  arc.c, 381  
embArc\_startAngle  
  arc.c, 381  
embArc\_updatePath  
  arc.c, 381  
embArc\_updateRubber  
  arc.c, 381  
EmbArcLengthDim  
  embroidery.h, 251  
EmbArcLengthDim\_, 81  
  position, 81  
EmbArray  
  embroidery.h, 251  
EmbArray\_, 81  
  count, 81  
  geometry, 81  
  length, 81  
  stitch, 82  
  thread, 82  
  type, 82  
embArray\_addArc  
  array.c, 232  
  embroidery.h, 253  
embArray\_addCircle  
  array.c, 232  
  embroidery.h, 253  
embArray\_addEllipse  
  array.c, 232  
  embroidery.h, 253  
embArray\_addFlag  
  array.c, 232  
  embroidery.h, 254  
embArray\_addLine  
  array.c, 233  
  embroidery.h, 254  
embArray\_addPath  
  array.c, 233  
  embroidery.h, 254  
embArray\_addPoint  
  array.c, 233  
  embroidery.h, 254  
embArray\_addPolygon  
  array.c, 233  
  embroidery.h, 254  
embArray\_addPolyline  
  array.c, 233  
  embroidery.h, 254  
embArray\_addRect  
  array.c, 233  
  embroidery.h, 254  
embArray\_addStitch  
  array.c, 233  
  embroidery.h, 254  
embArray\_addThread  
  embroidery.h, 254  
embArray\_addVector  
  array.c, 233  
  embroidery.h, 254  
embArray\_copy  
  array.c, 233  
  embroidery.h, 254  
embArray\_create  
  array.c, 233  
  embroidery.h, 255  
embArray\_free  
  array.c, 233

- embroidery.h, 255
- embArray\_resize
  - array.c, 234
  - embroidery.h, 255
- embBase\_setColorRGB
  - arc.c, 381
- EmbBezier
  - embroidery.h, 251
- EmbBezier\_, 82
  - control1, 82
  - control2, 82
  - end, 82
  - start, 82
- EmbBlock
  - embroidery.h, 251
- EmbBlock\_, 83
  - position, 83
- EmbCircle
  - embroidery.h, 251
- EmbCircle\_, 83
  - center, 83
  - radius, 83
- embCircle\_area
  - circle.c, 383
- embCircle\_circumference
  - circle.c, 383
- embCircle\_init
  - circle.c, 383
  - embroidery.h, 255
- embCircle\_prompt
  - arc.c, 381
- embCircle\_setArea
  - arc.c, 381
- embCircle\_setCircumference
  - arc.c, 382
- EmbColor
  - embroidery.h, 251
- EmbColor\_, 84
  - b, 84
  - g, 84
  - r, 84
- embColor\_create
  - embroidery.h, 255
- embColor\_distance
  - embroidery.h, 255
  - main.c, 399
- embColor\_fromHexStr
  - embroidery.h, 255
  - encoding.c, 324
- embColor\_make
  - embroidery.h, 255
- embColor\_read
  - embroidery\_internal.h, 298
  - main.c, 399
- embColor\_write
  - embroidery\_internal.h, 298
  - main.c, 399
- embConstantPi
  - embroidery.h, 268
  - main.c, 403
- EmbDetailsDialog, 84
  - ~EmbDetailsDialog, 85
  - boundingRect, 86
  - buttonBox, 86
  - colorChanges, 86
  - colorTotal, 86
  - createHistogram, 86
  - createMainWidget, 86
  - EmbDetailsDialog, 85
  - getInfo, 86
  - mainWidget, 86
  - stitchesJump, 86
  - stitchesReal, 87
  - stitchesTotal, 87
  - stitchesTrim, 87
- EmbDiameterDim
  - embroidery.h, 251
- EmbDiameterDim\_, 87
  - position, 87
- EmbEllipse
  - embroidery.h, 251
- EmbEllipse\_, 87
  - center, 88
  - radius, 88
  - rotation, 88
- embEllipse\_area
  - ellipse.c, 384
  - embroidery.h, 255
- embEllipse\_click
  - arc.c, 382
- embEllipse\_diameterX
  - ellipse.c, 384
  - embroidery.h, 255
- embEllipse\_diameterY
  - ellipse.c, 384
  - embroidery.h, 256
- embEllipse\_height
  - ellipse.c, 384
  - embroidery.h, 256
- embEllipse\_init
  - ellipse.c, 384
  - embroidery.h, 256
- embEllipse\_main
  - arc.c, 382
- embEllipse\_make
  - embroidery.h, 256
- embEllipse\_perimeter
  - ellipse.c, 384
  - embroidery.h, 256
- embEllipse\_setDiameterMajor
  - ellipse.c, 384
- embEllipse\_setDiameterMinor
  - ellipse.c, 385
- embEllipse\_setRadiusMajor
  - ellipse.c, 385
- embEllipse\_setRadiusMinor

- ellipse.c, 385
- embEllipse\_setSize
  - ellipse.c, 385
- embEllipse\_updatePath
  - ellipse.c, 385
- embEllipse\_width
  - ellipse.c, 385
- embroidery.h, 256
- EmbFlag
  - embroidery.h, 251
- embFormat\_getExtension
  - formats.c, 333
- EMBFORMAT\_MAXDESC
  - embroidery.h, 248
- EMBFORMAT\_MAXEXT
  - embroidery.h, 248
- EMBFORMAT\_OBJECTONLY
  - embroidery.h, 248
- EMBFORMAT\_STCHANDOBJ
  - embroidery.h, 248
- EMBFORMAT\_STITCHONLY
  - embroidery.h, 248
- EMBFORMAT\_UNSUPPORTED
  - embroidery.h, 248
- EmbFormatList
  - embroidery.h, 251
- EmbFormatList\_, 88
  - check\_for\_color\_file, 88
  - color\_only, 88
  - description, 88
  - extension, 89
  - reader\_state, 89
  - type, 89
  - write\_external\_color\_file, 89
  - writer\_state, 89
- EmbGeometry
  - embroidery.h, 251
- EmbGeometry\_, 89
  - arc, 90
  - circle, 90
  - color, 90
  - ellipse, 90
  - flag, 90
  - line, 90
  - lineType, 90
  - object, 90
  - path, 90
  - point, 90
  - polygon, 90
  - polyline, 91
  - rect, 91
  - spline, 91
  - stitch, 91
  - thread, 91
  - type, 91
  - vector, 91
- embGeometry\_boundingRect
  - embroidery.h, 256
- geometry.c, 378
- embGeometry\_free
  - embroidery.h, 256
- geometry.c, 378
- embGeometry\_init
  - embroidery.h, 256
- geometry.c, 378
- embGeometry\_move
  - embroidery.h, 256
- geometry.c, 378
- embGeometry\_vulcanize
  - embroidery.h, 257
- geometry.c, 378
- EmblImage
  - embroidery.h, 251
- EmblImage\_, 91
  - data, 92
  - dimensions, 92
  - height, 92
  - name, 92
  - path, 92
  - position, 92
  - width, 92
- emblImage\_create
  - embroidery.h, 257
- emblImage\_free
  - embroidery.h, 257
- emblImage\_read
  - embroidery.h, 257
- emblImage\_write
  - embroidery.h, 257
- EmblInfiniteLine
  - embroidery.h, 251
- EmblInfiniteLine\_, 92
  - position, 93
- emblInt\_read
  - embroidery\_internal.h, 299
- encoding.c, 325
- emblInt\_write
  - embroidery\_internal.h, 299
- encoding.c, 325
- Embird, 303, 341, 346
- EmbLayer
  - embroidery.h, 251
- EmbLayer\_, 93
  - geometry, 93
  - name, 93
- EmbLeaderDim
  - embroidery.h, 251
- EmbLeaderDim\_, 93
  - position, 94
- EmbLine
  - embroidery.h, 251
- EmbLine\_, 94
  - color, 94
  - end, 94
  - lineType, 94
  - start, 94



embLine\_intersectionPoint  
  embroidery.h, [257](#)  
  line.c, [386](#)

embLine\_make  
  embroidery.h, [257](#)

embLine\_normalVector  
  embroidery.h, [257](#)  
  line.c, [386](#)

embLine\_toVector  
  line.c, [386](#)

EmbLinearDim  
  embroidery.h, [252](#)

EmbLinearDim\_, [95](#)  
  position, [95](#)

EmbOrdinateDim  
  embroidery.h, [252](#)

EmbOrdinateDim\_, [95](#)  
  position, [95](#)

EmbPath  
  embroidery.h, [252](#)

EmbPath\_, [95](#)  
  color, [96](#)  
  flagList, [96](#)  
  lineType, [96](#)  
  pointList, [96](#)

EmbPattern  
  embroidery.h, [252](#)

EmbPattern\_, [96](#)  
  currentColorIndex, [97](#)  
  dstJumpsPerTrim, [97](#)  
  geometry, [97](#)  
  home, [97](#)  
  hoop\_height, [97](#)  
  hoop\_width, [97](#)  
  layer, [97](#)  
  stitch\_list, [97](#)  
  thread\_list, [97](#)

embPattern\_addCircleAbs  
  embroidery.h, [257](#)  
  pattern.c, [404](#)

embPattern\_addEllipseAbs  
  embroidery.h, [257](#)  
  pattern.c, [404](#)

embPattern\_addLineAbs  
  embroidery.h, [258](#)  
  pattern.c, [405](#)

embPattern\_addPathAbs  
  embroidery.h, [258](#)  
  pattern.c, [405](#)

embPattern\_addPointAbs  
  embroidery.h, [258](#)  
  pattern.c, [405](#)

embPattern\_addPolygonAbs  
  embroidery.h, [258](#)  
  pattern.c, [405](#)

embPattern\_addPolylineAbs  
  embroidery.h, [258](#)

embPattern\_addPolylineObjectAbs  
  pattern.c, [405](#)

embPattern\_addRectAbs  
  embroidery.h, [258](#)  
  pattern.c, [405](#)

embPattern\_addStitchAbs  
  embroidery.h, [258](#)  
  pattern.c, [405](#)

embPattern\_addStitchRel  
  embroidery.h, [258](#)  
  pattern.c, [405](#)

embPattern\_addThread  
  embroidery.h, [258](#)  
  pattern.c, [405](#)

embPattern\_calcBoundingBox  
  embroidery.h, [259](#)  
  pattern.c, [406](#)

embPattern\_center  
  embroidery.h, [259](#)  
  pattern.c, [406](#)

embPattern\_changeColor  
  embroidery.h, [259](#)  
  pattern.c, [406](#)

embPattern\_color\_count  
  embroidery.h, [259](#)  
  pattern.c, [406](#)

embPattern\_combine  
  embroidery.h, [259](#)  
  fill.c, [327](#)

embPattern\_combineJumpStitches  
  embroidery.h, [259](#)  
  pattern.c, [406](#)

embPattern\_convertGeometry  
  embroidery.h, [259](#)  
  fill.c, [327](#)

embPattern\_copyPolylinesToStitch\_list  
  pattern.c, [406](#)

embPattern\_copyPolylinesToStitchList  
  embroidery.h, [259](#)

embPattern\_copystitch\_listToPolylines  
  pattern.c, [406](#)

embPattern\_copyStitchListToPolylines  
  embroidery.h, [259](#)

embPattern\_correctForMaxStitchLength  
  embroidery.h, [259](#)  
  pattern.c, [406](#)

embPattern\_create  
  embroidery.h, [260](#)  
  pattern.c, [406](#)

embPattern\_crossstitch  
  embroidery.h, [260](#)  
  fill.c, [327](#)

embPattern\_designDetails  
  embroidery.h, [260](#)  
  pattern.c, [406](#)

embPattern\_end  
  embroidery.h, [260](#)  
  pattern.c, [407](#)

embPattern\_fixColorCount

- embroidery.h, 260
- pattern.c, 407
- embPattern\_flip
  - embroidery.h, 260
  - pattern.c, 407
- embPattern\_flipHorizontal
  - embroidery.h, 260
  - pattern.c, 407
- embPattern\_flipVertical
  - embroidery.h, 260
  - pattern.c, 407
- embPattern\_free
  - embroidery.h, 260
  - pattern.c, 407
- embPattern\_hideStitchesOverLength
  - embroidery.h, 261
  - pattern.c, 407
- embPattern\_horizontal\_fill
  - embroidery.h, 261
  - fill.c, 327
- embPattern\_jumpStitches
  - embroidery.h, 261
  - pattern.c, 407
- embPattern\_lengthHistogram
  - embroidery.h, 261
  - pattern.c, 407
- embPattern\_loadExternalColorFile
  - embroidery.h, 261
  - pattern.c, 407
- embPattern\_maximumStitchLength
  - embroidery.h, 261
  - pattern.c, 407
- embPattern\_minimumStitchLength
  - embroidery.h, 261
  - pattern.c, 407
- embPattern\_movePolylinesToStitch\_list
  - pattern.c, 408
- embPattern\_movePolylinesToStitchList
  - embroidery.h, 261
- embPattern\_movestitch\_listToPolylines
  - pattern.c, 408
- embPattern\_moveStitchListToPolylines
  - embroidery.h, 261
- embPattern\_read
  - embroidery.h, 261
  - formats.c, 333
- embPattern\_readAuto
  - embroidery.h, 262
  - formats.c, 333
- embPattern\_realStitches
  - embroidery.h, 262
  - pattern.c, 408
- embPattern\_render
  - embroidery.h, 262
- embPattern\_scale
  - embroidery.h, 262
  - pattern.c, 408
- embPattern\_simulate
  - embroidery.h, 262
- embPattern\_stitchArc
  - fill.c, 327
- embPattern\_stitchCircle
  - fill.c, 327
- embPattern\_stitchEllipse
  - fill.c, 328
- embPattern\_stitchPath
  - fill.c, 328
- embPattern\_stitchPolygon
  - fill.c, 328
- embPattern\_stitchPolyline
  - fill.c, 328
- embPattern\_stitchRect
  - fill.c, 329
- embPattern\_stitchText
  - fill.c, 329
- embPattern\_totalStitchLength
  - embroidery.h, 262
  - pattern.c, 408
- embPattern\_trimStitches
  - embroidery.h, 262
  - pattern.c, 408
- embPattern\_write
  - embroidery.h, 262
  - formats.c, 333
- embPattern\_writeAuto
  - embroidery.h, 263
  - formats.c, 333
- EmbPoint
  - embroidery.h, 252
- EmbPoint\_, 97
  - color, 98
  - lineType, 98
  - position, 98
- EmbPolygon
  - embroidery.h, 252
- embPolygon\_reduceByDistance
  - fill.c, 329
- embPolygon\_reduceByNth
  - fill.c, 329
- EmbPolyline
  - embroidery.h, 252
- EmbRadiusDim
  - embroidery.h, 252
- EmbRadiusDim\_, 98
  - position, 98
- EmbRay
  - embroidery.h, 252
- EmbRay\_, 99
  - position, 99
- EmbReal
  - embroidery.h, 252
- EmbRect
  - embroidery.h, 252
- EmbRect\_, 99
  - bottom, 99
  - left, 99

- radius, 99
- right, 100
- rotation, 100
- top, 100
- embRect\_area
  - embroidery.h, 263
  - rect.c, 387
- embRect\_bottomLeft
  - arc.c, 382
- embRect\_bottomRight
  - arc.c, 382
- embRect\_init
  - embroidery.h, 263
  - rect.c, 387
- embroidermodder.cpp
  - \_appVer\_, 411
  - exitApp, 411
  - main, 411
  - usage\_msg, 411
- embroidermodder.h
  - \_mainWin, 429
  - actionHash, 429
  - activeScene, 420
  - activeView, 420
  - actuator, 421
  - add\_polyline, 421
  - add\_to\_path, 421
  - BOOL\_TYPE, 418
  - checkBoxes, 429
  - comboBoxes, 429
  - Command, 419
  - config, 429
  - config\_tables, 429
  - construct\_command, 421
  - contains, 422
  - convert\_args\_to\_type, 422
  - create\_menu, 422
  - debug\_message, 422
  - degrees\_, 422
  - dialog, 429
  - Dictionary, 419
  - dockPropEdit, 429
  - dockUndoEdit, 429
  - doubleSpinBoxes, 430
  - emb\_constant\_pi, 430
  - fileExtension, 422
  - FUNCTION\_TYPE, 418
  - get\_bool, 423
  - get\_int, 423
  - get\_qstr, 423
  - get\_real, 423
  - get\_str, 423
  - get\_str\_list, 423
  - get\_uint, 423
  - groupBoxes, 430
  - INT\_TYPE, 418
  - labels, 430
  - lineEdits, 430
  - make\_checkbox, 423
  - make\_spinbox, 423
  - make\_ui\_element, 424
  - mdiArea, 430
  - menuHash, 430
  - Node, 419
  - node\_bool, 424
  - node\_int, 424
  - node\_qstr, 424
  - node\_real, 424
  - node\_str, 425
  - node\_str\_list, 425
  - node\_uint, 425
  - NodeList, 419
  - OBJ\_COLOR, 419
  - OBJ\_KEYS, 419
  - OBJ\_LAYER, 419
  - OBJ\_LTYPE, 419
  - OBJ\_LWT, 419
  - OBJ\_NAME, 419
  - OBJ\_RUBBER, 419
  - OBJ\_TYPE, 419
  - OBJ\_TYPE\_ARC, 420
  - OBJ\_TYPE\_BASE, 420
  - OBJ\_TYPE\_BLOCK, 420
  - OBJ\_TYPE\_CIRCLE, 420
  - OBJ\_TYPE\_DIMALIGNED, 420
  - OBJ\_TYPE\_DIMANGULAR, 420
  - OBJ\_TYPE\_DIMARCLENGTH, 420
  - OBJ\_TYPE\_DIMDIAMETER, 420
  - OBJ\_TYPE\_DIMLEADER, 420
  - OBJ\_TYPE\_DIMLINEAR, 420
  - OBJ\_TYPE\_DIMORDINATE, 420
  - OBJ\_TYPE\_DIMRADIUS, 420
  - OBJ\_TYPE\_ELLIPSE, 420
  - OBJ\_TYPE\_ELLIPSEARC, 420
  - OBJ\_TYPE\_GRID, 420
  - OBJ\_TYPE\_HATCH, 420
  - OBJ\_TYPE\_IMAGE, 420
  - OBJ\_TYPE\_INFINITELINE, 420
  - OBJ\_TYPE\_LINE, 420
  - OBJ\_TYPE\_NULL, 420
  - OBJ\_TYPE\_PATH, 420
  - OBJ\_TYPE\_POINT, 420
  - OBJ\_TYPE\_POLYGON, 420
  - OBJ\_TYPE\_POLYLINE, 420
  - OBJ\_TYPE\_RAY, 420
  - OBJ\_TYPE\_RECTANGLE, 420
  - OBJ\_TYPE\_RUBBER, 420
  - OBJ\_TYPE\_SLOT, 420
  - OBJ\_TYPE\_SPLINE, 420
  - OBJ\_TYPE\_TEXTMULTI, 420
  - OBJ\_TYPE\_TEXTSINGLE, 420
  - OBJ\_TYPE\_UNKNOWN, 420
  - OBJ\_TYPE\_VALUES, 420
  - operator\*, 425
  - operator+, 425
  - operator-, 426

- prompt, [430](#)
- radians\_\_, [426](#)
- read\_configuration, [426](#)
- read\_settings, [426](#)
- read\_string\_setting, [426](#)
- REAL\_TYPE, [418](#)
- rotate\_vector, [426](#)
- run\_script, [426](#)
- run\_script\_file, [427](#)
- scripts, [430](#)
- set\_enabled, [427](#)
- set\_visibility, [427](#)
- settings, [430](#)
- spinBoxes, [430](#)
- statusbar, [430](#)
- String, [419](#)
- STRING\_LIST\_TYPE, [418](#)
- STRING\_TYPE, [419](#)
- StringList, [419](#)
- subMenuHash, [430](#)
- to\_EmbVector, [427](#)
- to\_qlist, [428](#)
- to\_QPointF, [428](#)
- to\_string\_vector, [428](#)
- to\_vector, [428](#)
- tokenize, [428](#)
- toolbarHash, [430](#)
- toolButtons, [430](#)
- translate\_str, [429](#)
- UNKNOWN\_TYPE, [419](#)
- validFileFormat, [429](#)
- VECTOR\_TYPE, [419](#)
- write\_settings, [429](#)
- embroidery.h
  - \_dxfColorTable, [268](#)
  - Arc\_Polyester, [243](#)
  - Arc\_Rayon, [243](#)
  - black\_thread, [268](#)
  - CHUNK\_SIZE, [243](#)
  - CoatsAndClark\_Rayon, [243](#)
  - convert, [253](#)
  - degrees, [253](#)
  - dxf\_color, [243](#)
  - EMB\_ARC, [243](#)
  - EMB\_ARRAY, [244](#)
  - EMB\_CIRCLE, [244](#)
  - EMB\_DIM\_DIAMETER, [244](#)
  - EMB\_DIM\_LEADER, [244](#)
  - EMB\_ELLIPSE, [244](#)
  - emb\_error, [268](#)
  - EMB\_FLAG, [244](#)
  - EMB\_FORMAT\_100, [244](#)
  - EMB\_FORMAT\_10O, [244](#)
  - EMB\_FORMAT\_ART, [244](#)
  - EMB\_FORMAT\_BMC, [244](#)
  - EMB\_FORMAT\_BRO, [244](#)
  - EMB\_FORMAT\_CND, [244](#)
  - EMB\_FORMAT\_COL, [244](#)
  - EMB\_FORMAT\_CSD, [244](#)
  - EMB\_FORMAT\_CSV, [244](#)
  - EMB\_FORMAT\_DAT, [244](#)
  - EMB\_FORMAT\_DEM, [244](#)
  - EMB\_FORMAT\_DSB, [244](#)
  - EMB\_FORMAT\_DST, [245](#)
  - EMB\_FORMAT\_DSZ, [245](#)
  - EMB\_FORMAT\_DXF, [245](#)
  - EMB\_FORMAT\_EDR, [245](#)
  - EMB\_FORMAT\_EMD, [245](#)
  - EMB\_FORMAT\_EXP, [245](#)
  - EMB\_FORMAT\_EXY, [245](#)
  - EMB\_FORMAT\_EYS, [245](#)
  - EMB\_FORMAT\_FXY, [245](#)
  - EMB\_FORMAT\_GC, [245](#)
  - EMB\_FORMAT\_GNC, [245](#)
  - EMB\_FORMAT\_GT, [245](#)
  - EMB\_FORMAT\_HUS, [245](#)
  - EMB\_FORMAT\_INB, [245](#)
  - EMB\_FORMAT\_INF, [245](#)
  - EMB\_FORMAT\_JEF, [245](#)
  - EMB\_FORMAT\_KSM, [245](#)
  - EMB\_FORMAT\_MAX, [245](#)
  - EMB\_FORMAT\_MIT, [246](#)
  - EMB\_FORMAT\_NEW, [246](#)
  - EMB\_FORMAT\_OFM, [246](#)
  - EMB\_FORMAT\_PCD, [246](#)
  - EMB\_FORMAT\_PCM, [246](#)
  - EMB\_FORMAT\_PCQ, [246](#)
  - EMB\_FORMAT\_PCS, [246](#)
  - EMB\_FORMAT\_PEC, [246](#)
  - EMB\_FORMAT\_PEL, [246](#)
  - EMB\_FORMAT\_PEM, [246](#)
  - EMB\_FORMAT\_PES, [246](#)
  - EMB\_FORMAT\_PHB, [246](#)
  - EMB\_FORMAT\_PHC, [246](#)
  - EMB\_FORMAT\_PLT, [246](#)
  - EMB\_FORMAT\_RGB, [246](#)
  - EMB\_FORMAT\_SEW, [246](#)
  - EMB\_FORMAT\_SHV, [246](#)
  - EMB\_FORMAT\_SST, [246](#)
  - EMB\_FORMAT\_STX, [247](#)
  - EMB\_FORMAT\_SVG, [247](#)
  - EMB\_FORMAT\_T01, [247](#)
  - EMB\_FORMAT\_T09, [247](#)
  - EMB\_FORMAT\_TAP, [247](#)
  - EMB\_FORMAT\_THR, [247](#)
  - EMB\_FORMAT\_TXT, [247](#)
  - EMB\_FORMAT\_U00, [247](#)
  - EMB\_FORMAT\_U01, [247](#)
  - EMB\_FORMAT\_VIP, [247](#)
  - EMB\_FORMAT\_VP3, [247](#)
  - EMB\_FORMAT\_XXX, [247](#)
  - EMB\_FORMAT\_ZSK, [247](#)
  - emb\_identify\_format, [253](#)
  - EMB\_IMAGE, [247](#)
  - EMB\_LINE, [247](#)
  - EMB\_MAX\_LAYERS, [247](#)

EMB\_PATH, [247](#)  
EMB\_POINT, [247](#)  
EMB\_POLYGON, [248](#)  
EMB\_POLYLINE, [248](#)  
EMB\_PUBLIC, [248](#)  
EMB\_RECT, [248](#)  
emb\_round, [253](#)  
EMB\_SPLINE, [248](#)  
EMB\_STITCH, [248](#)  
EMB\_TEXT\_MULTI, [248](#)  
EMB\_TEXT\_SINGLE, [248](#)  
EMB\_THREAD, [248](#)  
EMB\_VECTOR, [248](#)  
emb\_verbose, [268](#)  
EmbAlignedDim, [250](#)  
EmbAngularDim, [251](#)  
EmbArc, [251](#)  
embArc\_clockwise, [253](#)  
embArc\_init, [253](#)  
EmbArcLengthDim, [251](#)  
EmbArray, [251](#)  
embArray\_addArc, [253](#)  
embArray\_addCircle, [253](#)  
embArray\_addEllipse, [253](#)  
embArray\_addFlag, [254](#)  
embArray\_addLine, [254](#)  
embArray\_addPath, [254](#)  
embArray\_addPoint, [254](#)  
embArray\_addPolygon, [254](#)  
embArray\_addPolyline, [254](#)  
embArray\_addRect, [254](#)  
embArray\_addStitch, [254](#)  
embArray\_addThread, [254](#)  
embArray\_addVector, [254](#)  
embArray\_copy, [254](#)  
embArray\_create, [255](#)  
embArray\_free, [255](#)  
embArray\_resize, [255](#)  
EmbBezier, [251](#)  
EmbBlock, [251](#)  
EmbCircle, [251](#)  
embCircle\_init, [255](#)  
EmbColor, [251](#)  
embColor\_create, [255](#)  
embColor\_distance, [255](#)  
embColor\_fromHexStr, [255](#)  
embColor\_make, [255](#)  
embConstantPi, [268](#)  
EmbDiameterDim, [251](#)  
EmbEllipse, [251](#)  
embEllipse\_area, [255](#)  
embEllipse\_diameterX, [255](#)  
embEllipse\_diameterY, [256](#)  
embEllipse\_height, [256](#)  
embEllipse\_init, [256](#)  
embEllipse\_make, [256](#)  
embEllipse\_perimeter, [256](#)  
embEllipse\_width, [256](#)  
EmbFlag, [251](#)  
EMBFORMAT\_MAXDESC, [248](#)  
EMBFORMAT\_MAXEXT, [248](#)  
EMBFORMAT\_OBJECTONLY, [248](#)  
EMBFORMAT\_STCHANDOBJ, [248](#)  
EMBFORMAT\_STITCHONLY, [248](#)  
EMBFORMAT\_UNSUPPORTED, [248](#)  
EmbFormatList, [251](#)  
EmbGeometry, [251](#)  
embGeometry\_boundingRect, [256](#)  
embGeometry\_free, [256](#)  
embGeometry\_init, [256](#)  
embGeometry\_move, [256](#)  
embGeometry\_vulcanize, [257](#)  
EmbImage, [251](#)  
embImage\_create, [257](#)  
embImage\_free, [257](#)  
embImage\_read, [257](#)  
embImage\_write, [257](#)  
EmbInfiniteLine, [251](#)  
EmbLayer, [251](#)  
EmbLeaderDim, [251](#)  
EmbLine, [251](#)  
embLine\_intersectionPoint, [257](#)  
embLine\_make, [257](#)  
embLine\_normalVector, [257](#)  
EmbLinearDim, [252](#)  
EmbOrdinateDim, [252](#)  
EmbPath, [252](#)  
EmbPattern, [252](#)  
embPattern\_addCircleAbs, [257](#)  
embPattern\_addEllipseAbs, [257](#)  
embPattern\_addLineAbs, [258](#)  
embPattern\_addPathAbs, [258](#)  
embPattern\_addPointAbs, [258](#)  
embPattern\_addPolygonAbs, [258](#)  
embPattern\_addPolylineAbs, [258](#)  
embPattern\_addRectAbs, [258](#)  
embPattern\_addStitchAbs, [258](#)  
embPattern\_addStitchRel, [258](#)  
embPattern\_addThread, [258](#)  
embPattern\_calcBoundingBox, [259](#)  
embPattern\_center, [259](#)  
embPattern\_changeColor, [259](#)  
embPattern\_color\_count, [259](#)  
embPattern\_combine, [259](#)  
embPattern\_combineJumpStitches, [259](#)  
embPattern\_convertGeometry, [259](#)  
embPattern\_copyPolylinesToStitchList, [259](#)  
embPattern\_copyStitchListToPolylines, [259](#)  
embPattern\_correctForMaxStitchLength, [259](#)  
embPattern\_create, [260](#)  
embPattern\_crossstitch, [260](#)  
embPattern\_designDetails, [260](#)  
embPattern\_end, [260](#)  
embPattern\_fixColorCount, [260](#)  
embPattern\_flip, [260](#)  
embPattern\_flipHorizontal, [260](#)

- embPattern\_flipVertical, 260
- embPattern\_free, 260
- embPattern\_hideStitchesOverLength, 261
- embPattern\_horizontal\_fill, 261
- embPattern\_jumpStitches, 261
- embPattern\_lengthHistogram, 261
- embPattern\_loadExternalColorFile, 261
- embPattern\_maximumStitchLength, 261
- embPattern\_minimumStitchLength, 261
- embPattern\_movePolylinesToStitchList, 261
- embPattern\_moveStitchListToPolylines, 261
- embPattern\_read, 261
- embPattern\_readAuto, 262
- embPattern\_realStitches, 262
- embPattern\_render, 262
- embPattern\_scale, 262
- embPattern\_simulate, 262
- embPattern\_totalStitchLength, 262
- embPattern\_trimStitches, 262
- embPattern\_write, 262
- embPattern\_writeAuto, 263
- EmbPoint, 252
- EmbPolygon, 252
- EmbPolyline, 252
- EmbRadiusDim, 252
- EmbRay, 252
- EmbReal, 252
- EmbRect, 252
- embRect\_area, 263
- embRect\_init, 263
- EmbSatinOutline, 252
- embSatinOutline\_generateSatinOutline, 263
- embSatinOutline\_renderStitches, 263
- EmbSpline, 252
- EmbStitch, 252
- EmbTextMulti, 252
- EmbTextSingle, 252
- EmbThread, 252
- embThread\_findNearestColor, 263
- embThread\_findNearestThread, 264
- embThread\_getRandom, 264
- EmbTime, 252
- embTime\_initNow, 264
- embTime\_time, 264
- EmbVector, 253
- embVector\_add, 264
- embVector\_angle, 264
- embVector\_average, 264
- embVector\_cross, 265
- embVector\_distance, 265
- embVector\_dot, 265
- embVector\_length, 265
- embVector\_multiply, 265
- embVector\_normalize, 265
- embVector\_relativeX, 266
- embVector\_relativeY, 266
- embVector\_subtract, 266
- embVector\_transpose\_product, 266
- embVector\_unit, 266
- END, 248
- Exquisite\_Polyester, 248
- formatTable, 268
- Fufu\_Polyester, 249
- Fufu\_Rayon, 249
- full\_test\_matrix, 266
- getArcCenter, 266
- getArcDataFromBulge, 266
- getCircleCircleIntersections, 267
- getCircleTangentPoints, 267
- Hemingworth\_Polyester, 249
- hilbert\_curve, 267
- hus\_thread, 249
- husThreads, 268
- Isacord\_Polyester, 249
- Isafil\_Rayon, 249
- jef\_thread, 249
- jefThreads, 268
- JUMP, 249
- L\_system, 253
- LIBEMBROIDERY\_EMBEDDED\_VERSION, 249
- lindenmayer\_system, 267
- Madeira\_Polyester, 249
- Madeira\_Rayon, 249
- Marathon\_Polyester, 249
- Marathon\_Rayon, 249
- MAX\_STITCHES, 249
- MAX\_THREADS, 249
- Metro\_Polyester, 249
- NORMAL, 249
- numberOfFormats, 249
- Pantone, 250
- pcm\_thread, 250
- pcmThreads, 268
- pec\_thread, 250
- pecThreadCount, 268
- pecThreads, 269
- radians, 267
- report, 267
- RobisonAnton\_Polyester, 250
- RobisonAnton\_Rayon, 250
- SEQUIN, 250
- shv\_thread, 250
- shvThreadCount, 269
- shvThreads, 269
- Sigma\_Polyester, 250
- STOP, 250
- Sulky\_Rayon, 250
- SVG\_Colors, 250
- testMain, 267
- thread\_color, 253
- ThreadArt\_Polyester, 250
- ThreadArt\_Rayon, 250
- threadColor, 268
- threadColorName, 268
- threadColorNum, 268
- ThreaDelight\_Polyester, 250

- TRIM, [250](#)
- vipDecodingTable, [269](#)
- Z102\_Isacord\_Polyester, [250](#)
- embroidery\_internal.h
  - bcf\_difat\_create, [293](#)
  - bcf\_directory, [292](#)
  - bcf\_directory\_entry, [292](#)
  - bcf\_directory\_free, [294](#)
  - bcf\_file, [292](#)
  - bcf\_file\_difat, [292](#)
  - bcf\_file\_difat\_free, [294](#)
  - bcf\_file\_fat, [292](#)
  - bcf\_file\_fat\_free, [294](#)
  - bcf\_file\_free, [294](#)
  - bcf\_file\_header, [292](#)
  - bcfFile\_read, [294](#)
  - bcfFileFat\_create, [294](#)
  - bcfFileHeader\_isValid, [294](#)
  - bcfFileHeader\_read, [294](#)
  - binaryReadString, [294](#)
  - binaryReadUnicodeString, [295](#)
  - binaryWriteInt, [295](#)
  - binaryWriteIntBE, [295](#)
  - binaryWriteShort, [295](#)
  - binaryWriteUInt, [295](#)
  - binaryWriteUIntBE, [295](#)
  - binaryWriteUShort, [295](#)
  - binaryWriteUShortBE, [296](#)
  - BULGETOCONTROL, [284](#)
  - BULGETOEND, [284](#)
  - check\_header\_present, [296](#)
  - CompoundFileDirectory, [296](#)
  - CompoundFileDirectoryEntry, [296](#)
  - CompoundFileSector\_DIFAT\_Sector, [284](#)
  - CompoundFileSector\_EndOfChain, [284](#)
  - CompoundFileSector\_FAT\_Sector, [285](#)
  - CompoundFileSector\_FreeSector, [285](#)
  - CompoundFileSector\_MaxRegSector, [285](#)
  - CompoundFileStreamId\_MaxRegularStreamId, [285](#)
  - CompoundFileStreamId\_NoStream, [285](#)
  - compress, [292](#)
  - compress\_get\_bits, [296](#)
  - compress\_get\_position, [296](#)
  - compress\_get\_token, [296](#)
  - compress\_load\_block, [297](#)
  - compress\_load\_character\_huffman, [297](#)
  - compress\_load\_character\_length\_huffman, [297](#)
  - compress\_load\_distance\_huffman, [297](#)
  - compress\_pop, [297](#)
  - compress\_read\_variable\_length, [297](#)
  - copy\_trim, [297](#)
  - create\_test\_file\_1, [297](#)
  - create\_test\_file\_2, [297](#)
  - create\_test\_file\_3, [297](#)
  - CSV\_EXPECT, [293](#)
  - CSV\_EXPECT\_COMMA, [293](#)
  - CSV\_EXPECT\_NULL, [293](#)
  - CSV\_EXPECT\_QUOTE1, [293](#)
  - CSV\_EXPECT\_QUOTE2, [293](#)
  - CSV\_MODE, [293](#)
  - CSV\_MODE\_COMMENT, [293](#)
  - CSV\_MODE\_NULL, [293](#)
  - CSV\_MODE\_STITCH, [293](#)
  - CSV\_MODE\_THREAD, [293](#)
  - CSV\_MODE\_VARIABLE, [293](#)
  - CUBICTOCONTROL1, [285](#)
  - CUBICTOCONTROL2, [285](#)
  - CUBICTOEND, [285](#)
  - decode\_t01\_record, [297](#)
  - decode\_tajima\_ternary, [298](#)
  - decodeNewStitch, [298](#)
  - DXF\_VERSION\_2000, [285](#)
  - DXF\_VERSION\_2002, [285](#)
  - DXF\_VERSION\_2004, [285](#)
  - DXF\_VERSION\_2006, [285](#)
  - DXF\_VERSION\_2007, [285](#)
  - DXF\_VERSION\_2009, [285](#)
  - DXF\_VERSION\_2010, [285](#)
  - DXF\_VERSION\_2013, [285](#)
  - DXF\_VERSION\_R10, [285](#)
  - DXF\_VERSION\_R11, [286](#)
  - DXF\_VERSION\_R12, [286](#)
  - DXF\_VERSION\_R13, [286](#)
  - DXF\_VERSION\_R14, [286](#)
  - DXF\_VERSION\_R15, [286](#)
  - DXF\_VERSION\_R18, [286](#)
  - DXF\_VERSION\_R21, [286](#)
  - DXF\_VERSION\_R24, [286](#)
  - DXF\_VERSION\_R27, [286](#)
  - ELEMENT\_A, [286](#)
  - ELEMENT\_ANIMATE, [286](#)
  - ELEMENT\_ANIMATECOLOR, [286](#)
  - ELEMENT\_ANIMATEMOTION, [286](#)
  - ELEMENT\_ANIMATETRANSFORM, [286](#)
  - ELEMENT\_ANIMATION, [286](#)
  - ELEMENT\_AUDIO, [286](#)
  - ELEMENT\_CIRCLE, [286](#)
  - ELEMENT\_DEFS, [286](#)
  - ELEMENT\_DESC, [287](#)
  - ELEMENT\_DISCARD, [287](#)
  - ELEMENT\_ELLIPSE, [287](#)
  - ELEMENT\_FONT, [287](#)
  - ELEMENT\_FONT\_FACE, [287](#)
  - ELEMENT\_FONT\_FACE\_SRC, [287](#)
  - ELEMENT\_FONT\_FACE\_URI, [287](#)
  - ELEMENT\_FOREIGN\_OBJECT, [287](#)
  - ELEMENT\_G, [287](#)
  - ELEMENT\_GLYPH, [287](#)
  - ELEMENT\_HANDLER, [287](#)
  - ELEMENT\_HKERN, [287](#)
  - ELEMENT\_IMAGE, [287](#)
  - ELEMENT\_LINE, [287](#)
  - ELEMENT\_LINEAR\_GRADIENT, [287](#)
  - ELEMENT\_LISTENER, [287](#)
  - ELEMENT\_METADATA, [287](#)



ELEMENT\_MISSING\_GLYPH, [287](#)  
ELEMENT\_MPATH, [288](#)  
ELEMENT\_PATH, [288](#)  
ELEMENT\_POLYGON, [288](#)  
ELEMENT\_POLYLINE, [288](#)  
ELEMENT\_PREFETCH, [288](#)  
ELEMENT\_RADIAL\_GRADIENT, [288](#)  
ELEMENT\_RECT, [288](#)  
ELEMENT\_SCRIPT, [288](#)  
ELEMENT\_SET, [288](#)  
ELEMENT\_SOLID\_COLOR, [288](#)  
ELEMENT\_STOP, [288](#)  
ELEMENT\_SVG, [288](#)  
ELEMENT\_SWITCH, [288](#)  
ELEMENT\_TBREAK, [288](#)  
ELEMENT\_TEXT, [288](#)  
ELEMENT\_TEXT\_AREA, [288](#)  
ELEMENT\_TITLE, [288](#)  
ELEMENT\_TSPAN, [288](#)  
ELEMENT\_USE, [289](#)  
ELEMENT\_VIDEO, [289](#)  
ELEMENT\_XML, [289](#)  
ELLIPSETOEND, [289](#)  
ELLIPSETORAD, [289](#)  
EMB\_BIG\_ENDIAN, [289](#)  
EMB\_INT16\_BIG, [289](#)  
EMB\_INT16\_LITTLE, [289](#)  
EMB\_INT32\_BIG, [289](#)  
EMB\_INT32\_LITTLE, [289](#)  
EMB\_LITTLE\_ENDIAN, [289](#)  
EMB\_MAX, [289](#)  
EMB\_MIN, [289](#)  
emb\_optOut, [298](#)  
emb\_readline, [298](#)  
embColor\_read, [298](#)  
embColor\_write, [298](#)  
embInt\_read, [299](#)  
embInt\_write, [299](#)  
encode\_t01\_record, [299](#)  
encode\_tajima\_ternary, [299](#)  
ENDIAN\_HOST, [289](#)  
entriesInDifatSector, [299](#)  
fpad, [299](#)  
fread\_int16, [299](#)  
fread\_int32\_be, [300](#)  
fread\_uint16, [300](#)  
GetFile, [300](#)  
GREEN\_TERM\_COLOR, [289](#)  
HOOP\_110X110, [289](#)  
HOOP\_126X110, [289](#)  
HOOP\_140X200, [290](#)  
HOOP\_230X200, [290](#)  
HOOP\_50X50, [290](#)  
huffman, [292](#)  
huffman\_build\_table, [300](#)  
huffman\_table\_lookup, [300](#)  
hus\_compress, [300](#)  
hus\_decompress, [300](#)  
imageWithFrame, [316](#)  
LINETO, [290](#)  
loadFatFromSector, [301](#)  
mitDecodeStitch, [301](#)  
mitEncodeStitch, [301](#)  
MOVETO, [290](#)  
N\_PES\_VERSIONS, [290](#)  
numberOfEntriesInDifatSector, [301](#)  
ObjectTypeRootEntry, [290](#)  
ObjectTypeStorage, [290](#)  
ObjectTypeStream, [290](#)  
ObjectTypeUnknown, [290](#)  
PES0001, [290](#)  
PES0020, [290](#)  
PES0022, [290](#)  
PES0030, [290](#)  
PES0040, [290](#)  
PES0050, [290](#)  
PES0055, [290](#)  
PES0056, [291](#)  
PES0060, [291](#)  
PES0070, [291](#)  
PES0080, [291](#)  
PES0090, [291](#)  
PES0100, [291](#)  
pfaffDecode, [301](#)  
pfaffEncode, [301](#)  
printArcResults, [301](#)  
QUADTOCONTROL, [291](#)  
QUADTOEND, [291](#)  
read100, [302](#)  
read10o, [302](#)  
readArt, [302](#)  
readBmc, [302](#)  
readBro, [302](#)  
readCnd, [302](#)  
readCol, [302](#)  
readCsd, [302](#)  
readCsv, [302](#)  
readDat, [302](#)  
readDem, [302](#)  
readDescriptions, [303](#)  
readDsb, [303](#)  
readDst, [303](#)  
readDsz, [303](#)  
readDxf, [303](#)  
readEdr, [303](#)  
readEmd, [303](#)  
readExp, [303](#)  
readExy, [303](#)  
readEys, [303](#)  
readFeatherPatterns, [304](#)  
readFullSector, [304](#)  
readFxy, [304](#)  
readGc, [304](#)  
readGnc, [304](#)  
readGt, [304](#)  
readHoopName, [304](#)



readHus, [304](#)  
readImageString, [304](#)  
readInb, [305](#)  
readInf, [305](#)  
readJef, [305](#)  
readKsm, [305](#)  
readMax, [305](#)  
readMit, [305](#)  
readMotifPatterns, [305](#)  
readNew, [305](#)  
readNextSector, [305](#)  
readOfm, [305](#)  
readPcd, [306](#)  
readPcm, [306](#)  
readPcq, [306](#)  
readPcs, [306](#)  
readPec, [306](#)  
readPecStitches, [306](#)  
readPel, [306](#)  
readPem, [306](#)  
readPes, [307](#)  
readPESHeaderV10, [307](#)  
readPESHeaderV5, [307](#)  
readPESHeaderV6, [307](#)  
readPESHeaderV7, [307](#)  
readPESHeaderV8, [307](#)  
readPESHeaderV9, [307](#)  
readPhb, [307](#)  
readPhc, [307](#)  
readPlt, [307](#)  
readProgrammableFills, [308](#)  
readRgb, [308](#)  
readSew, [308](#)  
readShv, [308](#)  
readSst, [308](#)  
readStx, [308](#)  
readSvg, [308](#)  
readT01, [308](#)  
readT09, [308](#)  
readTap, [308](#)  
readThr, [309](#)  
readThreads, [309](#)  
readTxt, [309](#)  
readU00, [309](#)  
readU01, [309](#)  
readVip, [309](#)  
readVp3, [309](#)  
readXxx, [309](#)  
readZsk, [309](#)  
RED\_TERM\_COLOR, [291](#)  
RESET\_TERM\_COLOR, [291](#)  
safe\_free, [309](#)  
stringInArray, [310](#)  
StxThread, [292](#)  
SubDescriptor, [293](#)  
SVG\_ATTRIBUTE, [291](#)  
SVG\_CATCH\_ALL, [291](#)  
SVG\_CREATOR\_EMBROIDERMODDER, [291](#)  
SVG\_CREATOR\_ILLUSTRATOR, [291](#)  
SVG\_CREATOR\_INKSCAPE, [291](#)  
SVG\_CREATOR\_NULL, [291](#)  
SVG\_ELEMENT, [291](#)  
SVG\_EXPECT\_ATTRIBUTE, [291](#)  
SVG\_EXPECT\_ELEMENT, [292](#)  
SVG\_EXPECT\_NULL, [292](#)  
SVG\_EXPECT\_VALUE, [292](#)  
SVG\_MEDIA\_PROPERTY, [292](#)  
SVG\_NULL, [292](#)  
SVG\_PROPERTY, [292](#)  
SvgAttribute, [293](#)  
testEmbCircle, [310](#)  
testEmbCircle\_2, [310](#)  
testEmbFormat, [310](#)  
testGeomArc, [310](#)  
testTangentPoints, [310](#)  
testThreadColor, [310](#)  
ThredExtension, [293](#)  
ThredHeader, [293](#)  
VipHeader, [293](#)  
vp3Hoop, [293](#)  
write100, [310](#)  
write10o, [310](#)  
write\_24bit, [310](#)  
writeArt, [311](#)  
writeBmc, [311](#)  
writeBro, [311](#)  
writeCnd, [311](#)  
writeCol, [311](#)  
writeCsd, [311](#)  
writeCsv, [311](#)  
writeDat, [311](#)  
writeDem, [311](#)  
writeDsb, [311](#)  
writeDst, [311](#)  
writeDsz, [312](#)  
writeDxf, [312](#)  
writeEdr, [312](#)  
writeEmd, [312](#)  
writeExp, [312](#)  
writeExy, [312](#)  
writeEys, [312](#)  
writeFxy, [312](#)  
writeGc, [312](#)  
writeGnc, [312](#)  
writeGt, [312](#)  
writeHus, [313](#)  
writeInb, [313](#)  
writeInf, [313](#)  
writeJef, [313](#)  
writeKsm, [313](#)  
writeMax, [313](#)  
writeMit, [313](#)  
writeNew, [313](#)  
writeOfm, [313](#)  
writePcd, [313](#)  
writePcm, [313](#)

- writePcq, 314
- writePcs, 314
- writePec, 314
- writePecStitches, 314
- writePel, 314
- writePem, 314
- writePes, 314
- writePhb, 314
- writePhc, 314
- writePlt, 314
- writeRgb, 315
- writeSew, 315
- writeShv, 315
- writeSst, 315
- writeStx, 315
- writeSvg, 315
- writeT01, 315
- writeT09, 315
- writeTap, 315
- writeThr, 315
- writeTxt, 315
- writeU00, 316
- writeU01, 316
- writeVip, 316
- writeVp3, 316
- writeXxx, 316
- writeZsk, 316
- YELLOW\_TERM\_COLOR, 292
- EmbSatinOutline
  - embroidery.h, 252
- EmbSatinOutline\_, 100
  - length, 100
  - side1, 100
  - side2, 100
- embSatinOutline\_generateSatinOutline
  - embroidery.h, 263
  - main.c, 399
- embSatinOutline\_renderStitches
  - embroidery.h, 263
  - main.c, 399
- EmbSpline
  - embroidery.h, 252
- EmbSpline\_, 101
  - beziers, 101
- EmbStitch
  - embroidery.h, 252
- EmbStitch\_, 101
  - color, 101
  - flags, 101
  - x, 101
  - y, 102
- EmbTextMulti
  - embroidery.h, 252
- EmbTextMulti\_, 102
  - position, 102
  - text, 102
- EmbTextSingle
  - embroidery.h, 252
- EmbTextSingle\_, 102
  - position, 103
  - text, 103
- EmbThread
  - embroidery.h, 252
- EmbThread\_, 103
  - catalogNumber, 103
  - color, 103
  - description, 103
- embThread\_findNearestColor
  - embroidery.h, 263
  - main.c, 400
- embThread\_findNearestThread
  - embroidery.h, 264
  - main.c, 400
- embThread\_getRandom
  - embroidery.h, 264
  - main.c, 400
- EmbTime
  - embroidery.h, 252
- EmbTime\_, 104
  - day, 104
  - hour, 104
  - minute, 104
  - month, 104
  - second, 104
  - year, 104
- embTime\_initNow
  - embroidery.h, 264
  - main.c, 400
- embTime\_time
  - embroidery.h, 264
  - main.c, 400
- EmbVector
  - embroidery.h, 253
- EmbVector\_, 105
  - x, 105
  - y, 105
- embVector\_add
  - embroidery.h, 264
  - vector.c, 389
- embVector\_angle
  - embroidery.h, 264
  - vector.c, 389
- embVector\_average
  - embroidery.h, 264
  - vector.c, 389
- embVector\_cross
  - embroidery.h, 265
  - vector.c, 389
- embVector\_distance
  - embroidery.h, 265
  - vector.c, 390
- embVector\_dot
  - embroidery.h, 265
  - vector.c, 390
- embVector\_length
  - embroidery.h, 265

- vector.c, 390
- embVector\_multiply
  - embroidery.h, 265
  - vector.c, 390
- embVector\_normalize
  - embroidery.h, 265
  - vector.c, 390
- embVector\_print
  - main.c, 400
- embVector\_relativeX
  - embroidery.h, 266
  - vector.c, 390
- embVector\_relativeY
  - embroidery.h, 266
  - vector.c, 391
- embVector\_subtract
  - embroidery.h, 266
  - vector.c, 391
- embVector\_transpose\_product
  - embroidery.h, 266
  - vector.c, 391
- embVector\_unit
  - embroidery.h, 266
  - vector.c, 391
- emd, 9, 347
- emdDecode
  - format\_emd.c, 347
- encode\_record
  - format\_dst.c, 345
- encode\_t01\_record
  - embroidery\_internal.h, 299
  - encoding.c, 325
- encode\_tajima\_ternary
  - embroidery\_internal.h, 299
  - encoding.c, 325
- encode\_tap\_record
  - format\_tap.c, 371
- encoding.c
  - decode\_t01\_record, 324
  - decode\_tajima\_ternary, 324
  - decodeNewStitch, 324
  - embColor\_fromHexStr, 324
  - embInt\_read, 325
  - embInt\_write, 325
  - encode\_t01\_record, 325
  - encode\_tajima\_ternary, 325
  - mitDecodeStitch, 325
  - mitEncodeStitch, 325
  - pfaffDecode, 325
  - pfaffEncode, 326
  - reverse\_byte\_order, 326
  - write\_24bit, 326
- END
  - embroidery.h, 248
- end
  - EmbArc\_, 80
  - EmbBezier\_, 82
  - EmbLine\_, 94
- end\_action
  - mainwindow.cpp, 473
- endCommand
  - CmdPromptInput, 72
- ENDIAN\_HOST
  - embroidery\_internal.h, 289
- enterEvent
  - View, 222
- entriesInDifatSector
  - embroidery\_internal.h, 299
  - main.c, 401
- error\_action
  - mainwindow.cpp, 473
- escapePressed
  - CmdPrompt, 57
  - CmdPromptInput, 72
  - MainWindow, 153
  - MdiWindow, 172
  - View, 222
- event
  - Application, 53
- eventFilter
  - CmdPromptInput, 72
  - PropertyEditor, 183
- exitApp
  - embroidermodder.cpp, 411
- exp, 9, 347
- expDecode
  - format\_exp.c, 347
- Exquisite\_Polyester
  - embroidery.h, 248
- extension
  - EmbFormatList\_, 89
- extensions
  - settings-dialog.cpp, 492
- extern/libembroidery/src/array.c, 232
- extern/libembroidery/src/compress.c, 234
- extern/libembroidery/src/embroidery.h, 236, 269
- extern/libembroidery/src/embroidery\_internal.h, 276, 316
- extern/libembroidery/src/encoding.c, 323
- extern/libembroidery/src/fill.c, 326
- extern/libembroidery/src/formats.c, 331
- extern/libembroidery/src/formats/format\_100.c, 335
- extern/libembroidery/src/formats/format\_10o.c, 335
- extern/libembroidery/src/formats/format\_art.c, 336
- extern/libembroidery/src/formats/format\_bmc.c, 336
- extern/libembroidery/src/formats/format\_bro.c, 337
- extern/libembroidery/src/formats/format\_cnd.c, 337
- extern/libembroidery/src/formats/format\_col.c, 338
- extern/libembroidery/src/formats/format\_csd.c, 339
- extern/libembroidery/src/formats/format\_csv.c, 340
- extern/libembroidery/src/formats/format\_dat.c, 341
- extern/libembroidery/src/formats/format\_dem.c, 342
- extern/libembroidery/src/formats/format\_dsb.c, 342
- extern/libembroidery/src/formats/format\_dst.c, 343
- extern/libembroidery/src/formats/format\_dsz.c, 345
- extern/libembroidery/src/formats/format\_dxf.c, 345

- extern/libembroidery/src/formats/format\_edr.c, 346
- extern/libembroidery/src/formats/format\_emd.c, 347
- extern/libembroidery/src/formats/format\_exp.c, 347
- extern/libembroidery/src/formats/format\_exy.c, 348
- extern/libembroidery/src/formats/format\_ey.c, 348
- extern/libembroidery/src/formats/format\_fxy.c, 349
- extern/libembroidery/src/formats/format\_gc.c, 349
- extern/libembroidery/src/formats/format\_gnc.c, 350
- extern/libembroidery/src/formats/format\_gt.c, 350
- extern/libembroidery/src/formats/format\_hus.c, 351
- extern/libembroidery/src/formats/format\_inb.c, 352
- extern/libembroidery/src/formats/format\_inf.c, 352
- extern/libembroidery/src/formats/format\_jef.c, 353
- extern/libembroidery/src/formats/format\_ksm.c, 354
- extern/libembroidery/src/formats/format\_max.c, 354
- extern/libembroidery/src/formats/format\_mit.c, 355
- extern/libembroidery/src/formats/format\_new.c, 356
- extern/libembroidery/src/formats/format\_ofm.c, 356
- extern/libembroidery/src/formats/format\_pcd.c, 357
- extern/libembroidery/src/formats/format\_pcm.c, 358
- extern/libembroidery/src/formats/format\_pcq.c, 358
- extern/libembroidery/src/formats/format\_pcs.c, 359
- extern/libembroidery/src/formats/format\_pec.c, 359
- extern/libembroidery/src/formats/format\_pel.c, 361
- extern/libembroidery/src/formats/format\_pem.c, 361
- extern/libembroidery/src/formats/format\_pes.c, 362
- extern/libembroidery/src/formats/format\_phb.c, 364
- extern/libembroidery/src/formats/format\_phc.c, 365
- extern/libembroidery/src/formats/format\_plt.c, 365
- extern/libembroidery/src/formats/format\_rgb.c, 366
- extern/libembroidery/src/formats/format\_sew.c, 366
- extern/libembroidery/src/formats/format\_shv.c, 367
- extern/libembroidery/src/formats/format\_sst.c, 367
- extern/libembroidery/src/formats/format\_stx.c, 368
- extern/libembroidery/src/formats/format\_svg.c, 368
- extern/libembroidery/src/formats/format\_t01.c, 370
- extern/libembroidery/src/formats/format\_t09.c, 370
- extern/libembroidery/src/formats/format\_tap.c, 371
- extern/libembroidery/src/formats/format\_thr.c, 371
- extern/libembroidery/src/formats/format\_txt.c, 372
- extern/libembroidery/src/formats/format\_u00.c, 372
- extern/libembroidery/src/formats/format\_u01.c, 373
- extern/libembroidery/src/formats/format\_vip.c, 373
- extern/libembroidery/src/formats/format\_vp3.c, 375
- extern/libembroidery/src/formats/format\_xxx.c, 376
- extern/libembroidery/src/formats/format\_zsk.c, 377
- extern/libembroidery/src/geometry.c, 377
- extern/libembroidery/src/geometry/arc.c, 379
- extern/libembroidery/src/geometry/circle.c, 382
- extern/libembroidery/src/geometry/ellipse.c, 383
- extern/libembroidery/src/geometry/functions.c, 385
- extern/libembroidery/src/geometry/line.c, 386
- extern/libembroidery/src/geometry/path.c, 386
- extern/libembroidery/src/geometry/polygon.c, 386
- extern/libembroidery/src/geometry/polyline.c, 386
- extern/libembroidery/src/geometry/rect.c, 387
- extern/libembroidery/src/geometry/text.c, 387
- extern/libembroidery/src/geometry/vector.c, 389
- extern/libembroidery/src/image.c, 391
- extern/libembroidery/src/main.c, 392
- extern/libembroidery/src/pattern.c, 403
- extern/libembroidery/src/thread-color.c, 408
- exy, 9, 348
- eys, 304, 349
- F10Pressed
  - CmdPrompt, 57
  - CmdPromptInput, 73
- F11Pressed
  - CmdPrompt, 57
  - CmdPromptInput, 73
- F12Pressed
  - CmdPrompt, 57
  - CmdPromptInput, 73
- F1Pressed
  - CmdPrompt, 57
  - CmdPromptInput, 73
- F2Pressed
  - CmdPrompt, 57
  - CmdPromptInput, 73
- F3Pressed
  - CmdPrompt, 57
  - CmdPromptInput, 73
- F4Pressed
  - CmdPrompt, 57
  - CmdPromptInput, 73
- F5Pressed
  - CmdPrompt, 57
  - CmdPromptInput, 73
- F6Pressed
  - CmdPrompt, 58
  - CmdPromptInput, 73
- F7Pressed
  - CmdPrompt, 58
  - CmdPromptInput, 73
- F8Pressed
  - CmdPrompt, 58
  - CmdPromptInput, 74
- F9Pressed
  - CmdPrompt, 58
  - CmdPromptInput, 74
- factor
  - UndoableCommand, 215
- fat
  - \_bcf\_file, 45
- fatEntries
  - \_bcf\_file\_fat, 47
- fatEntryCount
  - \_bcf\_file\_fat, 47
- fatSectorCount
  - \_bcf\_file\_difat, 46
- fatSectorEntries
  - \_bcf\_file\_difat, 46
- fieldEdited
  - PropertyEditor, 183
- fieldNewText
  - property-editor.cpp, 489

- fieldNoText
  - property-editor.cpp, [490](#)
- fieldOffText
  - property-editor.cpp, [490](#)
- fieldOldText
  - property-editor.cpp, [490](#)
- fieldOnText
  - property-editor.cpp, [490](#)
- fieldVariesText
  - property-editor.cpp, [490](#)
- fieldYesText
  - property-editor.cpp, [490](#)
- fileExtension
  - embroidermodder.h, [422](#)
  - mdiwindow.cpp, [487](#)
- fileWasLoaded
  - MdiWindow, [178](#)
- fill.c
  - dragon\_curve, [327](#)
  - embPattern\_combine, [327](#)
  - embPattern\_convertGeometry, [327](#)
  - embPattern\_crossstitch, [327](#)
  - embPattern\_horizontal\_fill, [327](#)
  - embPattern\_stitchArc, [327](#)
  - embPattern\_stitchCircle, [327](#)
  - embPattern\_stitchEllipse, [328](#)
  - embPattern\_stitchPath, [328](#)
  - embPattern\_stitchPolygon, [328](#)
  - embPattern\_stitchPolyline, [328](#)
  - embPattern\_stitchRect, [329](#)
  - embPattern\_stitchText, [329](#)
  - embPolygon\_reduceByDistance, [329](#)
  - embPolygon\_reduceByNth, [329](#)
  - generate\_dragon\_curve, [329](#)
  - greedy\_algorithm, [329](#)
  - hilbert\_curve, [329](#)
  - hilbert\_curve\_l\_system, [330](#)
  - join\_short\_stitches, [330](#)
  - lindenmayer\_system, [330](#)
  - rules, [330](#)
  - save\_points\_to\_pattern, [330](#)
  - threshold\_method, [330](#)
- filled
  - Geometry, [138](#)
- findIndex
  - Geometry, [115](#)
- findMdiWindow
  - MainWindow, [153](#)
- firstDifatSectorLocation
  - \_bcf\_file\_header, [48](#)
- firstDirectorySectorLocation
  - \_bcf\_file\_header, [48](#)
- firstMiniFATSectorLocation
  - \_bcf\_file\_header, [48](#)
- flag
  - EmbGeometry\_, [90](#)
- FLAG\_CIRCLE
  - main.c, [394](#)
- FLAG\_CIRCLE\_SHORT
  - main.c, [395](#)
- FLAG\_COMBINE
  - main.c, [395](#)
- FLAG\_CROSS\_STITCH
  - main.c, [395](#)
- FLAG\_ELLIPSE
  - main.c, [395](#)
- FLAG\_ELLIPSE\_SHORT
  - main.c, [395](#)
- FLAG\_FILL
  - main.c, [395](#)
- FLAG\_FILL\_SHORT
  - main.c, [395](#)
- FLAG\_FORMATS
  - main.c, [395](#)
- FLAG\_FORMATS\_SHORT
  - main.c, [395](#)
- FLAG\_FULL\_TEST\_SUITE
  - main.c, [395](#)
- FLAG\_HELP
  - main.c, [395](#)
- FLAG\_HELP\_SHORT
  - main.c, [395](#)
- FLAG\_HILBERT\_CURVE
  - main.c, [395](#)
- FLAG\_LINE
  - main.c, [395](#)
- FLAG\_LINE\_SHORT
  - main.c, [395](#)
- FLAG\_POLYGON
  - main.c, [395](#)
- FLAG\_POLYGON\_SHORT
  - main.c, [395](#)
- FLAG\_POLYLINE
  - main.c, [395](#)
- FLAG\_POLYLINE\_SHORT
  - main.c, [396](#)
- FLAG\_QUIET
  - main.c, [396](#)
- FLAG\_QUIET\_SHORT
  - main.c, [396](#)
- FLAG\_RENDER
  - main.c, [396](#)
- FLAG\_RENDER\_SHORT
  - main.c, [396](#)
- FLAG\_SATIN
  - main.c, [396](#)
- FLAG\_SATIN\_SHORT
  - main.c, [396](#)
- FLAG\_SIERPINSKI\_TRIANGLE
  - main.c, [396](#)
- FLAG\_SIMULATE
  - main.c, [396](#)
- FLAG\_STITCH
  - main.c, [396](#)
- FLAG\_STITCH\_SHORT
  - main.c, [396](#)

- FLAG\_TEST
  - main.c, [396](#)
- FLAG\_TO
  - main.c, [396](#)
- FLAG\_TO\_SHORT
  - main.c, [396](#)
- FLAG\_VERBOSE
  - main.c, [396](#)
- FLAG\_VERBOSE\_SHORT
  - main.c, [396](#)
- FLAG\_VERSION
  - main.c, [396](#)
- FLAG\_VERSION\_SHORT
  - main.c, [396](#)
- flagList
  - EmbPath\_, [96](#)
- flags
  - EmbStitch\_, [101](#)
- Flared
  - Geometry, [111](#)
- Fletching
  - Geometry, [111](#)
- floatingChanged
  - CmdPrompt, [58](#)
- floatingChangedToolBar
  - MainWindow, [154](#)
- focusWidget
  - PropertyEditor, [185](#)
  - UndoEditor, [217](#)
- forceRepaint
  - MdiArea, [166](#)
  - SelectBox, [198](#)
- format\_100.c
  - read100, [335](#)
  - write100, [335](#)
- format\_10o.c
  - read10o, [336](#)
  - write10o, [336](#)
- format\_art.c
  - readArt, [336](#)
  - writeArt, [336](#)
- format\_bmc.c
  - readBmc, [337](#)
  - writeBmc, [337](#)
- format\_bro.c
  - readBro, [337](#)
  - writeBro, [337](#)
- format\_cnd.c
  - readCnd, [338](#)
  - writeCnd, [338](#)
- format\_col.c
  - readCol, [339](#)
  - writeCol, [339](#)
- format\_csd.c
  - \_subMask, [340](#)
  - \_xorMask, [340](#)
  - BuildDecryptionTable, [339](#)
  - csd\_decryptArray, [340](#)
  - CsdSubMaskSize, [339](#)
  - CsdXorMaskSize, [339](#)
  - DecodeCsdByte, [340](#)
  - readCsd, [340](#)
  - writeCsd, [340](#)
- format\_csv.c
  - csvStitchFlagToStr, [341](#)
  - csvStrToStitchFlag, [341](#)
  - readCsv, [341](#)
  - writeCsv, [341](#)
- format\_dat.c
  - readDat, [341](#)
  - writeDat, [341](#)
- format\_dem.c
  - readDem, [342](#)
  - writeDem, [342](#)
- format\_dsb.c
  - readDsb, [343](#)
  - writeDsb, [343](#)
- format\_dst.c
  - cci, [344](#)
  - decode\_record\_flags, [344](#)
  - encode\_record, [345](#)
  - readDst, [345](#)
  - set\_dst\_variable, [345](#)
  - writeDst, [345](#)
- format\_dsz.c
  - readDsz, [345](#)
  - writeDsz, [345](#)
- format\_dxf.c
  - readDxf, [346](#)
  - readLine, [346](#)
  - writeDxf, [346](#)
- format\_edr.c
  - readEdr, [346](#)
  - writeEdr, [346](#)
- format\_emd.c
  - emdDecode, [347](#)
  - readEmd, [347](#)
  - writeEmd, [347](#)
- format\_exp.c
  - expDecode, [347](#)
  - readExp, [347](#)
  - writeExp, [348](#)
- format\_exy.c
  - decode\_exy\_flags, [348](#)
  - readExy, [348](#)
  - writeExy, [348](#)
- format\_eyc.c
  - readEys, [348](#)
  - writeEys, [349](#)
- format\_fxy.c
  - readFxy, [349](#)
  - writeFxy, [349](#)
- format\_gc.c
  - readGc, [349](#)
  - writeGc, [350](#)
- format\_gnc.c

- readGnc, [350](#)
- writeGnc, [350](#)
- format\_gt.c
  - readGt, [350](#)
  - writeGt, [351](#)
- format\_hus.c
  - husCompressData, [351](#)
  - husDecodeByte, [351](#)
  - husDecodeStitchType, [351](#)
  - husDecompressData, [351](#)
  - husEncodeByte, [351](#)
  - husEncodeStitchType, [351](#)
  - readHus, [352](#)
  - writeHus, [352](#)
- format\_inb.c
  - readInb, [352](#)
  - writeInb, [352](#)
- format\_inf.c
  - readInf, [352](#)
  - writeInf, [353](#)
- format\_jef.c
  - jefDecode, [353](#)
  - jefEncode, [353](#)
  - jefGetHoopSize, [353](#)
  - jefSetHoopFromId, [353](#)
  - read\_hoop, [353](#)
  - readJef, [354](#)
  - writeJef, [354](#)
- format\_ksm.c
  - ksmEncode, [354](#)
  - readKsm, [354](#)
  - writeKsm, [354](#)
- format\_max.c
  - max\_header, [355](#)
  - readMax, [355](#)
  - writeMax, [355](#)
- format\_mit.c
  - readMit, [355](#)
  - writeMit, [356](#)
- format\_new.c
  - readNew, [356](#)
  - writeNew, [356](#)
- format\_ofm.c
  - ofmDecode, [357](#)
  - ofmReadBlockHeader, [357](#)
  - ofmReadClass, [357](#)
  - ofmReadColorChange, [357](#)
  - ofmReadExpanded, [357](#)
  - ofmReadLibrary, [357](#)
  - ofmReadThreads, [357](#)
  - readOfm, [357](#)
  - writeOfm, [357](#)
- format\_pcd.c
  - readPcd, [358](#)
  - writePcd, [358](#)
- format\_pcm.c
  - readPcm, [358](#)
  - writePcm, [358](#)
- format\_pcq.c
  - readPcq, [359](#)
  - writePcq, [359](#)
- format\_pcs.c
  - readPcs, [359](#)
  - writePcs, [359](#)
- format\_pec.c
  - pecEncode, [360](#)
  - pecEncodeJump, [360](#)
  - pecEncodeStop, [360](#)
  - readPec, [360](#)
  - readPecStitches, [360](#)
  - writelnImage, [360](#)
  - writePec, [360](#)
  - writePecStitches, [360](#)
- format\_pel.c
  - readPel, [361](#)
  - writePel, [361](#)
- format\_pem.c
  - readPem, [361](#)
  - writePem, [361](#)
- format\_pes.c
  - pes\_version, [364](#)
  - pes\_version\_strings, [364](#)
  - pesWriteEmbOneSection, [362](#)
  - pesWriteSewSegSection, [362](#)
  - readDescriptions, [362](#)
  - readFeatherPatterns, [362](#)
  - readHoopName, [362](#)
  - readImageString, [363](#)
  - readMotifPatterns, [363](#)
  - readPes, [363](#)
  - readPESHeaderV10, [363](#)
  - readPESHeaderV5, [363](#)
  - readPESHeaderV6, [363](#)
  - readPESHeaderV7, [363](#)
  - readPESHeaderV8, [363](#)
  - readPESHeaderV9, [363](#)
  - readProgrammableFills, [363](#)
  - readThreads, [363](#)
  - writePes, [364](#)
- format\_phb.c
  - readPhb, [364](#)
  - writePhb, [364](#)
- format\_phc.c
  - readPhc, [365](#)
  - writePhc, [365](#)
- format\_plt.c
  - readPlt, [365](#)
  - writePlt, [365](#)
- format\_rgb.c
  - readRgb, [366](#)
  - writeRgb, [366](#)
- format\_sew.c
  - readSew, [366](#)
  - sewDecode, [366](#)
  - writeSew, [366](#)
- format\_shv.c



- readShv, [367](#)
- shvDecode, [367](#)
- shvDecodeShort, [367](#)
- writeShv, [367](#)
- format\_sst.c
  - readSst, [367](#)
  - writeSst, [368](#)
- format\_stx.c
  - readStx, [368](#)
  - stxReadThread, [368](#)
  - writeStx, [368](#)
- format\_svg.c
  - attributeList, [369](#)
  - current\_element\_id, [369](#)
  - currentAttribute, [369](#)
  - currentValue, [369](#)
  - n\_attributes, [369](#)
  - readSvg, [369](#)
  - svgCreator, [369](#)
  - svgExpect, [369](#)
  - svgMultiValue, [369](#)
  - writeSvg, [369](#)
- format\_t01.c
  - readT01, [370](#)
  - writeT01, [370](#)
- format\_t09.c
  - readT09, [370](#)
  - writeT09, [370](#)
- format\_tap.c
  - decode\_tap\_record\_flags, [371](#)
  - encode\_tap\_record, [371](#)
  - readTap, [371](#)
  - writeTap, [371](#)
- format\_thr.c
  - readThr, [371](#)
  - writeThr, [372](#)
- format\_txt.c
  - readTxt, [372](#)
  - writeTxt, [372](#)
- format\_u00.c
  - readU00, [372](#)
  - writeU00, [373](#)
- format\_u01.c
  - readU01, [373](#)
  - writeU01, [373](#)
- format\_vip.c
  - readVip, [374](#)
  - vipCompressData, [374](#)
  - vipDecodeByte, [374](#)
  - vipDecodeStitchType, [374](#)
  - vipDecodingTable, [374](#)
  - vipDecompressData, [374](#)
  - vipEncodeByte, [374](#)
  - vipEncodeStitchType, [374](#)
  - writeVip, [374](#)
- format\_vp3.c
  - readVp3, [375](#)
  - vp3Decode, [375](#)
  - vp3DecodeInt16, [375](#)
  - vp3PatchByteCount, [375](#)
  - vp3ReadHoopSection, [375](#)
  - vp3ReadString, [376](#)
  - vp3WriteString, [376](#)
  - vp3WriteStringLen, [376](#)
  - writeVp3, [376](#)
- format\_xxx.c
  - readXxx, [376](#)
  - writeXxx, [376](#)
  - xxxDecodeByte, [376](#)
  - xxxEncodeDesign, [377](#)
  - xxxEncodeStitch, [377](#)
  - xxxEncodeStop, [377](#)
- format\_zsk.c
  - readZsk, [377](#)
  - writeZsk, [377](#)
- formatFilterOpen
  - MainWindow, [163](#)
- formatFilterSave
  - MainWindow, [164](#)
- formats.c
  - binaryWriteInt, [332](#)
  - binaryWriteIntBE, [332](#)
  - binaryWriteShort, [332](#)
  - binaryWriteUInt, [332](#)
  - binaryWriteUIntBE, [332](#)
  - binaryWriteUShort, [332](#)
  - binaryWriteUShortBE, [332](#)
  - emb\_identify\_format, [333](#)
  - embFormat\_getExtension, [333](#)
  - embPattern\_read, [333](#)
  - embPattern\_readAuto, [333](#)
  - embPattern\_write, [333](#)
  - embPattern\_writeAuto, [333](#)
  - formatTable, [334](#)
  - fpad, [334](#)
  - fread\_int16, [334](#)
  - fread\_int32\_be, [334](#)
  - fread\_uint16, [334](#)
  - imageWithFrame, [335](#)
  - safe\_free, [334](#)
- formatTable
  - embroidery.h, [268](#)
  - formats.c, [334](#)
- formatType
  - SaveObject, [197](#)
- Fortron, [304](#), [349](#)
- fourier\_series
  - objects.cpp, [488](#)
- fpad
  - embroidery\_internal.h, [299](#)
  - formats.c, [334](#)
- fread\_int16
  - embroidery\_internal.h, [299](#)
  - formats.c, [334](#)
- fread\_int32\_be
  - embroidery\_internal.h, [300](#)



- formats.c, 334
- fread\_uint16
  - embroidery\_internal.h, 300
  - formats.c, 334
- fromCenter
  - UndoableCommand, 215
- fromTransform
  - UndoableCommand, 215
- Fufu\_Polyester
  - embroidery.h, 249
- Fufu\_Rayon
  - embroidery.h, 249
- full\_test\_matrix
  - embroidery.h, 266
- FUNCTION\_TYPE
  - embroidermodder.h, 418
- functions.c
  - degrees, 385
  - emb\_round, 385
  - radians, 385
- fx, 9, 304, 349
- g
  - EmbColor\_, 84
- general\_props
  - settings-dialog.cpp, 492
- generate\_dragon\_curve
  - fill.c, 329
- Geometry, 105
  - ~Geometry, 114
  - allGripPoints, 114
  - arcEndPoint, 138
  - arcMidPoint, 138
  - arcStartPoint, 138
  - ArrowStyle, 110
  - arrowStyleAngle, 138
  - arrowStyleLength, 138
  - arrowStylePath, 138
  - boundingRect, 114
  - Box, 110
  - calculateArcData, 115
  - circle\_click, 115
  - Closed, 110
  - curved, 138
  - Dot, 110
  - drawRubberLine, 115
  - filled, 138
  - findIndex, 115
  - Flared, 111
  - Fletching, 111
  - Geometry, 111–114
  - gripEdit, 116
  - gripIndex, 138
  - init, 116
  - init\_arc, 116
  - init\_circle, 116
  - init\_ellipse, 117
  - init\_line, 117
  - init\_path, 117
  - init\_point, 118
  - init\_rect, 118
  - init\_text\_single, 119
  - lineStyle, 110
  - lineStyleAngle, 138
  - lineStyleLength, 138
  - lineStylePath, 139
  - lwtPen, 139
  - mouseSnapPoint, 119
  - NoArrow, 110
  - NoLine, 111
  - normalPath, 139
  - objectAngle, 119
  - objectArcLength, 120
  - objectArea, 120
  - objectBottomLeft, 120
  - objectBottomRight, 120
  - objectCenter, 120
  - objectChord, 120
  - objectCircumference, 121
  - objectClockwise, 121
  - objectCopyPath, 121
  - objectDelta, 121
  - objectDiameter, 121
  - objectDiameterMajor, 121
  - objectDiameterMinor, 121
  - objectEndAngle, 122
  - objectEndPoint, 122
  - objectEndPoint1, 122
  - objectEndPoint2, 122
  - objectHeight, 122
  - objectIncludedAngle, 123
  - objectLength, 123
  - objectLineType, 123
  - objectLineWeight, 123
  - objectMidPoint, 123
  - objectPos, 123
  - objectQuadrant0, 123
  - objectQuadrant180, 124
  - objectQuadrant270, 124
  - objectQuadrant90, 124
  - objectRadius, 124
  - objectRadiusMajor, 124
  - objectRadiusMinor, 124
  - objectRubberPoint, 124
  - objectRubberText, 125
  - objectSavePath, 125
  - objectSavePathList, 125
  - objectStartAngle, 125
  - objectStartPoint, 125
  - objectTopLeft, 126
  - objectTopRight, 126
  - objectWidth, 126
  - objectX, 126
  - objectX1, 126
  - objectX2, 126
  - objectY, 126
  - objectY1, 127

- objectY2, [127](#)
- objID, [139](#)
- objLine, [139](#)
- objPen, [139](#)
- objRubberMode, [139](#)
- objRubberPoints, [139](#)
- objRubberTexts, [139](#)
- objText, [139](#)
- objTextBackward, [139](#)
- objTextFont, [140](#)
- objTextJustify, [140](#)
- objTextPath, [140](#)
- objTextUpsideDown, [140](#)
- Open, [110](#)
- paint, [127](#)
- properties, [140](#)
- realRender, [127](#)
- rect, [127](#)
- script\_click, [127](#)
- script\_context, [128](#)
- script\_main, [128](#)
- script\_prompt, [128](#)
- setLine, [128](#)
- setObjectArea, [128](#)
- setObjectCenter, [130](#)
- setObjectCenterX, [130](#)
- setObjectCenterY, [130](#)
- setObjectCircumference, [130](#)
- setObjectDiameter, [130](#)
- setObjectDiameterMajor, [130](#)
- setObjectDiameterMinor, [130](#)
- setObjectEndAngle, [131](#)
- setObjectEndPoint, [131](#)
- setObjectEndPoint1, [131](#)
- setObjectEndPoint2, [131](#)
- setObjectLineWeight, [132](#)
- setObjectMidPoint, [132](#)
- setObjectPos, [132](#)
- setObjectRadius, [132](#)
- setObjectRadiusMajor, [132](#)
- setObjectRadiusMinor, [133](#)
- setObjectRect, [133](#)
- setObjectSize, [133](#)
- setObjectStartAngle, [133](#)
- setObjectStartPoint, [133](#)
- setObjectText, [133](#)
- setObjectTextBackward, [133](#)
- setObjectTextBold, [134](#)
- setObjectTextFont, [134](#)
- setObjectTextItalic, [134](#)
- setObjectTextJustify, [134](#)
- setObjectTextOverline, [134](#)
- setObjectTextSize, [135](#)
- setObjectTextStrikeOut, [135](#)
- setObjectTextStyle, [135](#)
- setObjectTextUnderline, [135](#)
- setObjectTextUpsideDown, [136](#)
- setObjectX, [136](#)
- setObjectY, [136](#)
- setRect, [136](#)
- subPathList, [136](#)
- Tick, [110](#)
- Type, [140](#)
- type, [136](#)
- updateArcRect, [136](#)
- updateLeader, [137](#)
- updatePath, [137](#)
- updateRubber, [137](#)
- vulcanize, [137](#)
- x\_values, [140](#)
- y\_values, [140](#)
- geometry
  - EmbArray\_, [81](#)
  - EmbLayer\_, [93](#)
  - EmbPattern\_, [97](#)
- geometry.c
  - embGeometry\_boundingRect, [378](#)
  - embGeometry\_free, [378](#)
  - embGeometry\_init, [378](#)
  - embGeometry\_move, [378](#)
  - embGeometry\_vulcanize, [378](#)
- get\_bool
  - embroidermodder.h, [423](#)
  - interface.cpp, [450](#)
- get\_int
  - embroidermodder.h, [423](#)
  - interface.cpp, [450](#)
- get\_n\_reals
  - interface.cpp, [451](#)
- get\_qstr
  - embroidermodder.h, [423](#)
  - interface.cpp, [451](#)
- get\_real
  - embroidermodder.h, [423](#)
  - interface.cpp, [451](#)
- get\_str
  - embroidermodder.h, [423](#)
  - interface.cpp, [451](#)
- get\_str\_list
  - embroidermodder.h, [423](#)
  - interface.cpp, [451](#)
- get\_trim\_bounds
  - main.c, [401](#)
- get\_uint
  - embroidermodder.h, [423](#)
  - interface.cpp, [451](#)
- getArcCenter
  - arc.c, [382](#)
  - embroidery.h, [266](#)
- getArcDataFromBulge
  - arc.c, [382](#)
  - embroidery.h, [266](#)
- getCircleCircleIntersections
  - circle.c, [383](#)
  - embroidery.h, [267](#)
- getCircleTangentPoints

- circle.c, [383](#)
- embroidery.h, [267](#)
- getCurrentColor
  - MainWindow, [154](#)
- getCurrentLayer
  - MainWindow, [154](#)
- getCurrentLineType
  - MainWindow, [154](#)
- getCurrentLineWeight
  - MainWindow, [154](#)
- GetFile
  - embroidery\_internal.h, [300](#)
  - main.c, [401](#)
- getFileSeparator
  - MainWindow, [155](#)
- getInfo
  - EmbDetailsDialog, [86](#)
- getShortCurrentFile
  - MdiWindow, [172](#)
- getUndoStack
  - View, [222](#)
- gnc, [9](#), [304](#), [350](#)
- Gold Thread, [304](#), [351](#)
- Great Notions, [304](#), [350](#)
- greedy\_algorithm
  - fill.c, [329](#)
- GREEN\_TERM\_COLOR
  - embroidery\_internal.h, [289](#)
- gridColor
  - View, [227](#)
- gridPath
  - View, [227](#)
- gripBaseObj
  - View, [227](#)
- gripColorCool
  - View, [227](#)
- gripColorHot
  - View, [228](#)
- gripEdit
  - Geometry, [116](#)
- gripIndex
  - Geometry, [138](#)
- grippingActive
  - View, [228](#)
- gripSize
  - View, [228](#)
- group\_box\_data
  - property-editor.cpp, [490](#)
- group\_box\_types
  - property-editor.cpp, [490](#)
- groupBoxes
  - embroidermodder.h, [430](#)
  - mainwindow.cpp, [486](#)
- gscene
  - MdiWindow, [178](#)
  - SaveObject, [197](#)
  - View, [228](#)
- gt, [9](#), [304](#), [351](#)
- gview
  - MdiWindow, [178](#)
  - UndoableCommand, [215](#)
- handleMoved
  - CmdPromptHandle, [64](#)
- handlePressed
  - CmdPromptHandle, [64](#)
- handleReleased
  - CmdPromptHandle, [64](#)
- Happy, [371](#)
- hashDeletedObjects
  - View, [228](#)
- haveExtraDIFATSectors
  - main.c, [401](#)
- header
  - \_bcf\_file, [46](#)
  - em2\_dev\_script, [42](#)
- height
  - \_vp3Hoop, [50](#)
  - EmblImage\_, [92](#)
- help\_action
  - mainwindow.cpp, [473](#)
- Hemingworth\_Polyester
  - embroidery.h, [249](#)
- hex\_code
  - thread\_color\_, [211](#)
- hideAllGroups
  - PropertyEditor, [183](#)
- hideUnimplemented
  - MainWindow, [155](#)
- hilbert\_curve
  - embroidery.h, [267](#)
  - fill.c, [329](#)
- hilbert\_curve\_I\_system
  - fill.c, [330](#)
- historyAppended
  - CmdPrompt, [58](#)
  - CmdPromptHistory, [67](#)
- home
  - EmbPattern\_, [97](#)
- HOOP\_110X110
  - embroidery\_internal.h, [289](#)
- HOOP\_126X110
  - embroidery\_internal.h, [289](#)
- HOOP\_140X200
  - embroidery\_internal.h, [290](#)
- HOOP\_230X200
  - embroidery\_internal.h, [290](#)
- HOOP\_50X50
  - embroidery\_internal.h, [290](#)
- hoop\_height
  - EmbPattern\_, [97](#)
- hoop\_padding, [140](#)
  - bottom, [141](#)
  - left, [141](#)
  - right, [141](#)
  - top, [141](#)
- hoop\_width

- EmbPattern\_, 97
- hoopSize
  - ThredHeader\_, 212
- hoopX
  - ThredExtension\_, 211
- hoopY
  - ThredExtension\_, 211
- hour
  - EmbTime\_, 104
- Huffman, 141
  - default\_value, 141
  - lengths, 141
  - nlengths, 142
  - ntable, 142
  - table, 142
  - table\_width, 142
- huffman
  - embroidery\_internal.h, 292
- huffman\_build\_table
  - compress.c, 235
  - embroidery\_internal.h, 300
- huffman\_lookup
  - compress.c, 235
- huffman\_lookup\_data
  - compress.c, 236
- huffman\_table\_lookup
  - embroidery\_internal.h, 300
- hus, 9, 351
- hus\_compress
  - compress.c, 235
  - embroidery\_internal.h, 300
- hus\_decompress
  - compress.c, 236
  - embroidery\_internal.h, 300
- hus\_thread
  - embroidery.h, 249
- husCompressData
  - format\_hus.c, 351
- husDecodeByte
  - format\_hus.c, 351
- husDecodeStitchType
  - format\_hus.c, 351
- husDecompressData
  - format\_hus.c, 351
- husEncodeByte
  - format\_hus.c, 351
- husEncodeStitchType
  - format\_hus.c, 351
- Husqvarna Viking, 351, 367
- husThreads
  - embroidery.h, 268
  - thread-color.c, 409
- i
  - Node\_, 179
- icon\_action
  - mainwindow.cpp, 474
- iconDir
  - PropertyEditor, 185
- UndoEditor, 217
- iconResize
  - MainWindow, 155
- iconSize
  - PropertyEditor, 186
  - UndoEditor, 217
- id
  - UndoableCommand, 214
- image.c
  - image\_diff, 391
  - writelnImage, 392
- image\_diff
  - image.c, 391
- ImageWidget, 142
  - ~ImageWidget, 143
  - ImageWidget, 142
  - img, 144
  - load, 143
  - paintEvent, 143
  - save, 143
- imageWithFrame
  - embroidery\_internal.h, 316
  - formats.c, 335
- img
  - ImageWidget, 144
- imgWidget
  - PreviewDialog, 180
- inb, 9, 305, 352
- Inbro, 305, 352
- include\_action
  - mainwindow.cpp, 474
- inf, 305, 353
- init
  - Geometry, 116
- init\_action
  - mainwindow.cpp, 474
- init\_arc
  - Geometry, 116
- init\_circle
  - Geometry, 116
- init\_ellipse
  - Geometry, 117
- init\_line
  - Geometry, 117
- init\_path
  - Geometry, 117
- init\_point
  - Geometry, 118
- init\_rect
  - Geometry, 118
- init\_text\_single
  - Geometry, 119
- input\_data
  - Compress, 79
- input\_length
  - Compress, 79
- INT\_TYPE
  - embroidermodder.h, 418

- interface.cpp
  - add\_to\_path, [450](#)
  - debug\_message, [450](#)
  - degrees\_\_, [450](#)
  - get\_bool, [450](#)
  - get\_int, [450](#)
  - get\_n\_reals, [451](#)
  - get\_qstr, [451](#)
  - get\_real, [451](#)
  - get\_str, [451](#)
  - get\_str\_list, [451](#)
  - get\_uint, [451](#)
  - make\_checkbox, [451](#)
  - make\_spinbox, [451](#)
  - make\_ui\_element, [451](#)
  - node\_bool, [451](#)
  - node\_int, [452](#)
  - node\_qstr, [452](#)
  - node\_real, [452](#)
  - node\_str, [452](#)
  - node\_str\_list, [453](#)
  - node\_uint, [453](#)
  - operator\*, [453](#)
  - operator+, [453](#)
  - operator-, [453](#)
  - radians\_\_, [453](#)
  - set\_enabled, [455](#)
  - set\_visibility, [455](#)
  - to\_EmbVector, [455](#)
  - to\_qlist, [455](#)
  - to\_QPointF, [456](#)
  - to\_string\_vector, [456](#)
  - to\_vector, [456](#)
  - tokenize, [456](#)
  - translate\_str, [456](#)
- is\_int\_action
  - mainwindow.cpp, [474](#)
- Isacord\_Polyester
  - embroidery.h, [249](#)
- Isafil\_Rayon
  - embroidery.h, [249](#)
- isBlinking
  - CmdPromptInput, [76](#)
- isCommandActive
  - MainWindow, [155](#)
- isLwtEnabled
  - View, [222](#)
- isRealEnabled
  - View, [222](#)
- isShiftPressed
  - MainWindow, [155](#)
- Janome, [353](#), [366](#)
- jef, [9](#), [353](#)
- jef\_thread
  - embroidery.h, [249](#)
- jefDecode
  - format\_jef.c, [353](#)
- jefEncode
  - format\_jef.c, [353](#)
- jefGetHoopSize
  - format\_jef.c, [353](#)
- jefSetHoopFromId
  - format\_jef.c, [353](#)
- jefThreads
  - embroidery.h, [268](#)
  - thread-color.c, [409](#)
- join\_short\_stitches
  - fill.c, [330](#)
- JUMP
  - embroidery.h, [249](#)
- ksm, [9](#), [354](#)
- ksmEncode
  - format\_ksm.c, [354](#)
- L\_system
  - embroidery.h, [253](#)
- labels
  - embroidermodder.h, [430](#)
  - mainwindow.cpp, [486](#)
- labelTipOfTheDay
  - mainwindow.cpp, [486](#)
- lastCmd
  - CmdPromptInput, [76](#)
- layer
  - EmbPattern\_, [97](#)
- layer\_manager\_action
  - mainwindow.cpp, [474](#)
- layer\_previous\_action
  - mainwindow.cpp, [475](#)
- LayerManager, [144](#)
  - ~LayerManager, [144](#)
  - addLayer, [145](#)
  - LayerManager, [144](#)
  - layerModel, [145](#)
  - layerModelSorted, [145](#)
  - treeView, [145](#)
- layerModel
  - LayerManager, [145](#)
- layerModelSorted
  - LayerManager, [145](#)
- layerSelector
  - MainWindow, [164](#)
- layerSelectorIndexChanged
  - MainWindow, [155](#)
- layoutState
  - MainWindow, [164](#)
- left
  - \_vp3Hoop, [51](#)
  - EmbRect\_, [99](#)
  - hoop\_padding, [141](#)
- left2
  - \_vp3Hoop, [51](#)
- leftBrush
  - SelectBox, [199](#)
- leftBrushColor
  - SelectBox, [199](#)

- leftPen
  - SelectBox, [199](#)
- leftPenColor
  - SelectBox, [199](#)
- leftSiblingId
  - \_bcf\_directory\_entry, [44](#)
- length
  - EmbArray\_, [81](#)
  - EmbSatinOutline\_, [100](#)
  - ThredHeader\_, [212](#)
- lengths
  - Huffman, [141](#)
- LIBEMBROIDERY\_EMBEDDED\_VERSION
  - embroidery.h, [249](#)
- lindenmayer\_system
  - embroidery.h, [267](#)
  - fill.c, [330](#)
- line
  - EmbGeometry\_, [90](#)
- line.c
  - embLine\_intersectionPoint, [386](#)
  - embLine\_normalVector, [386](#)
  - embLine\_toVector, [386](#)
- lineEdits
  - embroidermodder.h, [430](#)
  - mainwindow.cpp, [486](#)
- lineStyle
  - Geometry, [110](#)
- lineStyleAngle
  - Geometry, [138](#)
- lineStyleLength
  - Geometry, [138](#)
- lineStylePath
  - Geometry, [139](#)
- LINETO
  - embroidery\_internal.h, [290](#)
- lineType
  - EmbGeometry\_, [90](#)
  - EmbLine\_, [94](#)
  - EmbPath\_, [96](#)
  - EmbPoint\_, [98](#)
- linetypeSelector
  - MainWindow, [164](#)
- linetypeSelectorIndexChanged
  - MainWindow, [156](#)
- lineweightSelector
  - MainWindow, [164](#)
- lineweightSelectorIndexChanged
  - MainWindow, [156](#)
- listMdiWin
  - MainWindow, [164](#)
- load
  - ImageWidget, [143](#)
- load\_group\_box\_data\_from\_table
  - property-editor.cpp, [489](#)
- loadFatFromSector
  - embroidery\_internal.h, [301](#)
  - main.c, [401](#)
- loadFile
  - MdiWindow, [172](#)
- loadFormats
  - MainWindow, [156](#)
- loadRulerSettings
  - View, [222](#)
- logPromptInput
  - MainWindow, [156](#)
  - MdiWindow, [174](#)
- LSYSTEM, [146](#)
  - alphabet, [146](#)
  - axiom, [146](#)
  - constants, [146](#)
  - rules, [146](#)
- lwtPen
  - Geometry, [139](#)
- Madeira\_Polyester
  - embroidery.h, [249](#)
- Madeira\_Rayon
  - embroidery.h, [249](#)
- magicCode
  - VipHeader\_, [231](#)
- main
  - embroidermodder.cpp, [411](#)
- main.c
  - bcf\_difat\_create, [397](#)
  - bcf\_directory\_free, [397](#)
  - bcf\_file\_free, [397](#)
  - bcfFile\_read, [397](#)
  - bcfFileFat\_create, [397](#)
  - bcfFileHeader\_read, [397](#)
  - binaryReadString, [397](#)
  - binaryReadUnicodeString, [398](#)
  - black\_thread, [403](#)
  - check\_header\_present, [398](#)
  - CompoundFileDirectory, [398](#)
  - CompoundFileDirectoryEntry, [398](#)
  - copy\_trim, [398](#)
  - difatEntriesInHeader, [403](#)
  - emb\_error, [403](#)
  - emb\_optOut, [398](#)
  - emb\_readline, [399](#)
  - emb\_verbose, [403](#)
  - embArc\_print, [399](#)
  - embColor\_distance, [399](#)
  - embColor\_read, [399](#)
  - embColor\_write, [399](#)
  - embConstantPi, [403](#)
  - embSatinOutline\_generateSatinOutline, [399](#)
  - embSatinOutline\_renderStitches, [399](#)
  - embThread\_findNearestColor, [400](#)
  - embThread\_findNearestThread, [400](#)
  - embThread\_getRandom, [400](#)
  - embTime\_initNow, [400](#)
  - embTime\_time, [400](#)
  - embVector\_print, [400](#)
  - entriesInDifatSector, [401](#)
  - FLAG\_CIRCLE, [394](#)

- FLAG\_CIRCLE\_SHORT, 395
- FLAG\_COMBINE, 395
- FLAG\_CROSS\_STITCH, 395
- FLAG\_ELLIPSE, 395
- FLAG\_ELLIPSE\_SHORT, 395
- FLAG\_FILL, 395
- FLAG\_FILL\_SHORT, 395
- FLAG\_FORMATS, 395
- FLAG\_FORMATS\_SHORT, 395
- FLAG\_FULL\_TEST\_SUITE, 395
- FLAG\_HELP, 395
- FLAG\_HELP\_SHORT, 395
- FLAG\_HILBERT\_CURVE, 395
- FLAG\_LINE, 395
- FLAG\_LINE\_SHORT, 395
- FLAG\_POLYGON, 395
- FLAG\_POLYGON\_SHORT, 395
- FLAG\_POLYLINE, 395
- FLAG\_POLYLINE\_SHORT, 396
- FLAG\_QUIET, 396
- FLAG\_QUIET\_SHORT, 396
- FLAG\_RENDER, 396
- FLAG\_RENDER\_SHORT, 396
- FLAG\_SATIN, 396
- FLAG\_SATIN\_SHORT, 396
- FLAG\_SIERPINSKI\_TRIANGLE, 396
- FLAG\_SIMULATE, 396
- FLAG\_STITCH, 396
- FLAG\_STITCH\_SHORT, 396
- FLAG\_TEST, 396
- FLAG\_TO, 396
- FLAG\_TO\_SHORT, 396
- FLAG\_VERBOSE, 396
- FLAG\_VERBOSE\_SHORT, 396
- FLAG\_VERSION, 396
- FLAG\_VERSION\_SHORT, 396
- get\_trim\_bounds, 401
- GetFile, 401
- haveExtraDIFATsectors, 401
- loadFatFromSector, 401
- NUM\_FLAGS, 397
- parseDIFATsectors, 401
- parseDirectoryEntryName, 401
- parseTime, 401
- readFullSector, 402
- readNextSector, 402
- sectorSize, 402
- seekToSector, 402
- sizeofChainingEntryAtEndOfDifatSector, 403
- sizeofDifatEntry, 403
- sizeofDirectoryEntry, 403
- sizeofFatEntry, 403
- stringInArray, 402
- WHITESPACE, 403
- write\_24bit, 402
- mainWidget
  - EmbDetailsDialog, 86
- MainWindow, 146
  - ~MainWindow, 150
  - about, 150
  - activeCommand, 151
  - activeMdiWindow, 151
  - activeUndoStack, 151
  - buttonTipOfTheDayClicked, 151
  - checkForUpdates, 151
  - closeEvent, 151
  - closeToolBar, 152
  - colorSelector, 163
  - colorSelectorIndexChanged, 152
  - create\_icon, 152
  - create\_toolbar, 152
  - createAllActions, 153
  - createAllMenus, 153
  - createAllToolbars, 153
  - cutCopyObjectList, 163
  - deletePressed, 153
  - docIndex, 163
  - escapePressed, 153
  - findMdiWindow, 153
  - floatingChangedToolBar, 154
  - formatFilterOpen, 163
  - formatFilterSave, 164
  - getCurrentColor, 154
  - getCurrentLayer, 154
  - getCurrentLineType, 154
  - getCurrentLineWeight, 154
  - getFileSeparator, 155
  - hideUnimplemented, 155
  - iconResize, 155
  - isCommandActive, 155
  - isShiftPressed, 155
  - layerSelector, 164
  - layerSelectorIndexChanged, 155
  - layoutState, 164
  - linetypeSelector, 164
  - linetypeSelectorIndexChanged, 156
  - linewidthSelector, 164
  - linewidthSelectorIndexChanged, 156
  - listMdiWin, 164
  - loadFormats, 156
  - logPromptInput, 156
  - MainWindow, 150
  - myFileSeparator, 164
  - newFile, 156
  - numOfDocs, 164
  - onCloseMdiWin, 156
  - onCloseWindow, 157
  - onWindowActivated, 157
  - openFile, 157
  - openFilesSelected, 157
  - openrecentfile, 157
  - pickAddModeToggled, 158
  - platformString, 158
  - promptHistoryAppended, 158
  - promptInputNext, 158
  - promptInputPrevious, 158

- quit, 158
- recentMenuAboutToShow, 158
- resizeEvent, 158
- saveasfile, 159
- savefile, 159
- setShiftPressed, 159
- setShiftReleased, 159
- setTextFont, 159
- setTextSize, 159
- settingsPrompt, 159
- setUndoCleanIcon, 160
- shiftKeyPressedState, 164
- stub\_testing, 160
- textFontSelector, 164
- textFontSelectorCurrentFontChanged, 160
- textSizeSelector, 164
- textSizeSelectorIndexChanged, 160
- tipOfTheDay, 160
- toggleGrid, 161
- toggleLwt, 161
- toggleRuler, 161
- updateAllViewBackgroundColors, 161
- updateAllViewCrossHairColors, 161
- updateAllViewGridColors, 161
- updateAllViewRulerColors, 162
- updateAllViewScrollBars, 162
- updateAllViewSelectBoxColors, 162
- updateMenuToolBarStatusbar, 162
- updatePickAddMode, 163
- windowMenuAboutToShow, 163
- windowMenuActivated, 163
- mainwindow-menus.cpp
  - create\_menu, 457
- mainwindow.cpp
  - \_mainWin, 485
  - about\_action, 464
  - actionHash, 485
  - activeScene, 464
  - activeView, 464
  - actuator, 464
  - add\_arc\_action, 465
  - add\_circle\_action, 465
  - add\_dim\_leader\_action, 465
  - add\_ellipse\_action, 465
  - add\_geometry\_action, 466
  - add\_horizontal\_dimension\_action, 466
  - add\_image\_action, 466
  - add\_infinite\_line\_action, 466
  - add\_line\_action, 466
  - add\_path\_action, 466
  - add\_point\_action, 466
  - add\_polygon\_action, 467
  - add\_polyline\_action, 467
  - add\_ray\_action, 467
  - add\_rectangle\_action, 467
  - add\_regular\_polygon\_action, 467
  - add\_rounded\_rectangle\_action, 467
  - add\_rubber\_action, 468
  - add\_slot\_action, 468
  - add\_text\_multi\_action, 468
  - add\_text\_single\_action, 468
  - add\_to\_selection\_action, 468
  - add\_triangle\_action, 469
  - add\_vertical\_dimension\_action, 469
  - alert\_action, 469
  - allow\_rubber\_action, 469
  - append\_history\_action, 469
  - append\_prompt\_history\_action, 470
  - blink\_prompt\_action, 470
  - calculate\_angle\_action, 470
  - calculate\_distance\_action, 470
  - changelog\_action, 470
  - checkBoxes, 485
  - checkBoxTipOfTheDay, 485
  - clear\_rubber\_action, 471
  - clear\_selection\_action, 471
  - comboBoxes, 485
  - command\_map, 486
  - config, 486
  - config\_tables, 486
  - construct\_command, 471
  - convert\_args\_to\_type, 471
  - copy\_action, 471
  - copy\_selected\_action, 472
  - cut\_action, 472
  - cut\_selected\_action, 472
  - day\_vision\_action, 472
  - debug\_action, 472
  - delete\_selected\_action, 472
  - design\_details\_action, 472
  - dialog, 486
  - disable\_action, 472
  - do\_nothing\_action, 473
  - dockPropEdit, 486
  - dockUndoEdit, 486
  - doubleSpinBoxes, 486
  - end\_action, 473
  - error\_action, 473
  - groupBoxes, 486
  - help\_action, 473
  - icon\_action, 474
  - include\_action, 474
  - init\_action, 474
  - is\_int\_action, 474
  - labels, 486
  - labelTipOfTheDay, 486
  - layer\_manager\_action, 474
  - layer\_previous\_action, 475
  - lineEdits, 486
  - make\_layer\_active\_action, 475
  - mdiArea, 486
  - menuHash, 486
  - messagebox\_action, 475
  - mirror\_selected\_action, 475
  - mouse\_x\_action, 475
  - mouse\_y\_action, 475



move\_selected\_action, 476  
new\_action, 476  
night\_vision\_action, 476  
no\_argument\_debug, 476  
num\_selected\_action, 476  
OBJ\_LTYPE\_CENTER, 463  
OBJ\_LTYPE\_CONT, 463  
OBJ\_LTYPE\_DOT, 463  
OBJ\_LTYPE\_FISHBONE, 463  
OBJ\_LTYPE\_HIDDEN, 463  
OBJ\_LTYPE\_PHANTOM, 463  
OBJ\_LTYPE\_RUNNING, 463  
OBJ\_LTYPE\_SATIN, 463  
OBJ\_LTYPE\_VALUES, 463  
OBJ\_LTYPE\_ZIGZAG, 463  
OBJ\_LWT\_01, 463  
OBJ\_LWT\_02, 463  
OBJ\_LWT\_03, 463  
OBJ\_LWT\_04, 463  
OBJ\_LWT\_05, 463  
OBJ\_LWT\_06, 463  
OBJ\_LWT\_07, 463  
OBJ\_LWT\_08, 463  
OBJ\_LWT\_09, 463  
OBJ\_LWT\_10, 463  
OBJ\_LWT\_11, 463  
OBJ\_LWT\_12, 463  
OBJ\_LWT\_13, 463  
OBJ\_LWT\_14, 463  
OBJ\_LWT\_15, 463  
OBJ\_LWT\_16, 463  
OBJ\_LWT\_17, 463  
OBJ\_LWT\_18, 463  
OBJ\_LWT\_19, 463  
OBJ\_LWT\_20, 463  
OBJ\_LWT\_21, 463  
OBJ\_LWT\_22, 464  
OBJ\_LWT\_23, 464  
OBJ\_LWT\_24, 464  
OBJ\_LWT\_BYBLOCK, 463  
OBJ\_LWT\_BYLAYER, 463  
OBJ\_LWT\_DEFAULT, 463  
OBJ\_LWT\_VALUES, 463  
OBJ\_SNAP\_APPINTERSECTION, 464  
OBJ\_SNAP\_CENTER, 464  
OBJ\_SNAP\_ENDPOINT, 464  
OBJ\_SNAP\_EXTENSION, 464  
OBJ\_SNAP\_INSERTION, 464  
OBJ\_SNAP\_INTERSECTION, 464  
OBJ\_SNAP\_MIDPOINT, 464  
OBJ\_SNAP\_NEAREST, 464  
OBJ\_SNAP\_NODE, 464  
OBJ\_SNAP\_NULL, 464  
OBJ\_SNAP\_PARALLEL, 464  
OBJ\_SNAP\_PERPENDICULAR, 464  
OBJ\_SNAP\_QUADRANT, 464  
OBJ\_SNAP\_TANGENT, 464  
OBJ\_SNAP\_VALUES, 464  
open\_action, 477  
pan\_action, 477  
paste\_action, 477  
paste\_selected\_action, 477  
perpendicular\_distance\_action, 477  
platform\_action, 477  
platformString, 478  
preview\_off\_action, 478  
preview\_on\_action, 478  
print\_action, 479  
print\_area\_action, 479  
prompt, 486  
qsnap\_x\_action, 479  
qsnap\_y\_action, 479  
quit\_action, 479  
read\_configuration, 479  
read\_string\_list\_setting, 480  
read\_string\_setting, 480  
redo\_action, 480  
rotate\_selected\_action, 480  
rubber\_action, 480  
rubber\_modes, 486  
run\_script, 480  
run\_script\_file, 480  
scale\_selected\_action, 481  
scripts, 487  
select\_all\_action, 481  
set\_background\_color\_action, 481  
set\_crosshair\_color\_action, 481  
set\_cursor\_shape\_action, 482  
set\_grid\_color\_action, 482  
set\_prompt\_prefix\_action, 482  
set\_rubber\_filter\_action, 482  
set\_rubber\_mode\_action, 482  
set\_rubber\_point\_action, 482  
set\_rubber\_text\_action, 482  
SetRubberText, 483  
SetTextAngle\_action, 483  
settings, 487  
settings\_dialog\_action, 483  
spare\_rubber\_action, 483  
spinBoxes, 487  
statusbar, 487  
subMenuHash, 487  
tip\_of\_the\_day\_action, 483  
todo\_action, 483  
toolbarHash, 487  
toolButtons, 487  
undo\_action, 484  
validFileFormat, 484  
validRGB, 484  
version\_action, 484  
vulcanize\_action, 484  
whats\_this\_action, 484  
window\_action, 485  
wizardTipOfTheDay, 487  
zoom\_action, 485  
majorVersion

- [\\_bcf\\_file\\_header](#), 48
- [make\\_checkbox](#)
  - [embroidermodder.h](#), 423
  - [interface.cpp](#), 451
- [make\\_editing\\_copy](#)
  - [settings-dialog.cpp](#), 491
- [make\\_layer\\_active\\_action](#)
  - [mainwindow.cpp](#), 475
- [make\\_spinbox](#)
  - [embroidermodder.h](#), 423
  - [interface.cpp](#), 451
- [make\\_ui\\_element](#)
  - [embroidermodder.h](#), 424
  - [interface.cpp](#), 451
- [manufacturer\\_code](#)
  - [thread\\_color\\_](#), 211
- [mapSignal](#)
  - [PropertyEditor](#), 184
- [Marathon\\_Polyester](#)
  - [embroidery.h](#), 249
- [Marathon\\_Rayon](#)
  - [embroidery.h](#), 249
- [max](#), 355
- [max\\_header](#)
  - [format\\_max.c](#), 355
- [MAX\\_STITCHES](#)
  - [embroidery.h](#), 249
- [MAX\\_THREADS](#)
  - [embroidery.h](#), 249
- [maxNumberOfDirectoryEntries](#)
  - [\\_bcf\\_directory](#), 43
- [MdiArea](#), 165
  - [~MdiArea](#), 166
  - [bgColor](#), 168
  - [bgLogo](#), 168
  - [bgTexture](#), 168
  - [cascade](#), 166
  - [forceRepaint](#), 166
  - [MdiArea](#), 166
  - [mouseDoubleClickEvent](#), 166
  - [paintEvent](#), 166
  - [setBackgroundColor](#), 167
  - [setBackgroundLogo](#), 167
  - [setBackgroundTexture](#), 167
  - [tile](#), 167
  - [useBackgroundColor](#), 167
  - [useBackgroundLogo](#), 168
  - [useBackgroundTexture](#), 168
  - [useColor](#), 168
  - [useLogo](#), 169
  - [useTexture](#), 169
  - [zoomExtentsAllSubWindows](#), 168
- [mdiArea](#)
  - [embroidermodder.h](#), 430
  - [mainwindow.cpp](#), 486
  - [MdiWindow](#), 178
- [MdiWindow](#), 169
  - [~MdiWindow](#), 171
- [closeEvent](#), 171
- [curColor](#), 177
- [curFile](#), 178
- [curLayer](#), 178
- [curLineType](#), 178
- [curLineWeight](#), 178
- [currentColorChanged](#), 171
- [currentLayerChanged](#), 171
- [currentLinetypeChanged](#), 172
- [currentLineweightChanged](#), 172
- [deletePressed](#), 172
- [designDetails](#), 172
- [escapePressed](#), 172
- [fileWasLoaded](#), 178
- [getShortCurrentFile](#), 172
- [gscene](#), 178
- [gview](#), 178
- [loadFile](#), 172
- [logPromptInput](#), 174
- [mdiArea](#), 178
- [MdiWindow](#), 170
- [myIndex](#), 178
- [onWindowActivated](#), 174
- [print](#), 174
- [printer](#), 178
- [promptHistory](#), 178
- [promptHistoryAppended](#), 174
- [promptInputList](#), 179
- [promptInputNext](#), 174
- [promptInputNum](#), 179
- [promptInputPrevious](#), 174
- [promptInputPrevNext](#), 175
- [saveBMC](#), 175
- [saveFile](#), 175
- [sendCloseMdiWin](#), 175
- [setCurrentFile](#), 176
- [setViewBackgroundColor](#), 176
- [setViewCrossHairColor](#), 176
- [setViewGridColor](#), 176
- [setViewRulerColor](#), 176
- [setViewSelectBoxColors](#), 177
- [showViewScrollBars](#), 177
- [sizeHint](#), 177
- [updateColorLinetypeLineweight](#), 177
- [mdiwindow.cpp](#)
  - [fileExtension](#), 487
- [Mega 2560 or another board with equal or](#), 16
- [Melco](#), 338, 342, 347, 357
- [menuHash](#)
  - [embroidermodder.h](#), 430
  - [mainwindow.cpp](#), 486
- [mergeWith](#)
  - [UndoableCommand](#), 214
- [messagebox\\_action](#)
  - [mainwindow.cpp](#), 475
- [Metro\\_Polyester](#)
  - [embroidery.h](#), 249
- [mid](#)

- EmbArc\_, 80
- miniSectorShift
  - \_bcf\_file\_header, 48
- miniStreamCutoffSize
  - \_bcf\_file\_header, 48
- minorVersion
  - \_bcf\_file\_header, 48
- minute
  - EmbTime\_, 104
- mirror
  - UndoableCommand, 214
- mirror\_selected\_action
  - mainwindow.cpp, 475
- mirrorLine
  - UndoableCommand, 215
- mirrorSelected
  - View, 222
- mit, 305, 356
- mitDecodeStitch
  - embroidery\_internal.h, 301
  - encoding.c, 325
- mitEncodeStitch
  - embroidery\_internal.h, 301
  - encoding.c, 325
- Mitsubishi, 305, 356
- modifiedTime
  - \_bcf\_directory\_entry, 44
- modifierName
  - ThredExtension\_, 211
- month
  - EmbTime\_, 104
- mouse\_x\_action
  - mainwindow.cpp, 475
- mouse\_y\_action
  - mainwindow.cpp, 475
- mouseDoubleClickEvent
  - MdiArea, 166
  - View, 223
- mouseMoveEvent
  - CmdPromptHandle, 64
  - View, 223
- mousePressEvent
  - CmdPromptHandle, 64
  - View, 223
- mouseReleaseEvent
  - CmdPromptHandle, 64
  - View, 223
- mouseSnapPoint
  - Geometry, 119
- move\_selected\_action
  - mainwindow.cpp, 476
- moveAction
  - View, 223
- movePoint
  - View, 228
- moveResizeHistory
  - CmdPromptSplitter, 77
- moveSelected
  - View, 223
- MOVETO
  - embroidery\_internal.h, 290
- moveY
  - CmdPromptHandle, 65
- movingActive
  - View, 228
- myFileSeparator
  - MainWindow, 164
- myIndex
  - MdiWindow, 178
- n\_attributes
  - format\_svg.c, 369
- N\_PES\_VERSIONS
  - embroidery\_internal.h, 290
- name
  - EmblImage\_, 92
  - Emblayer\_, 93
  - SvgAttribute\_, 210
  - thread\_color\_, 211
- navType
  - UndoableCommand, 215
- negativeXHoopSize
  - VipHeader\_, 231
- negativeYHoopSize
  - VipHeader\_, 231
- new, 305, 356
- new\_action
  - mainwindow.cpp, 476
- newFile
  - MainWindow, 156
- next
  - \_bcf\_directory\_entry, 44
- night\_vision\_action
  - mainwindow.cpp, 476
- nlengths
  - Huffman, 142
- no\_argument\_debug
  - mainwindow.cpp, 476
- NoArrow
  - Geometry, 110
- Node
  - embroidermodder.h, 419
- Node\_, 179
  - b, 179
  - i, 179
  - r, 179
  - s, 179
  - sl, 180
  - type, 180
- node\_bool
  - embroidermodder.h, 424
  - interface.cpp, 451
- node\_int
  - embroidermodder.h, 424
  - interface.cpp, 452
- node\_qstr
  - embroidermodder.h, 424

- interface.cpp, 452
- node\_real
  - embroidermodder.h, 424
  - interface.cpp, 452
- node\_str
  - embroidermodder.h, 425
  - interface.cpp, 452
- node\_str\_list
  - embroidermodder.h, 425
  - interface.cpp, 453
- node\_uint
  - embroidermodder.h, 425
  - interface.cpp, 453
- NodeList
  - embroidermodder.h, 419
- NoLine
  - Geometry, 111
- NORMAL
  - embroidery.h, 249
- normalPath
  - Geometry, 139
- ntable
  - Huffman, 142
- NUM\_FLAGS
  - main.c, 397
- num\_selected\_action
  - mainwindow.cpp, 476
- numberOfBytesRemaining
  - \_vp3Hoop, 51
- numberOfColors
  - \_vp3Hoop, 51
  - VipHeader\_, 231
- numberOfDifatSectors
  - \_bcf\_file\_header, 48
- numberOfDirectorySectors
  - \_bcf\_file\_header, 49
- numberOfEntriesInDifatSector
  - embroidery\_internal.h, 301
- numberOfEntriesInFatSector
  - \_bcf\_file\_fat, 47
- numberOfFATSectors
  - \_bcf\_file\_header, 49
- numberOfFormats
  - embroidery.h, 249
- numberOfMiniFatSectors
  - \_bcf\_file\_header, 49
- numberOfStitches
  - VipHeader\_, 231
- numOfDocs
  - MainWindow, 164
- numSelected
  - View, 223
- numStiches
  - ThredHeader\_, 212
- OBJ\_COLOR
  - embroidermodder.h, 419
- OBJ\_KEYS
  - embroidermodder.h, 419
- OBJ\_LAYER
  - embroidermodder.h, 419
- OBJ\_LTYPE
  - embroidermodder.h, 419
- OBJ\_LTYPE\_CENTER
  - mainwindow.cpp, 463
- OBJ\_LTYPE\_CONT
  - mainwindow.cpp, 463
- OBJ\_LTYPE\_DOT
  - mainwindow.cpp, 463
- OBJ\_LTYPE\_FISHBONE
  - mainwindow.cpp, 463
- OBJ\_LTYPE\_HIDDEN
  - mainwindow.cpp, 463
- OBJ\_LTYPE\_PHANTOM
  - mainwindow.cpp, 463
- OBJ\_LTYPE\_RUNNING
  - mainwindow.cpp, 463
- OBJ\_LTYPE\_SATIN
  - mainwindow.cpp, 463
- OBJ\_LTYPE\_VALUES
  - mainwindow.cpp, 463
- OBJ\_LTYPE\_ZIGZAG
  - mainwindow.cpp, 463
- OBJ\_LWT
  - embroidermodder.h, 419
- OBJ\_LWT\_01
  - mainwindow.cpp, 463
- OBJ\_LWT\_02
  - mainwindow.cpp, 463
- OBJ\_LWT\_03
  - mainwindow.cpp, 463
- OBJ\_LWT\_04
  - mainwindow.cpp, 463
- OBJ\_LWT\_05
  - mainwindow.cpp, 463
- OBJ\_LWT\_06
  - mainwindow.cpp, 463
- OBJ\_LWT\_07
  - mainwindow.cpp, 463
- OBJ\_LWT\_08
  - mainwindow.cpp, 463
- OBJ\_LWT\_09
  - mainwindow.cpp, 463
- OBJ\_LWT\_10
  - mainwindow.cpp, 463
- OBJ\_LWT\_11
  - mainwindow.cpp, 463
- OBJ\_LWT\_12
  - mainwindow.cpp, 463
- OBJ\_LWT\_13
  - mainwindow.cpp, 463
- OBJ\_LWT\_14
  - mainwindow.cpp, 463
- OBJ\_LWT\_15
  - mainwindow.cpp, 463
- OBJ\_LWT\_16
  - mainwindow.cpp, 463

- OBJ\_LWT\_17
  - mainwindow.cpp, [463](#)
- OBJ\_LWT\_18
  - mainwindow.cpp, [463](#)
- OBJ\_LWT\_19
  - mainwindow.cpp, [463](#)
- OBJ\_LWT\_20
  - mainwindow.cpp, [463](#)
- OBJ\_LWT\_21
  - mainwindow.cpp, [463](#)
- OBJ\_LWT\_22
  - mainwindow.cpp, [464](#)
- OBJ\_LWT\_23
  - mainwindow.cpp, [464](#)
- OBJ\_LWT\_24
  - mainwindow.cpp, [464](#)
- OBJ\_LWT\_BYBLOCK
  - mainwindow.cpp, [463](#)
- OBJ\_LWT\_BYLAYER
  - mainwindow.cpp, [463](#)
- OBJ\_LWT\_DEFAULT
  - mainwindow.cpp, [463](#)
- OBJ\_LWT\_VALUES
  - mainwindow.cpp, [463](#)
- OBJ\_NAME
  - embroidermodder.h, [419](#)
- OBJ\_RUBBER
  - embroidermodder.h, [419](#)
- OBJ\_SNAP\_APPINTERSECTION
  - mainwindow.cpp, [464](#)
- OBJ\_SNAP\_CENTER
  - mainwindow.cpp, [464](#)
- OBJ\_SNAP\_ENDPOINT
  - mainwindow.cpp, [464](#)
- OBJ\_SNAP\_EXTENSION
  - mainwindow.cpp, [464](#)
- OBJ\_SNAP\_INSERTION
  - mainwindow.cpp, [464](#)
- OBJ\_SNAP\_INTERSECTION
  - mainwindow.cpp, [464](#)
- OBJ\_SNAP\_MIDPOINT
  - mainwindow.cpp, [464](#)
- OBJ\_SNAP\_NEAREST
  - mainwindow.cpp, [464](#)
- OBJ\_SNAP\_NODE
  - mainwindow.cpp, [464](#)
- OBJ\_SNAP\_NULL
  - mainwindow.cpp, [464](#)
- OBJ\_SNAP\_PARALLEL
  - mainwindow.cpp, [464](#)
- OBJ\_SNAP\_PERPENDICULAR
  - mainwindow.cpp, [464](#)
- OBJ\_SNAP\_QUADRANT
  - mainwindow.cpp, [464](#)
- OBJ\_SNAP\_TANGENT
  - mainwindow.cpp, [464](#)
- OBJ\_SNAP\_VALUES
  - mainwindow.cpp, [464](#)
- OBJ\_TYPE
  - embroidermodder.h, [419](#)
- OBJ\_TYPE\_ARC
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_BASE
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_BLOCK
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_CIRCLE
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_DIMALIGNED
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_DIMANGULAR
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_DIMARCLENGTH
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_DIMDIAMETER
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_DIMLEADER
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_DIMLINEAR
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_DIMORDINATE
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_DIMRADIUS
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_ELLIPSE
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_ELLIPSEARC
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_GRID
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_HATCH
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_IMAGE
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_INFINITELINE
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_LINE
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_NULL
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_PATH
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_POINT
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_POLYGON
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_POLYLINE
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_RAY
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_RECTANGLE
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_RUBBER
  - embroidermodder.h, [420](#)
- OBJ\_TYPE\_SLOT
  - embroidermodder.h, [420](#)

- OBJ\_TYPE\_SPLINE
  - embroidermodder.h, 420
- OBJ\_TYPE\_TEXTMULTI
  - embroidermodder.h, 420
- OBJ\_TYPE\_TEXTSINGLE
  - embroidermodder.h, 420
- OBJ\_TYPE\_UNKNOWN
  - embroidermodder.h, 420
- OBJ\_TYPE\_VALUES
  - embroidermodder.h, 420
- object
  - EmbGeometry\_, 90
  - UndoableCommand, 215
- object\_names
  - property-editor.cpp, 490
- objectAngle
  - Geometry, 119
- objectArcLength
  - Geometry, 120
- objectArea
  - Geometry, 120
- objectBottomLeft
  - Geometry, 120
- objectBottomRight
  - Geometry, 120
- objectCenter
  - Geometry, 120
- objectChord
  - Geometry, 120
- objectCircumference
  - Geometry, 121
- objectClockwise
  - Geometry, 121
- objectCopyPath
  - Geometry, 121
- objectDelta
  - Geometry, 121
- objectDiameter
  - Geometry, 121
- objectDiameterMajor
  - Geometry, 121
- objectDiameterMinor
  - Geometry, 121
- objectEndAngle
  - Geometry, 122
- objectEndPoint
  - Geometry, 122
- objectEndPoint1
  - Geometry, 122
- objectEndPoint2
  - Geometry, 122
- objectHeight
  - Geometry, 122
- objectIncludedAngle
  - Geometry, 123
- objectLength
  - Geometry, 123
- objectLineType
  - Geometry, 123
- objectLineWeight
  - Geometry, 123
- objectMidPoint
  - Geometry, 123
- objectPos
  - Geometry, 123
- objectQuadrant0
  - Geometry, 123
- objectQuadrant180
  - Geometry, 124
- objectQuadrant270
  - Geometry, 124
- objectQuadrant90
  - Geometry, 124
- objectRadius
  - Geometry, 124
- objectRadiusMajor
  - Geometry, 124
- objectRadiusMinor
  - Geometry, 124
- objectRubberPoint
  - Geometry, 124
- objectRubberText
  - Geometry, 125
- objects.cpp
  - add\_polyline, 488
  - closest\_point, 488
  - fourier\_series, 488
  - rotate\_vector, 489
- objectSavePath
  - Geometry, 125
- objectSavePathList
  - Geometry, 125
- objectStartAngle
  - Geometry, 125
- objectStartPoint
  - Geometry, 125
- objectTopLeft
  - Geometry, 126
- objectTopRight
  - Geometry, 126
- objectType
  - \_bcf\_directory\_entry, 44
- ObjectTypeRootEntry
  - embroidery\_internal.h, 290
- ObjectTypeStorage
  - embroidery\_internal.h, 290
- ObjectTypeStream
  - embroidery\_internal.h, 290
- ObjectTypeUnknown
  - embroidery\_internal.h, 290
- objectWidth
  - Geometry, 126
- objectX
  - Geometry, 126
- objectX1
  - Geometry, 126

- objectX2
  - Geometry, [126](#)
- objectY
  - Geometry, [126](#)
- objectY1
  - Geometry, [127](#)
- objectY2
  - Geometry, [127](#)
- objID
  - Geometry, [139](#)
- objLine
  - Geometry, [139](#)
- objPen
  - Geometry, [139](#)
- objRubberMode
  - Geometry, [139](#)
- objRubberPoints
  - Geometry, [139](#)
- objRubberTexts
  - Geometry, [139](#)
- objText
  - Geometry, [139](#)
- objTextBackward
  - Geometry, [139](#)
- objTextFont
  - Geometry, [140](#)
- objTextJustify
  - Geometry, [140](#)
- objTextPath
  - Geometry, [140](#)
- objTextUpsideDown
  - Geometry, [140](#)
- ofm, [357](#)
- ofmDecode
  - format\_ofm.c, [357](#)
- ofmReadBlockHeader
  - format\_ofm.c, [357](#)
- ofmReadClass
  - format\_ofm.c, [357](#)
- ofmReadColorChange
  - format\_ofm.c, [357](#)
- ofmReadExpanded
  - format\_ofm.c, [357](#)
- ofmReadLibrary
  - format\_ofm.c, [357](#)
- ofmReadThreads
  - format\_ofm.c, [357](#)
- onCloseMdiWin
  - MainWindow, [156](#)
- onCloseWindow
  - MainWindow, [157](#)
- onWindowActivated
  - MainWindow, [157](#)
  - MdiWindow, [174](#)
- Open
  - Geometry, [110](#)
- open\_action
  - mainwindow.cpp, [477](#)
- openFile
  - MainWindow, [157](#)
- openFilesSelected
  - MainWindow, [157](#)
- openrecentfile
  - MainWindow, [157](#)
- opensave\_props
  - settings-dialog.cpp, [492](#)
- operator\*
  - embroidermodder.h, [425](#)
  - interface.cpp, [453](#)
- operator+
  - embroidermodder.h, [425](#)
  - interface.cpp, [453](#)
- operator-
  - embroidermodder.h, [426](#)
  - interface.cpp, [453](#)
- originPath
  - View, [228](#)
- paint
  - Geometry, [127](#)
- paintEvent
  - ImageWidget, [143](#)
  - MdiArea, [166](#)
  - SelectBox, [198](#)
- pan\_action
  - mainwindow.cpp, [477](#)
- panDistance
  - View, [228](#)
- panDown
  - View, [223](#)
- panLeft
  - View, [223](#)
- panningActive
  - View, [228](#)
- panningPointActive
  - View, [228](#)
- panningRealTimeActive
  - View, [228](#)
- panPoint
  - View, [223](#)
- panRealTime
  - View, [223](#)
- panRight
  - View, [223](#)
- panStart
  - View, [223](#)
- panStartX
  - View, [228](#)
- panStartY
  - View, [228](#)
- Pantone
  - embroidery.h, [250](#)
- panUp
  - View, [223](#)
- parseDIFATsectors
  - main.c, [401](#)
- parseDirectoryEntryName

- main.c, [401](#)
- parseTime
  - main.c, [401](#)
- paste
  - View, [224](#)
- paste\_action
  - mainwindow.cpp, [477](#)
- paste\_selected\_action
  - mainwindow.cpp, [477](#)
- pasteClip
  - CmdPromptInput, [74](#)
- pasteDelta
  - View, [228](#)
- pasteObjectItemGroup
  - View, [228](#)
- pastePressed
  - CmdPrompt, [58](#)
  - CmdPromptInput, [74](#)
- pastingActive
  - View, [228](#)
- path
  - EmbGeometry\_, [90](#)
  - EmblImage\_, [92](#)
- pattern.c
  - convert, [404](#)
  - embPattern\_addCircleAbs, [404](#)
  - embPattern\_addEllipseAbs, [404](#)
  - embPattern\_addLineAbs, [405](#)
  - embPattern\_addPathAbs, [405](#)
  - embPattern\_addPointAbs, [405](#)
  - embPattern\_addPolygonAbs, [405](#)
  - embPattern\_addPolylineObjectAbs, [405](#)
  - embPattern\_addRectAbs, [405](#)
  - embPattern\_addStitchAbs, [405](#)
  - embPattern\_addStitchRel, [405](#)
  - embPattern\_addThread, [405](#)
  - embPattern\_calcBoundingBox, [406](#)
  - embPattern\_center, [406](#)
  - embPattern\_changeColor, [406](#)
  - embPattern\_color\_count, [406](#)
  - embPattern\_combineJumpStitches, [406](#)
  - embPattern\_copyPolylinesTostitch\_list, [406](#)
  - embPattern\_copystitch\_listToPolylines, [406](#)
  - embPattern\_correctForMaxStitchLength, [406](#)
  - embPattern\_create, [406](#)
  - embPattern\_designDetails, [406](#)
  - embPattern\_end, [407](#)
  - embPattern\_fixColorCount, [407](#)
  - embPattern\_flip, [407](#)
  - embPattern\_flipHorizontal, [407](#)
  - embPattern\_flipVertical, [407](#)
  - embPattern\_free, [407](#)
  - embPattern\_hideStitchesOverLength, [407](#)
  - embPattern\_jumpStitches, [407](#)
  - embPattern\_lengthHistogram, [407](#)
  - embPattern\_loadExternalColorFile, [407](#)
  - embPattern\_maximumStitchLength, [407](#)
  - embPattern\_minimumStitchLength, [407](#)
  - embPattern\_movePolylinesTostitch\_list, [408](#)
  - embPattern\_movestitch\_listToPolylines, [408](#)
  - embPattern\_realStitches, [408](#)
  - embPattern\_scale, [408](#)
  - embPattern\_totalStitchLength, [408](#)
  - embPattern\_trimStitches, [408](#)
- pcd, [9](#), [306](#), [358](#)
- pcm, [9](#), [306](#), [358](#)
- pcm\_thread
  - embroidery.h, [250](#)
- pcmThreads
  - embroidery.h, [268](#)
  - thread-color.c, [409](#)
- pcq, [9](#), [306](#), [359](#)
- pcs, [9](#), [306](#), [359](#)
- pec, [9](#), [306](#), [360](#)
- pec\_thread
  - embroidery.h, [250](#)
- pecEncode
  - format\_pec.c, [360](#)
- pecEncodeJump
  - format\_pec.c, [360](#)
- pecEncodeStop
  - format\_pec.c, [360](#)
- pecThreadCount
  - embroidery.h, [268](#)
  - thread-color.c, [409](#)
- pecThreads
  - embroidery.h, [269](#)
  - thread-color.c, [410](#)
- pel, [9](#), [306](#), [361](#)
- pem, [9](#), [307](#), [361](#)
- perpendicular\_distance\_action
  - mainwindow.cpp, [477](#)
- pes, [9](#), [364](#)
- PES0001
  - embroidery\_internal.h, [290](#)
- PES0020
  - embroidery\_internal.h, [290](#)
- PES0022
  - embroidery\_internal.h, [290](#)
- PES0030
  - embroidery\_internal.h, [290](#)
- PES0040
  - embroidery\_internal.h, [290](#)
- PES0050
  - embroidery\_internal.h, [290](#)
- PES0055
  - embroidery\_internal.h, [290](#)
- PES0056
  - embroidery\_internal.h, [291](#)
- PES0060
  - embroidery\_internal.h, [291](#)
- PES0070
  - embroidery\_internal.h, [291](#)
- PES0080
  - embroidery\_internal.h, [291](#)
- PES0090



- embroidery\_internal.h, 291
- PES0100
  - embroidery\_internal.h, 291
- pes\_version
  - format\_pes.c, 364
- pes\_version\_strings
  - format\_pes.c, 364
- pesWriteEmbOneSection
  - format\_pes.c, 362
- pesWriteSewSegSection
  - format\_pes.c, 362
- Pfaff, 269, 306, 308, 354, 355, 358, 359, 370, 375, 376
- pfaffDecode
  - embroidery\_internal.h, 301
  - encoding.c, 325
- pfaffEncode
  - embroidery\_internal.h, 301
  - encoding.c, 326
- phb, 9, 307, 364
- phc, 9, 307, 365
- pickAdd
  - PropertyEditor, 186
- pickAddModeToggled
  - MainWindow, 158
  - PropertyEditor, 184
- pickBoxSize
  - View, 228
- pivot
  - UndoableCommand, 215
- platform\_action
  - mainwindow.cpp, 477
- platformString
  - MainWindow, 158
  - mainwindow.cpp, 478
- plt, 308, 365
- point
  - EmbGeometry\_, 90
- pointList
  - EmbPath\_, 96
- polygon
  - EmbGeometry\_, 90
- polyline
  - EmbGeometry\_, 91
- position
  - EmbAlignedDim\_, 79
  - EmbAngularDim\_, 80
  - EmbArcLengthDim\_, 81
  - EmbBlock\_, 83
  - EmbDiameterDim\_, 87
  - EmbImage\_, 92
  - EmbInfiniteLine\_, 93
  - EmbLeaderDim\_, 94
  - EmbLinearDim\_, 95
  - EmbOrdinateDim\_, 95
  - EmbPoint\_, 98
  - EmbRadiusDim\_, 98
  - EmbRay\_, 99
  - EmbTextMulti\_, 102
  - EmbTextSingle\_, 103
- positiveXHoopSize
  - VipHeader\_, 231
- positiveYHoopSize
  - VipHeader\_, 231
- precisionAngle
  - PropertyEditor, 186
- precisionLength
  - PropertyEditor, 186
- prefix
  - CmdPromptInput, 76
- pressPoint
  - View, 229
- pressResizeHistory
  - CmdPromptSplitter, 77
- pressY
  - CmdPromptHandle, 65
- preview
  - settings-dialog.cpp, 492
- preview\_off\_action
  - mainwindow.cpp, 478
- preview\_on\_action
  - mainwindow.cpp, 478
- previewActive
  - View, 229
- previewData
  - View, 229
- PreviewDialog, 180
  - ~PreviewDialog, 180
  - imgWidget, 180
  - PreviewDialog, 180
- previewMode
  - View, 229
- previewObjectItemGroup
  - View, 229
- previewObjectList
  - View, 229
- previewOff
  - View, 224
- previewOn
  - View, 224
- previewPoint
  - View, 229
- print
  - MdiWindow, 174
- print\_action
  - mainwindow.cpp, 479
- print\_area\_action
  - mainwindow.cpp, 479
- printArcResults
  - embroidery\_internal.h, 301
- printer
  - MdiWindow, 178
- privacy\_policy.md, 410
- processInput
  - CmdPromptInput, 74
- prompt
  - embroidermodder.h, 430

- mainwindow.cpp, 486
- prompt\_props
  - settings-dialog.cpp, 492
- promptDivider
  - CmdPrompt, 62
- promptHistory
  - CmdPrompt, 62
  - MdiWindow, 178
- promptHistoryAppended
  - MainWindow, 158
  - MdiWindow, 174
- promptInput
  - CmdPrompt, 62
- promptInputList
  - MdiWindow, 179
- promptInputNext
  - MainWindow, 158
  - MdiWindow, 174
- promptInputNum
  - MdiWindow, 179
- promptInputPrevious
  - MainWindow, 158
  - MdiWindow, 174
- promptInputPrevNext
  - MdiWindow, 175
- promptSplitter
  - CmdPrompt, 62
- promptVBoxLayout
  - CmdPrompt, 62
- properties
  - Geometry, 140
- property-editor.cpp
  - comboBoxTextSingleFont, 489
  - fieldNewText, 489
  - fieldNoText, 490
  - fieldOffText, 490
  - fieldOldText, 490
  - fieldOnText, 490
  - fieldVariesText, 490
  - fieldYesText, 490
  - group\_box\_data, 490
  - group\_box\_types, 490
  - load\_group\_box\_data\_from\_table, 489
  - object\_names, 490
- PropertyEditor, 181
  - ~PropertyEditor, 182
  - clearAllFields, 182
  - comboBoxSelected, 185
  - createComboBoxSelected, 182
  - createGroupBox, 183
  - createLineEdit, 183
  - createToolButton, 183
  - createToolButtonPickAdd, 183
  - createToolButtonQSelect, 183
  - eventFilter, 183
  - fieldEdited, 183
  - focusWidget, 185
  - hideAllGroups, 183
  - iconDir, 185
  - iconSize, 186
  - mapSignal, 184
  - pickAdd, 186
  - pickAddModeToggled, 184
  - precisionAngle, 186
  - precisionLength, 186
  - PropertyEditor, 182
  - propertyEditorButtonStyle, 186
  - selectedItemList, 186
  - setSelectedItems, 184
  - showGroups, 184
  - showOneType, 184
  - signalMapper, 186
  - togglePickAddMode, 184
  - toolButtonPickAdd, 186
  - toolButtonQSelect, 186
  - updateComboBoxBoolIfVaries, 184
  - updateComboBoxStrIfVaries, 185
  - updateFontComboBoxStrIfVaries, 185
  - updateLineEditNumIfVaries, 185
  - updateLineEditStrIfVaries, 185
  - updatePickAddModeButton, 185
- propertyEditorButtonStyle
  - PropertyEditor, 186
- qsnap\_x\_action
  - mainwindow.cpp, 479
- qsnap\_y\_action
  - mainwindow.cpp, 479
- qSnapActive
  - View, 229
- qsnapApertureSize
  - View, 229
- qsnapLocatorColor
  - View, 229
- qsnapLocatorSize
  - View, 229
- qSnapToggle
  - View, 229
- QUADTOCONTROL
  - embroidery\_internal.h, 291
- QUADTOEND
  - embroidery\_internal.h, 291
- quick\_snap\_props
  - settings-dialog.cpp, 492
- quit
  - MainWindow, 158
- quit\_action
  - mainwindow.cpp, 479
- r
  - EmbColor\_, 84
  - Node\_, 179
- radians
  - embroidery.h, 267
  - functions.c, 385
- radians\_\_
  - embroidermodder.h, 426

- interface.cpp, 453
- radius
  - EmbCircle\_, 83
  - EmbEllipse\_, 88
  - EmbRect\_, 99
- rapidFireEnabled
  - CmdPromptInput, 76
- rapidMoveActive
  - View, 229
- read100
  - embroidery\_internal.h, 302
  - format\_100.c, 335
- read10o
  - embroidery\_internal.h, 302
  - format\_10o.c, 336
- read\_configuration
  - embroidermodder.h, 426
  - mainwindow.cpp, 479
- read\_hoop
  - format\_jef.c, 353
- read\_settings
  - embroidermodder.h, 426
  - settings-dialog.cpp, 491
- read\_string\_list\_setting
  - mainwindow.cpp, 480
- read\_string\_setting
  - embroidermodder.h, 426
  - mainwindow.cpp, 480
- readArt
  - embroidery\_internal.h, 302
  - format\_art.c, 336
- readBmc
  - embroidery\_internal.h, 302
  - format\_bmc.c, 337
- readBro
  - embroidery\_internal.h, 302
  - format\_bro.c, 337
- readCnd
  - embroidery\_internal.h, 302
  - format\_cnd.c, 338
- readCol
  - embroidery\_internal.h, 302
  - format\_col.c, 339
- readCsd
  - embroidery\_internal.h, 302
  - format\_csd.c, 340
- readCsv
  - embroidery\_internal.h, 302
  - format\_csv.c, 341
- readDat
  - embroidery\_internal.h, 302
  - format\_dat.c, 341
- readDem
  - embroidery\_internal.h, 302
  - format\_dem.c, 342
- readDescriptions
  - embroidery\_internal.h, 303
  - format\_pes.c, 362
- readDsb
  - embroidery\_internal.h, 303
  - format\_dsb.c, 343
- readDst
  - embroidery\_internal.h, 303
  - format\_dst.c, 345
- readDsz
  - embroidery\_internal.h, 303
  - format\_dsz.c, 345
- readDxf
  - embroidery\_internal.h, 303
  - format\_dxf.c, 346
- readEdr
  - embroidery\_internal.h, 303
  - format\_edr.c, 346
- readEmd
  - embroidery\_internal.h, 303
  - format\_emd.c, 347
- reader\_state
  - EmbFormatList\_, 89
- readExp
  - embroidery\_internal.h, 303
  - format\_exp.c, 347
- readExy
  - embroidery\_internal.h, 303
  - format\_exy.c, 348
- readEys
  - embroidery\_internal.h, 303
  - format\_eyes.c, 348
- readFeatherPatterns
  - embroidery\_internal.h, 304
  - format\_pes.c, 362
- readFullSector
  - embroidery\_internal.h, 304
  - main.c, 402
- readFxy
  - embroidery\_internal.h, 304
  - format\_fxy.c, 349
- readGc
  - embroidery\_internal.h, 304
  - format\_gc.c, 349
- readGnc
  - embroidery\_internal.h, 304
  - format\_gnc.c, 350
- readGt
  - embroidery\_internal.h, 304
  - format\_gt.c, 350
- readHoopName
  - embroidery\_internal.h, 304
  - format\_pes.c, 362
- readHus
  - embroidery\_internal.h, 304
  - format\_hus.c, 352
- readImageString
  - embroidery\_internal.h, 304
  - format\_pes.c, 363
- readInb
  - embroidery\_internal.h, 305

format\_inb.c, 352  
readInf  
  embroidery\_internal.h, 305  
  format\_inf.c, 352  
readJef  
  embroidery\_internal.h, 305  
  format\_jef.c, 354  
readKsm  
  embroidery\_internal.h, 305  
  format\_ksm.c, 354  
readLine  
  format\_dxf.c, 346  
readMax  
  embroidery\_internal.h, 305  
  format\_max.c, 355  
readMit  
  embroidery\_internal.h, 305  
  format\_mit.c, 355  
readMotifPatterns  
  embroidery\_internal.h, 305  
  format\_pes.c, 363  
readNew  
  embroidery\_internal.h, 305  
  format\_new.c, 356  
readNextSector  
  embroidery\_internal.h, 305  
  main.c, 402  
readOfm  
  embroidery\_internal.h, 305  
  format\_ofm.c, 357  
readPcd  
  embroidery\_internal.h, 306  
  format\_pcd.c, 358  
readPcm  
  embroidery\_internal.h, 306  
  format\_pcm.c, 358  
readPcq  
  embroidery\_internal.h, 306  
  format\_pcq.c, 359  
readPcs  
  embroidery\_internal.h, 306  
  format\_pcs.c, 359  
readPec  
  embroidery\_internal.h, 306  
  format\_pec.c, 360  
readPecStitches  
  embroidery\_internal.h, 306  
  format\_pec.c, 360  
readPel  
  embroidery\_internal.h, 306  
  format\_pel.c, 361  
readPem  
  embroidery\_internal.h, 306  
  format\_pem.c, 361  
readPes  
  embroidery\_internal.h, 307  
  format\_pes.c, 363  
readPESHeaderV10  
  embroidery\_internal.h, 307  
  format\_pes.c, 363  
readPESHeaderV5  
  embroidery\_internal.h, 307  
  format\_pes.c, 363  
readPESHeaderV6  
  embroidery\_internal.h, 307  
  format\_pes.c, 363  
readPESHeaderV7  
  embroidery\_internal.h, 307  
  format\_pes.c, 363  
readPESHeaderV8  
  embroidery\_internal.h, 307  
  format\_pes.c, 363  
readPESHeaderV9  
  embroidery\_internal.h, 307  
  format\_pes.c, 363  
readPhb  
  embroidery\_internal.h, 307  
  format\_phb.c, 364  
readPhc  
  embroidery\_internal.h, 307  
  format\_phc.c, 365  
readPlt  
  embroidery\_internal.h, 307  
  format\_plt.c, 365  
readProgrammableFills  
  embroidery\_internal.h, 308  
  format\_pes.c, 363  
readRgb  
  embroidery\_internal.h, 308  
  format\_rgb.c, 366  
readSew  
  embroidery\_internal.h, 308  
  format\_sew.c, 366  
readShv  
  embroidery\_internal.h, 308  
  format\_shv.c, 367  
readSst  
  embroidery\_internal.h, 308  
  format\_sst.c, 367  
readStx  
  embroidery\_internal.h, 308  
  format\_stx.c, 368  
readSvg  
  embroidery\_internal.h, 308  
  format\_svg.c, 369  
readT01  
  embroidery\_internal.h, 308  
  format\_t01.c, 370  
readT09  
  embroidery\_internal.h, 308  
  format\_t09.c, 370  
readTap  
  embroidery\_internal.h, 308  
  format\_tap.c, 371  
readThr  
  embroidery\_internal.h, 309

- format\_thr.c, 371
- readThreads
  - embroidery\_internal.h, 309
  - format\_pes.c, 363
- readTxt
  - embroidery\_internal.h, 309
  - format\_txt.c, 372
- readU00
  - embroidery\_internal.h, 309
  - format\_u00.c, 372
- readU01
  - embroidery\_internal.h, 309
  - format\_u01.c, 373
- readVip
  - embroidery\_internal.h, 309
  - format\_vip.c, 374
- readVp3
  - embroidery\_internal.h, 309
  - format\_vp3.c, 375
- readXxx
  - embroidery\_internal.h, 309
  - format\_xxx.c, 376
- readZsk
  - embroidery\_internal.h, 309
  - format\_zsk.c, 377
- REAL\_TYPE
  - embroidermodder.h, 418
- realRender
  - Geometry, 127
- recalculateLimits
  - View, 224
- recentMenuAboutToShow
  - MainWindow, 158
- rect
  - EmbGeometry\_, 91
  - Geometry, 127
- rect.c
  - embRect\_area, 387
  - embRect\_init, 387
- RED\_TERM\_COLOR
  - embroidery\_internal.h, 291
- redo
  - UndoableCommand, 214
  - UndoEditor, 216
- redo\_action
  - mainwindow.cpp, 480
- redoPressed
  - CmdPrompt, 58
  - CmdPromptInput, 74
- redoText
  - UndoEditor, 216
- rejectChanges
  - Settings\_Dialog, 207
- releasePoint
  - View, 229
- releaseResizeHistory
  - CmdPromptSplitter, 78
- releaseY
  - CmdPromptHandle, 65
- repeatAction
  - View, 224
- report
  - embroidery.h, 267
- reserved
  - ThredExtension\_, 211
  - ThredHeader\_, 212
- reserved1
  - \_bcf\_file\_header, 49
- reserved2
  - \_bcf\_file\_header, 49
- RESET\_TERM\_COLOR
  - embroidery\_internal.h, 291
- resizeEvent
  - MainWindow, 158
- resizeHistory
  - CmdPromptHistory, 67
- reverse\_byte\_order
  - encoding.c, 326
- rgb, 9, 308, 366
- right
  - \_vp3Hoop, 51
  - EmbRect\_, 100
  - hoop\_padding, 141
- right2
  - \_vp3Hoop, 51
- rightBrush
  - SelectBox, 199
- rightBrushColor
  - SelectBox, 199
- rightPen
  - SelectBox, 199
- rightPenColor
  - SelectBox, 199
- rightSiblingId
  - \_bcf\_directory\_entry, 44
- RobisonAnton\_Polyester
  - embroidery.h, 250
- RobisonAnton\_Rayon
  - embroidery.h, 250
- rotate
  - UndoableCommand, 214
- rotate\_selected\_action
  - mainwindow.cpp, 480
- rotate\_vector
  - embroidermodder.h, 426
  - objects.cpp, 489
- rotateAction
  - View, 224
- rotateSelected
  - View, 224
- rotation
  - EmbEllipse\_, 88
  - EmbRect\_, 100
- roundToMultiple
  - View, 224
- rubber\_action

- mainwindow.cpp, 480
- rubber\_modes
  - mainwindow.cpp, 486
- rubberRoomList
  - View, 229
- rulerColor
  - View, 229
- rulerMetric
  - View, 229
- rulerPixelSize
  - View, 229
- rules
  - fill.c, 330
  - LSYSTEM, 146
- run\_script
  - embroidermodder.h, 426
  - mainwindow.cpp, 480
- run\_script\_file
  - embroidermodder.h, 427
  - mainwindow.cpp, 480
- runCommand
  - CmdPrompt, 58
  - CmdPromptInput, 74
- s
  - em2\_dev\_script, 42
  - Node\_, 179
- safe\_free
  - embroidery\_internal.h, 309
  - formats.c, 334
- save
  - ImageWidget, 143
  - SaveObject, 196
- save\_points\_to\_pattern
  - fill.c, 330
- saveasfile
  - MainWindow, 159
- saveBMC
  - MdiWindow, 175
- saveFile
  - MdiWindow, 175
- savefile
  - MainWindow, 159
- saveHistory
  - CmdPrompt, 59
- SaveObject, 187
  - ~SaveObject, 188
  - addArc, 189
  - addBlock, 189
  - addCircle, 189
  - addDimAligned, 189
  - addDimAngular, 190
  - addDimArcLength, 190
  - addDimDiameter, 190
  - addDimLeader, 190
  - addDimLinear, 191
  - addDimOrdinate, 191
  - addDimRadius, 191
  - addEllipse, 192
  - addEllipseArc, 192
  - addGrid, 192
  - addHatch, 192
  - addImage, 193
  - addInfiniteLine, 193
  - addLine, 193
  - addPath, 193
  - addPoint, 194
  - addPolygon, 194
  - addPolyline, 194
  - addRay, 195
  - addRectangle, 195
  - addSlot, 195
  - addSpline, 195
  - addTextMulti, 196
  - addTextSingle, 196
  - formatType, 197
  - gscene, 197
  - save, 196
  - SaveObject, 188
  - toPolyline, 196
- scale\_selected\_action
  - mainwindow.cpp, 481
- scaleAction
  - View, 224
- scaleSelected
  - View, 224
- sceneGripPoint
  - View, 230
- sceneMousePoint
  - View, 230
- sceneMovePoint
  - View, 230
- scenePressPoint
  - View, 230
- sceneReleasePoint
  - View, 230
- script\_click
  - Geometry, 127
- script\_context
  - Geometry, 128
- script\_main
  - Geometry, 128
- script\_prompt
  - Geometry, 128
- scripts
  - embroidermodder.h, 430
  - mainwindow.cpp, 487
- second
  - EmbTime\_, 104
- sectionName
  - StxThread\_, 209
- sectorShift
  - \_bcf\_file\_header, 49
- sectorSize
  - \_bcf\_file\_difat, 46
  - main.c, 402
- seekToSector

- main.c, 402
- select\_all\_action
  - mainwindow.cpp, 481
- selectAll
  - View, 224
- selectAllPressed
  - CmdPrompt, 59
  - CmdPromptInput, 74
- SelectBox, 197
  - alpha, 199
  - boxDir, 199
  - dirBrush, 199
  - dirPen, 199
  - forceRepaint, 198
  - leftBrush, 199
  - leftBrushColor, 199
  - leftPen, 199
  - leftPenColor, 199
  - paintEvent, 198
  - rightBrush, 199
  - rightBrushColor, 199
  - rightPen, 199
  - rightPenColor, 199
  - SelectBox, 198
  - setColors, 199
  - setDirection, 199
- selectBox
  - View, 230
- selected\_items
  - View, 224
- selectedItemList
  - PropertyEditor, 186
- selectingActive
  - View, 230
- selectionChanged
  - View, 224
- sendCloseMdiWin
  - MdiWindow, 175
- SEQUIN
  - embroidery.h, 250
- set\_background\_color\_action
  - mainwindow.cpp, 481
- set\_crosshair\_color\_action
  - mainwindow.cpp, 481
- set\_cursor\_shape\_action
  - mainwindow.cpp, 482
- set\_dst\_variable
  - format\_dst.c, 345
- set\_enabled
  - embroidermodder.h, 427
  - interface.cpp, 455
- set\_grid\_color\_action
  - mainwindow.cpp, 482
- set\_object\_color
  - arc.c, 382
- set\_prompt\_prefix\_action
  - mainwindow.cpp, 482
- set\_rubber\_filter\_action
  - mainwindow.cpp, 482
- set\_rubber\_mode\_action
  - mainwindow.cpp, 482
- set\_rubber\_point\_action
  - mainwindow.cpp, 482
- set\_rubber\_text\_action
  - mainwindow.cpp, 482
- set\_visibility
  - embroidermodder.h, 427
  - interface.cpp, 455
- setBackgroundColor
  - MdiArea, 167
  - View, 225
- setBackgroundLogo
  - MdiArea, 167
- setBackgroundTexture
  - MdiArea, 167
- setColors
  - SelectBox, 199
- setCornerButton
  - View, 225
- setCrossHairColor
  - View, 225
- setCrossHairSize
  - View, 225
- setCurrentFile
  - MdiWindow, 176
- setCurrentText
  - CmdPrompt, 59
- setDirection
  - SelectBox, 199
- setGridColor
  - View, 225
- setHistory
  - CmdPrompt, 59
- setLine
  - Geometry, 128
- setMainWin
  - Application, 53
- setMouseCoord
  - StatusBar, 208
- setObjectArea
  - Geometry, 128
- setObjectCenter
  - Geometry, 130
- setObjectCenterX
  - Geometry, 130
- setObjectCenterY
  - Geometry, 130
- setObjectCircumference
  - Geometry, 130
- setObjectDiameter
  - Geometry, 130
- setObjectDiameterMajor
  - Geometry, 130
- setObjectDiameterMinor
  - Geometry, 130
- setObjectEndAngle

- Geometry, [131](#)
- setObjectEndPoint
  - Geometry, [131](#)
- setObjectEndPoint1
  - Geometry, [131](#)
- setObjectEndPoint2
  - Geometry, [131](#)
- setObjectLineWeight
  - Geometry, [132](#)
- setObjectMidPoint
  - Geometry, [132](#)
- setObjectPos
  - Geometry, [132](#)
- setObjectRadius
  - Geometry, [132](#)
- setObjectRadiusMajor
  - Geometry, [132](#)
- setObjectRadiusMinor
  - Geometry, [133](#)
- setObjectRect
  - Geometry, [133](#)
- setObjectSize
  - Geometry, [133](#)
- setObjectStartAngle
  - Geometry, [133](#)
- setObjectStartPoint
  - Geometry, [133](#)
- setObjectText
  - Geometry, [133](#)
- setObjectTextBackward
  - Geometry, [133](#)
- setObjectTextBold
  - Geometry, [134](#)
- setObjectTextFont
  - Geometry, [134](#)
- setObjectTextItalic
  - Geometry, [134](#)
- setObjectTextJustify
  - Geometry, [134](#)
- setObjectTextOverline
  - Geometry, [134](#)
- setObjectTextSize
  - Geometry, [135](#)
- setObjectTextStrikeOut
  - Geometry, [135](#)
- setObjectTextStyle
  - Geometry, [135](#)
- setObjectTextUnderline
  - Geometry, [135](#)
- setObjectTextUpsideDown
  - Geometry, [136](#)
- setObjectX
  - Geometry, [136](#)
- setObjectY
  - Geometry, [136](#)
- setPrefix
  - CmdPrompt, [59](#)
- setPromptBackgroundColor
  - CmdPrompt, [59](#)
- setPromptFontFamily
  - CmdPrompt, [60](#)
- setPromptFontSize
  - CmdPrompt, [60](#)
- setPromptFontStyle
  - CmdPrompt, [60](#)
- setPromptTextColor
  - CmdPrompt, [60](#)
- setRect
  - Geometry, [136](#)
- setRubberMode
  - View, [225](#)
- setRubberPoint
  - View, [225](#)
- SetRubberText
  - mainwindow.cpp, [483](#)
- setRubberText
  - View, [225](#)
- setRulerColor
  - View, [225](#)
- setSelectBoxColors
  - View, [225](#)
- setSelectedItems
  - PropertyEditor, [184](#)
- setShiftPressed
  - MainWindow, [159](#)
- setShiftReleased
  - MainWindow, [159](#)
- SetTextAngle\_action
  - mainwindow.cpp, [483](#)
- setTextFont
  - MainWindow, [159](#)
- setTextSize
  - MainWindow, [159](#)
- settings
  - embroidermodder.h, [430](#)
  - mainwindow.cpp, [487](#)
- settings-dialog.cpp
  - accept\_, [491](#)
  - display\_props, [491](#)
  - extensions, [492](#)
  - general\_props, [492](#)
  - make\_editing\_copy, [491](#)
  - opensave\_props, [492](#)
  - preview, [492](#)
  - prompt\_props, [492](#)
  - quick\_snap\_props, [492](#)
  - read\_settings, [491](#)
  - write\_settings, [491](#)
- Settings\_Dialog, [200](#)
  - ~Settings\_Dialog, [201](#)
  - acceptChanges, [202](#)
  - addColorsToComboBox, [202](#)
  - buttonBox, [207](#)
  - buttonCustomFilterClearAll, [202](#)
  - buttonCustomFilterClearAllClicked, [202](#)
  - buttonCustomFilterSelectAll, [202](#)



- buttonCustomFilterSelectAllClicked, [202](#)
- buttonQSnapClearAll, [202](#)
- buttonQSnapClearAllClicked, [202](#)
- buttonQSnapSelectAll, [202](#)
- buttonQSnapSelectAllClicked, [202](#)
- checkBoxCustomFilterStateChanged, [202](#)
- checkBoxGeneralMdiBGUseColorStateChanged, [202](#)
- checkBoxGeneralMdiBGUseLogoStateChanged, [202](#)
- checkBoxGeneralMdiBGUseTextureStateChanged, [202](#)
- checkBoxGridCenterOnOriginStateChanged, [203](#)
- checkBoxGridColorMatchCrossHairStateChanged, [203](#)
- checkBoxGridLoadFromFileStateChanged, [203](#)
- checkBoxLwtRealRenderStateChanged, [203](#)
- checkBoxLwtShowLwtStateChanged, [203](#)
- checkBoxPromptSaveHistoryAsHtmlStateChanged, [203](#)
- checkBoxRulerShowOnLoadStateChanged, [203](#)
- checkBoxShowScrollBarsStateChanged, [203](#)
- chooseDisplayBackgroundColor, [203](#)
- chooseDisplayCrossHairColor, [203](#)
- chooseDisplaySelectBoxLeftColor, [203](#)
- chooseDisplaySelectBoxLeftFill, [203](#)
- chooseDisplaySelectBoxRightColor, [204](#)
- chooseDisplaySelectBoxRightFill, [204](#)
- chooseGeneralMdiBackgroundColor, [204](#)
- chooseGeneralMdiBackgroundLogo, [204](#)
- chooseGeneralMdiBackgroundTexture, [204](#)
- chooseGridColor, [204](#)
- choosePromptBackgroundColor, [204](#)
- choosePromptTextColor, [204](#)
- chooseRulerColor, [204](#)
- comboBoxGridTypeCurrentIndexChanged, [204](#)
- comboBoxIconSizeCurrentIndexChanged, [204](#)
- comboBoxPromptFontFamilyCurrentIndexChanged, [205](#)
- comboBoxPromptFontStyleCurrentIndexChanged, [205](#)
- comboBoxQSnapLocatorColorCurrentIndexChanged, [205](#)
- comboBoxRulerMetricCurrentIndexChanged, [205](#)
- comboBoxScrollBarWidgetCurrentIndexChanged, [205](#)
- comboBoxSelectionCoolGripColorCurrentIndexChanged, [205](#)
- comboBoxSelectionHotGripColorCurrentIndexChanged, [205](#)
- create\_checkbox, [205](#)
- create\_float\_spinbox, [205](#)
- createTabDisplay, [205](#)
- createTabFilesPaths, [205](#)
- createTabGeneral, [206](#)
- createTabGridRuler, [206](#)
- createTabLineWeight, [206](#)
- createTabOpenSave, [206](#)
- createTabOrthoPolar, [206](#)
- createTabPrinting, [206](#)
- createTabPrompt, [206](#)
- createTabQuickSnap, [206](#)
- createTabQuickTrack, [206](#)
- createTabSelection, [206](#)
- createTabSnap, [206](#)
- currentDisplayBackgroundColorChanged, [206](#)
- currentDisplayCrossHairColorChanged, [206](#)
- currentDisplaySelectBoxLeftColorChanged, [206](#)
- currentDisplaySelectBoxLeftFillChanged, [206](#)
- currentDisplaySelectBoxRightColorChanged, [206](#)
- currentDisplaySelectBoxRightFillChanged, [207](#)
- currentGeneralMdiBackgroundColorChanged, [207](#)
- currentGridColorChanged, [207](#)
- currentPromptBackgroundColorChanged, [207](#)
- currentPromptTextColorChanged, [207](#)
- currentRulerColorChanged, [207](#)
- rejectChanges, [207](#)
- Settings\_Dialog, [201](#)
- spinBoxDisplaySelectBoxAlphaValueChanged, [207](#)
- spinBoxPromptFontSizeValueChanged, [207](#)
- spinBoxRulerPixelSizeValueChanged, [207](#)
- tabWidget, [207](#)
- settings\_dialog\_action
  - mainwindow.cpp, [483](#)
- settingsPrompt
  - MainWindow, [159](#)
- setUndoCleanIcon
  - MainWindow, [160](#)
- setViewBackgroundColor
  - MdiWindow, [176](#)
- setViewCrossHairColor
  - MdiWindow, [176](#)
- setViewGridColor
  - MdiWindow, [176](#)
- setViewRulerColor
  - MdiWindow, [176](#)
- setViewSelectBoxColors
  - MdiWindow, [177](#)
- sew, [9](#), [366](#)
- sewDecode
  - format\_sew.c, [366](#)
- shiftKeyPressedState
  - MainWindow, [164](#)
- shiftPressed
  - CmdPrompt, [61](#)
  - CmdPromptInput, [74](#)
- shiftReleased
  - CmdPrompt, [61](#)
  - CmdPromptInput, [74](#)
- showGroups
  - PropertyEditor, [184](#)
- showOneType
  - PropertyEditor, [184](#)
- showScrollBars
  - View, [225](#)

- showSettings
  - CmdPrompt, 61
  - CmdPromptInput, 75
- showViewScrollBars
  - MdiWindow, 177
- shv, 9, 367
- shv\_thread
  - embroidery.h, 250
- shvDecode
  - format\_shv.c, 367
- shvDecodeShort
  - format\_shv.c, 367
- shvThreadCount
  - embroidery.h, 269
  - thread-color.c, 410
- shvThreads
  - embroidery.h, 269
  - thread-color.c, 410
- side1
  - EmbSatinOutline\_, 100
- side2
  - EmbSatinOutline\_, 100
- Sierra Expanded, 304, 349
- Sigma\_Polyester
  - embroidery.h, 250
- signalMapper
  - PropertyEditor, 186
- signature
  - \_bcf\_file\_header, 49
- sigVersion
  - ThredHeader\_, 212
- Singer, 339, 377
- sizeHint
  - MdiWindow, 177
- sizeOfChainingEntryAtEndOfDifatSector
  - main.c, 403
- sizeOfDifatEntry
  - main.c, 403
- sizeOfDirectoryEntry
  - main.c, 403
- sizeOfFatEntry
  - main.c, 403
- sl
  - Node\_, 180
- someInt
  - SubDescriptor\_, 210
- someNum
  - SubDescriptor\_, 210
- someOtherInt
  - SubDescriptor\_, 210
- spare\_rubber\_action
  - mainwindow.cpp, 483
- spareRubber
  - View, 225
- spareRubberList
  - View, 230
- spinBoxDisplaySelectBoxAlphaValueChanged
  - Settings\_Dialog, 207
- spinBoxes
  - embroidermodder.h, 430
  - mainwindow.cpp, 487
- spinBoxPromptFontSizeValueChanged
  - Settings\_Dialog, 207
- spinBoxRulerPixelSizeValueChanged
  - Settings\_Dialog, 207
- spline
  - EmbGeometry\_, 91
- src/cmdprompt.cpp, 410
- src/em2\_dev\_script.py, 410
- src/embdetails-dialog.cpp, 410
- src/embroidermodder.cpp, 410
- src/embroidermodder.h, 411, 431
- src/imagewidget.cpp, 448
- src/interface.cpp, 449
- src/layer-manager.cpp, 457
- src/mainwindow-menus.cpp, 457
- src/mainwindow-toolbars.cpp, 457
- src/mainwindow.cpp, 457
- src/mdiarea.cpp, 487
- src/mdiwindow.cpp, 487
- src/objects.cpp, 488
- src/preview-dialog.cpp, 489
- src/property-editor.cpp, 489
- src/README.md, 490
- src/selectbox.cpp, 490
- src/settings-dialog.cpp, 491
- src/statusbar.cpp, 493
- src/undo-commands.cpp, 493
- src/undo-editor.cpp, 493
- src/view.cpp, 493
- sst, 9, 308, 368
- start
  - EmbArc\_, 80
  - EmbBezier\_, 82
  - EmbLine\_, 94
- startBlinking
  - CmdPrompt, 61
- startCommand
  - CmdPrompt, 61
  - CmdPromptInput, 75
- startGripping
  - View, 225
- startingSectorLocation
  - \_bcf\_directory\_entry, 45
- startResizeHistory
  - CmdPromptHistory, 68
- state
  - View, 230
- stateBits
  - \_bcf\_directory\_entry, 45
- StatusBar, 208
  - buttons, 209
  - context\_menu\_action, 208
  - context\_menu\_event, 208
  - setMouseCoord, 208
  - StatusBar, 208

- statusBarMouseCoord, 209
  - toggle, 208
- statusbar
  - embroidermodder.h, 430
  - mainwindow.cpp, 487
- statusBarMouseCoord
  - StatusBar, 209
- stitch
  - EmbArray\_, 82
  - EmbGeometry\_, 91
- stitch\_list
  - EmbPattern\_, 97
- stitchesJump
  - EmbDetailsDialog, 86
- stitchesReal
  - EmbDetailsDialog, 87
- stitchesTotal
  - EmbDetailsDialog, 87
- stitchesTrim
  - EmbDetailsDialog, 87
- stitchGranularity
  - ThredExtension\_, 211
- STOP
  - embroidery.h, 250
- stopBlinking
  - CmdPrompt, 61
  - CmdPromptInput, 75
- stopGripping
  - View, 226
- stopResizeHistory
  - CmdPromptHistory, 68
- streamSize
  - \_bcf\_directory\_entry, 45
- streamSizeHigh
  - \_bcf\_directory\_entry, 45
- String
  - embroidermodder.h, 419
- STRING\_LIST\_TYPE
  - embroidermodder.h, 418
- STRING\_TYPE
  - embroidermodder.h, 419
- stringInArray
  - embroidery\_internal.h, 310
  - main.c, 402
- StringList
  - embroidermodder.h, 419
- stringVal
  - VipHeader\_, 231
- stub\_testing
  - MainWindow, 160
- stx, 368
- stxColor
  - StxThread\_, 209
- stxReadThread
  - format\_stx.c, 368
- StxThread
  - embroidery\_internal.h, 292
- StxThread\_, 209
- colorCode, 209
  - colorName, 209
  - sectionName, 209
  - stxColor, 209
  - subDescriptors, 209
- styleHash
  - CmdPrompt, 62
- SubDescriptor
  - embroidery\_internal.h, 293
- SubDescriptor\_, 209
  - colorCode, 210
  - colorName, 210
  - someInt, 210
  - someNum, 210
  - someOtherInt, 210
- subDescriptors
  - StxThread\_, 209
- subMenuHash
  - embroidermodder.h, 430
  - mainwindow.cpp, 487
- subPathList
  - Geometry, 136
- Sulky\_Rayon
  - embroidery.h, 250
- Sunstar, 308, 368
- svg, 9, 369
- SVG\_ATTRIBUTE
  - embroidery\_internal.h, 291
- SVG\_CATCH\_ALL
  - embroidery\_internal.h, 291
- SVG\_Colors
  - embroidery.h, 250
- SVG\_CREATOR\_EMBROIDERMODDER
  - embroidery\_internal.h, 291
- SVG\_CREATOR\_ILLUSTRATOR
  - embroidery\_internal.h, 291
- SVG\_CREATOR\_INKSCAPE
  - embroidery\_internal.h, 291
- SVG\_CREATOR\_NULL
  - embroidery\_internal.h, 291
- SVG\_ELEMENT
  - embroidery\_internal.h, 291
- SVG\_EXPECT\_ATTRIBUTE
  - embroidery\_internal.h, 291
- SVG\_EXPECT\_ELEMENT
  - embroidery\_internal.h, 292
- SVG\_EXPECT\_NULL
  - embroidery\_internal.h, 292
- SVG\_EXPECT\_VALUE
  - embroidery\_internal.h, 292
- SVG\_MEDIA\_PROPERTY
  - embroidery\_internal.h, 292
- SVG\_NULL
  - embroidery\_internal.h, 292
- SVG\_PROPERTY
  - embroidery\_internal.h, 292
- SvgAttribute
  - embroidery\_internal.h, 293

- SvgAttribute\_, 210
  - name, 210
  - value, 210
- svgCreator
  - format\_svg.c, 369
- svgExpect
  - format\_svg.c, 369
- svgMultiValue
  - format\_svg.c, 369
- t01, 308, 370
- t09, 308, 370
- table
  - Huffman, 142
- table\_width
  - Huffman, 142
- tabPressed
  - CmdPrompt, 61
  - CmdPromptInput, 75
- tabWidget
  - Settings\_Dialog, 207
- Tajima, 343
- tap, 9, 371
- tempBaseObj
  - View, 230
- testEmbCircle
  - embroidery\_internal.h, 310
- testEmbCircle\_2
  - embroidery\_internal.h, 310
- testEmbFormat
  - embroidery\_internal.h, 310
- testGeomArc
  - embroidery\_internal.h, 310
- testMain
  - embroidery.h, 267
- testTangentPoints
  - embroidery\_internal.h, 310
- testThreadColor
  - embroidery\_internal.h, 310
- text
  - EmbTextMulti\_, 102
  - EmbTextSingle\_, 103
- text.c
  - textSingle\_gripEdit, 387
  - textSingle\_mouseSnapPoint, 387
  - textSingle\_paint, 388
  - textSingle\_setJustify, 388
  - textSingle\_setTextBackward, 388
  - textSingle\_setTextBold, 388
  - textSingle\_setTextFont, 388
  - textSingle\_setTextItalic, 388
  - textSingle\_setTextOverline, 388
  - textSingle\_setTextSize, 388
  - textSingle\_setTextStrikeOut, 388
  - textSingle\_setTextStyle, 388
  - textSingle\_setTextUnderline, 388
  - textSingle\_setTextUpsideDown, 388
  - textSingle\_updateRubber, 388
- textFontSelector
  - MainWindow, 164
- textFontSelectorCurrentFontChanged
  - MainWindow, 160
- textSingle\_gripEdit
  - text.c, 387
- textSingle\_mouseSnapPoint
  - text.c, 387
- textSingle\_paint
  - text.c, 388
- textSingle\_setJustify
  - text.c, 388
- textSingle\_setTextBackward
  - text.c, 388
- textSingle\_setTextBold
  - text.c, 388
- textSingle\_setTextFont
  - text.c, 388
- textSingle\_setTextItalic
  - text.c, 388
- textSingle\_setTextOverline
  - text.c, 388
- textSingle\_setTextSize
  - text.c, 388
- textSingle\_setTextStrikeOut
  - text.c, 388
- textSingle\_setTextStyle
  - text.c, 388
- textSingle\_setTextUnderline
  - text.c, 388
- textSingle\_setTextUpsideDown
  - text.c, 388
- textSingle\_updateRubber
  - text.c, 388
- textSizeSelector
  - MainWindow, 164
- textSizeSelectorIndexChanged
  - MainWindow, 160
- thr, 309, 372
- thread
  - EmbArray\_, 82
  - EmbGeometry\_, 91
- thread-color.c
  - \_dxfColorTable, 409
  - brand\_codes, 409
  - brand\_codes\_files, 409
  - husThreads, 409
  - jefThreads, 409
  - pcmThreads, 409
  - pecThreadCount, 409
  - pecThreads, 410
  - shvThreadCount, 410
  - shvThreads, 410
  - threadColor, 409
  - threadColorName, 409
  - threadColorNum, 409
- thread\_color
  - embroidery.h, 253
- thread\_color\_, 210

- hex\_code, [211](#)
- manufacturer\_code, [211](#)
- name, [211](#)
- thread\_list
  - EmbPattern\_, [97](#)
- ThreadArt\_Polyester
  - embroidery.h, [250](#)
- ThreadArt\_Rayon
  - embroidery.h, [250](#)
- threadColor
  - embroidery.h, [268](#)
  - thread-color.c, [409](#)
- threadColorName
  - embroidery.h, [268](#)
  - thread-color.c, [409](#)
- threadColorNum
  - embroidery.h, [268](#)
  - thread-color.c, [409](#)
- ThreaDelight\_Polyester
  - embroidery.h, [250](#)
- threadLength
  - \_vp3Hoop, [51](#)
- ThreadWorks, [309](#), [372](#)
- ThredExtension
  - embroidery\_internal.h, [293](#)
- ThredExtension\_, [211](#)
  - auxFormat, [211](#)
  - creatorName, [211](#)
  - hoopX, [211](#)
  - hoopY, [211](#)
  - modifierName, [211](#)
  - reserved, [211](#)
  - stitchGranularity, [211](#)
- ThredHeader
  - embroidery\_internal.h, [293](#)
- ThredHeader\_, [212](#)
  - hoopSize, [212](#)
  - length, [212](#)
  - numStiches, [212](#)
  - reserved, [212](#)
  - sigVersion, [212](#)
- threshold\_method
  - fill.c, [330](#)
- Tick
  - Geometry, [110](#)
- tile
  - MdiArea, [167](#)
- tip\_of\_the\_day\_action
  - mainwindow.cpp, [483](#)
- tipOfTheDay
  - MainWindow, [160](#)
- tmpHeight
  - CmdPromptHistory, [68](#)
- to\_EmbVector
  - embroidermodder.h, [427](#)
  - interface.cpp, [455](#)
- to\_qlist
  - embroidermodder.h, [428](#)
- interface.cpp, [455](#)
- to\_QPointF
  - embroidermodder.h, [428](#)
  - interface.cpp, [456](#)
- to\_string\_vector
  - embroidermodder.h, [428](#)
  - interface.cpp, [456](#)
- to\_vector
  - embroidermodder.h, [428](#)
  - interface.cpp, [456](#)
- toCenter
  - UndoableCommand, [215](#)
- todo\_action
  - mainwindow.cpp, [483](#)
- toggle
  - StatusBar, [208](#)
- toggleGrid
  - MainWindow, [161](#)
  - View, [226](#)
- toggleLwt
  - MainWindow, [161](#)
  - View, [226](#)
- toggleOrtho
  - View, [226](#)
- togglePickAddMode
  - PropertyEditor, [184](#)
- togglePolar
  - View, [226](#)
- toggleQSnap
  - View, [226](#)
- toggleQTrack
  - View, [226](#)
- toggleReal
  - View, [226](#)
- toggleRuler
  - MainWindow, [161](#)
  - View, [226](#)
- toggleSnap
  - View, [226](#)
- tokenize
  - embroidermodder.h, [428](#)
  - interface.cpp, [456](#)
- toolbarHash
  - embroidermodder.h, [430](#)
  - mainwindow.cpp, [487](#)
- toolButtonPickAdd
  - PropertyEditor, [186](#)
- toolButtonQSelect
  - PropertyEditor, [186](#)
- toolButtons
  - embroidermodder.h, [430](#)
  - mainwindow.cpp, [487](#)
- top
  - \_vp3Hoop, [51](#)
  - EmbRect\_, [100](#)
  - hoop\_padding, [141](#)
- top2
  - \_vp3Hoop, [51](#)

- toPolyline
  - SaveObject, 196
- toTransform
  - UndoableCommand, 215
- Toyota, 335
- transactionSignatureNumber
  - \_bcf\_file\_header, 49
- translate\_str
  - embroidermodder.h, 429
  - interface.cpp, 456
- treeView
  - LayerManager, 145
- TRIM
  - embroidery.h, 250
- txt, 309, 372
- Type
  - Geometry, 140
- type
  - EmbArray\_, 82
  - EmbFormatList\_, 89
  - EmbGeometry\_, 91
  - Geometry, 136
  - Node\_, 180
- u00, 309, 373
- u01, 9, 309, 373
- undo
  - UndoableCommand, 214
  - UndoEditor, 216
- undo\_action
  - mainwindow.cpp, 484
- UndoableCommand, 212
  - after, 214
  - angle, 214
  - before, 214
  - command, 214
  - delta, 215
  - done, 215
  - factor, 215
  - fromCenter, 215
  - fromTransform, 215
  - gview, 215
  - id, 214
  - mergeWith, 214
  - mirror, 214
  - mirrorLine, 215
  - navType, 215
  - object, 215
  - pivot, 215
  - redo, 214
  - rotate, 214
  - toCenter, 215
  - toTransform, 215
  - undo, 214
  - UndoableCommand, 213, 214
- UndoEditor, 215
  - ~UndoEditor, 216
  - addStack, 216
  - canRedo, 216
  - canUndo, 216
  - focusWidget, 217
  - iconDir, 217
  - iconSize, 217
  - redo, 216
  - redoText, 216
  - undo, 216
  - UndoEditor, 216
  - undoGroup, 217
  - undoText, 216
  - undoView, 217
  - updateCleanIcon, 216
- undoGroup
  - UndoEditor, 217
- undoPressed
  - CmdPrompt, 61
  - CmdPromptInput, 75
- undoStack
  - View, 230
- undoText
  - UndoEditor, 216
- undoView
  - UndoEditor, 217
- unknown
  - VipHeader\_, 231
- unknown2
  - \_vp3Hoop, 51
- unknown3
  - \_vp3Hoop, 51
- unknown4
  - \_vp3Hoop, 52
- UNKNOWN\_TYPE
  - embroidermodder.h, 419
- updateAllViewBackgroundColors
  - MainWindow, 161
- updateAllViewCrossHairColors
  - MainWindow, 161
- updateAllViewGridColors
  - MainWindow, 161
- updateAllViewRulerColors
  - MainWindow, 162
- updateAllViewScrollBars
  - MainWindow, 162
- updateAllViewSelectBoxColors
  - MainWindow, 162
- updateArcRect
  - Geometry, 136
- updateCleanIcon
  - UndoEditor, 216
- updateColorLinetypeLineweight
  - MdiWindow, 177
- updateComboBoxBoolIfVaries
  - PropertyEditor, 184
- updateComboBoxStrIfVaries
  - PropertyEditor, 185
- updateCurrentText
  - CmdPromptInput, 75
- updateFontComboBoxStrIfVaries

- PropertyEditor, 185
- updateLeader
  - Geometry, 137
- updateLineEditNumIfVaries
  - PropertyEditor, 185
- updateLineEditStrIfVaries
  - PropertyEditor, 185
- updateMenuToolBarStatusBar
  - MainWindow, 162
- updateMouseCoords
  - View, 226
- updatePath
  - Geometry, 137
- updatePickAddMode
  - MainWindow, 163
- updatePickAddModeButton
  - PropertyEditor, 185
- updateRubber
  - Geometry, 137
- updateStyle
  - CmdPrompt, 61
- upPressed
  - CmdPrompt, 61
  - CmdPromptInput, 75
- usage\_msg
  - embroidermodder.cpp, 411
- useBackgroundColor
  - MdiArea, 167
- useBackgroundLogo
  - MdiArea, 168
- useBackgroundTexture
  - MdiArea, 168
- useColor
  - MdiArea, 168
- useLogo
  - MdiArea, 169
- useTexture
  - MdiArea, 169
- validFileFormat
  - embroidermodder.h, 429
  - mainwindow.cpp, 484
- validRGB
  - mainwindow.cpp, 484
- value
  - SvgAttribute\_, 210
- vector
  - EmbGeometry\_, 91
- vector.c
  - embVector\_add, 389
  - embVector\_angle, 389
  - embVector\_average, 389
  - embVector\_cross, 389
  - embVector\_distance, 390
  - embVector\_dot, 390
  - embVector\_length, 390
  - embVector\_multiply, 390
  - embVector\_normalize, 390
  - embVector\_relativeX, 390
  - embVector\_relativeY, 391
  - embVector\_subtract, 391
  - embVector\_transpose\_product, 391
  - embVector\_unit, 391
- VECTOR\_TYPE
  - embroidermodder.h, 419
- version\_action
  - mainwindow.cpp, 484
- View, 217
  - ~View, 220
  - addObject, 220
  - addToRubberRoom, 220
  - alignScenePointWithViewPoint, 220
  - allowRubber, 221
  - allowZoomIn, 221
  - allowZoomOut, 221
  - center, 221
  - centerAt, 221
  - clearRubberRoom, 221
  - clearSelection, 221
  - contextMenuEvent, 221
  - copy, 221
  - copySelected, 221
  - cornerButtonClicked, 221
  - createGrid, 221
  - createGridIso, 221
  - createGridPolar, 221
  - createGridRect, 221
  - createObjectList, 221
  - createOrigin, 222
  - createRulerTextPath, 222
  - crosshairColor, 227
  - crosshairSize, 227
  - cut, 222
  - cutCopyMousePoint, 227
  - deleteObject, 222
  - deletePressed, 222
  - deleteSelected, 222
  - drawBackground, 222
  - drawForeground, 222
  - enterEvent, 222
  - escapePressed, 222
  - getUndoStack, 222
  - gridColor, 227
  - gridPath, 227
  - gripBaseObj, 227
  - gripColorCool, 227
  - gripColorHot, 228
  - grippingActive, 228
  - gripSize, 228
  - gscene, 228
  - hashDeletedObjects, 228
  - isLwtEnabled, 222
  - isRealEnabled, 222
  - loadRulerSettings, 222
  - mirrorSelected, 222
  - mouseDoubleClickEvent, 223
  - mouseMoveEvent, 223



[mousePressEvent](#), 223  
[mouseReleaseEvent](#), 223  
[moveAction](#), 223  
[movePoint](#), 228  
[moveSelected](#), 223  
[movingActive](#), 228  
[numSelected](#), 223  
[originPath](#), 228  
[panDistance](#), 228  
[panDown](#), 223  
[panLeft](#), 223  
[panningActive](#), 228  
[panningPointActive](#), 228  
[panningRealTimeActive](#), 228  
[panPoint](#), 223  
[panRealTime](#), 223  
[panRight](#), 223  
[panStart](#), 223  
[panStartX](#), 228  
[panStartY](#), 228  
[panUp](#), 223  
[paste](#), 224  
[pasteDelta](#), 228  
[pasteObjectItemGroup](#), 228  
[pastingActive](#), 228  
[pickBoxSize](#), 228  
[pressPoint](#), 229  
[previewActive](#), 229  
[previewData](#), 229  
[previewMode](#), 229  
[previewObjectItemGroup](#), 229  
[previewObjectList](#), 229  
[previewOff](#), 224  
[previewOn](#), 224  
[previewPoint](#), 229  
[qSnapActive](#), 229  
[qsnapApertureSize](#), 229  
[qsnapLocatorColor](#), 229  
[qsnapLocatorSize](#), 229  
[qSnapToggle](#), 229  
[rapidMoveActive](#), 229  
[recalculateLimits](#), 224  
[releasePoint](#), 229  
[repeatAction](#), 224  
[rotateAction](#), 224  
[rotateSelected](#), 224  
[roundToMultiple](#), 224  
[rubberRoomList](#), 229  
[rulerColor](#), 229  
[rulerMetric](#), 229  
[rulerPixelSize](#), 229  
[scaleAction](#), 224  
[scaleSelected](#), 224  
[sceneGripPoint](#), 230  
[sceneMousePoint](#), 230  
[sceneMovePoint](#), 230  
[scenePressPoint](#), 230  
[sceneReleasePoint](#), 230  
[selectAll](#), 224  
[selectBox](#), 230  
[selected\\_items](#), 224  
[selectingActive](#), 230  
[selectionChanged](#), 224  
[setBackgroundColor](#), 225  
[setCornerButton](#), 225  
[setCrossHairColor](#), 225  
[setCrossHairSize](#), 225  
[setGridColor](#), 225  
[setRubberMode](#), 225  
[setRubberPoint](#), 225  
[setRubberText](#), 225  
[setRulerColor](#), 225  
[setSelectBoxColors](#), 225  
[showScrollBars](#), 225  
[spareRubber](#), 225  
[spareRubberList](#), 230  
[startGripping](#), 225  
[state](#), 230  
[stopGripping](#), 226  
[tempBaseObj](#), 230  
[toggleGrid](#), 226  
[toggleLwt](#), 226  
[toggleOrtho](#), 226  
[togglePolar](#), 226  
[toggleQSnap](#), 226  
[toggleQTrack](#), 226  
[toggleReal](#), 226  
[toggleRuler](#), 226  
[toggleSnap](#), 226  
[undoStack](#), 230  
[updateMouseCoords](#), 226  
[View](#), 220  
[viewMousePoint](#), 230  
[vulcanizeObject](#), 226  
[vulcanizeRubberRoom](#), 226  
[wheelEvent](#), 226  
[willOverflowInt32](#), 227  
[willUnderflowInt32](#), 227  
[zoomExtents](#), 227  
[zoomIn](#), 227  
[zoomOut](#), 227  
[zoomSelected](#), 227  
[zoomToPoint](#), 227  
[zoomWindow](#), 227  
[zoomWindowActive](#), 230  
[view.cpp](#)  
     [contains](#), 493  
[viewMousePoint](#)  
     [View](#), 230  
[vip](#), 9, 269, 375  
[vipCompressData](#)  
     [format\\_vip.c](#), 374  
[vipDecodeByte](#)  
     [format\\_vip.c](#), 374  
[vipDecodeStitchType](#)  
     [format\\_vip.c](#), 374



- vipDecodingTable
  - embroidery.h, [269](#)
  - format\_vip.c, [374](#)
- vipDecompressData
  - format\_vip.c, [374](#)
- vipEncodeByte
  - format\_vip.c, [374](#)
- vipEncodeStitchType
  - format\_vip.c, [374](#)
- VipHeader
  - embroidery\_internal.h, [293](#)
- VipHeader\_, [230](#)
  - attributeOffset, [231](#)
  - colorLength, [231](#)
  - magicCode, [231](#)
  - negativeXHoopSize, [231](#)
  - negativeYHoopSize, [231](#)
  - numberOfColors, [231](#)
  - numberOfStitches, [231](#)
  - postitiveXHoopSize, [231](#)
  - postitiveYHoopSize, [231](#)
  - stringVal, [231](#)
  - unknown, [231](#)
  - xOffset, [231](#)
  - yOffset, [231](#)
- vp3, [9](#), [376](#)
- vp3Decode
  - format\_vp3.c, [375](#)
- vp3DecodeInt16
  - format\_vp3.c, [375](#)
- vp3Hoop
  - embroidery\_internal.h, [293](#)
- vp3PatchByteCount
  - format\_vp3.c, [375](#)
- vp3ReadHoopSection
  - format\_vp3.c, [375](#)
- vp3ReadString
  - format\_vp3.c, [376](#)
- vp3WriteString
  - format\_vp3.c, [376](#)
- vp3WriteStringLen
  - format\_vp3.c, [376](#)
- vulcanize
  - Geometry, [137](#)
- vulcanize\_action
  - mainwindow.cpp, [484](#)
- vulcanizeObject
  - View, [226](#)
- vulcanizeRubberRoom
  - View, [226](#)
- whats\_this\_action
  - mainwindow.cpp, [484](#)
- wheelEvent
  - View, [226](#)
- WHITESPACE
  - main.c, [403](#)
- width
  - \_vp3Hoop, [52](#)
- EmblImage\_, [92](#)
- willOverflowInt32
  - View, [227](#)
- willUnderflowInt32
  - View, [227](#)
- window\_action
  - mainwindow.cpp, [485](#)
- windowMenuAboutToShow
  - MainWindow, [163](#)
- windowMenuActivated
  - MainWindow, [163](#)
- wizardTipOfTheDay
  - mainwindow.cpp, [487](#)
- write100
  - embroidery\_internal.h, [310](#)
  - format\_100.c, [335](#)
- write10o
  - embroidery\_internal.h, [310](#)
  - format\_10o.c, [336](#)
- write\_24bit
  - embroidery\_internal.h, [310](#)
  - encoding.c, [326](#)
  - main.c, [402](#)
- write\_external\_color\_file
  - EmbFormatList\_, [89](#)
- write\_settings
  - embroidermodder.h, [429](#)
  - settings-dialog.cpp, [491](#)
- writeArt
  - embroidery\_internal.h, [311](#)
  - format\_art.c, [336](#)
- writeBmc
  - embroidery\_internal.h, [311](#)
  - format\_bmc.c, [337](#)
- writeBro
  - embroidery\_internal.h, [311](#)
  - format\_bro.c, [337](#)
- writeCnd
  - embroidery\_internal.h, [311](#)
  - format\_cnd.c, [338](#)
- writeCol
  - embroidery\_internal.h, [311](#)
  - format\_col.c, [339](#)
- writeCsd
  - embroidery\_internal.h, [311](#)
  - format\_csd.c, [340](#)
- writeCsv
  - embroidery\_internal.h, [311](#)
  - format\_csv.c, [341](#)
- writeDat
  - embroidery\_internal.h, [311](#)
  - format\_dat.c, [341](#)
- writeDem
  - embroidery\_internal.h, [311](#)
  - format\_dem.c, [342](#)
- writeDsb
  - embroidery\_internal.h, [311](#)
  - format\_dsb.c, [343](#)

writeDst  
    embroidery\_internal.h, 311  
    format\_dst.c, 345

writeDsz  
    embroidery\_internal.h, 312  
    format\_dsz.c, 345

writeDxf  
    embroidery\_internal.h, 312  
    format\_dxf.c, 346

writeEdr  
    embroidery\_internal.h, 312  
    format\_edr.c, 346

writeEmd  
    embroidery\_internal.h, 312  
    format\_emd.c, 347

writeExp  
    embroidery\_internal.h, 312  
    format\_exp.c, 348

writeExy  
    embroidery\_internal.h, 312  
    format\_exy.c, 348

writeEys  
    embroidery\_internal.h, 312  
    format\_ey.c, 349

writeFxy  
    embroidery\_internal.h, 312  
    format\_fxy.c, 349

writeGc  
    embroidery\_internal.h, 312  
    format\_gc.c, 350

writeGnc  
    embroidery\_internal.h, 312  
    format\_gnc.c, 350

writeGt  
    embroidery\_internal.h, 312  
    format\_gt.c, 351

writeHus  
    embroidery\_internal.h, 313  
    format\_hus.c, 352

writelnImage  
    format\_pec.c, 360  
    image.c, 392

writelnb  
    embroidery\_internal.h, 313  
    format\_inb.c, 352

writelnf  
    embroidery\_internal.h, 313  
    format\_inf.c, 353

writeJef  
    embroidery\_internal.h, 313  
    format\_jef.c, 354

writeKsm  
    embroidery\_internal.h, 313  
    format\_ksm.c, 354

writeMax  
    embroidery\_internal.h, 313  
    format\_max.c, 355

writeMit  
    embroidery\_internal.h, 313  
    format\_mit.c, 356

writeNew  
    embroidery\_internal.h, 313  
    format\_new.c, 356

writeOfm  
    embroidery\_internal.h, 313  
    format\_ofm.c, 357

writePcd  
    embroidery\_internal.h, 313  
    format\_pcd.c, 358

writePcm  
    embroidery\_internal.h, 313  
    format\_pcm.c, 358

writePcq  
    embroidery\_internal.h, 314  
    format\_pcq.c, 359

writePcs  
    embroidery\_internal.h, 314  
    format\_pcs.c, 359

writePec  
    embroidery\_internal.h, 314  
    format\_pec.c, 360

writePecStitches  
    embroidery\_internal.h, 314  
    format\_pec.c, 360

writePel  
    embroidery\_internal.h, 314  
    format\_pel.c, 361

writePem  
    embroidery\_internal.h, 314  
    format\_pem.c, 361

writePes  
    embroidery\_internal.h, 314  
    format\_pes.c, 364

writePhb  
    embroidery\_internal.h, 314  
    format\_phb.c, 364

writePhc  
    embroidery\_internal.h, 314  
    format\_phc.c, 365

writePlt  
    embroidery\_internal.h, 314  
    format\_plt.c, 365

writer\_state  
    EmbFormatList\_, 89

writeRgb  
    embroidery\_internal.h, 315  
    format\_rgb.c, 366

writeSew  
    embroidery\_internal.h, 315  
    format\_sew.c, 366

writeShv  
    embroidery\_internal.h, 315  
    format\_shv.c, 367

writeSst  
    embroidery\_internal.h, 315  
    format\_sst.c, 368

- writeStx
  - embroidery\_internal.h, [315](#)
  - format\_stx.c, [368](#)
- writeSvg
  - embroidery\_internal.h, [315](#)
  - format\_svg.c, [369](#)
- writeT01
  - embroidery\_internal.h, [315](#)
  - format\_t01.c, [370](#)
- writeT09
  - embroidery\_internal.h, [315](#)
  - format\_t09.c, [370](#)
- writeTap
  - embroidery\_internal.h, [315](#)
  - format\_tap.c, [371](#)
- writeThr
  - embroidery\_internal.h, [315](#)
  - format\_thr.c, [372](#)
- writeTxt
  - embroidery\_internal.h, [315](#)
  - format\_txt.c, [372](#)
- writeU00
  - embroidery\_internal.h, [316](#)
  - format\_u00.c, [373](#)
- writeU01
  - embroidery\_internal.h, [316](#)
  - format\_u01.c, [373](#)
- writeVip
  - embroidery\_internal.h, [316](#)
  - format\_vip.c, [374](#)
- writeVp3
  - embroidery\_internal.h, [316](#)
  - format\_vp3.c, [376](#)
- writeXxx
  - embroidery\_internal.h, [316](#)
  - format\_xxx.c, [376](#)
- writeZsk
  - embroidery\_internal.h, [316](#)
  - format\_zsk.c, [377](#)
- x
  - EmbStitch\_, [101](#)
  - EmbVector\_, [105](#)
- x\_values
  - Geometry, [140](#)
- xOffset
  - \_vp3Hoop, [52](#)
  - VipHeader\_, [231](#)
- xxx, [9](#), [377](#)
- xxxDecodeByte
  - format\_xxx.c, [376](#)
- xxxEncodeDesign
  - format\_xxx.c, [377](#)
- xxxEncodeStitch
  - format\_xxx.c, [377](#)
- xxxEncodeStop
  - format\_xxx.c, [377](#)
- y
  - EmbStitch\_, [102](#)
  - EmbVector\_, [105](#)
  - y\_values
    - Geometry, [140](#)
  - year
    - EmbTime\_, [104](#)
  - YELLOW\_TERM\_COLOR
    - embroidery\_internal.h, [292](#)
  - yOffset
    - \_vp3Hoop, [52](#)
    - VipHeader\_, [231](#)
  - Z102\_Isacord\_Polyester
    - embroidery.h, [250](#)
  - zoom\_action
    - mainwindow.cpp, [485](#)
  - zoomExtents
    - View, [227](#)
  - zoomExtentsAllSubWindows
    - MdiArea, [168](#)
  - zoomIn
    - View, [227](#)
  - zoomOut
    - View, [227](#)
  - zoomSelected
    - View, [227](#)
  - zoomToPoint
    - View, [227](#)
  - zoomWindow
    - View, [227](#)
  - zoomWindowActive
    - View, [230](#)
  - zsk, [9](#), [377](#)
  - ZSK USA, [303](#), [345](#), [377](#)