

Embroidermodder

Generated by Doxygen 1.9.4

<b>1 Overview</b>	<b>1</b>
1.0.1 License	1
<b>2 About</b>	<b>2</b>
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
<b>3 Downloads</b>	<b>5</b>
3.1 Alpha Build	5
<b>4 Changelog</b>	<b>6</b>
<b>5 Ideas</b>	<b>6</b>
<b>6 Formats</b>	<b>6</b>
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
<b>7 Geometry and Algorithms</b>	<b>9</b>
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
<b>8 GNU Free Documentation License</b>	<b>18</b>
<b>9 Contributor Covenant Code of Conduct</b>	<b>24</b>
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
<b>10 Privacy Policy for Embroidery Viewer</b>	<b>26</b>
10.0.1 CONTACT US . . . . .	26
<b>11 Todo List</b>	<b>26</b>
<b>12 Namespace Index</b>	<b>30</b>
12.1 Namespace List . . . . .	30
<b>13 Hierarchical Index</b>	<b>30</b>
13.1 Class Hierarchy . . . . .	30
<b>14 Class Index</b>	<b>33</b>
14.1 Class List . . . . .	33
<b>15 File Index</b>	<b>35</b>
15.1 File List . . . . .	35
<b>16 Namespace Documentation</b>	<b>39</b>
16.1 em2_dev_script Namespace Reference . . . . .	39
16.1.1 Detailed Description . . . . .	39
16.1.2 Variable Documentation . . . . .	39
<b>17 Class Documentation</b>	<b>40</b>
17.1 _bcf_directory Struct Reference . . . . .	40
17.1.1 Detailed Description . . . . .	40

---

17.1.2 Member Data Documentation . . . . .	40
17.2 _bcf_directory_entry Struct Reference . . . . .	41
17.2.1 Member Data Documentation . . . . .	41
17.3 _bcf_file Struct Reference . . . . .	42
17.3.1 Member Data Documentation . . . . .	43
17.4 _bcf_file_difat Struct Reference . . . . .	43
17.4.1 Member Data Documentation . . . . .	43
17.5 _bcf_file_fat Struct Reference . . . . .	44
17.5.1 Member Data Documentation . . . . .	44
17.6 _bcf_file_header Struct Reference . . . . .	44
17.6.1 Detailed Description . . . . .	45
17.6.2 Member Data Documentation . . . . .	45
17.7 _vp3Hoop Struct Reference . . . . .	47
17.7.1 Member Data Documentation . . . . .	47
17.8 Application Class Reference . . . . .	49
17.8.1 Detailed Description . . . . .	50
17.8.2 Constructor & Destructor Documentation . . . . .	50
17.8.3 Member Function Documentation . . . . .	50
17.8.4 Member Data Documentation . . . . .	51
17.9 CmdPrompt Class Reference . . . . .	51
17.9.1 Detailed Description . . . . .	53
17.9.2 Constructor & Destructor Documentation . . . . .	53
17.9.3 Member Function Documentation . . . . .	53
17.9.4 Member Data Documentation . . . . .	59
17.10 CmdPromptHandle Class Reference . . . . .	60
17.10.1 Detailed Description . . . . .	60
17.10.2 Constructor & Destructor Documentation . . . . .	60
17.10.3 Member Function Documentation . . . . .	61
17.10.4 Member Data Documentation . . . . .	62
17.11 CmdPromptHistory Class Reference . . . . .	62
17.11.1 Detailed Description . . . . .	63
17.11.2 Constructor & Destructor Documentation . . . . .	63
17.11.3 Member Function Documentation . . . . .	64
17.11.4 Member Data Documentation . . . . .	65
17.12 CmdPromptInput Class Reference . . . . .	66
17.12.1 Constructor & Destructor Documentation . . . . .	67
17.12.2 Member Function Documentation . . . . .	68
17.12.3 Member Data Documentation . . . . .	73
17.13 CmdPromptSplitter Class Reference . . . . .	74
17.13.1 Detailed Description . . . . .	74
17.13.2 Constructor & Destructor Documentation . . . . .	74
17.13.3 Member Function Documentation . . . . .	75

17.14 Compress Struct Reference . . . . .	75
17.14.1 Member Data Documentation . . . . .	76
17.15 EmbAlignedDim_ Struct Reference . . . . .	77
17.15.1 Member Data Documentation . . . . .	77
17.16 EmbAngularDim_ Struct Reference . . . . .	77
17.16.1 Member Data Documentation . . . . .	77
17.17 EmbArc_ Struct Reference . . . . .	78
17.17.1 Detailed Description . . . . .	78
17.17.2 Member Data Documentation . . . . .	78
17.18 EmbArcLengthDim_ Struct Reference . . . . .	78
17.18.1 Member Data Documentation . . . . .	79
17.19 EmbArray_ Struct Reference . . . . .	79
17.19.1 Member Data Documentation . . . . .	79
17.20 EmbBezier_ Struct Reference . . . . .	80
17.20.1 Member Data Documentation . . . . .	80
17.21 EmbBlock_ Struct Reference . . . . .	81
17.21.1 Member Data Documentation . . . . .	81
17.22 EmbCircle_ Struct Reference . . . . .	81
17.22.1 Member Data Documentation . . . . .	81
17.23 EmbColor_ Struct Reference . . . . .	81
17.23.1 Detailed Description . . . . .	82
17.23.2 Member Data Documentation . . . . .	82
17.24 EmbDetailsDialog Class Reference . . . . .	82
17.24.1 Detailed Description . . . . .	83
17.24.2 Constructor & Destructor Documentation . . . . .	83
17.24.3 Member Function Documentation . . . . .	83
17.24.4 Member Data Documentation . . . . .	84
17.25 EmbDiameterDim_ Struct Reference . . . . .	85
17.25.1 Member Data Documentation . . . . .	85
17.26 EmbEllipse_ Struct Reference . . . . .	85
17.26.1 Member Data Documentation . . . . .	85
17.27 EmbFormatList_ Struct Reference . . . . .	86
17.27.1 Member Data Documentation . . . . .	86
17.28 EmbGeometry_ Struct Reference . . . . .	87
17.28.1 Member Data Documentation . . . . .	87
17.29 EmbImage_ Struct Reference . . . . .	89
17.29.1 Member Data Documentation . . . . .	89
17.30 EmbInfiniteLine_ Struct Reference . . . . .	90
17.30.1 Member Data Documentation . . . . .	90
17.31 EmbLayer_ Struct Reference . . . . .	90
17.31.1 Member Data Documentation . . . . .	91
17.32 EmbLeaderDim_ Struct Reference . . . . .	91

17.32.1 Member Data Documentation . . . . .	91
17.33 EmbLine_ Struct Reference . . . . .	91
17.33.1 Member Data Documentation . . . . .	92
17.34 EmbLinearDim_ Struct Reference . . . . .	92
17.34.1 Member Data Documentation . . . . .	92
17.35 EmbOrdinateDim_ Struct Reference . . . . .	93
17.35.1 Member Data Documentation . . . . .	93
17.36 EmbPath_ Struct Reference . . . . .	93
17.36.1 Member Data Documentation . . . . .	93
17.37 EmbPattern_ Struct Reference . . . . .	94
17.37.1 Member Data Documentation . . . . .	94
17.38 EmbPoint_ Struct Reference . . . . .	95
17.38.1 Member Data Documentation . . . . .	95
17.39 EmbRadiusDim_ Struct Reference . . . . .	96
17.39.1 Member Data Documentation . . . . .	96
17.40 EmbRay_ Struct Reference . . . . .	96
17.40.1 Member Data Documentation . . . . .	96
17.41 EmbRect_ Struct Reference . . . . .	96
17.41.1 Member Data Documentation . . . . .	97
17.42 EmbSatinOutline_ Struct Reference . . . . .	97
17.42.1 Member Data Documentation . . . . .	98
17.43 EmbSpline_ Struct Reference . . . . .	98
17.43.1 Member Data Documentation . . . . .	98
17.44 EmbStitch_ Struct Reference . . . . .	99
17.44.1 Member Data Documentation . . . . .	99
17.45 EmbTextMulti_ Struct Reference . . . . .	99
17.45.1 Member Data Documentation . . . . .	100
17.46 EmbTextSingle_ Struct Reference . . . . .	100
17.46.1 Member Data Documentation . . . . .	100
17.47 EmbThread_ Struct Reference . . . . .	100
17.47.1 Member Data Documentation . . . . .	101
17.48 EmbTime_ Struct Reference . . . . .	101
17.48.1 Member Data Documentation . . . . .	101
17.49 EmbVector_ Struct Reference . . . . .	102
17.49.1 Detailed Description . . . . .	102
17.49.2 Member Data Documentation . . . . .	102
17.50 Geometry Class Reference . . . . .	103
17.50.1 Detailed Description . . . . .	107
17.50.2 Member Enumeration Documentation . . . . .	108
17.50.3 Constructor & Destructor Documentation . . . . .	109
17.50.4 Member Function Documentation . . . . .	113
17.50.5 Member Data Documentation . . . . .	135

17.51 hoop_padding Struct Reference . . . . .	138
17.51.1 Member Data Documentation . . . . .	138
17.52 Huffman Struct Reference . . . . .	139
17.52.1 Member Data Documentation . . . . .	139
17.53 ImageWidget Class Reference . . . . .	140
17.53.1 Detailed Description . . . . .	140
17.53.2 Constructor & Destructor Documentation . . . . .	140
17.53.3 Member Function Documentation . . . . .	141
17.53.4 Member Data Documentation . . . . .	141
17.54 LayerManager Class Reference . . . . .	142
17.54.1 Detailed Description . . . . .	142
17.54.2 Constructor & Destructor Documentation . . . . .	142
17.54.3 Member Function Documentation . . . . .	142
17.54.4 Member Data Documentation . . . . .	143
17.55 LSYSTEM Struct Reference . . . . .	143
17.55.1 Member Data Documentation . . . . .	143
17.56 MainWindow Class Reference . . . . .	144
17.56.1 Detailed Description . . . . .	147
17.56.2 Constructor & Destructor Documentation . . . . .	148
17.56.3 Member Function Documentation . . . . .	148
17.56.4 Member Data Documentation . . . . .	161
17.57 MdiArea Class Reference . . . . .	162
17.57.1 Constructor & Destructor Documentation . . . . .	163
17.57.2 Member Function Documentation . . . . .	164
17.57.3 Member Data Documentation . . . . .	166
17.58 MdiWindow Class Reference . . . . .	167
17.58.1 Constructor & Destructor Documentation . . . . .	168
17.58.2 Member Function Documentation . . . . .	169
17.58.3 Member Data Documentation . . . . .	175
17.59 Node_ Struct Reference . . . . .	177
17.59.1 Member Data Documentation . . . . .	177
17.60 PreviewDialog Class Reference . . . . .	178
17.60.1 Constructor & Destructor Documentation . . . . .	178
17.60.2 Member Data Documentation . . . . .	178
17.61 PropertyEditor Class Reference . . . . .	179
17.61.1 Constructor & Destructor Documentation . . . . .	180
17.61.2 Member Function Documentation . . . . .	180
17.61.3 Member Data Documentation . . . . .	183
17.62 SaveObject Class Reference . . . . .	185
17.62.1 Constructor & Destructor Documentation . . . . .	186
17.62.2 Member Function Documentation . . . . .	187
17.62.3 Member Data Documentation . . . . .	195

17.63 SelectBox Class Reference . . . . .	195
17.63.1 Constructor & Destructor Documentation . . . . .	196
17.63.2 Member Function Documentation . . . . .	196
17.63.3 Member Data Documentation . . . . .	197
17.64 Settings_Dialog Class Reference . . . . .	198
17.64.1 Constructor & Destructor Documentation . . . . .	199
17.64.2 Member Function Documentation . . . . .	200
17.64.3 Member Data Documentation . . . . .	205
17.65 StatusBar Class Reference . . . . .	206
17.65.1 Detailed Description . . . . .	206
17.65.2 Constructor & Destructor Documentation . . . . .	206
17.65.3 Member Function Documentation . . . . .	206
17.65.4 Member Data Documentation . . . . .	207
17.66 StxThread_ Struct Reference . . . . .	207
17.66.1 Member Data Documentation . . . . .	207
17.67 SubDescriptor_ Struct Reference . . . . .	207
17.67.1 Member Data Documentation . . . . .	208
17.68 SvgAttribute_ Struct Reference . . . . .	208
17.68.1 Member Data Documentation . . . . .	208
17.69 thread_color_ Struct Reference . . . . .	208
17.69.1 Member Data Documentation . . . . .	209
17.70 ThredExtension_ Struct Reference . . . . .	209
17.70.1 Member Data Documentation . . . . .	209
17.71 ThredHeader_ Struct Reference . . . . .	210
17.71.1 Member Data Documentation . . . . .	210
17.72 UndoableCommand Class Reference . . . . .	210
17.72.1 Constructor & Destructor Documentation . . . . .	211
17.72.2 Member Function Documentation . . . . .	212
17.72.3 Member Data Documentation . . . . .	212
17.73 UndoEditor Class Reference . . . . .	213
17.73.1 Constructor & Destructor Documentation . . . . .	214
17.73.2 Member Function Documentation . . . . .	214
17.73.3 Member Data Documentation . . . . .	215
17.74 View Class Reference . . . . .	215
17.74.1 Constructor & Destructor Documentation . . . . .	218
17.74.2 Member Function Documentation . . . . .	218
17.74.3 Member Data Documentation . . . . .	225
17.75 VipHeader_ Struct Reference . . . . .	228
17.75.1 Member Data Documentation . . . . .	229
<b>18 File Documentation</b>	<b>230</b>
18.1 CODE_OF_CONDUCT.md File Reference . . . . .	230



18.2 embroidermodder2/cmdprompt.cpp File Reference	230
18.2.1 Detailed Description	230
18.3 embroidermodder2/em2_dev_script.py File Reference	230
18.4 embroidermodder2/embdetails-dialog.cpp File Reference	230
18.5 embroidermodder2/embroidermodder.cpp File Reference	230
18.5.1 Function Documentation	230
18.5.2 Variable Documentation	231
18.6 embroidermodder2/embroidermodder.h File Reference	231
18.6.1 Detailed Description	235
18.6.2 Macro Definition Documentation	235
18.6.3 Typedef Documentation	236
18.6.4 Enumeration Type Documentation	236
18.6.5 Function Documentation	237
18.6.6 Variable Documentation	246
18.7 embroidermodder.h	247
18.8 embroidermodder2/imagewidget.cpp File Reference	265
18.9 embroidermodder2/interface.cpp File Reference	265
18.9.1 Detailed Description	267
18.9.2 Function Documentation	267
18.10 embroidermodder2/layer-manager.cpp File Reference	272
18.10.1 Detailed Description	272
18.11 embroidermodder2/mainwindow-menus.cpp File Reference	273
18.11.1 Function Documentation	273
18.12 embroidermodder2/mainwindow-toolbars.cpp File Reference	273
18.13 embroidermodder2/mainwindow.cpp File Reference	273
18.13.1 Enumeration Type Documentation	278
18.13.2 Function Documentation	280
18.13.3 Variable Documentation	302
18.14 embroidermodder2/mdiarea.cpp File Reference	304
18.15 embroidermodder2/mdiwindow.cpp File Reference	304
18.15.1 Function Documentation	304
18.16 embroidermodder2/objects.cpp File Reference	305
18.16.1 Function Documentation	305
18.17 embroidermodder2/preview-dialog.cpp File Reference	306
18.18 embroidermodder2/property-editor.cpp File Reference	306
18.18.1 Function Documentation	306
18.18.2 Variable Documentation	306
18.19 embroidermodder2/README.md File Reference	307
18.20 embroidermodder2/selectbox.cpp File Reference	307
18.21 embroidermodder2/settings-dialog.cpp File Reference	308
18.21.1 Function Documentation	308
18.21.2 Variable Documentation	308

18.22 embroidermodder2/statusbar.cpp File Reference	310
18.23 embroidermodder2/undo-commands.cpp File Reference	310
18.24 embroidermodder2/undo-editor.cpp File Reference	310
18.24.1 Detailed Description	310
18.25 embroidermodder2/view.cpp File Reference	310
18.25.1 Detailed Description	310
18.25.2 Function Documentation	310
18.26 extern/libembroidery/src/array.c File Reference	311
18.26.1 Function Documentation	311
18.27 extern/libembroidery/src/compress.c File Reference	313
18.27.1 Detailed Description	313
18.27.2 Function Documentation	313
18.27.3 Variable Documentation	315
18.28 extern/libembroidery/src/embroidery.h File Reference	315
18.28.1 Macro Definition Documentation	322
18.28.2 Typedef Documentation	329
18.28.3 Function Documentation	332
18.28.4 Variable Documentation	347
18.29 embroidery.h	348
18.30 extern/libembroidery/src/embroidery_internal.h File Reference	355
18.30.1 Macro Definition Documentation	363
18.30.2 Typedef Documentation	371
18.30.3 Enumeration Type Documentation	372
18.30.4 Function Documentation	372
18.30.5 Variable Documentation	395
18.31 embroidery_internal.h	395
18.32 extern/libembroidery/src/encoding.c File Reference	402
18.32.1 Detailed Description	403
18.32.2 Function Documentation	403
18.33 extern/libembroidery/src/fill.c File Reference	405
18.33.1 Function Documentation	406
18.33.2 Variable Documentation	409
18.34 extern/libembroidery/src/formats.c File Reference	410
18.34.1 Function Documentation	411
18.34.2 Variable Documentation	413
18.35 extern/libembroidery/src/formats/format_100.c File Reference	414
18.35.1 Detailed Description	414
18.35.2 Function Documentation	414
18.36 extern/libembroidery/src/formats/format_10o.c File Reference	414
18.36.1 Detailed Description	414
18.36.2 Function Documentation	415
18.37 extern/libembroidery/src/formats/format_art.c File Reference	415

18.37.1 Detailed Description	415
18.37.2 Function Documentation	415
18.38 extern/libembroidery/src/formats/format_bmc.c File Reference	415
18.38.1 Detailed Description	416
18.38.2 Function Documentation	416
18.39 extern/libembroidery/src/formats/format_bro.c File Reference	416
18.39.1 Detailed Description	416
18.39.2 Function Documentation	416
18.40 extern/libembroidery/src/formats/format_cnd.c File Reference	416
18.40.1 Detailed Description	417
18.40.2 Function Documentation	417
18.41 extern/libembroidery/src/formats/format_col.c File Reference	417
18.41.1 Detailed Description	417
18.41.2 Function Documentation	418
18.42 extern/libembroidery/src/formats/format_csd.c File Reference	418
18.42.1 Detailed Description	418
18.42.2 Macro Definition Documentation	418
18.42.3 Function Documentation	418
18.42.4 Variable Documentation	419
18.43 extern/libembroidery/src/formats/format_csv.c File Reference	419
18.43.1 Detailed Description	420
18.43.2 Function Documentation	420
18.44 extern/libembroidery/src/formats/format_dat.c File Reference	420
18.44.1 Function Documentation	420
18.45 extern/libembroidery/src/formats/format_dem.c File Reference	421
18.45.1 Detailed Description	421
18.45.2 Function Documentation	421
18.46 extern/libembroidery/src/formats/format_dsb.c File Reference	421
18.46.1 Detailed Description	421
18.46.2 Function Documentation	422
18.47 extern/libembroidery/src/formats/format_dst.c File Reference	422
18.47.1 Detailed Description	422
18.47.2 Macro Definition Documentation	423
18.47.3 Function Documentation	423
18.48 extern/libembroidery/src/formats/format_dsz.c File Reference	424
18.48.1 Function Documentation	424
18.49 extern/libembroidery/src/formats/format_dxf.c File Reference	424
18.49.1 Function Documentation	425
18.50 extern/libembroidery/src/formats/format_edr.c File Reference	425
18.50.1 Function Documentation	425
18.51 extern/libembroidery/src/formats/format_emd.c File Reference	426
18.51.1 Detailed Description	426

18.51.2 Function Documentation . . . . .	426
18.52 extern/libembroidery/src/formats/format_exp.c File Reference . . . . .	426
18.52.1 Function Documentation . . . . .	426
18.53 extern/libembroidery/src/formats/format_exy.c File Reference . . . . .	427
18.53.1 Function Documentation . . . . .	427
18.54 extern/libembroidery/src/formats/format_ey.c File Reference . . . . .	427
18.54.1 Function Documentation . . . . .	427
18.55 extern/libembroidery/src/formats/format_fxy.c File Reference . . . . .	428
18.55.1 Function Documentation . . . . .	428
18.56 extern/libembroidery/src/formats/format_gc.c File Reference . . . . .	428
18.56.1 Function Documentation . . . . .	428
18.57 extern/libembroidery/src/formats/format_gnc.c File Reference . . . . .	429
18.57.1 Function Documentation . . . . .	429
18.58 extern/libembroidery/src/formats/format_gt.c File Reference . . . . .	429
18.58.1 Function Documentation . . . . .	429
18.59 extern/libembroidery/src/formats/format_hus.c File Reference . . . . .	430
18.59.1 Function Documentation . . . . .	430
18.60 extern/libembroidery/src/formats/format_inb.c File Reference . . . . .	431
18.60.1 Function Documentation . . . . .	431
18.61 extern/libembroidery/src/formats/format_inf.c File Reference . . . . .	431
18.61.1 Function Documentation . . . . .	431
18.62 extern/libembroidery/src/formats/format_jef.c File Reference . . . . .	432
18.62.1 Function Documentation . . . . .	432
18.63 extern/libembroidery/src/formats/format_ksm.c File Reference . . . . .	433
18.63.1 Function Documentation . . . . .	433
18.64 extern/libembroidery/src/formats/format_max.c File Reference . . . . .	433
18.64.1 Function Documentation . . . . .	434
18.64.2 Variable Documentation . . . . .	434
18.65 extern/libembroidery/src/formats/format_mit.c File Reference . . . . .	434
18.65.1 Function Documentation . . . . .	434
18.66 extern/libembroidery/src/formats/format_new.c File Reference . . . . .	435
18.66.1 Function Documentation . . . . .	435
18.67 extern/libembroidery/src/formats/format_ofm.c File Reference . . . . .	435
18.67.1 Function Documentation . . . . .	436
18.68 extern/libembroidery/src/formats/format_pcd.c File Reference . . . . .	436
18.68.1 Function Documentation . . . . .	437
18.69 extern/libembroidery/src/formats/format_pcm.c File Reference . . . . .	437
18.69.1 Function Documentation . . . . .	437
18.70 extern/libembroidery/src/formats/format_pcq.c File Reference . . . . .	437
18.70.1 Function Documentation . . . . .	438
18.71 extern/libembroidery/src/formats/format_pcs.c File Reference . . . . .	438
18.71.1 Function Documentation . . . . .	438

18.72 extern/libembroidery/src/formats/format_pec.c File Reference . . . . .	438
18.72.1 Function Documentation . . . . .	439
18.73 extern/libembroidery/src/formats/format_pel.c File Reference . . . . .	440
18.73.1 Function Documentation . . . . .	440
18.74 extern/libembroidery/src/formats/format_pem.c File Reference . . . . .	440
18.74.1 Function Documentation . . . . .	440
18.75 extern/libembroidery/src/formats/format_pes.c File Reference . . . . .	441
18.75.1 Function Documentation . . . . .	441
18.75.2 Variable Documentation . . . . .	443
18.76 extern/libembroidery/src/formats/format_phb.c File Reference . . . . .	443
18.76.1 Function Documentation . . . . .	443
18.77 extern/libembroidery/src/formats/format_phc.c File Reference . . . . .	444
18.77.1 Function Documentation . . . . .	444
18.78 extern/libembroidery/src/formats/format_plt.c File Reference . . . . .	444
18.78.1 Function Documentation . . . . .	444
18.79 extern/libembroidery/src/formats/format_rgb.c File Reference . . . . .	445
18.79.1 Function Documentation . . . . .	445
18.80 extern/libembroidery/src/formats/format_sew.c File Reference . . . . .	445
18.80.1 Function Documentation . . . . .	445
18.81 extern/libembroidery/src/formats/format_shv.c File Reference . . . . .	446
18.81.1 Function Documentation . . . . .	446
18.82 extern/libembroidery/src/formats/format_sst.c File Reference . . . . .	446
18.82.1 Function Documentation . . . . .	446
18.83 extern/libembroidery/src/formats/format_stx.c File Reference . . . . .	447
18.83.1 Function Documentation . . . . .	447
18.84 extern/libembroidery/src/formats/format_svg.c File Reference . . . . .	447
18.84.1 Function Documentation . . . . .	448
18.84.2 Variable Documentation . . . . .	448
18.85 extern/libembroidery/src/formats/format_t01.c File Reference . . . . .	449
18.85.1 Function Documentation . . . . .	449
18.86 extern/libembroidery/src/formats/format_t09.c File Reference . . . . .	449
18.86.1 Function Documentation . . . . .	449
18.87 extern/libembroidery/src/formats/format_tap.c File Reference . . . . .	450
18.87.1 Function Documentation . . . . .	450
18.88 extern/libembroidery/src/formats/format_thr.c File Reference . . . . .	450
18.88.1 Function Documentation . . . . .	450
18.89 extern/libembroidery/src/formats/format_txt.c File Reference . . . . .	451
18.89.1 Function Documentation . . . . .	451
18.90 extern/libembroidery/src/formats/format_u00.c File Reference . . . . .	451
18.90.1 Function Documentation . . . . .	451
18.91 extern/libembroidery/src/formats/format_u01.c File Reference . . . . .	452
18.91.1 Function Documentation . . . . .	452

18.92 extern/libembroidery/src/formats/format_vip.c File Reference	452
18.92.1 Function Documentation	453
18.92.2 Variable Documentation	453
18.93 extern/libembroidery/src/formats/format_vp3.c File Reference	454
18.93.1 Function Documentation	454
18.94 extern/libembroidery/src/formats/format_xxx.c File Reference	455
18.94.1 Function Documentation	455
18.95 extern/libembroidery/src/formats/format_zsk.c File Reference	456
18.95.1 Detailed Description	456
18.95.2 Function Documentation	456
18.96 extern/libembroidery/src/geometry.c File Reference	456
18.96.1 Function Documentation	457
18.97 extern/libembroidery/src/geometry/arc.c File Reference	458
18.97.1 Function Documentation	458
18.98 extern/libembroidery/src/geometry/circle.c File Reference	461
18.98.1 Function Documentation	462
18.99 extern/libembroidery/src/geometry/ellipse.c File Reference	462
18.99.1 Function Documentation	463
18.100 extern/libembroidery/src/geometry/functions.c File Reference	464
18.100.1 Function Documentation	464
18.101 extern/libembroidery/src/geometry/line.c File Reference	465
18.101.1 Function Documentation	465
18.102 extern/libembroidery/src/geometry/path.c File Reference	465
18.103 extern/libembroidery/src/geometry/polygon.c File Reference	465
18.104 extern/libembroidery/src/geometry/polyline.c File Reference	465
18.105 extern/libembroidery/src/geometry/rect.c File Reference	466
18.105.1 Function Documentation	466
18.106 extern/libembroidery/src/geometry/text.c File Reference	466
18.106.1 Function Documentation	466
18.107 extern/libembroidery/src/geometry/vector.c File Reference	468
18.107.1 Function Documentation	468
18.108 extern/libembroidery/src/image.c File Reference	470
18.108.1 Detailed Description	470
18.108.2 Function Documentation	470
18.109 extern/libembroidery/src/main.c File Reference	471
18.109.1 Macro Definition Documentation	473
18.109.2 Function Documentation	476
18.109.3 Variable Documentation	482
18.110 extern/libembroidery/src/pattern.c File Reference	482
18.110.1 Detailed Description	483
18.110.2 Function Documentation	483
18.111 extern/libembroidery/src/thread-color.c File Reference	487

18.111.1 Function Documentation . . . . .	488
18.111.2 Variable Documentation . . . . .	488
18.112 privacy_policy.md File Reference . . . . .	489
<b>Bibliography</b>	<b>490</b>
<b>Index</b>	<b>491</b>

## 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](http://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](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 Fritzing[8](8), upload netlists to embroiderbot when they can run simulations using the asm in `libembroidery`.
- Create a common assembly for data that is the same across chipsets `libembroidery\_data\_internal.s`.
- Make the defines part of `embroidery.h` all systems and the function list `c code only`. That way we can share some development between assembly and C versions.

## 7.1.10 Development

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

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

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

```
#!/bin/bash

rm -fr libembroidery-debug

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

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

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

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

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

## 7.2 Embroiderbot and Libembroidery on Embedded Systems

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

### 7.2.1 Compatible Boards

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

### 7.2.2 Arduino Considerations

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

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

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

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

### 7.2.3 Space

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

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

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

### 7.2.4 Tables

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

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

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

### 7.2.5 Current Pattern Memory Management

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

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

### 7.2.6 Special Notes

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

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

### 7.2.7 The Assembly Split

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

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

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

## 7.3 The Embroider Command Line Program

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

### 7.3.1 Embroider pipeline

Adjectives apply to every following noun so

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

Creates:

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

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

### 7.3.2 embroider CLI

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

## 8 GNU Free Documentation License

Version 1.3, 3 November 2008

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

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

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

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

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



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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## 9 Contributor Covenant Code of Conduct

### 9.1 Our Pledge

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

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

### 9.2 Our Standards

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

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

Examples of unacceptable behavior include:

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

### 9.3 Enforcement Responsibilities

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

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

### 9.4 Scope

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

## 9.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported to the community leaders responsible for enforcement at [embroidermodder@gmail.com](mailto: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](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](mailto:Embroidermodder@gmail.com)

## 11 Todo List

### Member [about\\_action](#) (String args)

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

### Member [bcf\\_directory](#)

possibly add a directory tree in the future.

### Member [bcf\\_file\\_header](#)

CLSID should be a separate type.

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

replace with embInt\_read



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

replace with embInt\_read

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

replace with embInt\_read

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

replace with embInt\_read

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

replace with embInt\_read

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

replace with embInt\_read

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

replace with embInt\_read

Member [copy\\_trim](#) (char const \*s)

decription

Member [day\\_vision\\_action](#) (String args)

Make day vision color settings.

Member [decode\\_t01\\_record](#) (unsigned char b[3], int \*x, int \*y, int \*flags)

remove the unused return argument.

Member [embArc\\_print](#) (EmbArc arc)

move to [arc.c](#)

Member [embGeometry\\_vulcanize](#) (EmbGeometry \*obj)

Review. This could be controlled by a simple flag.

Member [embPattern\\_correctForMaxStitchLength](#) (EmbPattern \*p, EmbReal maxStitchLength, EmbReal maxJumpLength)

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

Member [embPattern\\_stitchEllipse](#) (EmbPattern \*p, EmbEllipse ellipse, int thread\_index, int style)

finish stitchEllipse

Member [embPattern\\_stitchPath](#) (EmbPattern \*p, EmbPath path, int thread\_index, int style)

finish stitch path

Member [embPattern\\_stitchPolygon](#) (EmbPattern \*p, EmbPolygon polygon, int thread\_index, int style)

finish stitch polygon

Member [embPattern\\_stitchPolyline](#) (EmbPattern \*p, EmbPolyline polyline, int thread\_index, int style)

finish stitch polyline

Member [embVector\\_multiply](#) (EmbVector vector, EmbReal magnitude, EmbVector \*result)

make result return argument.

Member [embVector\\_normalize](#) (EmbVector vector, EmbVector \*result)

make result return argument.

File [format\\_art.c](#)

Find a source.

File [format\\_bmc.c](#)

Find a source.

File [format\\_cnd.c](#)

Find a source.

## Page **Formats**

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

Support for Singer FHE, CHE (Compucon) formats?

### Member **formatTable [numberOfFormats]**

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

### Member **fread\_int32\_be (FILE \*f)**

replace with embInt\_read

### Member **fread\_uint16 (FILE \*f)**

replace with embInt\_read

### Member **generate\_dragon\_curve (char \*state, int iterations)**

find citation for paper folding method

## Page **Geometry and Algorithms**

(Arduino) Fix thread-color files

(Arduino) Fix emb-outline files

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

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

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

Replace EmbArray functions with embPattern load functions.

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

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

(Arduino) Smoothieboard experiments

(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

### 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**

Serif font for printed docs.

Bibliography style to plainnat.

US letter paper version of printed docs.

### Member **MainWindow::createAllActions ()**

Set What's This Context Help to statusTip for now so there is some info there. Make custom what's 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 QStringList 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)`**

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

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

**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](#) 39

## 13 Hierarchical Index

### 13.1 Class Hierarchy

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

<a href="#">_bcf_directory</a>	40
<a href="#">_bcf_directory_entry</a>	41
<a href="#">_bcf_file</a>	42
<a href="#">_bcf_file_difat</a>	43
<a href="#">_bcf_file_fat</a>	44
<a href="#">_bcf_file_header</a>	44
<a href="#">_vp3Hoop</a>	47
<a href="#">Compress</a>	75
<a href="#">EmbAlignedDim_</a>	77
<a href="#">EmbAngularDim_</a>	77
<a href="#">EmbArc_</a>	78
<a href="#">EmbArcLengthDim_</a>	78
<a href="#">EmbArray_</a>	79
<a href="#">EmbBezier_</a>	80
<a href="#">EmbBlock_</a>	81

EmbCircle_	81
EmbColor_	81
EmbDiameterDim_	85
EmbEllipse_	85
EmbFormatList_	86
EmbGeometry_	87
EmbImage_	89
EmbInfiniteLine_	90
EmbLayer_	90
EmbLeaderDim_	91
EmbLine_	91
EmbLinearDim_	92
EmbOrdinateDim_	93
EmbPath_	93
EmbPattern_	94
EmbPoint_	95
EmbRadiusDim_	96
EmbRay_	96
EmbRect_	96
EmbSatinOutline_	97
EmbSpline_	98
EmbStitch_	99
EmbTextMulti_	99
EmbTextSingle_	100
EmbThread_	100
EmbTime_	101
EmbVector_	102
hoop_padding	138
Huffman	139
LSYSTEM	143
Node_	177
QApplication	

<b>Application</b>	<b>49</b>
QDialog	
<b>EmbDetailsDialog</b>	<b>82</b>
<b>LayerManager</b>	<b>142</b>
<b>Settings_Dialog</b>	<b>198</b>
QDockWidget	
<b>PropertyEditor</b>	<b>179</b>
<b>UndoEditor</b>	<b>213</b>
QFileDialog	
<b>PreviewDialog</b>	<b>178</b>
QGraphicsPathItem	
<b>Geometry</b>	<b>103</b>
QGraphicsView	
<b>View</b>	<b>215</b>
QLineEdit	
<b>CmdPromptInput</b>	<b>66</b>
QMainWindow	
<b>MainWindow</b>	<b>144</b>
QMdiArea	
<b>MdiArea</b>	<b>162</b>
QMdiSubWindow	
<b>MdiWindow</b>	<b>167</b>
QObject	
<b>SaveObject</b>	<b>185</b>
QRubberBand	
<b>SelectBox</b>	<b>195</b>
QSplitter	
<b>CmdPromptSplitter</b>	<b>74</b>
QSplitterHandle	
<b>CmdPromptHandle</b>	<b>60</b>
QStatusBar	
<b>StatusBar</b>	<b>206</b>
QTextBrowser	
<b>CmdPromptHistory</b>	<b>62</b>
QUndoCommand	
<b>UndoableCommand</b>	<b>210</b>
QWidget	
<b>CmdPrompt</b>	<b>51</b>
<b>ImageWidget</b>	<b>140</b>
<b>StxThread_</b>	<b>207</b>

SubDescriptor_	207
SvgAttribute_	208
thread_color_	208
ThredExtension_	209
ThredHeader_	210
VipHeader_	228

## 14 Class Index

### 14.1 Class List

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

<a href="#">_bcf_directory</a>	40
<a href="#">_bcf_directory_entry</a>	41
<a href="#">_bcf_file</a>	42
<a href="#">_bcf_file_difat</a>	43
<a href="#">_bcf_file_fat</a>	44
<a href="#">_bcf_file_header</a>	44
<a href="#">_vp3Hoop</a>	47
<a href="#">Application</a>	49
<a href="#">CmdPrompt</a> 51	
<a href="#">CmdPromptHandle</a> 60	
<a href="#">CmdPromptHistory</a> The Command Prompt History class	62
<a href="#">CmdPromptInput</a>	66
<a href="#">CmdPromptSplitter</a> 74	
<a href="#">Compress</a>	75
<a href="#">EmbAlignedDim_</a>	77
<a href="#">EmbAngularDim_</a>	77
<a href="#">EmbArc_</a> Absolute position (not relative)	78
<a href="#">EmbArcLengthDim_</a>	78

EmbArray_	79
EmbBezier_	80
EmbBlock_	81
EmbCircle_	81
EmbColor_	81
EmbDetailsDialog 82	
EmbDiameterDim_	85
EmbEllipse_	85
EmbFormatList_	86
EmbGeometry_	87
EmbImage_	89
EmbInfiniteLine_	90
EmbLayer_	90
EmbLeaderDim_	91
EmbLine_	91
EmbLinearDim_	92
EmbOrdinateDim_	93
EmbPath_	93
EmbPattern_	94
EmbPoint_	95
EmbRadiusDim_	96
EmbRay_	96
EmbRect_	96
EmbSatinOutline_	97
EmbSpline_	98
EmbStitch_	99
EmbTextMulti_	99
EmbTextSingle_	100
EmbThread_	100
EmbTime_	101
EmbVector_	102



<a href="#">Geometry</a>	
The <a href="#">Geometry</a> class	103
<a href="#">hoop_padding</a>	138
<a href="#">Huffman</a>	139
<a href="#">ImageWidget</a>	
140	
<a href="#">LayerManager</a>	
142	
<a href="#">LSYSTEM</a>	143
<a href="#">MainWindow</a>	
The <a href="#">MainWindow</a> class	144
<a href="#">MdiArea</a>	162
<a href="#">MdiWindow</a>	167
<a href="#">Node_</a>	177
<a href="#">PreviewDialog</a>	178
<a href="#">PropertyEditor</a>	179
<a href="#">SaveObject</a>	185
<a href="#">SelectBox</a>	195
<a href="#">Settings_Dialog</a>	198
<a href="#">StatusBar</a>	206
<a href="#">StxThread_</a>	207
<a href="#">SubDescriptor_</a>	207
<a href="#">SvgAttribute_</a>	208
<a href="#">thread_color_</a>	208
<a href="#">ThredExtension_</a>	209
<a href="#">ThredHeader_</a>	210
<a href="#">UndoableCommand</a>	210
<a href="#">UndoEditor</a>	213
<a href="#">View</a>	215
<a href="#">VipHeader_</a>	228

## 15 File Index

### 15.1 File List

Here is a list of all files with brief descriptions:

embroidermodder2/cmdprompt.cpp	230
embroidermodder2/em2_dev_script.py	230
embroidermodder2/embdetails-dialog.cpp	230
embroidermodder2/embroidermodder.cpp	230
embroidermodder2/embroidermodder.h	231
embroidermodder2/imagewidget.cpp	265
embroidermodder2/interface.cpp	265
embroidermodder2/layer-manager.cpp	272
embroidermodder2/mainwindow-menus.cpp	273
embroidermodder2/mainwindow-toolbars.cpp	273
embroidermodder2/mainwindow.cpp	273
embroidermodder2/mdiarea.cpp	304
embroidermodder2/mdiwindow.cpp	304
embroidermodder2/objects.cpp	305
embroidermodder2/preview-dialog.cpp	306
embroidermodder2/property-editor.cpp	306
embroidermodder2/selectbox.cpp	307
embroidermodder2/settings-dialog.cpp	308
embroidermodder2/statusbar.cpp	310
embroidermodder2/undo-commands.cpp	310
embroidermodder2/undo-editor.cpp	310
embroidermodder2/view.cpp	310
extern/libembroidery/src/array.c	311
extern/libembroidery/src/compress.c	313
extern/libembroidery/src/embroidery.h	315
extern/libembroidery/src/embroidery_internal.h	355
extern/libembroidery/src/encoding.c	402
extern/libembroidery/src/fill.c	405
extern/libembroidery/src/formats.c	410
extern/libembroidery/src/geometry.c	456
extern/libembroidery/src/image.c	470
extern/libembroidery/src/main.c	471

extern/libembroidery/src/ <a href="#">pattern.c</a>	482
extern/libembroidery/src/ <a href="#">thread-color.c</a>	487
extern/libembroidery/src/formats/ <a href="#">format_100.c</a>	414
extern/libembroidery/src/formats/ <a href="#">format_10o.c</a>	414
extern/libembroidery/src/formats/ <a href="#">format_art.c</a>	415
extern/libembroidery/src/formats/ <a href="#">format_bmc.c</a>	415
extern/libembroidery/src/formats/ <a href="#">format_bro.c</a>	416
extern/libembroidery/src/formats/ <a href="#">format_cnd.c</a>	416
extern/libembroidery/src/formats/ <a href="#">format_col.c</a>	417
extern/libembroidery/src/formats/ <a href="#">format_csd.c</a>	418
extern/libembroidery/src/formats/ <a href="#">format_csv.c</a>	419
extern/libembroidery/src/formats/ <a href="#">format_dat.c</a>	420
extern/libembroidery/src/formats/ <a href="#">format_dem.c</a>	421
extern/libembroidery/src/formats/ <a href="#">format_dsb.c</a>	421
extern/libembroidery/src/formats/ <a href="#">format_dst.c</a>	422
extern/libembroidery/src/formats/ <a href="#">format_dsz.c</a>	424
extern/libembroidery/src/formats/ <a href="#">format_dxf.c</a>	424
extern/libembroidery/src/formats/ <a href="#">format_edr.c</a>	425
extern/libembroidery/src/formats/ <a href="#">format_emd.c</a>	426
extern/libembroidery/src/formats/ <a href="#">format_exp.c</a>	426
extern/libembroidery/src/formats/ <a href="#">format_exy.c</a>	427
extern/libembroidery/src/formats/ <a href="#">format_eyc.c</a>	427
extern/libembroidery/src/formats/ <a href="#">format_eyc.c</a>	427
extern/libembroidery/src/formats/ <a href="#">format_fxy.c</a>	428
extern/libembroidery/src/formats/ <a href="#">format_gc.c</a>	428
extern/libembroidery/src/formats/ <a href="#">format_gnc.c</a>	429
extern/libembroidery/src/formats/ <a href="#">format_gt.c</a>	429
extern/libembroidery/src/formats/ <a href="#">format_hus.c</a>	430
extern/libembroidery/src/formats/ <a href="#">format_inb.c</a>	431
extern/libembroidery/src/formats/ <a href="#">format_inf.c</a>	431
extern/libembroidery/src/formats/ <a href="#">format_jef.c</a>	432
extern/libembroidery/src/formats/ <a href="#">format_ksm.c</a>	433
extern/libembroidery/src/formats/ <a href="#">format_max.c</a>	433

extern/libembroidery/src/formats/ <a href="#">format_mit.c</a>	434
extern/libembroidery/src/formats/ <a href="#">format_new.c</a>	435
extern/libembroidery/src/formats/ <a href="#">format_ofm.c</a>	435
extern/libembroidery/src/formats/ <a href="#">format_pcd.c</a>	436
extern/libembroidery/src/formats/ <a href="#">format_pcm.c</a>	437
extern/libembroidery/src/formats/ <a href="#">format_pcq.c</a>	437
extern/libembroidery/src/formats/ <a href="#">format_pcs.c</a>	438
extern/libembroidery/src/formats/ <a href="#">format_pec.c</a>	438
extern/libembroidery/src/formats/ <a href="#">format_pel.c</a>	440
extern/libembroidery/src/formats/ <a href="#">format_pem.c</a>	440
extern/libembroidery/src/formats/ <a href="#">format_pes.c</a>	441
extern/libembroidery/src/formats/ <a href="#">format_phb.c</a>	443
extern/libembroidery/src/formats/ <a href="#">format_phc.c</a>	444
extern/libembroidery/src/formats/ <a href="#">format_plt.c</a>	444
extern/libembroidery/src/formats/ <a href="#">format_rgb.c</a>	445
extern/libembroidery/src/formats/ <a href="#">format_sew.c</a>	445
extern/libembroidery/src/formats/ <a href="#">format_shv.c</a>	446
extern/libembroidery/src/formats/ <a href="#">format_sst.c</a>	446
extern/libembroidery/src/formats/ <a href="#">format_stx.c</a>	447
extern/libembroidery/src/formats/ <a href="#">format_svg.c</a>	447
extern/libembroidery/src/formats/ <a href="#">format_t01.c</a>	449
extern/libembroidery/src/formats/ <a href="#">format_t09.c</a>	449
extern/libembroidery/src/formats/ <a href="#">format_tap.c</a>	450
extern/libembroidery/src/formats/ <a href="#">format_thr.c</a>	450
extern/libembroidery/src/formats/ <a href="#">format_txt.c</a>	451
extern/libembroidery/src/formats/ <a href="#">format_u00.c</a>	451
extern/libembroidery/src/formats/ <a href="#">format_u01.c</a>	452
extern/libembroidery/src/formats/ <a href="#">format_vip.c</a>	452
extern/libembroidery/src/formats/ <a href="#">format_vp3.c</a>	454
extern/libembroidery/src/formats/ <a href="#">format_xxx.c</a>	455
extern/libembroidery/src/formats/ <a href="#">format_zsk.c</a>	456
extern/libembroidery/src/geometry/ <a href="#">arc.c</a>	458

<a href="#">extern/libembroidery/src/geometry/circle.c</a>	461
<a href="#">extern/libembroidery/src/geometry/ellipse.c</a>	462
<a href="#">extern/libembroidery/src/geometry/functions.c</a>	464
<a href="#">extern/libembroidery/src/geometry/line.c</a>	465
<a href="#">extern/libembroidery/src/geometry/path.c</a>	465
<a href="#">extern/libembroidery/src/geometry/polygon.c</a>	465
<a href="#">extern/libembroidery/src/geometry/polyline.c</a>	465
<a href="#">extern/libembroidery/src/geometry/rect.c</a>	466
<a href="#">extern/libembroidery/src/geometry/text.c</a>	466
<a href="#">extern/libembroidery/src/geometry/vector.c</a>	468

## 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:

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

## 17.9 CmdPrompt Class Reference

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

### Public Slots

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

## Signals

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

## Public Member Functions

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

## Public Attributes

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

### 17.9.1 Detailed Description

### 17.9.2 Constructor & Destructor Documentation

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

[CmdPrompt::CmdPrompt.](#)

Parameters

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

**17.9.2.2 ~CmdPrompt()** [~CmdPrompt](#) ( )

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

### 17.9.3 Member Function Documentation

**17.9.3.1 alert** void alert (  
    QString *txt* ) [slot]

[CmdPrompt::alert.](#)

Parameters

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

**17.9.3.2 appendHistory** void appendHistory (  
    QString *txt* ) [slot]

[CmdPrompt::appendHistory.](#)

Parameters

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

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

**17.9.3.4 blink** void blink ( ) [slot]

[CmdPrompt::blink](#).

**17.9.3.5 copyPressed** void copyPressed ( ) [signal]

**17.9.3.6 cutPressed** void cutPressed ( ) [signal]

**17.9.3.7 deletePressed** void deletePressed ( ) [signal]

**17.9.3.8 downPressed** void downPressed ( ) [signal]

**17.9.3.9 escapePressed** void escapePressed ( ) [signal]

**17.9.3.10 F10Pressed** void F10Pressed ( ) [signal]

**17.9.3.11 F11Pressed** void F11Pressed ( ) [signal]

**17.9.3.12 F12Pressed** void F12Pressed ( ) [signal]

**17.9.3.13 F1Pressed** void F1Pressed ( ) [signal]

**17.9.3.14 F2Pressed** void F2Pressed ( ) [signal]

**17.9.3.15 F3Pressed** void F3Pressed ( ) [signal]

**17.9.3.16 F4Pressed** void F4Pressed ( ) [signal]

**17.9.3.17 F5Pressed** void F5Pressed ( ) [signal]

**17.9.3.18 F6Pressed** void F6Pressed ( ) [signal]

**17.9.3.19 F7Pressed** void F7Pressed ( ) [signal]

**17.9.3.20 F8Pressed** void F8Pressed ( ) [signal]

**17.9.3.21 F9Pressed** void F9Pressed ( ) [signal]

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

[CmdPrompt::floatingChanged.](#)

Parameters

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

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

**17.9.3.24 pastePressed** void pastePressed ( ) [signal]

**17.9.3.25 redoPressed** void redoPressed ( ) [signal]

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

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

[CmdPrompt::saveHistory.](#)

Parameters

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

**17.9.3.28 selectAllPressed** void selectAllPressed ( ) [signal]

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

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

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

[CmdPrompt::setPrefix.](#)

## Parameters

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

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

[CmdPrompt::setPromptBackgroundColor.](#)

## Parameters

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

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

[CmdPrompt::setPromptFontFamily.](#)

## Parameters

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

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

[CmdPrompt::setPromptFontSize.](#)

## Parameters

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

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

[CmdPrompt::setPromptFontStyle.](#)

## Parameters

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

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

[CmdPrompt::setPromptTextColor.](#)

Parameters

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

**17.9.3.37 shiftPressed** `void shiftPressed ( ) [signal]`

**17.9.3.38 shiftReleased** `void shiftReleased ( ) [signal]`

**17.9.3.39 showSettings** `void showSettings ( ) [signal]`

**17.9.3.40 startBlinking** `void startBlinking ( ) [slot]`

[CmdPrompt::startBlinking.](#)

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

**17.9.3.42 stopBlinking** `void stopBlinking ( ) [slot]`

[CmdPrompt::stopBlinking.](#)

**17.9.3.43 tabPressed** `void tabPressed ( ) [signal]`



**17.9.3.44 undoPressed** void undoPressed ( ) [signal]

**17.9.3.45 updateStyle()** void updateStyle ( )

[CmdPrompt::updateStyle](#).

**17.9.3.46 upPressed** void upPressed ( ) [signal]

## 17.9.4 Member Data Documentation

**17.9.4.1 blinkState** bool blinkState

**17.9.4.2 blinkTimer** QTimer\* blinkTimer

**17.9.4.3 promptDivider** QFrame\* promptDivider

**17.9.4.4 promptHistory** [CmdPromptHistory](#)\* promptHistory

**17.9.4.5 promptInput** [CmdPromptInput](#)\* promptInput

**17.9.4.6 promptSplitter** [CmdPromptSplitter](#)\* promptSplitter

**17.9.4.7 promptVBoxLayout** QVBoxLayout\* promptVBoxLayout

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

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

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

### 17.10 CmdPromptHandle Class Reference

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

#### Signals

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

#### Public Member Functions

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

#### Public Attributes

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

#### Protected Member Functions

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

#### 17.10.1 Detailed Description

#### 17.10.2 Constructor & Destructor Documentation

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

[CmdPromptHandle::CmdPromptHandle.](#)

## Parameters

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

**17.10.2.2** [~CmdPromptHandle\(\)](#) [~CmdPromptHandle \( \)](#)

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

### 17.10.3 Member Function Documentation

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

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

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

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

[CmdPromptHandle::mouseMoveEvent.](#)

## Parameters

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

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

[CmdPromptHandle::mousePressEvent.](#)

**Parameters**

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

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

[CmdPromptHandle::mouseReleaseEvent.](#)

**Parameters**

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

**17.10.4 Member Data Documentation**

**17.10.4.1 moveY** `int moveY`

**17.10.4.2 pressY** `int pressY`

**17.10.4.3 releaseY** `int releaseY`

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

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

**17.11 CmdPromptHistory Class Reference**

The Command Prompt History class.

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

## Public Slots

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

## Signals

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

## Public Member Functions

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

## Public Attributes

- int [tmpHeight](#)

## Protected Member Functions

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

### 17.11.1 Detailed Description

The Command Prompt History class.

### 17.11.2 Constructor & Destructor Documentation

#### 17.11.2.1 CmdPromptHistory()

```
CmdPromptHistory (  
    QWidget * parent = 0 )
```

[CmdPromptHistory::CmdPromptHistory.](#)

**Parameters**

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

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

[CmdPromptHistory::~CmdPromptHistory.](#)

**17.11.3 Member Function Documentation****17.11.3.1** **appendHistory** void appendHistory (  
    QString *txt*,  
    int *prefixLength* ) [slot]

[CmdPromptHistory::appendHistory.](#)

**Parameters**

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

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

[CmdPromptHistory::applyFormatting.](#)

**Parameters**

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

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

[CmdPromptHistory::contextMenuEvent.](#)

## Parameters

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

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

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

[CmdPromptHistory::resizeHistory.](#)

## Parameters

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

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

[CmdPromptHistory::startResizeHistory.](#)

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

[CmdPromptHistory::stopResizeHistory.](#)

## 17.11.4 Member Data Documentation

**17.11.4.1 tmpHeight** int tmpHeight

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

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

## 17.12 CmdPromptInput Class Reference

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

### Public Slots

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

### Signals

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



## Public Member Functions

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

## Public Attributes

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

## Protected Member Functions

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

## Private Slots

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

### 17.12.1 Constructor & Destructor Documentation

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

[CmdPromptInput::CmdPromptInput.](#)

**Parameters**

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

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

**17.12.2 Member Function Documentation**

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

**17.12.2.2** **applyFormatting()** void applyFormatting ( )

[CmdPromptInput::applyFormatting](#).

**17.12.2.3** **changeFormatting()** void changeFormatting (   
    std::vector< QTextLayout::FormatRange > *formats* )

[CmdPromptInput::changeFormatting](#).

**Parameters**

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

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

[CmdPromptInput::checkChangedText](#).

**Parameters**

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

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

[CmdPromptInput::checkCursorPosition.](#)

Parameters

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

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

[CmdPromptInput::checkEditedText.](#)

Parameters

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

**17.12.2.7 checkSelection** `void checkSelection ( ) [slot]`

[CmdPromptInput::checkSelection.](#)

**17.12.2.8 clearFormatting()** `void clearFormatting ( )`

[CmdPromptInput::clearFormatting.](#)

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

[CmdPromptInput::contextMenuEvent.](#)

Parameters

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

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

[CmdPromptInput::copyClip.](#)

**17.12.2.11 copyPressed** `void copyPressed ( ) [signal]`

**17.12.2.12 cutPressed** `void cutPressed ( ) [signal]`

**17.12.2.13 deletePressed** `void deletePressed ( ) [signal]`

**17.12.2.14 downPressed** `void downPressed ( ) [signal]`

**17.12.2.15 endCommand** `void endCommand ( ) [slot]`

[CmdPromptInput::endCommand.](#)

**17.12.2.16 escapePressed** `void escapePressed ( ) [signal]`

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

[CmdPromptInput::eventFilter.](#)

#### Parameters

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

#### Returns

**17.12.2.18 F10Pressed** void F10Pressed ( ) [signal]

**17.12.2.19 F11Pressed** void F11Pressed ( ) [signal]

**17.12.2.20 F12Pressed** void F12Pressed ( ) [signal]

**17.12.2.21 F1Pressed** void F1Pressed ( ) [signal]

**17.12.2.22 F2Pressed** void F2Pressed ( ) [signal]

**17.12.2.23 F3Pressed** void F3Pressed ( ) [signal]

**17.12.2.24 F4Pressed** void F4Pressed ( ) [signal]

**17.12.2.25 F5Pressed** void F5Pressed ( ) [signal]

**17.12.2.26 F6Pressed** void F6Pressed ( ) [signal]

**17.12.2.27 F7Pressed** void F7Pressed ( ) [signal]

**17.12.2.28 F8Pressed** void F8Pressed ( ) [signal]

**17.12.2.29 F9Pressed** `void F9Pressed ( ) [signal]`

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

[CmdPromptInput::pasteClip.](#)

**17.12.2.31 pastePressed** `void pastePressed ( ) [signal]`

**17.12.2.32 processInput** `void processInput (`  
    `void ) [slot]`

[CmdPromptInput::processInput.](#)

**17.12.2.33 redoPressed** `void redoPressed ( ) [signal]`

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

**17.12.2.35 selectAllPressed** `void selectAllPressed ( ) [signal]`

**17.12.2.36 shiftPressed** `void shiftPressed ( ) [signal]`

**17.12.2.37 shiftReleased** `void shiftReleased ( ) [signal]`

**17.12.2.38 showSettings** `void showSettings ( ) [signal]`

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

**17.12.2.40 stopBlinking** void stopBlinking ( ) [signal]

**17.12.2.41 tabPressed** void tabPressed ( ) [signal]

**17.12.2.42 undoPressed** void undoPressed ( ) [signal]

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

[CmdPromptInput::updateCurrentText.](#)

Parameters

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

**17.12.2.44 upPressed** void upPressed ( ) [signal]

## 17.12.3 Member Data Documentation

**17.12.3.1 cmdActive** bool cmdActive

**17.12.3.2 curCmd** QString curCmd

**17.12.3.3 curText** QString curText

**17.12.3.4 defaultPrefix** `QString defaultPrefix`

**17.12.3.5 isBlinking** `bool isBlinking`

**17.12.3.6 lastCmd** `QString lastCmd`

**17.12.3.7 prefix** `QString prefix`

**17.12.3.8 rapidFireEnabled** `bool rapidFireEnabled`

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

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

## 17.13 CmdPromptSplitter Class Reference

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

### Signals

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

### Public Member Functions

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

### Protected Member Functions

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

#### 17.13.1 Detailed Description

#### 17.13.2 Constructor & Destructor Documentation

**17.13.2.1 CmdPromptSplitter()** [CmdPromptSplitter](#) (  
    QWidget \* parent = 0 )  
[CmdPromptSplitter::CmdPromptSplitter.](#)



## Parameters

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

**17.13.2.2** `~CmdPromptSplitter()` `~CmdPromptSplitter ( )`

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

### 17.13.3 Member Function Documentation

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

[CmdPromptSplitter::createHandle.](#)

## Returns

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

**17.13.3.3** `pressResizeHistory` `void pressResizeHistory (`  
`int y )` [signal]

**17.13.3.4** `releaseResizeHistory` `void releaseResizeHistory (`  
`int y )` [signal]

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

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

## 17.14 Compress Struct Reference

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

## Public Attributes

- int [bit\\_position](#)
- char \* [input\\_data](#)
- int [input\\_length](#)
- int [bits\\_total](#)
- int [block\\_elements](#)
- [huffman](#) [character\\_length\\_huffman](#)
- [huffman](#) [character\\_huffman](#)
- [huffman](#) [distance\\_huffman](#)

### 17.14.1 Member Data Documentation

**17.14.1.1 [bit\\_position](#)**   `int bit_position`

**17.14.1.2 [bits\\_total](#)**   `int bits_total`

**17.14.1.3 [block\\_elements](#)**   `int block_elements`

**17.14.1.4 [character\\_huffman](#)**   `huffman character_huffman`

**17.14.1.5 [character\\_length\\_huffman](#)**   `huffman character_length_huffman`

**17.14.1.6 [distance\\_huffman](#)**   `huffman distance_huffman`

**17.14.1.7 [input\\_data](#)**   `char* input_data`

#### 17.14.1.8 input\_length `int input_length`

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

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

### 17.15 EmbAlignedDim\_ Struct Reference

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

#### Public Attributes

- [EmbVector position](#)

#### 17.15.1 Member Data Documentation

##### 17.15.1.1 position `EmbVector position`

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

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

### 17.16 EmbAngularDim\_ Struct Reference

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

#### Public Attributes

- [EmbVector position](#)

#### 17.16.1 Member Data Documentation

##### 17.16.1.1 position `EmbVector position`

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

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

## 17.17 EmbArc\_Struct Reference

absolute position (not relative)

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

### Public Attributes

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

### 17.17.1 Detailed Description

absolute position (not relative)

### 17.17.2 Member Data Documentation

**17.17.2.1** `end` [EmbVector](#) `end`

**17.17.2.2** `mid` [EmbVector](#) `mid`

**17.17.2.3** `start` [EmbVector](#) `start`

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

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

## 17.18 EmbArcLengthDim\_Struct Reference

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

### Public Attributes

- [EmbVector position](#)

### 17.18.1 Member Data Documentation

#### 17.18.1.1 position [EmbVector](#) position

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

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

## 17.19 EmbArray\_ Struct Reference

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

### Public Attributes

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

### 17.19.1 Member Data Documentation

#### 17.19.1.1 count [int](#) count

#### 17.19.1.2 geometry [EmbGeometry](#)\* geometry

#### 17.19.1.3 length [int](#) length

#### 17.19.1.4 stitch [EmbStitch](#)\* stitch

**17.19.1.5 thread** `EmbThread* thread`

**17.19.1.6 type** `int type`

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

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

## 17.20 EmbBezier\_ Struct Reference

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

### Public Attributes

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

### 17.20.1 Member Data Documentation

**17.20.1.1 control1** `EmbVector control1`

**17.20.1.2 control2** `EmbVector control2`

**17.20.1.3 end** `EmbVector end`

**17.20.1.4 start** `EmbVector start`

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

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

## 17.21 EmbBlock\_ Struct Reference

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

### Public Attributes

- [EmbVector](#) position

#### 17.21.1 Member Data Documentation

**17.21.1.1 position** [EmbVector](#) position

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

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

## 17.22 EmbCircle\_ Struct Reference

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

### Public Attributes

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

#### 17.22.1 Member Data Documentation

**17.22.1.1 center** [EmbVector](#) center

**17.22.1.2 radius** [EmbReal](#) radius

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

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

## 17.23 EmbColor\_ Struct Reference

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

### Public Attributes

- unsigned char [r](#)
- unsigned char [g](#)
- unsigned char [b](#)

#### 17.23.1 Detailed Description

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

#### 17.23.2 Member Data Documentation

**17.23.2.1** [b](#) unsigned char b

**17.23.2.2** [g](#) unsigned char g

**17.23.2.3** [r](#) unsigned char r

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

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

### 17.24 EmbDetailsDialog Class Reference

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

#### Public Member Functions

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



**Public Attributes**

- QWidget \* [mainWidget](#)
- QDialogButtonBox \* [buttonBox](#)
- uint32\_t [stitchesTotal](#)
- uint32\_t [stitchesReal](#)
- uint32\_t [stitchesJump](#)
- uint32\_t [stitchesTrim](#)
- uint32\_t [colorTotal](#)
- uint32\_t [colorChanges](#)
- QRectF [boundingRect](#)

**17.24.1 Detailed Description****17.24.2 Constructor & Destructor Documentation**

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

[EmbDetailsDialog::EmbDetailsDialog.](#)

**Parameters**

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

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

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

**17.24.3 Member Function Documentation**

**17.24.3.1 createHistogram()** [QWidget \\* createHistogram](#) ( )

**17.24.3.2 createMainWidget()** [QWidget \\* createMainWidget](#) ( )

[EmbDetailsDialog::createMainWidget.](#)

**Returns**

**17.24.3.3** `getInfo()` `void getInfo ( )`

[EmbDetailsDialog::getInfo.](#)

## 17.24.4 Member Data Documentation

**17.24.4.1** `boundingRect` `QRectF boundingRect`

**17.24.4.2** `buttonBox` `QDialogButtonBox* buttonBox`

**17.24.4.3** `colorChanges` `uint32_t colorChanges`

**17.24.4.4** `colorTotal` `uint32_t colorTotal`

**17.24.4.5** `mainWidget` `QWidget* mainWidget`

**17.24.4.6** `stitchesJump` `uint32_t stitchesJump`

**17.24.4.7** `stitchesReal` `uint32_t stitchesReal`

**17.24.4.8** `stitchesTotal` `uint32_t stitchesTotal`

**17.24.4.9** `stitchesTrim` `uint32_t stitchesTrim`

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

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

## 17.25 EmbDiameterDim\_ Struct Reference

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

### Public Attributes

- [EmbVector](#) position

### 17.25.1 Member Data Documentation

#### 17.25.1.1 position [EmbVector](#) position

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

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

## 17.26 EmbEllipse\_ Struct Reference

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

### Public Attributes

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

### 17.26.1 Member Data Documentation

#### 17.26.1.1 center [EmbVector](#) center

#### 17.26.1.2 radius [EmbVector](#) radius

#### 17.26.1.3 rotation [EmbReal](#) rotation

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

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

## 17.27 EmbFormatList\_ Struct Reference

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

### Public Attributes

- char [extension](#) [2+EMBFORMAT\_MAXEXT]
- char [description](#) [EMBFORMAT\_MAXDESC]
- char [reader\\_state](#)
- char [writer\\_state](#)
- int [type](#)
- int [color\\_only](#)
- int [check\\_for\\_color\\_file](#)
- int [write\\_external\\_color\\_file](#)

### 17.27.1 Member Data Documentation

**17.27.1.1 [check\\_for\\_color\\_file](#)**   int [check\\_for\\_color\\_file](#)

**17.27.1.2 [color\\_only](#)**   int [color\\_only](#)

**17.27.1.3 [description](#)**   char [description](#)[EMBFORMAT\_MAXDESC]

**17.27.1.4 [extension](#)**   char [extension](#)[2+EMBFORMAT\_MAXEXT]

**17.27.1.5 [reader\\_state](#)**   char [reader\\_state](#)

**17.27.1.6 [type](#)**   int [type](#)

**17.27.1.7 [write\\_external\\_color\\_file](#)**   int [write\\_external\\_color\\_file](#)

### 17.27.1.8 writer\_state `char writer_state`

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

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

## 17.28 EmbGeometry\_ Struct Reference

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

### Public Attributes

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

### 17.28.1 Member Data Documentation

#### 17.28.1.1 arc `EmbArc` `arc`

#### 17.28.1.2 circle `EmbCircle` `circle`

#### 17.28.1.3 color `EmbColor` `color`

**17.28.1.4 ellipse** `EmbEllipse` ellipse

**17.28.1.5 flag** `int` flag

**17.28.1.6 line** `EmbLine` line

**17.28.1.7 lineType** `int` lineType

**17.28.1.8** `union { ... }` object

**17.28.1.9 path** `EmbPath` path

**17.28.1.10 point** `EmbPoint` point

**17.28.1.11 polygon** `EmbPolygon` polygon

**17.28.1.12 polyline** `EmbPolyline` polyline

**17.28.1.13 rect** `EmbRect` rect

**17.28.1.14 spline** `EmbSpline` spline

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

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

**17.28.1.17 type** int type

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

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

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

## 17.29 EmblImage\_ Struct Reference

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

### Public Attributes

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

### 17.29.1 Member Data Documentation

**17.29.1.1 data** unsigned char\* data

**17.29.1.2 dimensions** [EmbVector](#) dimensions

**17.29.1.3 height** `int height`

**17.29.1.4 name** `char name[200]`

**17.29.1.5 path** `char path[200]`

**17.29.1.6 position** `EmbVector position`

**17.29.1.7 width** `int width`

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

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

## 17.30 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 &amp; 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,`  
`QRgb 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_,`  
`QRgb rgb,`  
`Qt::PenStyle lineType,`  
`QGraphicsItem * parent )`

Construct a new `Geometry::Geometry` object.

#### Parameters

<i>line</i>	
<i>Type_</i>	

**Parameters**

<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 QPointF & rubLine,   
 QPainter * painter = 0,   
 const char * colorFromScene = 0 )`

[Geometry::drawRubberLine.](#)

Parameters

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

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

[Geometry::findIndex.](#)

Parameters

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

Returns

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

DimLeaderObject::gripEdit.

Parameters

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

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

**17.50.4.9 init\_arc()** `void init_arc (`  
`EmbArc arc,`  
`QRgb rgb,`  
`Qt::PenStyle lineType )`

[Geometry::init.](#)

Parameters

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



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

**17.50.4.10 init\_circle()** `void init_circle (`  
    [EmbCircle](#) *circle*,  
    QRgb *rgb*,  
    Qt::PenStyle *lineType* )

[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::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.

**17.50.4.15 `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

area	
------	--

**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

circumference	
---------------	--

**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:

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

## 17.51 hoop\_padding Struct Reference

### Public Attributes

- `int` [left](#)
- `int` [right](#)
- `int` [top](#)
- `int` [bottom](#)

### 17.51.1 Member Data Documentation

**17.51.1.1 bottom** `int bottom`

**17.51.1.2 left** `int left`

**17.51.1.3 right** `int right`

**17.51.1.4 top** `int top`

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

- [extern/libembroidery/src/formats/format\\_jef.c](#)



## 17.52 Huffman Struct Reference

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

### Public Attributes

- int [default\\_value](#)
- int [lengths](#) [1000]
- int [nlengths](#)
- int [table](#) [1000]
- int [table\\_width](#)
- int [ntable](#)

### 17.52.1 Member Data Documentation

**17.52.1.1 [default\\_value](#)**   int default\_value

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

**17.52.1.3 [nlengths](#)**   int nlengths

**17.52.1.4 [ntable](#)**   int ntable

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

**17.52.1.6 [table\\_width](#)**   int table\_width

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

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

## 17.53 ImageWidget Class Reference

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

### Public Member Functions

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

### Public Attributes

- QImage [img](#)

### Protected Member Functions

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

#### 17.53.1 Detailed Description

#### 17.53.2 Constructor & Destructor Documentation

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

[ImageWidget::ImageWidget.](#)

##### Parameters

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

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

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

### 17.53.3 Member Function Documentation

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

[ImageWidget::load.](#)

#### Parameters

<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:

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

## 17.54 LayerManager Class Reference

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

### Public Member Functions

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

### Public Attributes

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

#### 17.54.1 Detailed Description

#### 17.54.2 Constructor & Destructor Documentation

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

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

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

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

#### 17.54.3 Member Function Documentation

**17.54.3.1 addLayer()** void addLayer (  
    QString name,  
    const bool visible,  
    const bool frozen,  
    const [EmbReal](#) zValue,  
    const QColor color,  
    QString lineType,  
    QString lineWeight,  
    const bool print )

[LayerManager::addLayer.](#)

**Parameters**

<i>name</i>	
<i>visible</i>	
<i>frozen</i>	
<i>zValue</i>	
<i>color</i>	
<i>lineType</i>	
<i>lineWeight</i>	
<i>print</i>	

**17.54.4 Member Data Documentation**

**17.54.4.1 layerModel** `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:

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

**17.55 LSYSTEM Struct Reference**

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

**Public Attributes**

- char [axiom](#)
- char \* [alphabet](#)
- char \* [constants](#)
- char \*\* [rules](#)

**17.55.1 Member Data Documentation**

**17.55.1.1 alphabet** `char* alphabet`

**17.55.1.2 axiom** `char axiom`

**17.55.1.3 constants** `char* constants`

**17.55.1.4 rules** `char** rules`

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

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

## 17.56 MainWindow Class Reference

The [MainWindow](#) class.

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

### Public Slots

- void [onCloseWindow](#) ()  
*MainWindow::onCloseWindow.*
- virtual void [onCloseMdiWin](#) (MdiWindow \*)  
*MainWindow::onCloseMdiWin.*
- void [recentMenuAboutToShow](#) ()  
*MainWindow::recentMenuAboutToShow.*
- void [onWindowActivated](#) (QMdiSubWindow \*w)  
*MainWindow::onWindowActivated.*
- void [windowMenuAboutToShow](#) ()  
*MainWindow::windowMenuAboutToShow.*
- void [windowMenuActivated](#) (bool checked)  
*MainWindow::windowMenuActivated.*
- void [updateAllViewScrollBars](#) (bool val)  
*MainWindow::updateAllViewScrollBars.*
- void [updateAllViewCrossHairColors](#) (QRgb color)  
*MainWindow::updateAllViewCrossHairColors.*
- void [updateAllViewBackgroundColors](#) (QRgb color)  
*MainWindow::updateAllViewBackgroundColors.*
- void [updateAllViewSelectBoxColors](#) (QRgb colorL, QRgb fillL, QRgb colorR, QRgb fillR, int alpha)  
*MainWindow::updateAllViewSelectBoxColors.*
- void [updateAllViewGridColors](#) (QRgb color)  
*MainWindow::updateAllViewGridColors.*

- void [updateAllViewRulerColors](#) (QRgb color)  
*MainWindow::updateAllViewRulerColors.*
- void [updatePickAddMode](#) (bool val)  
*MainWindow::updatePickAddMode.*
- void [pickAddModeToggled](#) ()  
*MainWindow::pickAddModeToggled.*
- void [settingsPrompt](#) ()  
*MainWindow::settingsPrompt.*
- void [stub\\_testing](#) ()  
*MainWindow::stub\_testing.*
- void [promptHistoryAppended](#) (QString txt)  
*MainWindow::promptHistoryAppended.*
- void [logPromptInput](#) (QString txt)  
*MainWindow::logPromptInput.*
- void [promptInputPrevious](#) ()  
*MainWindow::promptInputPrevious.*
- void [promptInputNext](#) ()
- void [about](#) (void)  
*about\_action*
- void [tipOfTheDay](#) (void)  
*MainWindow::tipOfTheDay.*
- void [newFile](#) ()  
*MainWindow::newFile.*
- void [openFile](#) (bool recent=false, [String](#) recentFile="")  
*MainWindow::openFile.*
- void [openFilesSelected](#) ([StringList](#) files)  
*MainWindow::openFilesSelected.*
- void [openrecentfile](#) ()  
*MainWindow::openrecentfile.*
- void [savefile](#) ()  
*MainWindow::savefile.*
- void [saveasfile](#) ()  
*MainWindow::saveasfile.*
- void [quit](#) ()  
*MainWindow::quit.*
- void [checkForUpdates](#) ()  
*MainWindow::checkForUpdates.*
- void [buttonTipOfTheDayClicked](#) (int)
- void [closeToolBar](#) (QAction \*)  
*MainWindow::closeToolBar.*
- void [floatingChangedToolBar](#) (bool)  
*MainWindow::floatingChangedToolBar.*
- void [toggleGrid](#) ()  
*MainWindow::toggleGrid.*
- void [toggleRuler](#) ()  
*MainWindow::toggleRuler.*
- void [toggleLwt](#) ()  
*MainWindow::toggleLwt.*
- void [iconResize](#) (int iconSize)
- void [layerSelectorIndexChanged](#) (int index)  
*MainWindow::layerSelectorIndexChanged.*

- void [colorSelectorIndexChanged](#) (int index)  
*MainWindow::colorSelectorIndexChanged.*
- void [linetypeSelectorIndexChanged](#) (int index)
- void [lineweightSelectorIndexChanged](#) (int index)  
*MainWindow::lineweightSelectorIndexChanged.*
- void [textFontSelectorCurrentFontChanged](#) (const QFont &font)  
*MainWindow::textFontSelectorCurrentFontChanged.*
- void [textSizeSelectorIndexChanged](#) (int index)  
*MainWindow::textSizeSelectorIndexChanged.*
- void [setTextFont](#) (QString str)  
*MainWindow::setTextFont.*
- void [setTextSize](#) (EmbReal num)  
*MainWindow::setTextSize.*
- QString [getCurrentLayer](#) ()  
*MainWindow::getCurrentLayer.*
- QColor [getCurrentColor](#) ()  
*MainWindow::getCurrentColor.*
- QString [getCurrentLineType](#) ()  
*MainWindow::getCurrentLineType.*
- QString [getCurrentLineWeight](#) ()  
*MainWindow::getCurrentLineWeight.*
- bool [isShiftPressed](#) ()
- void [setShiftPressed](#) ()
- void [setShiftReleased](#) ()
- void [deletePressed](#) ()  
*MainWindow::deletePressed.*
- void [escapePressed](#) ()  
*MainWindow::escapePressed.*

## Public Member Functions

- [MainWindow](#) ()  
*MainWindow::MainWindow.*
- [~MainWindow](#) ()  
*MainWindow::~~MainWindow.*
- [MdiWindow \\* activeMdiWindow](#) ()  
*MainWindow::activeMdiWindow.*
- [QUndoStack \\* activeUndoStack](#) ()  
*MainWindow::activeUndoStack.*
- void [setUndoCleanIcon](#) (bool opened)  
*MainWindow::setUndoCleanIcon.*
- virtual void [updateMenuToolbarStatusbar](#) ()  
*MainWindow::updateMenuToolbarStatusbar.*
- bool [isCommandActive](#) ()
- QString [activeCommand](#) ()
- QIcon [create\\_icon](#) (QString stub)  
*MainWindow::create\_icon.*
- void [create\\_toolbar](#) (String toolbar, String label, QStringList entries)  
*MainWindow::create\_toolbar.*
- QString [platformString](#) ()



## Public Attributes

- `std::vector< QGraphicsItem * >` [cutCopyObjectList](#)
- `QString` [formatFilterOpen](#)
- `QString` [formatFilterSave](#)

## Protected Member Functions

- virtual void [resizeEvent](#) (`QResizeEvent *`)  
*[MainWindow::resizeEvent](#).*
- void [closeEvent](#) (`QCloseEvent *event`)  
*[MainWindow::closeEvent](#).*
- `QAction *` [getFileSeparator](#) ()  
*[MainWindow::getFileSeparator](#).*
- void [loadFormats](#) ()  
*[MainWindow::loadFormats](#).*
- `QMdiSubWindow *` [findMdiWindow](#) (`String fileName`)  
*[MainWindow::findMdiWindow](#).*
- void [createAllActions](#) ()  
*[MainWindow::createAllActions](#).*
- void [createAllMenus](#) ()  
*[MainWindow::createAllMenus](#).*
- void [createAllToolbars](#) ()  
*[MainWindow::createAllToolbars](#).*

## Protected Attributes

- bool [shiftKeyPressedState](#)
- `QByteArray` [layoutState](#)
- int [numOfDocs](#)
- int [docIndex](#)
- `std::vector< MdiWindow * >` [listMdiWin](#)
- `QAction *` [myFileSeparator](#)
- `QComboBox *` [layerSelector](#)
- `QComboBox *` [colorSelector](#)
- `QComboBox *` [linetypeSelector](#)
- `QComboBox *` [lineweightSelector](#)
- `QFontComboBox *` [textFontSelector](#)
- `QComboBox *` [textSizeSelector](#)

## Private Slots

- void [hideUnimplemented](#) ()  
*[MainWindow::hideUnimplemented](#).*

### 17.56.1 Detailed Description

The [MainWindow](#) class.

## 17.56.2 Constructor & Destructor Documentation

### 17.56.2.1 MainWindow() `MainWindow ( )`

[MainWindow::MainWindow.](#)

### 17.56.2.2 ~MainWindow() `~MainWindow ( )`

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

## 17.56.3 Member Function Documentation

### 17.56.3.1 **about** `void about ( void ) [slot]`

`about_action`

Parameters

<i>args</i>	
-------------	--

Returns

### 17.56.3.2 **activeCommand()** `QString activeCommand ( ) [inline]`

### 17.56.3.3 **activeMdiWindow()** `MdiWindow * activeMdiWindow ( )`

[MainWindow::activeMdiWindow.](#)

Returns

**17.56.3.4 activeUndoStack()** `QUndoStack * activeUndoStack ( )`

[MainWindow::activeUndoStack.](#)

Returns

**17.56.3.5 buttonTipOfTheDayClicked** `void buttonTipOfTheDayClicked (   
int button ) [slot]`

**17.56.3.6 checkForUpdates** `void checkForUpdates ( ) [slot]`

[MainWindow::checkForUpdates.](#)

**17.56.3.7 closeEvent()** `void closeEvent (   
QCloseEvent * event ) [protected]`

[MainWindow::closeEvent.](#)

Parameters

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

**17.56.3.8 closeToolBar** `void closeToolBar (   
QAction * action ) [slot]`

[MainWindow::closeToolBar.](#)

Parameters

<i>action</i>	
---------------	--

**17.56.3.9 colorSelectorIndexChanged** `void colorSelectorIndexChanged (   
int index ) [slot]`

[MainWindow::colorSelectorIndexChanged.](#)

**Parameters**

<i>index</i>	
--------------	--

**17.56.3.10 create\_icon()** `QIcon create_icon (`  
`QString stub )`

[MainWindow::create\\_icon.](#)

**Parameters**

<i>stub</i>	
-------------	--

**Returns**

**17.56.3.11 create\_toolbar()** `void create_toolbar (`  
`String toolbar,`  
`String label,`  
`StringList entries )`

[MainWindow::create\\_toolbar.](#)

**Parameters**

<i>toolbar</i>	
<i>label</i>	
<i>entries</i>	

**17.56.3.12 createAllActions()** `void createAllActions ( ) [protected]`

[MainWindow::createAllActions.](#)

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

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

An alias is another entry in config that is also a StringList containing just the name of the command it aliases.

icon: The stub used for the icon and the basic command. command: tooltip: The label in the menus and the message that appears when you hover over an icon. statustip: The message that appears at the bottom of the . shortcut: The keyboard shortcut for this action.

**17.56.3.13 createAllMenus()** `void createAllMenus ( ) [protected]`

[MainWindow::createAllMenus.](#)

**17.56.3.14 createAllToolbars()** `void createAllToolbars ( ) [protected]`

[MainWindow::createAllToolbars.](#)

**17.56.3.15 deletePressed** `void deletePressed ( ) [slot]`

[MainWindow::deletePressed.](#)

**17.56.3.16 escapePressed** `void escapePressed ( ) [slot]`

[MainWindow::escapePressed.](#)

**17.56.3.17 findMdiWindow()** `QMdiSubWindow * findMdiWindow (   
 String fileName ) [protected]`

[MainWindow::findMdiWindow.](#)

Parameters

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

Returns

**17.56.3.18 floatingChangedToolBar** `void floatingChangedToolBar (   
 bool isFloating ) [slot]`

[MainWindow::floatingChangedToolBar.](#)

Parameters

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

**17.56.3.19 getCurrentColor** `QRgb getCurrentColor ( ) [slot]`

[MainWindow::getCurrentColor.](#)

Returns

**17.56.3.20 getCurrentLayer** `QString getCurrentLayer ( ) [slot]`

[MainWindow::getCurrentLayer.](#)

Returns

**17.56.3.21 getCurrentLineType** `QString getCurrentLineType ( ) [slot]`

[MainWindow::getCurrentLineType.](#)

Returns

**17.56.3.22 getCurrentLineWeight** `QString getCurrentLineWeight ( ) [slot]`

[MainWindow::getCurrentLineWeight.](#)

Returns

**17.56.3.23 getFileSeparator()** `QAction * getFileSeparator ( ) [protected]`

[MainWindow::getFileSeparator.](#)

Returns

**17.56.3.24 hideUnimplemented** void hideUnimplemented ( ) [private], [slot]

[MainWindow::hideUnimplemented.](#)

**17.56.3.25 iconResize** void iconResize (   
 int *iconSize* ) [slot]

**17.56.3.26 isCommandActive()** bool isCommandActive ( ) [inline]

**17.56.3.27 isShiftPressed** bool isShiftPressed ( ) [slot]

**17.56.3.28 layerSelectorIndexChanged** void layerSelectorIndexChanged (   
 int *index* ) [slot]

[MainWindow::layerSelectorIndexChanged.](#)

Parameters

<i>index</i>	
--------------	--

**17.56.3.29 linetypeSelectorIndexChanged** void linetypeSelectorIndexChanged (   
 int *index* ) [slot]

**17.56.3.30 linewidthSelectorIndexChanged** void linewidthSelectorIndexChanged (   
 int *index* ) [slot]

[MainWindow::linewidthSelectorIndexChanged.](#)

Parameters

<i>index</i>	
--------------	--

**17.56.3.31 loadFormats()** void loadFormats ( ) [protected]

[MainWindow::loadFormats.](#)

**17.56.3.32 logPromptInput** `void logPromptInput (`  
    `QString txt ) [slot]`

[MainWindow::logPromptInput.](#)

Parameters

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

**17.56.3.33 newFile** `void newFile ( ) [slot]`

[MainWindow::newFile.](#)

**17.56.3.34 onCloseMdiWin** `void onCloseMdiWin (`  
    `MdiWindow * theMdiWin ) [virtual], [slot]`

[MainWindow::onCloseMdiWin.](#)

Parameters

<i>theMdiWin</i>	
------------------	--

**17.56.3.35 onCloseWindow** `void onCloseWindow ( ) [slot]`

[MainWindow::onCloseWindow.](#)

**17.56.3.36 onWindowActivated** `void onWindowActivated (`  
    `QMdiSubWindow * w ) [slot]`

[MainWindow::onWindowActivated.](#)

Parameters

<i>w</i>	
----------	--



**17.56.3.37 openFile** void openFile (   
 bool *recent* = false,   
 String *recentFile* = "" ) [slot]

[MainWindow::openFile.](#)

Parameters

<i>recent</i>	
<i>recentFile</i>	

**17.56.3.38 openFilesSelected** void openFilesSelected (   
 QStringList *filesToOpen* ) [slot]

[MainWindow::openFilesSelected.](#)

Parameters

<i>filesToOpen</i>	
--------------------	--

**17.56.3.39 openrecentfile** void openrecentfile ( ) [slot]

[MainWindow::openrecentfile.](#)

**17.56.3.40 pickAddModeToggled** void pickAddModeToggled ( ) [slot]

[MainWindow::pickAddModeToggled.](#)

**17.56.3.41 platformString()** QString platformString ( )

**17.56.3.42 promptHistoryAppended** void promptHistoryAppended (   
 QString *txt* ) [slot]

[MainWindow::promptHistoryAppended.](#)

Parameters

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

**17.56.3.43 promptInputNext** `void promptInputNext ( ) [slot]`

**17.56.3.44 promptInputPrevious** `void promptInputPrevious ( ) [slot]`

[MainWindow::promptInputPrevious.](#)

**17.56.3.45 quit** `void quit ( ) [slot]`

[MainWindow::quit.](#)

**17.56.3.46 recentMenuAboutToShow** `void recentMenuAboutToShow ( ) [slot]`

[MainWindow::recentMenuAboutToShow.](#)

**17.56.3.47 resizeEvent()** `void resizeEvent (   
 QResizeEvent * e ) [protected], [virtual]`

[MainWindow::resizeEvent.](#)

Parameters

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

**17.56.3.48 saveasfile** `void saveasfile ( ) [slot]`

[MainWindow::saveasfile.](#)

**17.56.3.49 savefile** `void savefile ( ) [slot]`

[MainWindow::savefile.](#)

**17.56.3.50 setShiftPressed** void setShiftPressed ( ) [slot]

**17.56.3.51 setShiftReleased** void setShiftReleased ( ) [slot]

**17.56.3.52 setTextFont** void setTextFont (   
QString *str* ) [slot]

[MainWindow::setFont.](#)

Parameters

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

**17.56.3.53 setTextSize** void setTextSize (   
EmbReal *num* ) [slot]

[MainWindow::setTextSize.](#)

Parameters

<i>num</i>	
------------	--

**17.56.3.54 settingsPrompt** void settingsPrompt ( ) [slot]

[MainWindow::settingsPrompt.](#)

**17.56.3.55 setUndoCleanIcon()** void setUndoCleanIcon (   
bool *opened* )

[MainWindow::setUndoCleanIcon.](#)

Parameters

<i>opened</i>	
---------------	--

**17.56.3.56 stub\_testing** void stub\_testing ( ) [slot]

[MainWindow::stub\\_testing.](#)

**17.56.3.57 textFontSelectorCurrentFontChanged** `void textFontSelectorCurrentFontChanged ( const QFont & font ) [slot]`

[MainWindow::textFontSelectorCurrentFontChanged.](#)

Parameters

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

**17.56.3.58 textSizeSelectorIndexChanged** `void textSizeSelectorIndexChanged ( int index ) [slot]`

[MainWindow::textSizeSelectorIndexChanged.](#)

Parameters

<i>index</i>	
--------------	--

**17.56.3.59 tipOfTheDay** `void tipOfTheDay ( void ) [slot]`

[MainWindow::tipOfTheDay.](#)

**17.56.3.60 toggleGrid** `void toggleGrid ( ) [slot]`

[MainWindow::toggleGrid.](#)

**17.56.3.61 toggleLwt** `void toggleLwt ( ) [slot]`

[MainWindow::toggleLwt.](#)

**17.56.3.62 toggleRuler** `void toggleRuler ( ) [slot]`

[MainWindow::toggleRuler.](#)

**17.56.3.63 updateAllViewBackgroundColors** `void updateAllViewBackgroundColors ( QRgb color ) [slot]`

[MainWindow::updateAllViewBackgroundColors.](#)

## Parameters

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

**17.56.3.64 updateAllViewCrossHairColors** void updateAllViewCrossHairColors (   
 QRgb *color* ) [slot]

[MainWindow::updateAllViewCrossHairColors.](#)

## Parameters

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

**17.56.3.65 updateAllViewGridColors** void updateAllViewGridColors (   
 QRgb *color* ) [slot]

[MainWindow::updateAllViewGridColors.](#)

## Parameters

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

**17.56.3.66 updateAllViewRulerColors** void updateAllViewRulerColors (   
 QRgb *color* ) [slot]

[MainWindow::updateAllViewRulerColors.](#)

## Parameters

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

**17.56.3.67 updateAllViewScrollBars** void updateAllViewScrollBars (   
 bool *val* ) [slot]

[MainWindow::updateAllViewScrollBars.](#)

## Parameters

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

**17.56.3.68 updateAllViewSelectBoxColors** `void updateAllViewSelectBoxColors (`  
    `QRgb colorL,`  
    `QRgb fillL,`  
    `QRgb colorR,`  
    `QRgb fillR,`  
    `int alpha ) [slot]`

[MainWindow::updateAllViewSelectBoxColors.](#)

Parameters

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

**17.56.3.69 updateMenuToolbarStatusbar()** `void updateMenuToolbarStatusbar ( ) [virtual]`

[MainWindow::updateMenuToolbarStatusbar.](#)

**17.56.3.70 updatePickAddMode** `void updatePickAddMode (`  
    `bool val ) [slot]`

[MainWindow::updatePickAddMode.](#)

Parameters

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

**17.56.3.71 windowMenuAboutToShow** `void windowMenuAboutToShow ( ) [slot]`

[MainWindow::windowMenuAboutToShow.](#)

**17.56.3.72 windowMenuActivated** `void windowMenuActivated (`  
    `bool checked ) [slot]`

[MainWindow::windowMenuActivated.](#)

## Parameters

<i>checked</i>	<input type="checkbox"/>
----------------	--------------------------

#### 17.56.4 Member Data Documentation

**17.56.4.1 colorSelector** `QComboBox* colorSelector` [protected]

**17.56.4.2 cutCopyObjectList** `std::vector<QGraphicsItem*> cutCopyObjectList`

**17.56.4.3 docIndex** `int docIndex` [protected]

**17.56.4.4 formatFilterOpen** `QString formatFilterOpen`

**17.56.4.5 formatFilterSave** `QString formatFilterSave`

**17.56.4.6 layerSelector** `QComboBox* layerSelector` [protected]

**17.56.4.7 layoutState** `QByteArray layoutState` [protected]

**17.56.4.8 linetypeSelector** `QComboBox* linetypeSelector` [protected]

**17.56.4.9 lineweightSelector** `QComboBox* lineweightSelector` [protected]

**17.56.4.10 listMdiWin** `std::vector<MdiWindow*> listMdiWin` [protected]

**17.56.4.11 myFileSeparator** `QAction* myFileSeparator` [protected]

**17.56.4.12 numOfDocs** `int numOfDocs` [protected]

**17.56.4.13 shiftKeyPressedState** `bool shiftKeyPressedState` [protected]

**17.56.4.14 textFontSelector** `QFontComboBox* textFontSelector` [protected]

**17.56.4.15 textSizeSelector** `QComboBox* textSizeSelector` [protected]

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

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

## 17.57 MdiArea Class Reference

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

### Public Slots

- void [cascade](#) ()  
*MdiArea::cascade.*
- void [tile](#) ()  
*MdiArea::tile.*



## Public Member Functions

- void [zoomExtentsAllSubWindows](#) ()  
*MdiArea::zoomExtentsAllSubWindows.*
- void [forceRepaint](#) ()  
*MdiArea::forceRepaint.*
- [MdiArea](#) (QWidget \*parent=0)  
*MdiArea::MdiArea.*
- [~MdiArea](#) ()  
*MdiArea::~~MdiArea.*
- void [useBackgroundLogo](#) (bool use)  
*MdiArea::useBackgroundLogo.*
- void [useBackgroundTexture](#) (bool use)  
*MdiArea::useBackgroundTexture.*
- void [useBackgroundColor](#) (bool use)
- void [setBackgroundLogo](#) (QString fileName)  
*MdiArea::setBackgroundLogo.*
- void [setBackgroundTexture](#) (QString fileName)  
*MdiArea::setBackgroundTexture.*
- void [setBackgroundColor](#) (const QColor &color)  
*MdiArea::setBackgroundColor.*

## Public Attributes

- bool [useLogo](#)
- bool [useTexture](#)
- bool [useColor](#)
- QPixmap [bgLogo](#)
- QPixmap [bgTexture](#)
- QColor [bgColor](#)

## Protected Member Functions

- virtual void [mouseDoubleClickEvent](#) (QMouseEvent \*e)  
*MdiArea::mouseDoubleClickEvent.*
- virtual void [paintEvent](#) (QPaintEvent \*e)  
*MdiArea::paintEvent.*

### 17.57.1 Constructor & Destructor Documentation

**17.57.1.1 MdiArea()** [MdiArea](#) (  
    QWidget \* parent = 0 )

[MdiArea::MdiArea.](#)

## Parameters

<i>mw</i>	
<i>parent</i>	

**17.57.1.2** `~MdiArea()` `~MdiArea ( )`

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

## 17.57.2 Member Function Documentation

**17.57.2.1** `cascade` `void cascade ( )` [slot]

[MdiArea::cascade.](#)

**17.57.2.2** `forceRepaint()` `void forceRepaint ( )`

[MdiArea::forceRepaint.](#)

**17.57.2.3** `mouseDoubleClickEvent()` `void mouseDoubleClickEvent (`  
`QMouseEvent * e )` [protected], [virtual]

[MdiArea::mouseDoubleClickEvent.](#)

**17.57.2.4** `paintEvent()` `void paintEvent (`  
`QPaintEvent * e )` [protected], [virtual]

[MdiArea::paintEvent.](#)

**17.57.2.5** `setBackgroundColor()` `void setBackgroundColor (`  
`const QColor & color )`

[MdiArea::setBackgroundColor.](#)

## Parameters

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

**17.57.2.6 setBackgroundLogo()** `void setBackgroundLogo (   
          QString fileName )`

[MdiArea::setBackgroundLogo.](#)

## Parameters

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

**17.57.2.7 setBackgroundTexture()** `void setBackgroundTexture (   
          QString fileName )`

[MdiArea::setBackgroundTexture.](#)

## Parameters

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

**17.57.2.8 tile** `void tile ( ) [slot]`

[MdiArea::tile.](#)

**17.57.2.9 useBackgroundColor()** `void useBackgroundColor (   
          bool use )`

## Parameters

<i>use</i>	
------------	--

**17.57.2.10 useBackgroundLogo()** `void useBackgroundLogo (   
          bool use )`

[MdiArea::useBackgroundLogo.](#)

## Parameters

<i>use</i>	
------------	--

**17.57.2.11 useBackgroundTexture()** `void useBackgroundTexture (`  
    `bool use )`

[MdiArea::useBackgroundTexture.](#)

## Parameters

<i>use</i>	
------------	--

**17.57.2.12 zoomExtentsAllSubWindows()** `void zoomExtentsAllSubWindows ( )`

[MdiArea::zoomExtentsAllSubWindows.](#)

### 17.57.3 Member Data Documentation

**17.57.3.1 bgColor** `QColor bgColor`

**17.57.3.2 bgLogo** `QPixmap bgLogo`

**17.57.3.3 bgTexture** `QPixmap bgTexture`

**17.57.3.4 useColor** `bool useColor`

**17.57.3.5 useLogo** `bool useLogo`

**17.57.3.6 useTexture** `bool useTexture`

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

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

**17.58 MdiWindow Class Reference**

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

**Public Slots**

- void [closeEvent](#) (QCloseEvent \*e)  
*MdiWindow::closeEvent.*
- void [onWindowActivated](#) ()  
*MdiWindow::onWindowActivated.*
- void [currentLayerChanged](#) (QString layer)  
*MdiWindow::currentLayerChanged.*
- void [currentColorChanged](#) (const QColor &color)  
*MdiWindow::currentColorChanged.*
- void [currentLinetypeChanged](#) (QString type)  
*MdiWindow::currentLinetypeChanged.*
- void [currentLineweightChanged](#) (QString weight)  
*MdiWindow::currentLineweightChanged.*
- void [updateColorLinetypeLineweight](#) ()
- void [deletePressed](#) ()
- void [escapePressed](#) ()
- void [showViewScrollBars](#) (bool val)
- void [setViewCrossHairColor](#) (QColor color)
- void [setViewBackgroundColor](#) (QColor color)
- void [setViewSelectBoxColors](#) (QColor colorL, QColor fillL, QColor colorR, QColor fillR, int alpha)
- void [setViewGridColor](#) (QColor color)
- void [setViewRulerColor](#) (QColor color)
- void [print](#) ()  
*MdiWindow::print.*
- void [saveBMC](#) ()  
*MdiWindow::saveBMC.*
- void [promptHistoryAppended](#) (QString txt)
- void [logPromptInput](#) (QString txt)
- void [promptInputPrevious](#) ()
- void [promptInputNext](#) ()  
*MdiWindow::promptInputNext.*

**Signals**

- void [sendCloseMdiWin](#) (MdiWindow \*)

## Public Member Functions

- [MdiWindow](#) (const int theIndex, QMdiArea \*parent, Qt::WindowFlags wflags)  
*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](#)
- QColor [curColor](#)
- QString [curLineType](#)
- QString [curLineWeight](#)

## 17.58.1 Constructor & Destructor Documentation

**17.58.1.1 MdiWindow()** [MdiWindow](#) (  
const int *theIndex*,  
QMdiArea \* *parent*,  
Qt::WindowFlags *wflags* )

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

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

**17.58.2.22 setViewBackgroundColor** `void setViewBackgroundColor (`  
    `QRgb color ) [slot]`

## Parameters

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

**17.58.2.23 setViewCrossHairColor** `void setViewCrossHairColor (`  
    `QRgb color ) [slot]`

## Parameters

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

**17.58.2.24 setViewGridColor** `void setViewGridColor (`  
    `QRgb color ) [slot]`

## Parameters

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

**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:

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

**17.59 Node\_Struct Reference**

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

**Public Attributes**

- [String](#) `s`
- [EmbReal](#) `r`
- `int32_t` `i`
- `bool` `b`
- [StringList](#) `sl`
- `int` `type`

**17.59.1 Member Data Documentation****17.59.1.1 b** `bool b`**17.59.1.2 i** `int32_t i`**17.59.1.3 r** [EmbReal](#) `r`**17.59.1.4 s** [String](#) `s`**17.59.1.5 sl** [StringList](#) `sl`

### 17.59.1.6 `type` `int type`

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

- `embroidermodder2/embroidermodder.h`

## 17.60 PreviewDialog Class Reference

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

### Public Member Functions

- [PreviewDialog](#) (`QWidget *parent=0`, `QString caption=QString()`, `QString directory=QString()`, `QString filter=QString()`)
- [~PreviewDialog](#) ()

### Public Attributes

- `ImageWidget *imgWidget`

### 17.60.1 Constructor & Destructor Documentation

**17.60.1.1 PreviewDialog()** `PreviewDialog` (  
    `QWidget *parent = 0`,  
    `QString caption = QString()`,  
    `QString directory = QString()`,  
    `QString filter = QString()` )

**17.60.1.2 ~PreviewDialog()** `~PreviewDialog` ( )

### 17.60.2 Member Data Documentation

**17.60.2.1 imgWidget** `ImageWidget* imgWidget`

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

- `embroidermodder2/embroidermodder.h`
- `embroidermodder2/preview-dialog.cpp`



## 17.61 PropertyEditor Class Reference

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

### Public Slots

- void [setSelectedItems](#) (std::vector< QGraphicsItem \* > itemList)
- void [updatePickAddModeButton](#) (bool pickAddMode)

### Signals

- void [pickAddModeToggled](#) ()

### Public Member Functions

- [PropertyEditor](#) (QString iconDirectory=QString(), bool pickAddMode=true, QWidget \*widgetToFocus=0, QWidget \*parent=0)
- [~PropertyEditor](#) ()
- QPushButton \* [createToolButton](#) (QString iconName, QString txt)
- QLineEdit \* [createLineEdit](#) (QString validatorType=QString(), bool readOnly=false)
- void [updateLineEditStrlfVaries](#) (QLineEdit \*lineEdit, QString str)
- void [updateLineEditNumIfVaries](#) (QLineEdit \*lineEdit, [EmbReal](#) num, bool useAnglePrecision)
- void [updateFontComboBoxStrlfVaries](#) (QFontComboBox \*fontComboBox, QString str)
- void [updateComboBoxStrlfVaries](#) (QComboBox \*comboBox, QString str, [StringList](#) strList)
- void [updateComboBoxBoolIfVaries](#) (QComboBox \*comboBox, bool val, bool yesOrNoText)
- void [mapSignal](#) (QObject \*fieldObj, QString name, QVariant value)  
*PropertyEditor::mapSignal.*
- QComboBox \* [createComboBoxSelected](#) ()
- QPushButton \* [createToolButtonQSelect](#) ()
- QPushButton \* [createToolButtonPickAdd](#) ()
- void [createGroupBox](#) (String group\_box\_key, const char \*title)

### Public Attributes

- QWidget \* [focusWidget](#)
- QString [iconDir](#)
- int [iconSize](#)
- Qt::ToolButtonStyle [propertyEditorButtonStyle](#)
- bool [pickAdd](#)
- std::vector< QGraphicsItem \* > [selectedItemList](#)
- int [precisionAngle](#)
- int [precisionLength](#)
- QSignalMapper \* [signalMapper](#)
- QComboBox \* [comboBoxSelected](#)
- QPushButton \* [toolButtonQSelect](#)
- QPushButton \* [toolButtonPickAdd](#)

### Protected Member Functions

- bool [eventFilter](#) (QObject \*obj, QEvent \*event)

## Private Slots

- void `fieldEdited` (QObject \*fieldObj)
- void `showGroups` (int objType)
- void `showOneType` (int index)
- void `hideAllGroups` ()
- void `clearAllFields` ()
- void `togglePickAddMode` ()

## 17.61.1 Constructor & Destructor Documentation

**17.61.1.1 `PropertyEditor()`** `PropertyEditor` (  
    QString *iconDirectory* = QString(),  
    bool *pickAddMode* = true,  
    QWidget \* *widgetToFocus* = 0,  
    QWidget \* *parent* = 0 )

**17.61.1.2 `~PropertyEditor()`** `~PropertyEditor` ( )

**Todo** document this

## 17.61.2 Member Function Documentation

**17.61.2.1 `clearAllFields`** void `clearAllFields` ( ) [private], [slot]

**Todo** DimAligned  
DimAngular  
DimArcLength  
DimDiameter  
DimLeader  
DimLinear  
DimOrdinate  
DimRadius

**17.61.2.2 createComboBoxSelected()** `QComboBox * createComboBoxSelected ( )`

**Todo** document this

**17.61.2.3 createGroupBox()** `void createGroupBox (`  
    `String group_box_key,`  
    `const char * title )`

**17.61.2.4 createLineEdit()** `QLineEdit * createLineEdit (`  
    `QString validatorType = QString(),`  
    `bool readOnly = false )`

**17.61.2.5 createToolButton()** `QToolButton * createToolButton (`  
    `QString iconName,`  
    `QString txt )`

**17.61.2.6 createToolButtonPickAdd()** `QToolButton * createToolButtonPickAdd ( )`

**17.61.2.7 createToolButtonQSelect()** `QToolButton * createToolButtonQSelect ( )`

**Todo** document this

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

**Todo** document this

**17.61.2.9 fieldEdited** `void fieldEdited (`  
    `QObject * fieldObj ) [private], [slot]`

**17.61.2.10 hideAllGroups** void hideAllGroups ( ) [private], [slot]

#### Note

General group will never be hidden.

**17.61.2.11 mapSignal()** void mapSignal (   
    QObject \* *fieldObj*,   
    QString *name*,   
    QVariant *value* )

[PropertyEditor::mapSignal.](#)

#### Parameters

<i>fieldObj</i>	
<i>name</i>	
<i>value</i>	

**17.61.2.12 pickAddModeToggled** void pickAddModeToggled ( ) [signal]

**17.61.2.13 setSelectedItems** void setSelectedItems (   
    std::vector< QGraphicsItem \* > *itemList* ) [slot]

**17.61.2.14 showGroups** void showGroups (   
    int *objType* ) [private], [slot]

**17.61.2.15 showOneType** void showOneType (   
    int *index* ) [private], [slot]

**17.61.2.16 togglePickAddMode** void togglePickAddMode ( ) [private], [slot]

**17.61.2.17 updateComboBoxBoolIfVaries()** `void updateComboBoxBoolIfVaries (`  
    `QComboBox * comboBox,`  
    `bool val,`  
    `bool yesOrNoText )`

**17.61.2.18 updateComboBoxStrIfVaries()** `void updateComboBoxStrIfVaries (`  
    `QComboBox * comboBox,`  
    `QString str,`  
    `StringList strList )`

**17.61.2.19 updateFontComboBoxStrIfVaries()** `void updateFontComboBoxStrIfVaries (`  
    `QFontComboBox * fontComboBox,`  
    `QString str )`

**17.61.2.20 updateLineEditNumIfVaries()** `void updateLineEditNumIfVaries (`  
    `QLineEdit * lineEdit,`  
    `EmbReal num,`  
    `bool useAnglePrecision )`

**17.61.2.21 updateLineEditStrIfVaries()** `void updateLineEditStrIfVaries (`  
    `QLineEdit * lineEdit,`  
    `QString str )`

**17.61.2.22 updatePickAddModeButton** `void updatePickAddModeButton (`  
    `bool pickAddMode ) [slot]`

### 17.61.3 Member Data Documentation

**17.61.3.1 comboBoxSelected** `QComboBox* comboBoxSelected`

**17.61.3.2 focusWidget** `QWidget* focusWidget`

**17.61.3.3 iconDir** `QString iconDir`

**17.61.3.4 iconSize** `int iconSize`

**17.61.3.5 pickAdd** `bool pickAdd`

**17.61.3.6 precisionAngle** `int precisionAngle`

**17.61.3.7 precisionLength** `int precisionLength`

**17.61.3.8 propertyEditorButtonStyle** `Qt::ToolButtonStyle propertyEditorButtonStyle`

**17.61.3.9 selectedItemList** `std::vector<QGraphicsItem*> selectedItemList`

**17.61.3.10 signalMapper** `QSignalMapper* signalMapper`

**17.61.3.11 toolButtonPickAdd** `QToolButton* toolButtonPickAdd`

**17.61.3.12 toolButtonQSelect** `QToolButton* toolButtonQSelect`

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

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

## 17.62 SaveObject Class Reference

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

### Public Member Functions

- [SaveObject](#) (QGraphicsScene \*theScene, QObject \*parent=0)  
*SaveObject::SaveObject.*
- [~SaveObject](#) ()  
*SaveObject::~~SaveObject.*
- bool [save](#) (QString fileName)
- void [addArc](#) (EmbPattern \*pattern, QGraphicsItem \*item)  
*SaveObject::addArc.*
- void [addBlock](#) (EmbPattern \*pattern, QGraphicsItem \*item)  
*SaveObject::addBlock.*
- void [addCircle](#) (EmbPattern \*pattern, QGraphicsItem \*item)  
*SaveObject::addCircle.*
- void [addDimAligned](#) (EmbPattern \*pattern, QGraphicsItem \*item)  
*SaveObject::addDimAligned.*
- void [addDimAngular](#) (EmbPattern \*pattern, QGraphicsItem \*item)  
*SaveObject::addDimAngular.*
- void [addDimArcLength](#) (EmbPattern \*pattern, QGraphicsItem \*item)  
*SaveObject::addDimArcLength.*
- void [addDimDiameter](#) (EmbPattern \*pattern, QGraphicsItem \*item)  
*SaveObject::addDimDiameter.*
- void [addDimLeader](#) (EmbPattern \*pattern, QGraphicsItem \*item)  
*SaveObject::addDimLeader.*
- void [addDimLinear](#) (EmbPattern \*pattern, QGraphicsItem \*item)  
*SaveObject::addDimLinear.*
- void [addDimOrdinate](#) (EmbPattern \*pattern, QGraphicsItem \*item)  
*SaveObject::addDimOrdinate.*
- void [addDimRadius](#) (EmbPattern \*pattern, QGraphicsItem \*item)  
*SaveObject::addDimRadius.*
- void [addEllipse](#) (EmbPattern \*pattern, QGraphicsItem \*item)  
*SaveObject::addEllipse.*
- void [addEllipseArc](#) (EmbPattern \*pattern, QGraphicsItem \*item)  
*SaveObject::addEllipseArc.*
- void [addGrid](#) (EmbPattern \*pattern, QGraphicsItem \*item)  
*SaveObject::addGrid.*
- void [addHatch](#) (EmbPattern \*pattern, QGraphicsItem \*item)  
*SaveObject::addHatch.*
- void [addImage](#) (EmbPattern \*pattern, QGraphicsItem \*item)  
*SaveObject::addImage.*
- void [addInfiniteLine](#) (EmbPattern \*pattern, QGraphicsItem \*item)  
*SaveObject::addInfiniteLine.*
- void [addLine](#) (EmbPattern \*pattern, QGraphicsItem \*item)  
*SaveObject::addLine.*
- void [addPath](#) (EmbPattern \*pattern, QGraphicsItem \*item)  
*SaveObject::addPath.*
- void [addPoint](#) (EmbPattern \*pattern, QGraphicsItem \*item)

- [SaveObject::addPoint.](#)
- void [addPolygon](#) ([EmbPattern](#) \*pattern, QGraphicsItem \*item)
- [SaveObject::addPolygon.](#)
- void [addPolyline](#) ([EmbPattern](#) \*pattern, QGraphicsItem \*item)
- [SaveObject::addPolyline.](#)
- void [addRay](#) ([EmbPattern](#) \*pattern, QGraphicsItem \*item)
- [SaveObject::addRay.](#)
- void [addRectangle](#) ([EmbPattern](#) \*pattern, QGraphicsItem \*item)
- [SaveObject::addRectangle.](#)
- void [addSlot](#) ([EmbPattern](#) \*pattern, QGraphicsItem \*item)
- [SaveObject::addSlot.](#)
- void [addSpline](#) ([EmbPattern](#) \*pattern, QGraphicsItem \*item)
- [SaveObject::addSpline.](#)
- void [addTextMulti](#) ([EmbPattern](#) \*pattern, QGraphicsItem \*item)
- [SaveObject::addTextMulti.](#)
- void [addTextSingle](#) ([EmbPattern](#) \*pattern, QGraphicsItem \*item)
- [SaveObject::addTextSingle.](#)
- void [toPolyline](#) ([EmbPattern](#) \*pattern, const QPointF &objPos, const QPainterPath &objPath, QString layer, const QColor &color, QString lineType, QString lineWeight)
- [SaveObject::toPolyline.](#)

## Public Attributes

- QGraphicsScene \* [gscene](#)
- int [formatType](#)

## 17.62.1 Constructor & Destructor Documentation

**17.62.1.1 SaveObject()** [SaveObject](#) (  
     QGraphicsScene \* *theScene*,  
     QObject \* *parent* = 0 )

[SaveObject::SaveObject.](#)

### Parameters

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

**17.62.1.2 ~SaveObject()** [~SaveObject](#) ( )

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



## 17.62.2 Member Function Documentation

**17.62.2.1 addArc()** `void addArc (`  
    `EmbPattern * pattern,`  
    `QGraphicsItem * item )`

[SaveObject::addArc.](#)

### Parameters

<i>pattern</i>	
<i>item</i>	

**17.62.2.2 addBlock()** `void addBlock (`  
    `EmbPattern * pattern,`  
    `QGraphicsItem * item )`

[SaveObject::addBlock.](#)

### Parameters

<i>pattern</i>	
<i>item</i>	

**17.62.2.3 addCircle()** `void addCircle (`  
    `EmbPattern * pattern,`  
    `QGraphicsItem * item )`

[SaveObject::addCircle.](#)

### Parameters

<i>pattern</i>	
<i>item</i>	

**17.62.2.4 addDimAligned()** `void addDimAligned (`  
    `EmbPattern * pattern,`  
    `QGraphicsItem * item )`

[SaveObject::addDimAligned.](#)

## Parameters

<i>pattern</i>	
<i>item</i>	

**17.62.2.5 addDimAngular()** `void addDimAngular (`  
    `EmbPattern * pattern,`  
    `QGraphicsItem * item )`

[SaveObject::addDimAngular.](#)

## Parameters

<i>pattern</i>	
<i>item</i>	

**17.62.2.6 addDimArcLength()** `void addDimArcLength (`  
    `EmbPattern * pattern,`  
    `QGraphicsItem * item )`

[SaveObject::addDimArcLength.](#)

## Parameters

<i>pattern</i>	
<i>item</i>	

**17.62.2.7 addDimDiameter()** `void addDimDiameter (`  
    `EmbPattern * pattern,`  
    `QGraphicsItem * item )`

[SaveObject::addDimDiameter.](#)

## Parameters

<i>pattern</i>	
<i>item</i>	

**17.62.2.8 addDimLeader()** `void addDimLeader (`

```
    EmbPattern * pattern,  
    QGraphicsItem * item )
```

[SaveObject::addDimLeader.](#)

#### Parameters

<i>pattern</i>	
<i>item</i>	

**17.62.2.9 addDimLinear()** void addDimLinear (  
 EmbPattern \* pattern,  
 QGraphicsItem \* item )

[SaveObject::addDimLinear.](#)

#### Parameters

<i>pattern</i>	
<i>item</i>	

**17.62.2.10 addDimOrdinate()** void addDimOrdinate (  
 EmbPattern \* pattern,  
 QGraphicsItem \* item )

[SaveObject::addDimOrdinate.](#)

#### Parameters

<i>pattern</i>	
<i>item</i>	

**17.62.2.11 addDimRadius()** void addDimRadius (  
 EmbPattern \* pattern,  
 QGraphicsItem \* item )

[SaveObject::addDimRadius.](#)

#### Parameters

<i>pattern</i>	
<i>item</i>	

**17.62.2.12 addEllipse()** `void addEllipse (`  
    `EmbPattern * pattern,`  
    `QGraphicsItem * item )`

[SaveObject::addEllipse.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

**17.62.2.13 addEllipseArc()** `void addEllipseArc (`  
    `EmbPattern * pattern,`  
    `QGraphicsItem * item )`

[SaveObject::addEllipseArc.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

**17.62.2.14 addGrid()** `void addGrid (`  
    `EmbPattern * pattern,`  
    `QGraphicsItem * item )`

[SaveObject::addGrid.](#)

Parameters

<i>pattern</i>	
<i>item</i>	

**17.62.2.15 addHatch()** `void addHatch (`  
    `EmbPattern * pattern,`  
    `QGraphicsItem * item )`

[SaveObject::addHatch.](#)

## Parameters

<i>pattern</i>	
<i>item</i>	

**17.62.2.16 addImage()** void addImage (   
 [EmbPattern](#) \* *pattern*,   
 QGraphicsItem \* *item* )

[SaveObject::addImage.](#)

## Parameters

<i>pattern</i>	
<i>item</i>	

**17.62.2.17 addInfiniteLine()** void addInfiniteLine (   
 [EmbPattern](#) \* *pattern*,   
 QGraphicsItem \* *item* )

[SaveObject::addInfiniteLine.](#)

## Parameters

<i>pattern</i>	
<i>item</i>	

**17.62.2.18 addLine()** void addLine (   
 [EmbPattern](#) \* *pattern*,   
 QGraphicsItem \* *item* )

[SaveObject::addLine.](#)

## Parameters

<i>pattern</i>	
<i>item</i>	

**17.62.2.19 addPath()** void addPath (

```
    EmbPattern * pattern,  
    QGraphicsItem * item )
```

[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:

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

## 17.63 SelectBox Class Reference

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

## Public Slots

- void [setDirection](#) (int dir)
- void [setColors](#) (const QColor &colorL, const QColor &fillL, const QColor &colorR, const QColor &fillR, int newAlpha)

## Public Member Functions

- [SelectBox](#) (Shape s, QWidget \*parent=0)
- void [forceRepaint](#) ()

## Public Attributes

- QColor [leftBrushColor](#)
- QColor [rightBrushColor](#)
- QColor [leftPenColor](#)
- QColor [rightPenColor](#)
- uint8\_t [alpha](#)
- QBrush [dirBrush](#)
- QBrush [leftBrush](#)
- QBrush [rightBrush](#)
- QPen [dirPen](#)
- QPen [leftPen](#)
- QPen [rightPen](#)
- bool [boxDir](#)

## Protected Member Functions

- void [paintEvent](#) (QPaintEvent \*)

## 17.63.1 Constructor & Destructor Documentation

**17.63.1.1 [SelectBox\(\)](#)** [SelectBox](#) (  
    Shape s,  
    QWidget \* parent = 0 )

Embroidermodder 2.

Copyright 2013-2022 The Embroidermodder Team Embroidermodder 2 is Open Source Software. See LICENSE for licensing terms.

Use Python's PEP7 style guide. <https://peps.python.org/pep-0007/>

---

## 17.63.2 Member Function Documentation

**17.63.2.1 [forceRepaint\(\)](#)** void [forceRepaint](#) ( )

**17.63.2.2 paintEvent()** void paintEvent (  
    QPaintEvent \* ) [protected]

**17.63.2.3 setColors** void setColors (  
    const QColor & *colorL*,  
    const QColor & *fillL*,  
    const QColor & *colorR*,  
    const QColor & *fillR*,  
    int *newAlpha* ) [slot]

**17.63.2.4 setDirection** void setDirection (  
    int *dir* ) [slot]

### 17.63.3 Member Data Documentation

**17.63.3.1 alpha** uint8\_t alpha

**17.63.3.2 boxDir** bool boxDir

**17.63.3.3 dirBrush** QBrush dirBrush

**17.63.3.4 dirPen** QPen dirPen

**17.63.3.5 leftBrush** QBrush leftBrush

**17.63.3.6 leftBrushColor** QColor leftBrushColor

**17.63.3.7 leftPen** QPen leftPen

**17.63.3.8 leftPenColor** QColor leftPenColor

**17.63.3.9 rightBrush** QBrush rightBrush

**17.63.3.10 rightBrushColor** QColor rightBrushColor

**17.63.3.11 rightPen** QPen rightPen

**17.63.3.12 rightPenColor** `QColor rightPenColor`

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

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

**17.64 Settings\_Dialog Class Reference**

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

**Signals**

- void [buttonCustomFilterSelectAll](#) (bool)
- void [buttonCustomFilterClearAll](#) (bool)
- void [buttonQSnapSelectAll](#) (bool)
- void [buttonQSnapClearAll](#) (bool)

**Public Member Functions**

- [Settings\\_Dialog](#) (QString showTab=QString(), QWidget \*parent=0)
- [~Settings\\_Dialog](#) ()
- QWidget \* [createTabGeneral](#) ()
- QWidget \* [createTabFilesPaths](#) ()
- QWidget \* [createTabDisplay](#) ()
- QWidget \* [createTabPrompt](#) ()
- QWidget \* [createTabOpenSave](#) ()
- QWidget \* [createTabPrinting](#) ()
- QWidget \* [createTabSnap](#) ()
- QWidget \* [createTabGridRuler](#) ()
- QWidget \* [createTabOrthoPolar](#) ()
- QWidget \* [createTabQuickSnap](#) ()
- QWidget \* [createTabQuickTrack](#) ()
- QWidget \* [createTabLineWeight](#) ()
- QWidget \* [createTabSelection](#) ()
- void [addColorsToComboBox](#) (QComboBox \*comboBox)
- void [create\\_float\\_spinbox](#) (QGroupBox \*gb, QGridLayout \*gridLayout, const char \*label\_in, [EmbReal](#) single\_step, [EmbReal](#) lower, [EmbReal](#) upper, [String](#), int row)
- QCheckBox \* [create\\_checkbox](#) (QGroupBox \*groupbox, [String](#) label)

**Public Attributes**

- QTabWidget \* [tabWidget](#)
- QDialogButtonBox \* [buttonBox](#)

**Private Slots**

- void [comboBoxIconSizeCurrentIndexChanged](#) (int)  
*[Settings\\_Dialog::comboBoxIconSizeCurrentIndexChanged.](#)*
- void [checkBoxGeneralMdiBGUseLogoStateChanged](#) (int)
- void [chooseGeneralMdiBackgroundLogo](#) ()
- void [checkBoxGeneralMdiBGUseTextureStateChanged](#) (int)  
*[Settings\\_Dialog::checkBoxGeneralMdiBGUseTextureStateChanged.](#)*
- void [chooseGeneralMdiBackgroundTexture](#) ()
- void [checkBoxGeneralMdiBGUseColorStateChanged](#) (int)
- void [chooseGeneralMdiBackgroundColor](#) ()
- void [currentGeneralMdiBackgroundColorChanged](#) (const QColor &)
- void [checkBoxShowScrollBarsStateChanged](#) (int)

- void [comboBoxScrollBarWidgetCurrentIndexChanged](#) (int)
- void [chooseDisplayCrossHairColor](#) ()
- void [currentDisplayCrossHairColorChanged](#) (const QColor &)
- void [chooseDisplayBackgroundColor](#) ()
- void [currentDisplayBackgroundColorChanged](#) (const QColor &)
- void [chooseDisplaySelectBoxLeftColor](#) ()
- void [currentDisplaySelectBoxLeftColorChanged](#) (const QColor &)
- void [chooseDisplaySelectBoxLeftFill](#) ()
- void [currentDisplaySelectBoxLeftFillChanged](#) (const QColor &)
- void [chooseDisplaySelectBoxRightColor](#) ()
- void [currentDisplaySelectBoxRightColorChanged](#) (const QColor &)
- void [chooseDisplaySelectBoxRightFill](#) ()
- void [currentDisplaySelectBoxRightFillChanged](#) (const QColor &)
- void [comboBoxSelectionCoolGripColorCurrentIndexChanged](#) (int index)
- void [comboBoxSelectionHotGripColorCurrentIndexChanged](#) (int index)
- void [spinBoxDisplaySelectBoxAlphaValueChanged](#) (int)
- void [choosePromptTextColor](#) ()
- void [currentPromptTextColorChanged](#) (const QColor &)
- void [choosePromptBackgroundColor](#) ()
- void [currentPromptBackgroundColorChanged](#) (const QColor &)
- void [comboBoxPromptFontFamilyCurrentIndexChanged](#) (QString)
- void [comboBoxPromptFontStyleCurrentIndexChanged](#) (QString)
- void [spinBoxPromptFontSizeValueChanged](#) (int)
- void [checkBoxPromptSaveHistoryAsHtmlStateChanged](#) (int)
- void [checkBoxCustomFilterStateChanged](#) (int)
- void [buttonCustomFilterSelectAllClicked](#) ()
- void [buttonCustomFilterClearAllClicked](#) ()
- void [checkBoxGridColorMatchCrossHairStateChanged](#) (int)
- void [chooseGridColor](#) ()
- void [currentGridColorChanged](#) (const QColor &)
- void [checkBoxGridLoadFromFileStateChanged](#) (int)
- void [comboBoxGridTypeCurrentIndexChanged](#) (QString)
- [Settings\\_Dialog::comboBoxGridTypeCurrentIndexChanged.](#)
- void [checkBoxGridCenterOnOriginStateChanged](#) (int)
- void [checkBoxRulerShowOnLoadStateChanged](#) (int)
- void [comboBoxRulerMetricCurrentIndexChanged](#) (int)
- void [chooseRulerColor](#) ()
- void [currentRulerColorChanged](#) (const QColor &)
- void [spinBoxRulerPixelSizeValueChanged](#) (double)
- void [buttonQSnapSelectAllClicked](#) ()
- void [buttonQSnapClearAllClicked](#) ()
- void [comboBoxQSnapLocatorColorCurrentIndexChanged](#) (int)
- void [checkBoxLwtShowLwtStateChanged](#) (int)
- void [checkBoxLwtRealRenderStateChanged](#) (int)
- void [acceptChanges](#) ()
- void [rejectChanges](#) ()
- [Settings\\_Dialog::rejectChanges.](#)

### 17.64.1 Constructor & Destructor Documentation

**17.64.1.1 Settings\_Dialog()** [Settings\\_Dialog](#) (  
     QString *showTab* = *QString()*,  
     QWidget \* *parent* = 0 )

**17.64.1.2** `~Settings_Dialog()` `~Settings_Dialog ( )`

## **17.64.2 Member Function Documentation**

**17.64.2.1** `acceptChanges` `void acceptChanges ( ) [private], [slot]`

**17.64.2.2** `addColorsToComboBox()` `void addColorsToComboBox (`  
`QComboBox * comboBox )`

**17.64.2.3** `buttonCustomFilterClearAll` `void buttonCustomFilterClearAll (`  
`bool ) [signal]`

**17.64.2.4** `buttonCustomFilterClearAllClicked` `void buttonCustomFilterClearAllClicked ( ) [private],`  
`[slot]`

**17.64.2.5** `buttonCustomFilterSelectAll` `void buttonCustomFilterSelectAll (`  
`bool ) [signal]`

**17.64.2.6** `buttonCustomFilterSelectAllClicked` `void buttonCustomFilterSelectAllClicked ( ) [private],`  
`[slot]`

**17.64.2.7** `buttonQSnapClearAll` `void buttonQSnapClearAll (`  
`bool ) [signal]`

**17.64.2.8** `buttonQSnapClearAllClicked` `void buttonQSnapClearAllClicked ( ) [private], [slot]`

**17.64.2.9** `buttonQSnapSelectAll` `void buttonQSnapSelectAll (`  
`bool ) [signal]`

**17.64.2.10** `buttonQSnapSelectAllClicked` `void buttonQSnapSelectAllClicked ( ) [private], [slot]`

**17.64.2.11** `checkBoxCustomFilterStateChanged` `void checkBoxCustomFilterStateChanged (`  
`int checked ) [private], [slot]`

**17.64.2.12** `checkBoxGeneralMdiBGUseColorStateChanged` `void checkBoxGeneralMdiBGUseColor↵`  
`StateChanged (`  
`int checked ) [private], [slot]`

**17.64.2.13** `checkBoxGeneralMdiBGUseLogoStateChanged` `void checkBoxGeneralMdiBGUseLogoState↵`  
`Changed (`  
`int checked ) [private], [slot]`

**17.64.2.14 checkBoxGeneralMdiBGUseTextureStateChanged** void checkBoxGeneralMdiBGUseTexture↔  
StateChanged (   
 int *checked* ) [private], [slot]  
[Settings\\_Dialog::checkBoxGeneralMdiBGUseTextureStateChanged](#).

#### Parameters

<i>checked</i>	
----------------	--

**17.64.2.15 checkBoxGridCenterOnOriginStateChanged** void checkBoxGridCenterOnOriginStateChanged (   
 int *checked* ) [private], [slot]

**17.64.2.16 checkBoxGridColorMatchCrossHairStateChanged** void checkBoxGridColorMatchCrossHair↔  
StateChanged (   
 int *checked* ) [private], [slot]

**17.64.2.17 checkBoxGridLoadFromFileStateChanged** void checkBoxGridLoadFromFileStateChanged (   
 int *checked* ) [private], [slot]

**17.64.2.18 checkBoxLwtRealRenderStateChanged** void checkBoxLwtRealRenderStateChanged (   
 int *checked* ) [private], [slot]

**17.64.2.19 checkBoxLwtShowLwtStateChanged** void checkBoxLwtShowLwtStateChanged (   
 int *checked* ) [private], [slot]

**17.64.2.20 checkBoxPromptSaveHistoryAsHtmlStateChanged** void checkBoxPromptSaveHistoryAs↔  
HtmlStateChanged (   
 int *checked* ) [private], [slot]

**17.64.2.21 checkBoxRulerShowOnLoadStateChanged** void checkBoxRulerShowOnLoadStateChanged (   
 int *checked* ) [private], [slot]

**17.64.2.22 checkBoxShowScrollBarsStateChanged** void checkBoxShowScrollBarsStateChanged (   
 int *checked* ) [private], [slot]

**17.64.2.23 chooseDisplayBackgroundColor** void chooseDisplayBackgroundColor ( ) [private],  
[slot]

**17.64.2.24 chooseDisplayCrossHairColor** void chooseDisplayCrossHairColor ( ) [private], [slot]

**17.64.2.25 chooseDisplaySelectBoxLeftColor** void chooseDisplaySelectBoxLeftColor ( ) [private],  
[slot]

**17.64.2.26 chooseDisplaySelectBoxLeftFill** void chooseDisplaySelectBoxLeftFill ( ) [private],  
[slot]

**17.64.2.27 chooseDisplaySelectBoxRightColor** void chooseDisplaySelectBoxRightColor ( ) [private],  
[slot]

**17.64.2.28 chooseDisplaySelectBoxRightFill** void chooseDisplaySelectBoxRightFill ( ) [private],  
[slot]

**17.64.2.29 chooseGeneralMdiBackgroundColor** void chooseGeneralMdiBackgroundColor ( ) [private],  
[slot]

**17.64.2.30 chooseGeneralMdiBackgroundLogo** void chooseGeneralMdiBackgroundLogo ( ) [private],  
[slot]

**17.64.2.31 chooseGeneralMdiBackgroundTexture** void chooseGeneralMdiBackgroundTexture ( ) [private],  
[slot]

**17.64.2.32 chooseGridColor** void chooseGridColor ( ) [private], [slot]

**17.64.2.33 choosePromptBackgroundColor** void choosePromptBackgroundColor ( ) [private],  
[slot]

**17.64.2.34 choosePromptTextColor** void choosePromptTextColor ( ) [private], [slot]

**17.64.2.35 chooseRulerColor** void chooseRulerColor ( ) [private], [slot]

**17.64.2.36 comboBoxGridTypeCurrentIndexChanged** void comboBoxGridTypeCurrentIndexChanged (   
QString type ) [private], [slot]  
[Settings\\_Dialog::comboBoxGridTypeCurrentIndexChanged](#).

#### Parameters

<i>type</i>	
-------------	--

**17.64.2.37 comboBoxIconSizeCurrentIndexChanged** void comboBoxIconSizeCurrentIndexChanged (   
int index ) [private], [slot]  
[Settings\\_Dialog::comboBoxIconSizeCurrentIndexChanged](#).

#### Parameters

<i>index</i>	
--------------	--



**17.64.2.38 comboBoxPromptFontFamilyCurrentIndexChanged** void comboBoxPromptFontFamily↔  
CurrentIndexChanged (   
    QString *family* ) [private], [slot]

**17.64.2.39 comboBoxPromptFontStyleCurrentIndexChanged** void comboBoxPromptFontStyleCurrent↔  
IndexChanged (   
    QString *style* ) [private], [slot]

**17.64.2.40 comboBoxQSnapLocatorColorCurrentIndexChanged** void comboBoxQSnapLocatorColor↔  
CurrentIndexChanged (   
    int *index* ) [private], [slot]

**17.64.2.41 comboBoxRulerMetricCurrentIndexChanged** void comboBoxRulerMetricCurrentIndex↔  
Changed (   
    int *index* ) [private], [slot]

**17.64.2.42 comboBoxScrollBarWidgetCurrentIndexChanged** void comboBoxScrollBarWidgetCurrent↔  
IndexChanged (   
    int *index* ) [private], [slot]

**17.64.2.43 comboBoxSelectionCoolGripColorCurrentIndexChanged** void comboBoxSelectionCoolGrip↔  
ColorCurrentIndexChanged (   
    int *index* ) [private], [slot]

**17.64.2.44 comboBoxSelectionHotGripColorCurrentIndexChanged** void comboBoxSelectionHotGrip↔  
ColorCurrentIndexChanged (   
    int *index* ) [private], [slot]

**17.64.2.45 create\_checkbox()** QCheckBox \* create\_checkbox (   
    QGroupBox \* *groupbox*,   
    String *label* )

**17.64.2.46 create\_float\_spinbox()** void create\_float\_spinbox (   
    QGroupBox \* *gb*,   
    QGridLayout \* *gridLayout*,   
    const char \* *label\_in*,   
    EmbReal *single\_step*,   
    EmbReal *lower*,   
    EmbReal *upper*,   
    String *key*,   
    int *row* )

**17.64.2.47 createTabDisplay()** QWidget \* createTabDisplay ( )

**17.64.2.48 createTabFilePaths()** `QWidget * createTabFilePaths ( )`

**17.64.2.49 createTabGeneral()** `QWidget * createTabGeneral ( )`

**17.64.2.50 createTabGridRuler()** `QWidget * createTabGridRuler ( )`

**17.64.2.51 createTabLineWeight()** `QWidget * createTabLineWeight ( )`

**17.64.2.52 createTabOpenSave()** `QWidget * createTabOpenSave ( )`

**17.64.2.53 createTabOrthoPolar()** `QWidget * createTabOrthoPolar ( )`

**17.64.2.54 createTabPrinting()** `QWidget * createTabPrinting ( )`

**17.64.2.55 createTabPrompt()** `QWidget * createTabPrompt ( )`

**17.64.2.56 createTabQuickSnap()** `QWidget * createTabQuickSnap ( )`

**17.64.2.57 createTabQuickTrack()** `QWidget * createTabQuickTrack ( )`

**17.64.2.58 createTabSelection()** `QWidget * createTabSelection ( )`

**17.64.2.59 createTabSnap()** `QWidget * createTabSnap ( )`

**17.64.2.60 currentDisplayBackgroundColorChanged** `void currentDisplayBackgroundColorChanged ( const QColor & color ) [private], [slot]`

**17.64.2.61 currentDisplayCrossHairColorChanged** `void currentDisplayCrossHairColorChanged ( const QColor & color ) [private], [slot]`

**17.64.2.62 currentDisplaySelectBoxLeftColorChanged** `void currentDisplaySelectBoxLeftColor↔  
Changed ( const QColor & color ) [private], [slot]`

**17.64.2.63 currentDisplaySelectBoxLeftFillChanged** `void currentDisplaySelectBoxLeftFillChanged ( const QColor & color ) [private], [slot]`

**17.64.2.64 currentDisplaySelectBoxRightColorChanged** void currentDisplaySelectBoxRightColor↔  
Changed (   
     const QColor & *color* )   [private], [slot]

**17.64.2.65 currentDisplaySelectBoxRightFillColorChanged** void currentDisplaySelectBoxRightFillColorChanged (   
     const QColor & *color* )   [private], [slot]

**17.64.2.66 currentGeneralMdiBackgroundColorChanged** void currentGeneralMdiBackgroundColor↔  
Changed (   
     const QColor & *color* )   [private], [slot]

**17.64.2.67 currentGridColorChanged** void currentGridColorChanged (   
     const QColor & *color* )   [private], [slot]

**17.64.2.68 currentPromptBackgroundColorChanged** void currentPromptBackgroundColorChanged (   
     const QColor & *color* )   [private], [slot]

**17.64.2.69 currentPromptTextColorChanged** void currentPromptTextColorChanged (   
     const QColor & *color* )   [private], [slot]

**17.64.2.70 currentRulerColorChanged** void currentRulerColorChanged (   
     const QColor & *color* )   [private], [slot]

**17.64.2.71 rejectChanges** void rejectChanges ( )   [private], [slot]  
[Settings\\_Dialog::rejectChanges](#).

**17.64.2.72 spinBoxDisplaySelectBoxAlphaValueChanged** void spinBoxDisplaySelectBoxAlphaValue↔  
Changed (   
     int *value* )   [private], [slot]

**17.64.2.73 spinBoxPromptFontSizeValueChanged** void spinBoxPromptFontSizeValueChanged (   
     int *value* )   [private], [slot]

**17.64.2.74 spinBoxRulerPixelSizeValueChanged** void spinBoxRulerPixelSizeValueChanged (   
     double *value* )   [private], [slot]

### 17.64.3 Member Data Documentation

**17.64.3.1 buttonBox** QDialogButtonBox\* buttonBox

### 17.64.3.2 **tabWidget** `QTabWidget* tabWidget`

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

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

## 17.65 **StatusBar Class Reference**

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

### Public Member Functions

- [StatusBar](#) (`QWidget *parent=0`)
- void [setMouseCoord](#) (`EmbReal x`, `EmbReal y`)
- void [context\\_menu\\_action](#) (`QToolButton *button`, `const char *icon`, `const char *label`, `QMenu *menu`, [String](#) `setting_page`)
- void [toggle](#) ([String](#) `key`, `bool on`)
- void [context\\_menu\\_event](#) (`QContextMenuEvent *event`, `QToolButton *button`)

### Public Attributes

- `std::unordered_map< String, QToolButton * >` [buttons](#)
- `QLabel *` [statusBarMouseCoord](#)

### 17.65.1 Detailed Description

### 17.65.2 Constructor & Destructor Documentation

**17.65.2.1 [StatusBar\(\)](#)** `StatusBar (  
    QWidget * parent = 0 )`

### 17.65.3 Member Function Documentation

**17.65.3.1 [context\\_menu\\_action\(\)](#)** `void context_menu_action (  
    QToolButton * button,  
    const char * icon,  
    const char * label,  
    QMenu * menu,  
    String setting_page )`

**17.65.3.2 [context\\_menu\\_event\(\)](#)** `void context_menu_event (  
    QContextMenuEvent * event,  
    QToolButton * button )`

**17.65.3.3 [setMouseCoord\(\)](#)** `void setMouseCoord (  
    EmbReal x,  
    EmbReal y )`

**17.65.3.4 [toggle\(\)](#)** `void toggle (  
    String key,  
    bool on )`

### 17.65.4 Member Data Documentation

**17.65.4.1 buttons** `std::unordered_map<String, QPushButton*> buttons`

**17.65.4.2 statusBarMouseCoord** `QLabel* statusBarMouseCoord`

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

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

## 17.66 StxThread\_ Struct Reference

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

### Public Attributes

- char \* [colorCode](#)
- char \* [colorName](#)
- char \* [sectionName](#)
- [SubDescriptor](#) \* [subDescriptors](#)
- [EmbColor](#) [stxColor](#)

### 17.66.1 Member Data Documentation

**17.66.1.1 colorCode** `char* colorCode`

**17.66.1.2 colorName** `char* colorName`

**17.66.1.3 sectionName** `char* sectionName`

**17.66.1.4 stxColor** [EmbColor](#) [stxColor](#)

**17.66.1.5 subDescriptors** [SubDescriptor](#)\* [subDescriptors](#)

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

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

## 17.67 SubDescriptor\_ Struct Reference

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

### Public Attributes

- int [someNum](#)
- int [someInt](#)
- int [someOtherInt](#)
- char \* [colorCode](#)
- char \* [colorName](#)

### 17.67.1 Member Data Documentation

**17.67.1.1 colorCode** `char* colorCode`

**Todo** better variable naming

**17.67.1.2 colorName** `char* colorName`

**17.67.1.3 someInt** `int someInt`

**Todo** better variable naming

**17.67.1.4 someNum** `int someNum`

**17.67.1.5 someOtherInt** `int someOtherInt`

**Todo** better variable naming

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

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

## 17.68 SvgAttribute\_ Struct Reference

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

### Public Attributes

- `char * name`
- `char * value`

### 17.68.1 Member Data Documentation

**17.68.1.1 name** `char* name`

**17.68.1.2 value** `char* value`

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

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

## 17.69 thread\_color\_ Struct Reference

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

### Public Attributes

- `char name [22]`
- `unsigned int hex_code`
- `int manufacturer_code`

### 17.69.1 Member Data Documentation

**17.69.1.1 hex\_code** unsigned int hex\_code

**17.69.1.2 manufacturer\_code** int manufacturer\_code

**17.69.1.3 name** char name[22]

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

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

## 17.70 ThredExtension\_ Struct Reference

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

### Public Attributes

- float [hoopX](#)
- float [hoopY](#)
- float [stitchGranularity](#)
- char [creatorName](#) [50]
- char [modifierName](#) [50]
- char [auxFormat](#)
- char [reserved](#) [31]

### 17.70.1 Member Data Documentation

**17.70.1.1 auxFormat** char auxFormat

**17.70.1.2 creatorName** char creatorName[50]

**17.70.1.3 hoopX** float hoopX

**17.70.1.4 hoopY** float hoopY

**17.70.1.5 modifierName** char modifierName[50]

**17.70.1.6 reserved** char reserved[31]

**17.70.1.7 stitchGranularity** float stitchGranularity

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

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

## 17.71 ThredHeader\_ Struct Reference

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

### Public Attributes

- unsigned int [sigVersion](#)
- unsigned int [length](#)
- unsigned short [numStiches](#)
- unsigned short [hoopSize](#)
- unsigned short [reserved](#) [7]

### 17.71.1 Member Data Documentation

**17.71.1.1 [hoopSize](#)** unsigned short `hoopSize`

**17.71.1.2 [length](#)** unsigned int `length`

**17.71.1.3 [numStiches](#)** unsigned short `numStiches`

**17.71.1.4 [reserved](#)** unsigned short `reserved[7]`

**17.71.1.5 [sigVersion](#)** unsigned int `sigVersion`

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

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

## 17.72 UndoableCommand Class Reference

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

### Public Member Functions

- [UndoableCommand](#) ([String](#) `command`, [QString](#) `text`, [Geometry](#) \*`obj`, [View](#) \*`v`, [QUndoCommand](#) \*`parent`=0)
- [UndoableCommand](#) ([EmbVector](#) `d`, [QString](#) `text`, [Geometry](#) \*`obj`, [View](#) \*`v`, [QUndoCommand](#) \*`parent`=0)
- [UndoableCommand](#) ([String](#) `command`, [EmbVector](#) `pivot`, [EmbReal](#) `angle`, [QString](#) `text`, [Geometry](#) \*`obj`, [View](#) \*`v`, [QUndoCommand](#) \*`parent`=0)
- [UndoableCommand](#) ([QString](#) `type`, [View](#) \*`v`, [QUndoCommand](#) \*`parent`=0)
- [UndoableCommand](#) (const [QPointF](#) `beforePoint`, const [QPointF](#) `afterPoint`, [QString](#) `text`, [Geometry](#) \*`obj`, [View](#) \*`v`, [QUndoCommand](#) \*`parent`=0)
- [UndoableCommand](#) ([EmbReal](#) `x1`, [EmbReal](#) `y1`, [EmbReal](#) `x2`, [EmbReal](#) `y2`, [QString](#) `text`, [Geometry](#) \*`obj`, [View](#) \*`v`, [QUndoCommand](#) \*`parent`=0)
- int [id](#) ()
- bool [mergeWith](#) (const [QUndoCommand](#) \*`command`)
- void [undo](#) ()
- void [redo](#) ()
- void [mirror](#) ()
- void [rotate](#) ([EmbVector](#) `pivot`, [EmbReal](#) `rot`)



**Public Attributes**

- [Geometry](#) \* [object](#)
- [View](#) \* [gview](#)
- [String](#) [command](#)
- [EmbVector](#) [delta](#)
- [EmbVector](#) [pivot](#)
- [QPointF](#) [before](#)
- [QPointF](#) [after](#)
- [EmbReal](#) [angle](#)
- [EmbReal](#) [factor](#)
- [QString](#) [navType](#)
- [QTransform](#) [fromTransform](#)
- [QTransform](#) [toTransform](#)
- [QPointF](#) [fromCenter](#)
- [QPointF](#) [toCenter](#)
- [QLineF](#) [mirrorLine](#)
- [bool](#) [done](#)

**17.72.1 Constructor & Destructor Documentation**

**17.72.1.1 UndoableCommand()** [1/6] [UndoableCommand](#) (  
[String](#) *command*,  
[QString](#) *text*,  
[Geometry](#) \* *obj*,  
[View](#) \* *v*,  
[QUndoCommand](#) \* *parent* = 0 )

**17.72.1.2 UndoableCommand()** [2/6] [UndoableCommand](#) (  
[EmbVector](#) *d*,  
[QString](#) *text*,  
[Geometry](#) \* *obj*,  
[View](#) \* *v*,  
[QUndoCommand](#) \* *parent* = 0 )

**17.72.1.3 UndoableCommand()** [3/6] [UndoableCommand](#) (  
[String](#) *command*,  
[EmbVector](#) *pivot*,  
[EmbReal](#) *angle*,  
[QString](#) *text*,  
[Geometry](#) \* *obj*,  
[View](#) \* *v*,  
[QUndoCommand](#) \* *parent* = 0 )

**17.72.1.4 UndoableCommand()** [4/6] [UndoableCommand](#) (  
[QString](#) *type*,  
[View](#) \* *v*,  
[QUndoCommand](#) \* *parent* = 0 )

**17.72.1.5 UndoableCommand()** [5/6] `UndoableCommand` (

```
    const QPointF beforePoint,
    const QPointF afterPoint,
    QString text,
    Geometry * obj,
    View * v,
    QUndoCommand * parent = 0 )
```

**17.72.1.6 UndoableCommand()** [6/6] `UndoableCommand` (

```
    EmbReal x1,
    EmbReal y1,
    EmbReal x2,
    EmbReal y2,
    QString text,
    Geometry * obj,
    View * v,
    QUndoCommand * parent = 0 )
```

## 17.72.2 Member Function Documentation

**17.72.2.1 id()** `int id ( )` [inline]

**17.72.2.2 mergeWith()** `bool mergeWith (`  
 `const QUndoCommand * command )`

**17.72.2.3 mirror()** `void mirror ( )`

**17.72.2.4 redo()** `void redo ( )`

**17.72.2.5 rotate()** `void rotate (`  
 `EmbVector pivot,`  
 `EmbReal rot )`

**17.72.2.6 undo()** `void undo ( )`

## 17.72.3 Member Data Documentation

**17.72.3.1 after** `QPointF after`

**17.72.3.2 angle** `EmbReal angle`

**17.72.3.3 before** `QPointF before`

**17.72.3.4 command** [String](#) command

**17.72.3.5 delta** [EmbVector](#) delta

**17.72.3.6 done** bool done

**17.72.3.7 factor** [EmbReal](#) factor

**17.72.3.8 fromCenter** [QPointF](#) fromCenter

**17.72.3.9 fromTransform** [QTransform](#) fromTransform

**17.72.3.10 gview** [View\\*](#) gview

**17.72.3.11 mirrorLine** [QLineF](#) mirrorLine

**17.72.3.12 navType** [QString](#) navType

**17.72.3.13 object** [Geometry\\*](#) object

**17.72.3.14 pivot** [EmbVector](#) pivot

**17.72.3.15 toCenter** [QPointF](#) toCenter

**17.72.3.16 toTransform** [QTransform](#) toTransform

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

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

## 17.73 UndoEditor Class Reference

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

### Public Slots

- void [undo](#) ()
- void [redo](#) ()
- void [updateCleanIcon](#) (bool opened)

## Public Member Functions

- [UndoEditor](#) (QString iconDirectory=QString(), QWidget \*widgetToFocus=0, QWidget \*parent=0)
- [~UndoEditor](#) ()
- void [addStack](#) (QUndoStack \*stack)
- bool [canUndo](#) ()
- bool [canRedo](#) ()
- QString [undoText](#) ()
- QString [redoText](#) ()

## Public Attributes

- QWidget \* [focusWidget](#)
- QString [iconDir](#)
- int [iconSize](#)
- QUndoGroup \* [undoGroup](#)
- QUndoView \* [undoView](#)

## 17.73.1 Constructor & Destructor Documentation

**17.73.1.1 UndoEditor()** [UndoEditor](#) (  
    QString *iconDirectory* = QString(),  
    QWidget \* *widgetToFocus* = 0,  
    QWidget \* *parent* = 0 )

**17.73.1.2 ~UndoEditor()** [~UndoEditor](#) ( )

## 17.73.2 Member Function Documentation

**17.73.2.1 addStack()** void [addStack](#) (  
    QUndoStack \* *stack* )

**17.73.2.2 canRedo()** bool [canRedo](#) ( )

**17.73.2.3 canUndo()** bool [canUndo](#) ( )

**17.73.2.4 redo** void [redo](#) ( ) [slot]

**17.73.2.5 redoText()** QString [redoText](#) ( )

**17.73.2.6 undo** void [undo](#) ( ) [slot]

**17.73.2.7 undoText()** QString [undoText](#) ( )

**17.73.2.8 updateCleanIcon** void updateCleanIcon (  
     bool *opened* ) [slot]

### 17.73.3 Member Data Documentation

**17.73.3.1 focusWidget** QWidget\* focusWidget

**17.73.3.2 iconDir** QString iconDir

**17.73.3.3 iconSize** int iconSize

**17.73.3.4 undoGroup** QUndoGroup\* undoGroup

**17.73.3.5 undoView** QUndoView\* undoView

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

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

## 17.74 View Class Reference

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

### Public Slots

- void [zoomIn](#) ()
- void [zoomOut](#) ()
- void [zoomWindow](#) ()
- void [zoomSelected](#) ()
- void [zoomExtents](#) ()
- void [panRealTime](#) ()
- void [panPoint](#) ()
- void [panLeft](#) ()
- void [panRight](#) ()
- void [panUp](#) ()
- void [panDown](#) ()
- void [selectAll](#) ()
- void [selectionChanged](#) ()
- void [clearSelection](#) ()
- void [deleteSelected](#) ()
- void [moveSelected](#) (EmbReal dx, EmbReal dy)
- void [cut](#) ()
- void [copy](#) ()
- void [paste](#) ()
- void [repeatAction](#) ()
- void [moveAction](#) ()
- void [scaleAction](#) ()
- void [scaleSelected](#) (EmbReal x, EmbReal y, EmbReal factor)
- void [rotateAction](#) ()
- void [rotateSelected](#) (EmbReal x, EmbReal y, EmbReal rot)

- void `mirrorSelected` (`EmbReal` x1, `EmbReal` y1, `EmbReal` x2, `EmbReal` y2)
- int `numSelected` ()
- void `deletePressed` ()
- void `escapePressed` ()
- void `cornerButtonClicked` ()
- void `showScrollBars` (bool val)
- void `setCornerButton` ()
- void `setCrossHairColor` (QRgb color)
- void `setCrossHairSize` (uint8\_t percent)
- void `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:

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

```
#include "embroidermodder.h"
```

#### 18.2.1 Detailed Description

Embroidermodder 2.

Copyright 2013-2022 The Embroidermodder Team Embroidermodder 2 is Open Source Software. See LICENSE for licensing terms.

Use Python's PEP7 style guide. <https://peps.python.org/pep-0007/>

---

### 18.3 embroidermodder2/em2\_dev\_script.py File Reference

#### Namespaces

- namespace [em2\\_dev\\_script](#)

#### Variables

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

### 18.4 embroidermodder2/embdetails-dialog.cpp File Reference

```
#include "embroidermodder.h"
```

### 18.5 embroidermodder2/embroidermodder.cpp File Reference

```
#include "embroidermodder.h"
```

#### Functions

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

#### Variables

- static const char \* [\\_appVer\\_](#) = "v2.0.0-alpha3"
- static bool [exitApp](#) = false
- const char \* [usage\\_msg](#)

#### 18.5.1 Function Documentation



## 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](#)

### 18.6.1 Detailed Description

The only header for the GUI part: a good overview of this source code.

### 18.6.2 Macro Definition Documentation

**18.6.2.1 `BOOL_TYPE`** `#define BOOL_TYPE 4`

**18.6.2.2 `FUNCTION_TYPE`** `#define FUNCTION_TYPE 5`

**18.6.2.3 `INT_TYPE`** `#define INT_TYPE 3`

**18.6.2.4 `REAL_TYPE`** `#define REAL_TYPE 2`

**18.6.2.5 `STRING_LIST_TYPE`** `#define STRING_LIST_TYPE 1`

**18.6.2.6 `STRING_TYPE`** `#define STRING_TYPE 0`

**18.6.2.7 `UNKNOWN_TYPE`** `#define UNKNOWN_TYPE 7`

**18.6.2.8 VECTOR\_TYPE** `#define VECTOR_TYPE 6`

### 18.6.3 Typedef Documentation

**18.6.3.1 Command** `typedef String(* Command) (String)`

**18.6.3.2 Dictionary** `typedef std::unordered_map<String, Node> Dictionary`

**18.6.3.3 Node** `typedef struct Node_ Node`

**18.6.3.4 NodeList** `typedef std::vector<Node> NodeList`

**18.6.3.5 String** `typedef std::string String`

**18.6.3.6 StringList** `typedef std::vector<String> StringList`

### 18.6.4 Enumeration Type Documentation

**18.6.4.1 OBJ\_KEYS** `enum OBJ_KEYS`

Custom Data used in QGraphicsItems

( int, const QVariant)

I.E. object.setData(OBJ\_TYPE, OBJ\_TYPE\_LINE); I.E. object.setData(OBJ\_LAYER, "OUTLINE"); I.E. object.setData(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 <b>Todo</b> Use color chart in formats/format-dxf.h for this
OBJ_LWT	
OBJ_RUBBER	

**18.6.4.2 OBJ\_TYPE\_VALUES** `enum OBJ_TYPE_VALUES`

#### Enumerator

OBJ_TYPE_NULL	
OBJ_TYPE_BASE	

## Enumerator

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	

## 18.6.5 Function Documentation

**18.6.5.1 activeScene()** `QGraphicsScene * activeScene ( )`

MainWindow::activeScene.

Returns

**18.6.5.2 activeView()** `View * activeView (`  
`void )`

activeView

Returns

**18.6.5.3 actuator()** `String actuator (`  
`String line )`

MainWindow::actuator.

Parameters

<i>command</i>	
----------------	--

**18.6.5.4 RUN COMMAND** `QAction* act = qobject_cast<QAction*>(sender()); if (act) { prompt->endCommand(); prompt->setCurrentText(act->objectName()); prompt->processInput(); }`

**18.6.5.5 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);`

**18.6.5.6 PROMPT** `QString fileName = "commands/" + cmd + "/" + cmd + ".js"; NOTE: Replace any special characters that will cause a syntax error QString safeStr = str; safeStr.replace("\\", "\\"); safeStr.replace("'", "\'"); if (prompt->promptInput->rapidFireEnabled) { engine->evaluate(cmd + "_prompt(" + safeStr + ")", fileName); } else { engine->evaluate(cmd + "_prompt(" + safeStr.toUpperCase() + ")", fileName); }`

**18.6.5.7 add\_polyline()** `void add_polyline (`  
`QPainterPath p,`  
`String rubberMode )`

add\_polyline

Parameters

<i>p</i>	
<i>rubberMode</i>	

**18.6.5.8 add\_to\_path()** `QPainterPath add_to_path (`  
`QPainterPath path,`  
`EmbVector scale,`  
`String command )`

**18.6.5.9 construct\_command()** `String construct_command (`  
`String command,`  
`const char * fmt,`  
`... )`

construct\_command

Parameters

<i>command</i>	
<i>fmt</i>	

Returns

**18.6.5.10 contains()** `bool contains (`

```
StringList list,
String entry )
```

**18.6.5.11 convert\_args\_to\_type()** `String` convert\_args\_to\_type (   
`String` label,   
`StringList` args,   
const char \* args\_template,   
`NodeList` a )

**18.6.5.12 create\_menu()** void create\_menu (   
std::string menu,   
`StringList` def,   
bool topLevel )

create\_menu

Parameters

<i>menu</i>	
<i>def</i>	
<i>topLevel</i>	

**18.6.5.13 debug\_message()** void debug\_message (   
std::string msg )

debug\_message

Parameters

<i>msg</i>	
------------	--

**18.6.5.14 degrees\_\_()** `EmbReal` degrees\_\_ (   
`EmbReal` radian )

degrees\_\_

Parameters

<i>radian</i>	
---------------	--

Returns

**18.6.5.15 fileExtension()** `QString` fileExtension (   
`String` fileName )

MdiWindow::fileExtension.

Parameters

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

## Returns

**18.6.5.16** **get\_bool()** `bool get_bool (`  
    `Dictionary d,`  
    `String key )`

**18.6.5.17** **get\_int()** `int32_t get_int (`  
    `Dictionary d,`  
    `String key )`

**18.6.5.18** **get\_qstr()** `QString get_qstr (`  
    `Dictionary d,`  
    `String key )`

**18.6.5.19** **get\_real()** `EmbReal get_real (`  
    `Dictionary d,`  
    `String key )`

**18.6.5.20** **get\_str()** `String get_str (`  
    `Dictionary d,`  
    `String key )`

**18.6.5.21** **get\_str\_list()** `StringList get_str_list (`  
    `Dictionary d,`  
    `String key )`

**18.6.5.22** **get\_uint()** `uint32_t get_uint (`  
    `Dictionary d,`  
    `String key )`

**18.6.5.23** **make\_checkbox()** `QCheckBox * make_checkbox (`  
    `QGroupBox * gb,`  
    `String dictionary,`  
    `const char * label,`  
    `const char * icon,`  
    `String key )`

**18.6.5.24** **make\_spinbox()** `QDoubleSpinBox * make_spinbox (`  
    `QGroupBox * gb,`  
    `String dictionary,`  
    `QString object_name,`  
    `EmbReal single_step,`  
    `EmbReal lower,`  
    `EmbReal upper,`  
    `String key )`

**18.6.5.25 make\_ui\_element()** void make\_ui\_element (   
String description )

**18.6.5.26 node\_bool()** Node node\_bool (   
bool value )

set\_node

Parameters

node	
value	

**18.6.5.27 node\_int()** Node node\_int (   
int32\_t value )

create\_node

Parameters

mode	
------	--

Returns

**18.6.5.28 node\_qstr()** Node node\_qstr (   
QString value )

set\_node

Parameters

node	
value	

**18.6.5.29 node\_real()** Node node\_real (   
EmbReal value )

set\_node

Parameters

node	
value	

**18.6.5.30 node\_str()** Node node\_str (   
String value )

set\_node

**Parameters**

<i>node</i>	
<i>value</i>	

**18.6.5.31 node\_str\_list()** `Node` node\_str\_list (   
 `StringList` value )

set\_node

**Parameters**

<i>node</i>	
<i>value</i>	

**18.6.5.32 node\_uint()** `Node` node\_uint (   
 `uint32_t` value )

create\_node

**Parameters**

<i>mode</i>	
-------------	--

**Returns**

**18.6.5.33 operator\*()** `EmbVector` operator\* (   
 `EmbVector` v,   
 `EmbReal` s )

operator \*

**Parameters**

<i>v</i>	
<i>s</i>	

**Returns**

**18.6.5.34 operator+()** `EmbVector` operator+ (   
 `EmbVector` a,   
 `EmbVector` b )

operator + Wrapper for embVector\_add to use the syntax *a + b*.

**18.6.5.35 operator-()** `EmbVector` operator- (   
 `EmbVector` a,   
 `EmbVector` b )



operator - Wrapper for `embVector_subtract` to use the syntax  $a - b$ .

**18.6.5.36 radians\_\_()** `EmbReal` radians\_\_ (   
 `EmbReal` degrees )

radians\_\_

Parameters

<i>degrees</i>	
----------------	--

Returns

**18.6.5.37 read\_configuration()** `int` read\_configuration (   
 `const char *` file )

**18.6.5.38 read\_settings()** `void` read\_settings (   
 `void` )

read\_settings

This file needs to be read from the users home directory to ensure it is writable.

**18.6.5.39 read\_string\_setting()** `String` read\_string\_setting (   
 `toml_table_t *` table,   
 `const char *` key )

**18.6.5.40 rotate\_vector()** `EmbVector` rotate\_vector (   
 `EmbVector` v,   
 `EmbReal` alpha )

Returns

**18.6.5.41 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
-----
```

**18.6.5.42 run\_script\_file()** `String run_script_file (`  
                                  `String fname )`

MainWindow::run\_script\_file.

Parameters

<i>fname</i>	The path of the script to run.
--------------	--------------------------------

**18.6.5.43 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.

**18.6.5.44 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.

**18.6.5.45 to\_EmbVector()** `EmbVector to_EmbVector (`  
                          `QPointF a )`

**18.6.5.46 to\_qlist()** `QList< QGraphicsItem * > to_qlist (`  
                          `std::vector< QGraphicsItem * > list )`

to\_qlist

Parameters

<i>list</i>	
-------------	--

Returns

**18.6.5.47 to\_QPointF()** QPointF to\_QPointF (   
 EmbVector a )

**18.6.5.48 to\_string\_vector()** StringList to\_string\_vector (   
 QStringList list )   
 to\_string\_vector

Parameters

<i>list</i>	
-------------	--

Returns

**18.6.5.49 to\_vector()** std::vector< QGraphicsItem \* > to\_vector (   
 QList< QGraphicsItem \* > list )   
 to\_vector

Parameters

<i>list</i>	
-------------	--

Returns

**18.6.5.50 tokenize()** StringList tokenize (   
 String str,   
 const char delim )   
 tokenize

Parameters

<i>str</i>	
<i>delim</i>	

Returns

**18.6.5.51 translate\_str()** QString translate\_str (   
 const char \* str )

```
MainWindow::validFileFormat.
```

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

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

MainWindow::writeSettings.

This file needs to be read from the users home directory to ensure it is writable

### 18.6.6.1 `_mainWin` `MainWindow*` `_mainWin` [extern]

### 18.6.6.2 actionHash

`std::unordered_map<String, QAction*> actionHash` [extern]

### 18.6.6.3 checkBoxes

`std::unordered_map<String, QCheckBox *> checkBoxes` [extern]

#### 18.6.6.4 comboBoxes

#### 18.6.6.5 config Dictionary config

#### 18.6.6.6 config tables

#### 18.6.6.7 dialog Dictionary dialog

#### 18.6.6.8 dockPropEdit [PropertyEditor\\*](#) dockPropEdit [extern]

#### 18.6.6.9 dockUndoEdit UndoEditor\* dockUndoEdit [extern]

### 18.6.6.10 doubleSpinBoxes

**18.6.6.11 emb\_constant\_pi** `const EmbReal emb_constant_pi = 3.14159265358979323846 [static]`

**18.6.6.12 groupBoxes** `std::unordered_map<String, QGroupBox *> groupBoxes [extern]`

**18.6.6.13 labels** `std::unordered_map<String, QLabel *> labels [extern]`

**18.6.6.14 lineEdits** `std::unordered_map<String, QLineEdit *> lineEdits [extern]`

**18.6.6.15 mdiArea** `MdiArea* mdiArea [extern]`

**18.6.6.16 menuHash** `std::unordered_map<String, QMenu*> menuHash [extern]`

**18.6.6.17 prompt** `CmdPrompt* prompt [extern]`

**18.6.6.18 scripts** `std::unordered_map<String, StringList> scripts [extern]`

**18.6.6.19 settings** `Dictionary settings [extern]`

Settings System.

Rather than pollute the global namespace, we collect together all the global settings into a structure that stores them. This also allows us to create a complete copy of the settings for the purpose of restoring them if the user cancels out of the Settings Dialog.

**18.6.6.20 spinBoxes** `std::unordered_map<String, QSpinBox *> spinBoxes [extern]`

**18.6.6.21 statusbar** `StatusBar* statusbar [extern]`

**18.6.6.22 subMenuHash** `std::unordered_map<String, QMenu*> subMenuHash [extern]`

**18.6.6.23 toolbarHash** `std::unordered_map<String, QToolBar*> toolbarHash [extern]`

**18.6.6.24 toolButtons** `std::unordered_map<String, QToolButton *> toolButtons [extern]`

## 18.7 embroidermodder.h

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

```
1 /*
2  * Embroidermodder 2.
3  *
4  * -----
5  *
6  * Copyright 2013-2023 The Embroidermodder Team
7  * Embroidermodder 2 is Open Source Software.
8  * See LICENSE for licensing terms.
9  *
10 * -----
11 *
12 * Use Python's PEP7 style guide.
```

```

13 *      https://peps.python.org/pep-0007/
14 */
15
21 #ifndef __EMBROIDERMODDER_UTILITY_H__
22 #define __EMBROIDERMODDER_UTILITY_H__
23
24 /*
25 * C/C++ Standard Libraries.
26 */
27 #include <cstdio>
28 #include <cmath>
29 #include <ctime>
30 #include <cinttypes>
31 #include <cstdlib>
32 #include <vector>
33 #include <unordered_map>
34 #include <string>
35 #include <filesystem>
36
37 /*
38 * Libraries included in "extern/".
39 */
40 #include "embroidery.h"
41 #include "toml.h"
42
43 /*
44 * Qt 6.0+ libraries.
45 */
46 #include <QAction>
47 #include <QApplication>
48
49 #include <QtPrintSupport>
50
51 #define STRING_TYPE          0
52 #define STRING_LIST_TYPE    1
53 #define REAL_TYPE           2
54 #define INT_TYPE             3
55 #define BOOL_TYPE           4
56 #define FUNCTION_TYPE       5
57 #define VECTOR_TYPE         6
58 #define UNKNOWN_TYPE        7
59
60 class ImageWidget;
61 class MdiArea;
62 class MdiWindow;
63 class View;
64 class StatusBar;
65 class CmdPrompt;
66 class PropertyEditor;
67 class UndoEditor;
68 class MainWindow;
69 class Geometry;
70
71 typedef std::string String;
72 typedef std::vector<String> StringList;
73
74 typedef struct Node_ {
75     String s;
76     EmbReal r;
77     int32_t i;
78     bool b;
79     StringList sl;
80     int type;
81 } Node;
82
83 typedef String (*Command)(String);
84 typedef std::vector<Node> NodeList;
85 typedef std::unordered_map<String, Node> Dictionary;
86
87 //Values
88 enum OBJ_TYPE_VALUES {
89     OBJ_TYPE_NULL = 0,
90     /*< NOTE: Allow this enum to evaluate false */
91     OBJ_TYPE_BASE = 100000,
92     /*< NOTE: Values >= 65536 ensure compatibility with qgraphicsitem_cast() */
93     OBJ_TYPE_ARC = 100001,
94     OBJ_TYPE_BLOCK = 100002,
95     /*< For the block type, that has to exist for SVG. */
96     OBJ_TYPE_CIRCLE = 100003,
97     OBJ_TYPE_DIMALIGNED = 100004,
98     /*< For the Aligned Dimension, that has to exist for DXF drawings. */
99     OBJ_TYPE_DIMANGULAR = 100005,
100    /*< For the Angular Dimension, that has to exist for DXF drawings. */
101    OBJ_TYPE_DIMARCLENGTH = 100006,
102    /*< For the Arc Length Dimension, that has to exist for DXF drawings. */
103    OBJ_TYPE_DIMDIAMETER = 100007,
104    OBJ_TYPE_DIMLEADER = 100008,

```

```

105     OBJ_TYPE_DIMLINEAR = 100009,
106     /*< For the Linear Dimension, that has to exist for DXF drawings. */
107     OBJ_TYPE_DIMORDINATE = 100010,
108     /*< For the Ordinate Dimension, that has to exist for DXF drawings. */
109     OBJ_TYPE_DIMRADIUS = 100011,
110     /*< For the Radial Dimension, that has to exist for DXF drawings. */
111     OBJ_TYPE_ELLIPSE = 100012,
112     OBJ_TYPE_ELLIPSEARC = 100013,
113     OBJ_TYPE_RUBBER = 100014,
114     OBJ_TYPE_GRID = 100015,
115     OBJ_TYPE_HATCH = 100016,
116     OBJ_TYPE_IMAGE = 100017,
117     OBJ_TYPE_INFINITELINE = 100018,
118     /*< For the Infinite Line object. Which should be removed from output as it exists
119 for drafting reasons. */
120     OBJ_TYPE_LINE = 100019,
121     OBJ_TYPE_PATH = 100020,
122     OBJ_TYPE_POINT = 100021,
123     OBJ_TYPE_POLYGON = 100022,
124     OBJ_TYPE_POLYLINE = 100023,
125     OBJ_TYPE_RAY = 100024,
126     /*< For the Ray object. */
127     OBJ_TYPE_RECTANGLE = 100025,
128     OBJ_TYPE_SLOT = 100026,
129     OBJ_TYPE_SPLINE = 100027,
130     OBJ_TYPE_TEXTMULTI = 100028,
131     OBJ_TYPE_TEXTSINGLE = 100029,
132     OBJ_TYPE_UNKNOWN = 100030
133 };
134
135 enum OBJ_KEYS {
136     OBJ_TYPE = 0,
137     /*< value type - int: See OBJ_TYPE_VALUES */
138     OBJ_NAME = 1,
139     /*< value type - str: See OBJ_NAME_VALUES */
140     OBJ_LAYER = 2,
141     /*< value type - str: "USER", "DEFINED", "STRINGS", etc... */
142     OBJ_COLOR = 3,
143     OBJ_LTYPE = 4,
144     /*< value type - int: See OBJ_LTYPE_VALUES */
145     OBJ_LWT = 5, //value type - int: 0-27
146     OBJ_RUBBER = 6 //value type - int: See OBJ_RUBBER_VALUES
147 };
148
149 static const EmbReal emb_constant_pi = 3.14159265358979323846;
150
151 /* Global variables
152 * -----
153 */
154 extern MdiArea* mdiArea;
155
156 extern Dictionary settings, dialog, config;
157 extern std::unordered_map<String, QStringList> scripts;
158 extern std::unordered_map<String, QGroupBox*> groupBoxes;
159 extern std::unordered_map<String, QCheckBox*> checkBoxes;
160 extern std::unordered_map<String, QSpinBox*> spinBoxes;
161 extern std::unordered_map<String, QDoubleSpinBox*> doubleSpinBoxes;
162 extern std::unordered_map<String, QLabel*> labels;
163 extern std::unordered_map<String, QComboBox*> comboBoxes;
164 extern std::unordered_map<String, QLineEdit*> lineEdits;
165 extern std::unordered_map<String, QToolButton*> toolButtons;
166 extern std::unordered_map<String, Dictionary> config_tables;
167 extern std::unordered_map<String, QAction*> actionHash;
168 extern std::unordered_map<String, QToolBar*> toolbarHash;
169 extern std::unordered_map<String, QMenu*> menuHash;
170 extern std::unordered_map<String, QMenu*> subMenuHash;
171
172 extern MainWindow* _mainWin;
173 extern CmdPrompt* prompt;
174 extern PropertyEditor* dockPropEdit;
175 extern UndoEditor* dockUndoEdit;
176 extern StatusBar* statusBar;
177
178 /* Functions in the global namespace
179 * -----
180 */
181 int read_configuration(const char *file);
182 void read_settings(void);
183 void write_settings(void);
184 EmbVector rotate_vector(EmbVector v, EmbReal alpha);
185
186 QString translate_str(const char *str);
187 bool contains(StringList, String);
188 bool validFileFormat(String fileName);
189 QString fileExtension(String fileName);
190
191 void add_polyline(QPainterPath p, String rubberMode);

```

```

215
216 String read_string_setting(toml_table_t *table, const char *key);
217 StringList tokenize(String str, const char delim);
218 String convert_args_to_type(String label, StringList args,
219     const char *args_template, NodeList a);
220
221 View *activeView(void);
222 QGraphicsScene* activeScene();
223
224 void debug_message(String msg);
225 void set_enabled(QObject *parent, const char *key, bool enabled);
226 void set_visibility(QObject *parent, const char *name, bool visibility);
227 QPainterPath add_to_path(QPainterPath path, EmbVector scale, String s);
228
229 String actuator(String line);
230 String run_script_file(String fname);
231 String run_script(StringList script);
232 String construct_command(String command, const char *fmt, ...);
233
234 void create_menu(String menu, StringList def, bool topLevel);
235
236 QPointF to_QPointF(EmbVector a);
237 EmbVector to_EmbVector(QPointF a);
238 EmbVector operator+(EmbVector a, EmbVector b);
239 EmbVector operator-(EmbVector a, EmbVector b);
240 EmbVector operator*(EmbVector v, EmbReal s);
241 EmbReal radians__(EmbReal degrees);
242 EmbReal degrees__(EmbReal radian);
243
244 std::vector<QGraphicsItem*> to_vector(QList<QGraphicsItem*> list);
245 QList<QGraphicsItem*> to_qlist(std::vector<QGraphicsItem*> list);
246
247 StringList to_string_vector(QStringList list);
248
249 /* Interface creation functions.
250 */
251 void make_ui_element(String description);
252 QDoubleSpinBox *make_spinbox(QGroupBox *gb, String d,
253     QString object_name, EmbReal single_step, EmbReal lower, EmbReal upper, String key);
254 QCheckBox *make_checkbox(QGroupBox *gb, String d,
255     const char *label, const char *icon, String key);
256
257 /* Dictionary management functions.
258 */
259 Node node_bool(bool value);
260 Node node_int(int32_t value);
261 Node node_uint(uint32_t value);
262 Node node_real(EmbReal value);
263 Node node_str(String value);
264 Node node_qstr(QString value);
265 Node node_str_list(StringList value);
266
267 bool get_bool(Dictionary d, String key);
268 int32_t get_int(Dictionary d, String key);
269 uint32_t get_uint(Dictionary d, String key);
270 EmbReal get_real(Dictionary d, String key);
271 String get_str(Dictionary d, String key);
272 QString get_qstr(Dictionary d, String key);
273 StringList get_str_list(Dictionary d, String key);
274
275 class Geometry : public QGraphicsPathItem
276 {
277 public:
278     enum ArrowStyle {
279         NoArrow, //NOTE: Allow this enum to evaluate false
280         Open,
281         Closed,
282         Dot,
283         Box,
284         Tick
285     };
286
287     enum lineStyle {
288         NoLine, //NOTE: Allow this enum to evaluate false
289         Flared,
290         Fletching
291     };
292
293     Dictionary properties;
294
295     QPen objPen;
296     QPen lwtPen;
297     QLineF objLine;
298     String objRubberMode = "OBJ_RUBBER_OFF";
299     QHash<QString, QPointF> objRubberPoints;
300     QHash<QString, QString> objRubberTexts;
301     int64_t objID;

```



```

309
310     QPointF arcStartPoint;
311     QPointF arcMidPoint;
312     QPointF arcEndPoint;
313
314     bool curved;
315     bool filled;
316     QPainterPath lineStylePath;
317     QPainterPath arrowStylePath;
318     EmbReal arrowStyleAngle;
319     EmbReal arrowStyleLength;
320     EmbReal lineStyleAngle;
321     EmbReal lineStyleLength;
322
323     QPainterPath normalPath;
324
325     QString objText;
326     QString objTextFont;
327     QString objTextJustify;
328     bool objTextBackward;
329     bool objTextUpsideDown;
330     QPainterPath objTextPath;
331
332     std::vector<EmbReal> x_values;
333     std::vector<EmbReal> y_values;
334
335     int gripIndex;
336
337     int Type = OBJ_TYPE_BASE;
338     virtual int type(){ return Type; }
339
340     Geometry(int object_type = OBJ_TYPE_BASE, QGraphicsItem* parent = 0);
341     Geometry(Geometry *obj, QGraphicsItem* parent = 0);
342     Geometry(EmbArc arc, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
343     Geometry(EmbCircle circle, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
344     Geometry(EmbLine line, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
345     Geometry(EmbEllipse ellipse, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
346     Geometry(EmbRect rect, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
347     Geometry(QString str, EmbVector position, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent =
0);
348     Geometry(EmbLine line, int Type_, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent);
349     Geometry(QPainterPath p, int type_, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
350     Geometry(EmbVector pos, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
351
352     void init_arc(EmbArc arc, QRgb rgb, Qt::PenStyle lineType);
353     void init_circle(EmbCircle circle, QRgb rgb, Qt::PenStyle lineType);
354     void init_line(EmbLine line, QRgb rgb, Qt::PenStyle lineType);
355     void init_ellipse(EmbEllipse ellipse, QRgb rgb, Qt::PenStyle lineType);
356     void init_rect(EmbRect rect, QRgb rgb, Qt::PenStyle lineType);
357     void init_text_single(QString str, EmbVector position, QRgb rgb, Qt::PenStyle lineType);
358     void init_path(QPainterPath p, QRgb rgb, Qt::PenStyle lineType);
359     void init_point(EmbVector pos, QRgb rgb, Qt::PenStyle lineType);
360
361     void init(void);
362
363     ~Geometry();
364
365     /* Getters */
366     Qt::PenStyle objectLineType() { return objPen.style(); }
367     EmbReal objectLineWeight() { return lwtPen.widthF(); }
368     QPointF objectRubberPoint(QString key);
369     QString objectRubberText(QString key);
370
371     QPointF objectCenter() { return scenePos(); }
372     QPointF objectPos() { return scenePos(); }
373     EmbReal objectX(){ return scenePos().x(); }
374     EmbReal objectY(){ return scenePos().y(); }
375
376     QPointF objectTopLeft();
377     QPointF objectTopRight();
378     QPointF objectBottomLeft();
379     QPointF objectBottomRight();
380     EmbReal objectArea();
381     QPointF objectStartPoint();
382     QPointF objectMidPoint();
383     QPointF objectEndPoint();
384
385     QRectF rect();
386     void circle_click(Dictionary global, EmbVector v);
387     EmbReal objectWidth();
388     EmbReal objectHeight();
389     EmbReal objectRadiusMajor();
390     EmbReal objectRadiusMinor();
391     EmbReal objectDiameterMajor();
392     EmbReal objectDiameterMinor();
393     QPointF objectEndPoint1();
394     QPointF objectEndPoint2();

```

```

395     EmbReal objectStartAngle();
396     EmbReal objectEndAngle();
397     EmbReal objectArcLength();
398     EmbReal objectChord();
399     EmbReal objectIncludedAngle();
400     bool objectClockwise();
401     EmbReal objectX1() { return objectEndPoint1().x(); }
402     EmbReal objectY1() { return objectEndPoint1().y(); }
403     EmbReal objectX2() { return objectEndPoint2().x(); }
404     EmbReal objectY2() { return objectEndPoint2().y(); }
405     EmbReal objectAngle();
406     QPointF objectDelta() { return objectEndPoint2() - objectEndPoint1(); }
407     EmbReal objectLength() { return objLine.length()*scale(); }
408     EmbReal objectRadius();
409     EmbReal objectDiameter();
410     EmbReal objectCircumference();
411     QPointF objectQuadrant0();
412     QPointF objectQuadrant90();
413     QPointF objectQuadrant180();
414     QPointF objectQuadrant270();
415     QPainterPath objectCopyPath();
416     QPainterPath objectSavePath();
417
418     std::vector<QPainterPath> objectSavePathList() { return subPathList(); }
419     std::vector<QPainterPath> subPathList();
420
421     int findIndex(const QPointF& point);
422
423     void setObjectEndPoint1(EmbVector endPt1);
424     void setObjectEndPoint2(EmbVector endPt2);
425
426     void updatePath();
427     void updatePath(const QPainterPath& p);
428     void updateLeader(void);
429
430     virtual QRectF boundingRect();
431
432     void drawRubberLine(const QLineF& rubLine, QPainter* painter = 0, const char* colorFromScene = 0);
433
434     void updateRubber(QPainter* painter = 0);
435     void vulcanize(void);
436     QPointF mouseSnapPoint(const QPointF& mousePoint);
437     std::vector<QPointF> allGripPoints();
438     void gripEdit(const QPointF& before, const QPointF& after);
439
440     void realRender(QPainter* painter, const QPainterPath& renderPath);
441     void paint(QPainter*, const QStyleOptionGraphicsItem*, QWidget*);
442
443     /* Updaters, todo: combine */
444     void calculateArcData(EmbArc arc);
445     void updateArcRect(EmbReal radius);
446
447     /* Setters */
448     void setObjectPos(const QPointF& point) { setPos(point.x(), point.y()); }
449     void setObjectX(EmbReal x) { setPos(x, objectY()); }
450     void setObjectY(EmbReal y) { setPos(objectX(), y); }
451     void setObjectCenter(EmbVector center);
452     void setObjectCenterX(EmbReal centerX);
453     void setObjectCenterY(EmbReal centerY);
454     void setObjectSize(EmbReal width, EmbReal height);
455     void setObjectRect(EmbReal x, EmbReal y, EmbReal w, EmbReal h);
456     void setRect(const QRectF& r);
457     void setRect(EmbReal x, EmbReal y, EmbReal w, EmbReal h);
458     void setLine(const QLineF& li);
459     void setLine(EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2);
460     void setObjectLineWeight(QString lineWeight);
461     void setObjectRadius(EmbReal radius);
462     void setObjectStartAngle(EmbReal angle);
463     void setObjectEndAngle(EmbReal angle);
464     void setObjectStartPoint(EmbVector point);
465     void setObjectMidPoint(EmbVector point);
466     void setObjectEndPoint(EmbVector point);
467     void setObjectDiameter(EmbReal diameter);
468     void setObjectArea(EmbReal area);
469     void setObjectCircumference(EmbReal circumference);
470     void setObjectPos(EmbReal x, EmbReal y) { setPos(x, y); }
471     void setObjectText(QString str);
472     void setObjectTextFont(QString font);
473     void setObjectTextJustify(QString justify);
474     void setObjectTextSize(EmbReal size);
475     void setObjectTextStyle(bool bold, bool italic, bool under, bool strike, bool over);
476     void setObjectTextBold(bool val);
477     void setObjectTextItalic(bool val);
478     void setObjectTextUnderline(bool val);
479     void setObjectTextStrikeOut(bool val);
480     void setObjectTextOverline(bool val);
481     void setObjectTextBackward(bool val);

```

```

482 void setObjectTextUpsideDown(bool val);
483 void setObjectRadiusMajor(EmbReal radius);
484 void setObjectRadiusMinor(EmbReal radius);
485 void setObjectDiameterMajor(EmbReal diameter);
486 void setObjectDiameterMinor(EmbReal diameter);
487
488 /* Scripted commands, uses the script string in */
489 void script_main(void);
490 void script_click(EmbVector v);
491 void script_context(String str);
492 void script_prompt(String str);
493 };
494
495 class SaveObject : public QObject
496 {
497     Q_OBJECT
498 public:
499     SaveObject(QGraphicsScene* theScene, QObject* parent = 0);
500     ~SaveObject();
501
502     bool save(QString fileName);
503
504     void addArc(EmbPattern* pattern, QGraphicsItem* item);
505     void addBlock(EmbPattern* pattern, QGraphicsItem* item);
506     void addCircle(EmbPattern* pattern, QGraphicsItem* item);
507     void addDimAligned(EmbPattern* pattern, QGraphicsItem* item);
508     void addDimAngular(EmbPattern* pattern, QGraphicsItem* item);
509     void addDimArcLength(EmbPattern* pattern, QGraphicsItem* item);
510     void addDimDiameter(EmbPattern* pattern, QGraphicsItem* item);
511     void addDimLeader(EmbPattern* pattern, QGraphicsItem* item);
512     void addDimLinear(EmbPattern* pattern, QGraphicsItem* item);
513     void addDimOrdinate(EmbPattern* pattern, QGraphicsItem* item);
514     void addDimRadius(EmbPattern* pattern, QGraphicsItem* item);
515     void addEllipse(EmbPattern* pattern, QGraphicsItem* item);
516     void addEllipseArc(EmbPattern* pattern, QGraphicsItem* item);
517     void addGrid(EmbPattern* pattern, QGraphicsItem* item);
518     void addHatch(EmbPattern* pattern, QGraphicsItem* item);
519     void addImage(EmbPattern* pattern, QGraphicsItem* item);
520     void addInfiniteLine(EmbPattern* pattern, QGraphicsItem* item);
521     void addLine(EmbPattern* pattern, QGraphicsItem* item);
522     void addPath(EmbPattern* pattern, QGraphicsItem* item);
523     void addPoint(EmbPattern* pattern, QGraphicsItem* item);
524     void addPolygon(EmbPattern* pattern, QGraphicsItem* item);
525     void addPolyline(EmbPattern* pattern, QGraphicsItem* item);
526     void addRay(EmbPattern* pattern, QGraphicsItem* item);
527     void addRectangle(EmbPattern* pattern, QGraphicsItem* item);
528     void addSlot(EmbPattern* pattern, QGraphicsItem* item);
529     void addSpline(EmbPattern* pattern, QGraphicsItem* item);
530     void addTextMulti(EmbPattern* pattern, QGraphicsItem* item);
531     void addTextSingle(EmbPattern* pattern, QGraphicsItem* item);
532
533     QGraphicsScene* gscene;
534     int formatType;
535
536     void toPolyline(EmbPattern* pattern, const QPointF& objPos, const QPainterPath& objPath, QString
        layer, const QColor& color, QString lineType, QString lineWeight);
537 };
538
539 class Application : public QApplication
540 {
541     Q_OBJECT
542 public:
543     Application(int argc, char **argv);
544     void setMainWin(MainWindow* mainWin) { __mainWin = _mainWin; }
545     MainWindow* __mainWin;
546 protected:
547     virtual bool event(QEvent *e);
548 };
549
550 class CmdPromptInput : public QLineEdit
551 {
552     Q_OBJECT
553 public:
554     CmdPromptInput(QWidget* parent = 0);
555     ~CmdPromptInput() {}
556
557     QString curText;
558     QString defaultPrefix;
559     QString prefix;
560
561     QString lastCmd;
562     QString curCmd;
563     bool cmdActive;
564 };

```

```

579     bool rapidFireEnabled;
580     bool isBlinking;
581
582     void changeFormatting(std::vector<QTextLayout::FormatRange> formats);
583     void clearFormatting();
584     void applyFormatting();
585
586 protected:
587     void contextMenuEvent(QContextMenuEvent *event);
588     bool eventFilter(QObject *obj, QEvent *event);
589
590 signals:
591     void appendHistory(QString txt, int prefixLength);
592
593     //These connect to the CmdPrompt signals
594     void startCommand(QString cmd);
595     void runCommand(QString cmd, QString cmdtxt);
596     void deletePressed();
597     void tabPressed();
598     void escapePressed();
599     void upPressed();
600     void downPressed();
601     void F1Pressed();
602     void F2Pressed();
603     void F3Pressed();
604     void F4Pressed();
605     void F5Pressed();
606     void F6Pressed();
607     void F7Pressed();
608     void F8Pressed();
609     void F9Pressed();
610     void F10Pressed();
611     void F11Pressed();
612     void F12Pressed();
613     void cutPressed();
614     void copyPressed();
615     void pastePressed();
616     void selectAllPressed();
617     void undoPressed();
618     void redoPressed();
619
620     void shiftPressed();
621     void shiftReleased();
622
623     void showSettings();
624
625     void stopBlinking();
626
627 public slots:
628     void endCommand();
629     void processInput(void);
630     void checkSelection();
631     void updateCurrentText(QString txt);
632     void checkEditedText(QString txt);
633     void checkChangedText(QString txt);
634     void checkCursorPosition(int oldpos, int newpos);
635 private slots:
636     void copyClip();
637     void pasteClip();
638 };
639
640 class CmdPromptHistory : public QTextBrowser
641 {
642     Q_OBJECT
643
644 public:
645     CmdPromptHistory(QWidget* parent = 0);
646     ~CmdPromptHistory();
647
648     int tmpHeight;
649     QString applyFormatting(QString txt, int prefixLength);
650
651 protected:
652     void contextMenuEvent(QContextMenuEvent* event);
653
654 public slots:
655     void appendHistory(QString txt, int prefixLength);
656     void startResizeHistory(int y);
657     void stopResizeHistory(int y);
658     void resizeHistory(int y);
659
660 signals:
661     void historyAppended(QString txt);
662 };
663
664 class CmdPromptSplitter : public QSplitter
665 {

```

```

672     Q_OBJECT
673
674 public:
675     CmdPromptSplitter(QWidget* parent = 0);
676     ~CmdPromptSplitter();
677
678 protected:
679     QSplitterHandle* createHandle();
680
681 signals:
682     void pressResizeHistory(int y);
683     void releaseResizeHistory(int y);
684     void moveResizeHistory(int y);
685 };
686
687 class CmdPromptHandle : public QSplitterHandle
688 {
689     Q_OBJECT
690
691 public:
692     CmdPromptHandle(Qt::Orientation orientation, QSplitter* parent);
693     ~CmdPromptHandle();
694
695     int pressY;
696     int releaseY;
697     int moveY;
698
699 protected:
700     void mousePressEvent(QMouseEvent* e);
701     void mouseReleaseEvent(QMouseEvent* e);
702     void mouseMoveEvent(QMouseEvent* e);
703
704 signals:
705     void handlePressed(int y);
706     void handleReleased(int y);
707     void handleMoved(int y);
708 };
709
710 class CmdPrompt : public QWidget
711 {
712     Q_OBJECT
713
714 public:
715     CmdPrompt(QWidget* parent = 0);
716     ~CmdPrompt();
717
718     CmdPromptInput* promptInput;
719     CmdPromptHistory* promptHistory;
720     QVBoxLayout* promptVBoxLayout;
721     QFrame* promptDivider;
722
723     CmdPromptSplitter* promptSplitter;
724
725     QHash<QString, QString>* styleHash;
726     void updateStyle();
727     QTimer* blinkTimer;
728     bool blinkState;
729
730 public slots:
731     void setCurrentText(QString txt) {
732         promptInput->curText = promptInput->prefix + txt;
733         promptInput->setText(promptInput->curText);
734     }
735     void setHistory(QString txt) {
736         promptHistory->setHtml(txt);
737         promptHistory->moveCursor(QTextCursor::End, QTextCursor::MoveAnchor);
738     }
739     void setPrefix(QString txt);
740     void appendHistory(QString txt);
741
742     void alert(QString txt);
743
744     void startBlinking();
745     void stopBlinking();
746     void blink();
747
748     void setPromptTextColor(const QColor&);
749     void setPromptBackgroundColor(const QColor&);
750     void setPromptFontFamily(QString );
751     void setPromptFontStyle(QString );
752     void setPromptFontSize(int);
753
754     void floatingChanged(bool);
755
756     void saveHistory(QString fileName, bool html);
757
758 signals:

```

```

765     void appendTheHistory(QString txt, int prefixLength);
766
767     //For connecting outside of command prompt
768     void startCommand(QString cmd);
769     void runCommand(QString cmd, QString cmdtxt);
770     void deletePressed();
771     void tabPressed();
772     void escapePressed();
773     void upPressed();
774     void downPressed();
775     void F1Pressed();
776     void F2Pressed();
777     void F3Pressed();
778     void F4Pressed();
779     void F5Pressed();
780     void F6Pressed();
781     void F7Pressed();
782     void F8Pressed();
783     void F9Pressed();
784     void F10Pressed();
785     void F11Pressed();
786     void F12Pressed();
787     void cutPressed();
788     void copyPressed();
789     void pastePressed();
790     void selectAllPressed();
791     void undoPressed();
792     void redoPressed();
793
794     void shiftPressed();
795     void shiftReleased();
796
797     void showSettings();
798
799     void historyAppended(QString txt);
800 };
801
802 class EmbDetailsDialog : public QDialog
803 {
804     Q_OBJECT
805 public:
806     EmbDetailsDialog(QGraphicsScene* theScene, QWidget *parent = 0);
807     ~EmbDetailsDialog();
808
809     QWidget* mainWidget;
810
811     void getInfo();
812     QWidget* createMainWidget();
813     QWidget* createHistogram();
814
815     QDialogButtonBox* buttonBox;
816
817     uint32_t stitchesTotal;
818     uint32_t stitchesReal;
819     uint32_t stitchesJump;
820     uint32_t stitchesTrim;
821     uint32_t colorTotal;
822     uint32_t colorChanges;
823
824     QRectF boundingRect;
825 };
826
827 class ImageWidget : public QWidget
828 {
829     Q_OBJECT
830 public:
831     QImage img;
832     ImageWidget(QString filename, QWidget* parent = 0);
833     ~ImageWidget();
834
835     bool load(QString fileName);
836     bool save(QString fileName);
837
838 protected:
839     void paintEvent(QPaintEvent* event);
840 };
841
842 class LayerManager : public QDialog
843 {
844     Q_OBJECT
845 public:
846     QStandardItemModel* layerModel;
847     QSortFilterProxyModel* layerModelSorted;
848     QTreeView* treeView;

```

```

861
862     LayerManager(QWidget *parent = 0);
863     ~LayerManager();
864
865     void addLayer(QString name, const bool visible, const bool frozen,
866                  const EmbReal zValue, const QRgb color, QString lineType,
867                  QString lineWeight, const bool print);
868 };
869
870 class MainWindow: public QMainWindow
871 {
872     Q_OBJECT
873
874 public:
875     MainWindow();
876     ~MainWindow();
877
878     MdiWindow* activeMdiWindow();
879     QUndoStack* activeUndoStack();
880
881     void setUndoCleanIcon(bool opened);
882
883     virtual void updateMenuToolbarStatusbar();
884
885     std::vector<QGraphicsItem*> cutCopyObjectList;
886
887     QString formatFilterOpen;
888     QString formatFilterSave;
889
890     bool isCommandActive() { return prompt->promptInput->cmdActive; }
891     QString activeCommand() { return prompt->promptInput->curCmd; }
892     QIcon create_icon(QString stub);
893     void create_toolbar(String toolbar, String label, QStringList entries);
894
895     QString platformString();
896
897 public slots:
898
899     void onCloseWindow();
900     virtual void onCloseMdiWin(MdiWindow*);
901
902     void recentMenuAboutToShow();
903
904     void onWindowActivated(QMdiSubWindow* w);
905     void windowMenuAboutToShow();
906     void windowMenuActivated( bool checked/*int id*/ );
907
908     void updateAllViewScrollBars(bool val);
909     void updateAllViewCrossHairColors(QRgb color);
910     void updateAllViewBackgroundColors(QRgb color);
911     void updateAllViewSelectBoxColors(QRgb colorL, QRgb fillL, QRgb colorR, QRgb fillR, int alpha);
912     void updateAllViewGridColors(QRgb color);
913     void updateAllViewRulerColors(QRgb color);
914
915     void updatePickAddMode(bool val);
916     void pickAddModeToggled();
917
918     void settingsPrompt();
919
920 protected:
921     virtual void resizeEvent(QResizeEvent*);
922     void closeEvent(QCloseEvent *event);
923     QAction* getFileSeparator();
924     void loadFormats();
925
926     bool shiftKeyPressedState;
927
928     QByteArray layoutState;
929
930     int numOfDocs;
931     int docIndex;
932
933     std::vector<MdiWindow*> listMdiWin;
934     QMdiSubWindow* findMdiWindow(String fileName);
935
936     QAction* myFileSeparator;
937
938     void createAllActions();
939     void createAllMenus();
940     void createAllToolbars();
941
942     // Selectors
943     QComboBox* layerSelector;
944     QComboBox* colorSelector;
945     QComboBox* linetypeSelector;
946     QComboBox* linewidthSelector;
947     QFontComboBox* textFontSelector;

```

```

951     QComboBox* textSizeSelector;
952
953 private slots:
954     void hideUnimplemented();
955
956 public slots:
957     void stub_testing();
958
959     void promptHistoryAppended(QString txt);
960     void logPromptInput(QString txt);
961     void promptInputPrevious();
962     void promptInputNext();
963
964     void about(void);
965     void tipOfTheDay(void);
966
967     void newFile();
968     void openFile(bool recent = false, String recentFile = "");
969     void openFilesSelected(StringList files);
970     void openrecentfile();
971     void savefile();
972     void saveasfile();
973     void quit();
974     void checkForUpdates();
975     // Help Menu
976     void buttonTipOfTheDayClicked(int);
977
978     void closeToolBar(QAction*);
979     void floatingChangedToolBar(bool);
980
981     void toggleGrid();
982     void toggleRuler();
983     void toggleLwt();
984
985     // Icons
986     void iconResize(int iconSize);
987
988     //Selectors
989     void layerSelectorIndexChanged(int index);
990     void colorSelectorIndexChanged(int index);
991     void linetypeSelectorIndexChanged(int index);
992     void linewidthSelectorIndexChanged(int index);
993     void textFontSelectorCurrentFontChanged(const QFont& font);
994     void textSizeSelectorIndexChanged(int index);
995
996     void setTextFont(QString str);
997     void setTextSize(EmbReal num);
998
999     QString getCurrentLayer();
1000     QRgb getCurrentColor();
1001     QString getCurrentLineType();
1002     QString getCurrentLineWeight();
1003
1004     bool isShiftPressed();
1005     void setShiftPressed();
1006     void setShiftReleased();
1007
1008     void deletePressed();
1009     void escapePressed();
1010 };
1011
1012 class MdiWindow: public QMdiSubWindow
1013 {
1014     Q_OBJECT
1015
1016 public:
1017     MdiWindow(const int theIndex, QMdiArea* parent, Qt::WindowFlags wflags);
1018     ~MdiWindow();
1019
1020     QMdiArea* mdiArea;
1021     QGraphicsScene* gscene;
1022     View* gview;
1023
1024     bool fileWasLoaded;
1025
1026     QString promptHistory;
1027     std::vector<QString> promptInputList;
1028     int promptInputNum;
1029
1030     QPrinter printer;
1031
1032     QString curFile;
1033     void setCurrentFile(QString fileName);
1034
1035     int myIndex;
1036
1037     QString curLayer;

```



```

1038     QRgb curColor;
1039     QString curLineType;
1040     QString curLineWeight;
1041
1042     void promptInputPrevNext(bool prev);
1043
1044     virtual QSize sizeHint();
1045     QString getShortCurrentFile();
1046     void designDetails();
1047     bool loadFile(QString fileName);
1048     bool saveFile(QString fileName);
1049 signals:
1050     void sendCloseMdiWin(MdiWindow*);
1051
1052 public slots:
1053     void closeEvent(QCloseEvent* e);
1054     void onWindowActivated();
1055
1056     void currentLayerChanged(QString layer);
1057     void currentColorChanged(const QRgb& color);
1058     void currentLinetypeChanged(QString type);
1059     void currentLineweightChanged(QString weight);
1060
1061     void updateColorLinetypeLineweight();
1062     void deletePressed();
1063     void escapePressed();
1064
1065     void showViewScrollBars(bool val);
1066     void setViewCrossHairColor(QRgb color);
1067     void setViewBackgroundColor(QRgb color);
1068     void setViewSelectBoxColors(QRgb colorL, QRgb fillL, QRgb colorR, QRgb fillR, int alpha);
1069     void setViewGridColor(QRgb color);
1070     void setViewRulerColor(QRgb color);
1071
1072     void print();
1073     void saveBMC();
1074
1075     void promptHistoryAppended(QString txt);
1076     void logPromptInput(QString txt);
1077     void promptInputPrevious();
1078     void promptInputNext();
1079 };
1080
1081 class MdiArea : public QMdiArea
1082 {
1083     Q_OBJECT
1084
1085 public:
1086     bool useLogo;
1087     bool useTexture;
1088     bool useColor;
1089
1090     QPixmap bgLogo;
1091     QPixmap bgTexture;
1092     QColor bgColor;
1093
1094     void zoomExtentsAllSubWindows();
1095     void forceRepaint();
1096
1097     MdiArea(QWidget* parent = 0);
1098     ~MdiArea();
1099
1100     void useBackgroundLogo(bool use);
1101     void useBackgroundTexture(bool use);
1102     void useBackgroundColor(bool use);
1103
1104     void setBackgroundLogo(QString fileName);
1105     void setBackgroundTexture(QString fileName);
1106     void setBackgroundColor(const QColor& color);
1107
1108 public slots:
1109     void cascade();
1110     void tile();
1111 protected:
1112     virtual void mouseDoubleClickEvent(QMouseEvent* e);
1113     virtual void paintEvent(QPaintEvent* e);
1114 };
1115
1116 class PreviewDialog : public QFileDialog
1117 {
1118     Q_OBJECT
1119
1120 public:
1121     PreviewDialog(QWidget* parent = 0,
1122         QString caption = QString(),
1123         QString directory = QString(),
1124         QString filter = QString());

```

```

1131     ~PreviewDialog();
1132
1133     ImageWidget* imgWidget;
1134 };
1135
1136
1137 class PropertyEditor : public QDockWidget
1138 {
1139     Q_OBJECT
1140
1141 public:
1142     PropertyEditor(QString iconDirectory = QString(), bool pickAddMode = true, QWidget* widgetToFocus
1143 = 0, QWidget* parent = 0); //, Qt::WindowFlags flags = 0);
1144     ~PropertyEditor();
1145
1146     QWidget* focusWidget;
1147
1148     QString iconDir;
1149     int iconSize;
1150     Qt::ToolButtonStyle propertyEditorButtonStyle;
1151
1152     bool pickAdd;
1153
1154     std::vector<QGraphicsItem*> selectedItemList;
1155
1156     QToolButton* createToolButton(QString iconName, QString txt);
1157     QLineEdit* createLineEdit(QString validatorType = QString(), bool readOnly = false);
1158
1159     int precisionAngle;
1160     int precisionLength;
1161
1162     void updateLineEditStrIfVaries(QLineEdit* lineEdit, QString str);
1163     void updateLineEditNumIfVaries(QLineEdit* lineEdit, EmbReal num, bool useAnglePrecision);
1164     void updateFontComboBoxStrIfVaries(QFontComboBox* fontComboBox, QString str);
1165     void updateComboBoxStrIfVaries(QComboBox* comboBox, QString str, QStringList strList);
1166     void updateComboBoxBoolIfVaries(QComboBox* comboBox, bool val, bool yesOrNoText);
1167
1168     QSignalMapper* signalMapper;
1169     void mapSignal(QObject* fieldObj, QString name, QVariant value);
1170
1171     // Selection
1172     // =====
1173     QComboBox* createComboBoxSelected();
1174     QToolButton* createToolButtonQSelect();
1175     QToolButton* createToolButtonPickAdd();
1176
1177     QComboBox* comboBoxSelected;
1178     QToolButton* toolButtonQSelect;
1179     QToolButton* toolButtonPickAdd;
1180
1181     //TODO: Alphabetic/Categorized TabWidget
1182
1183     void createGroupBox(String group_box_key, const char *title);
1184
1185 protected:
1186     bool eventFilter(QObject *obj, QEvent *event);
1187
1188 signals:
1189     void pickAddModeToggled();
1190
1191 public slots:
1192     void setSelectedItems(std::vector<QGraphicsItem*> itemList);
1193     void updatePickAddModeButton(bool pickAddMode);
1194
1195 private slots:
1196     void fieldEdited(QObject* fieldObj);
1197     void showGroups(int objType);
1198     void showOneType(int index);
1199     void hideAllGroups();
1200     void clearAllFields();
1201     void togglePickAddMode();
1202 };
1203
1204 class SelectBox : public QRubberBand
1205 {
1206     Q_OBJECT
1207
1208 public:
1209     SelectBox(Shape s, QWidget* parent = 0);
1210
1211     QColor leftBrushColor;
1212     QColor rightBrushColor;
1213     QColor leftPenColor;
1214     QColor rightPenColor;
1215     uint8_t alpha;
1216

```

```

1217     QBrush dirBrush;
1218     QBrush leftBrush;
1219     QBrush rightBrush;
1220
1221     QPen dirPen;
1222     QPen leftPen;
1223     QPen rightPen;
1224
1225     bool boxDir;
1226
1227     void forceRepaint();
1228
1229 public slots:
1230     void setDirection(int dir);
1231     void setColors(const QColor& colorL, const QColor& fillL, const QColor& colorR, const QColor&
        fillR, int newAlpha);
1232
1233 protected:
1234     void paintEvent(QPaintEvent*);
1235 };
1236
1240 class Settings_Dialog : public QDialog
1241 {
1242     Q_OBJECT
1243
1244 public:
1245     Settings_Dialog(QString showTab = QString(), QWidget *parent = 0);
1246     ~Settings_Dialog();
1247
1248     QTabWidget* tabWidget;
1249
1250     QWidget* createTabGeneral();
1251     QWidget* createTabFilesPaths();
1252     QWidget* createTabDisplay();
1253     QWidget* createTabPrompt();
1254     QWidget* createTabOpenSave();
1255     QWidget* createTabPrinting();
1256     QWidget* createTabSnap();
1257     QWidget* createTabGridRuler();
1258     QWidget* createTabOrthoPolar();
1259     QWidget* createTabQuickSnap();
1260     QWidget* createTabQuickTrack();
1261     QWidget* createTabLineWeight();
1262     QWidget* createTabSelection();
1263
1264     QDialogButtonBox* buttonBox;
1265
1266     void addColorsToComboBox(QComboBox* comboBox);
1267
1268     void create_float_spinbox(
1269         QGroupBox *gb,
1270         QGridLayout* gridLayout,
1271         const char *label_in,
1272         EmbReal single_step,
1273         EmbReal lower,
1274         EmbReal upper,
1275         String,
1276         int row);
1277     QCheckBox* create_checkbox(QGroupBox *groupbox, String label);
1278
1279 private slots:
1280     void comboBoxIconSizeCurrentIndexChanged(int);
1281     void checkBoxGeneralMdiBGUseLogoStateChanged(int);
1282     void chooseGeneralMdiBackgroundLogo();
1283     void checkBoxGeneralMdiBGUseTextureStateChanged(int);
1284     void chooseGeneralMdiBackgroundTexture();
1285     void checkBoxGeneralMdiBGUseColorStateChanged(int);
1286     void chooseGeneralMdiBackgroundColor();
1287     void currentGeneralMdiBackgroundColorChanged(const QColor&);
1288     void checkBoxShowScrollBarsStateChanged(int);
1289     void comboBoxScrollBarWidgetCurrentIndexChanged(int);
1290     void chooseDisplayCrossHairColor();
1291     void currentDisplayCrossHairColorChanged(const QColor&);
1292     void chooseDisplayBackgroundColor();
1293     void currentDisplayBackgroundColorChanged(const QColor&);
1294     void chooseDisplaySelectBoxLeftColor();
1295     void currentDisplaySelectBoxLeftColorChanged(const QColor&);
1296     void chooseDisplaySelectBoxLeftFill();
1297     void currentDisplaySelectBoxLeftFillChanged(const QColor&);
1298     void chooseDisplaySelectBoxRightColor();
1299     void currentDisplaySelectBoxRightColorChanged(const QColor&);
1300     void chooseDisplaySelectBoxRightFill();
1301     void currentDisplaySelectBoxRightFillChanged(const QColor&);
1302     void comboBoxSelectionCoolGripColorCurrentIndexChanged(int index);
1303     void comboBoxSelectionHotGripColorCurrentIndexChanged(int index);
1304     void spinBoxDisplaySelectBoxAlphaValueChanged(int);
1305     void choosePromptTextColor();

```

```

1306     void currentPromptTextColorChanged(const QColor&);
1307     void choosePromptBackgroundColor();
1308     void currentPromptBackgroundColorChanged(const QColor&);
1309     void comboBoxPromptFontFamilyCurrentIndexChanged(QString );
1310     void comboBoxPromptFontStyleCurrentIndexChanged(QString );
1311     void spinBoxPromptFontSizeValueChanged(int);
1312     void checkBoxPromptSaveHistoryAsHtmlStateChanged(int);
1313     void checkBoxCustomFilterStateChanged(int);
1314     void buttonCustomFilterSelectAllClicked();
1315     void buttonCustomFilterClearAllClicked();
1316     void checkBoxGridColorMatchCrossHairStateChanged(int);
1317     void chooseGridColor();
1318     void currentGridColorChanged(const QColor&);
1319     void checkBoxGridLoadFromFileStateChanged(int);
1320     void comboBoxGridTypeCurrentIndexChanged(QString );
1321     void checkBoxGridCenterOnOriginStateChanged(int);
1322     void checkBoxRulerShowOnLoadStateChanged(int);
1323     void comboBoxRulerMetricCurrentIndexChanged(int);
1324     void chooseRulerColor();
1325     void currentRulerColorChanged(const QColor&);
1326     void spinBoxRulerPixelSizeValueChanged(double);
1327     void buttonQSnapSelectAllClicked();
1328     void buttonQSnapClearAllClicked();
1329     void comboBoxQSnapLocatorColorCurrentIndexChanged(int);
1330     void checkBoxLwtShowLwtStateChanged(int);
1331     void checkBoxLwtRealRenderStateChanged(int);
1332
1333     void acceptChanges();
1334     void rejectChanges();
1335
1336 signals:
1337     void buttonCustomFilterSelectAll(bool);
1338     void buttonCustomFilterClearAll(bool);
1339     void buttonQSnapSelectAll(bool);
1340     void buttonQSnapClearAll(bool);
1341 };
1342
1343 class StatusBar : public QStatusBar
1344 {
1345     Q_OBJECT
1346 public:
1347     StatusBar(QWidget* parent = 0);
1348     std::unordered_map<String, QToolButton*> buttons;
1349     QLabel* statusBarMouseCoord;
1350     void setMouseCoord(EmbReal x, EmbReal y);
1351     void context_menu_action(QToolButton *button, const char *icon, const char *label, QMenu *menu,
1352         String setting_page);
1353     void toggle(String key, bool on);
1354     void context_menu_event(QContextMenuEvent *event, QToolButton *button);
1355 };
1356
1357 class UndoEditor : public QDockWidget
1358 {
1359     Q_OBJECT
1360 public:
1361     UndoEditor(QString iconDirectory = QString(), QWidget* widgetToFocus = 0, QWidget* parent = 0);
1362     //, Qt::WindowFlags flags = 0);
1363     ~UndoEditor();
1364
1365     void addStack(QUndoStack* stack);
1366
1367     bool canUndo();
1368     bool canRedo();
1369
1370     QWidget* focusWidget;
1371
1372     QString iconDir;
1373     int iconSize;
1374
1375     QUndoGroup* undoGroup;
1376     QUndoView* undoView;
1377
1378     QString undoText();
1379     QString redoText();
1380 protected:
1381
1382 public slots:
1383     void undo();
1384     void redo();
1385
1386     void updateCleanIcon(bool opened);
1387 };
1388
1389 class UndoableCommand : public QUndoCommand
1390 {

```

```

1400 public:
1401     UndoableCommand(String command, QString text, Geometry* obj, View* v, QUndoCommand* parent = 0);
1402     UndoableCommand(EmbVector d, QString text, Geometry* obj, View* v, QUndoCommand* parent = 0);
1403     UndoableCommand(String command, EmbVector pivot, EmbReal angle, QString text, Geometry* obj, View*
v, QUndoCommand* parent = 0);
1404     UndoableCommand(QString type, View* v, QUndoCommand* parent = 0);
1405     UndoableCommand(const QPointF beforePoint, const QPointF afterPoint, QString text, Geometry* obj,
View* v, QUndoCommand* parent = 0);
1406     UndoableCommand(EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, QString text, Geometry* obj, View*
v, QUndoCommand* parent = 0);
1407
1408     int id(){ return 1234; }
1409     bool mergeWith(const QUndoCommand* command);
1410     void undo();
1411     void redo();
1412     void mirror();
1413     void rotate(EmbVector pivot, EmbReal rot);
1414
1415     Geometry* object;
1416     View* gview;
1417     String command;
1418     EmbVector delta;
1419     EmbVector pivot;
1420     QPointF before;
1421     QPointF after;
1422     EmbReal angle;
1423     EmbReal factor;
1424     QString navType;
1425     QTransform fromTransform;
1426     QTransform toTransform;
1427     QPointF fromCenter;
1428     QPointF toCenter;
1429     QLineF mirrorLine;
1430     bool done;
1431 };
1432
1433 class View : public QGraphicsView
1434 {
1435     Q_OBJECT
1436
1437 public:
1438     View(QGraphicsScene* theScene, QWidget* parent);
1439     ~View();
1440
1441     Dictionary state;
1442
1443     std::vector<QGraphicsItem*> selected_items();
1444
1445     bool allowZoomIn();
1446     bool allowZoomOut();
1447
1448     QColor gridColor;
1449     QPainterPath gridPath;
1450     QPainterPath originPath;
1451     bool rulerMetric;
1452     QColor rulerColor;
1453     uint8_t rulerPixelSize;
1454
1455     bool grippingActive;
1456     bool rapidMoveActive;
1457     bool previewActive;
1458     bool pastingActive;
1459     bool movingActive;
1460     bool selectingActive;
1461     bool zoomWindowActive;
1462     bool panningRealTimeActive;
1463     bool panningPointActive;
1464     bool panningActive;
1465     bool qSnapActive;
1466     bool qSnapToggle;
1467
1468     Geometry* gripBaseObj;
1469     Geometry* tempBaseObj;
1470
1471     QGraphicsScene* gscene;
1472     QUndoStack* undoStack;
1473
1474     SelectBox* selectBox;
1475     QPointF scenePressPoint;
1476     QPoint pressPoint;
1477     QPointF sceneMovePoint;
1478     QPoint movePoint;
1479     QPointF sceneReleasePoint;
1480     QPoint releasePoint;
1481     QPointF sceneGripPoint;
1482
1483     void updateMouseCoords(int x, int y);

```

```
1487     QPoint viewMousePoint;
1488     QPointF sceneMousePoint;
1489     QRgb qsnapLocatorColor;
1490     uint8_t qsnapLocatorSize;
1491     uint8_t qsnapApertureSize;
1492     QRgb gripColorCool;
1493     QRgb gripColorHot;
1494     uint8_t gripSize;
1495     uint8_t pickBoxSize;
1496     QRgb crosshairColor;
1497     uint32_t crosshairSize;
1498
1499     void recalculateLimits();
1500     void zoomToPoint(const QPoint& mousePoint, int zoomDir);
1501     void centerAt(const QPointF& centerPoint);
1502     QPointF center() { return mapToScene(rect().center()); }
1503
1504     QUndoStack* getUndoStack() { return undoStack; }
1505     void addObject(Geometry* obj);
1506     void deleteObject(Geometry* obj);
1507     void vulcanizeObject(Geometry* obj);
1508
1509 public slots:
1510     void zoomIn();
1511     void zoomOut();
1512     void zoomWindow();
1513     void zoomSelected();
1514     void zoomExtents();
1515     void panRealTime();
1516     void panPoint();
1517     void panLeft();
1518     void panRight();
1519     void panUp();
1520     void panDown();
1521     void selectAll();
1522     void selectionChanged();
1523     void clearSelection();
1524     void deleteSelected();
1525     void moveSelected(EmbReal dx, EmbReal dy);
1526     void cut();
1527     void copy();
1528     void paste();
1529     void repeatAction();
1530     void moveAction();
1531     void scaleAction();
1532     void scaleSelected(EmbReal x, EmbReal y, EmbReal factor);
1533     void rotateAction();
1534     void rotateSelected(EmbReal x, EmbReal y, EmbReal rot);
1535     void mirrorSelected(EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2);
1536     int numSelected();
1537
1538     void deletePressed();
1539     void escapePressed();
1540
1541     void cornerButtonClicked();
1542
1543     void showScrollBars(bool val);
1544     void setCornerButton();
1545     void setCrossHairColor(QRgb color);
1546     void setCrossHairSize(uint8_t percent);
1547     void setBackgroundColor(QRgb color);
1548     void setSelectBoxColors(QRgb colorL, QRgb fillL, QRgb colorR, QRgb fillR, int alpha);
1549     void toggleSnap(bool on);
1550     void toggleGrid(bool on);
1551     void toggleRuler(bool on);
1552     void toggleOrtho(bool on);
1553     void togglePolar(bool on);
1554     void toggleQSnap(bool on);
1555     void toggleQTrack(bool on);
1556     void toggleLwt(bool on);
1557     void toggleReal(bool on);
1558     bool isLwtEnabled();
1559     bool isRealEnabled();
1560
1561     void setGridColor(QRgb color);
1562     void createGrid(QString gridType);
1563     void setRulerColor(QRgb color);
1564
1565     void previewOn(String clone, String mode, EmbReal x, EmbReal y, EmbReal data);
1566     void previewOff();
1567
1568     bool allowRubber();
1569     void addToRubberRoom(QGraphicsItem* item);
1570     void vulcanizeRubberRoom();
1571     void clearRubberRoom();
1572     void spareRubber(int64_t id);
1573     void setRubberMode(String mode);
```

```

1574     void setRubberPoint(QString key, const QPointF& point);
1575     void setRubberText(QString key, QString txt);
1576
1577 protected:
1578     void mouseDoubleClickEvent(QMouseEvent* event);
1579     void mousePressEvent(QMouseEvent* event);
1580     void mouseMoveEvent(QMouseEvent* event);
1581     void mouseReleaseEvent(QMouseEvent* event);
1582     void wheelEvent(QWheelEvent* event);
1583     void contextMenuEvent(QContextMenuEvent* event);
1584     void drawBackground(QPainter* painter, const QRectF& rect);
1585     void drawForeground(QPainter* painter, const QRectF& rect);
1586     void enterEvent(QEvent* event);
1587
1588 private:
1589     QHash<int64_t, QGraphicsItem*> hashDeletedObjects;
1590
1591     QStringList spareRubberList;
1592
1593     void createGridRect();
1594     void createGridPolar();
1595     void createGridIso();
1596     void createOrigin();
1597
1598     void loadRulerSettings();
1599
1600     bool willUnderflowInt32(int64_t a, int64_t b);
1601     bool willOverflowInt32(int64_t a, int64_t b);
1602     int roundToMultiple(bool roundUp, int numToRound, int multiple);
1603     QPainterPath createRulerTextPath(EmbVector position, QString str, EmbReal height);
1604
1605     QList<QGraphicsItem*> previewObjectList;
1606     QGraphicsItemGroup* previewObjectItemGroup;
1607     QPointF previewPoint;
1608     EmbReal previewData;
1609     String previewMode;
1610
1611     std::vector<QGraphicsItem*> createObjectList(std::vector<QGraphicsItem*> list);
1612     QPointF cutCopyMousePoint;
1613     QGraphicsItemGroup* pasteObjectItemGroup;
1614     QPointF pasteDelta;
1615
1616     std::vector<QGraphicsItem*> rubberRoomList;
1617
1618     void copySelected();
1619
1620     void startGripping(Geometry* obj);
1621     void stopGripping(bool accept = false);
1622
1623     void panStart(const QPoint& point);
1624     int panDistance;
1625     int panStartX;
1626     int panStartY;
1627
1628     void alignScenePointWithViewPoint(const QPointF& scenePoint, const QPoint& viewPoint);
1629 };
1630
1631 #endif

```

## 18.8 embroidermodder2/imagewidget.cpp File Reference

```
#include "embroidermodder.h"
```

## 18.9 embroidermodder2/interface.cpp File Reference

```
#include "embroidermodder.h"
```

### Functions

- QString [translate\\_str](#) (const char \*str)
- Node [node\\_bool](#) (bool value)
  - set\_node*
- Node [node\\_int](#) (int32\_t value)
  - create\_node*
- Node [node\\_uint](#) (uint32\_t value)

- create\_node*
- [Node node\\_real](#) ([EmbReal](#) value)
- set\_node*
- [Node node\\_str](#) ([String](#) value)
- set\_node*
- [Node node\\_qstr](#) ([QString](#) value)
- set\_node*
- [Node node\\_str\\_list](#) ([StringList](#) value)
- set\_node*
- [bool get\\_bool](#) ([Dictionary](#) d, [String](#) key)
- [int get\\_int](#) ([Dictionary](#) d, [String](#) key)
- [uint32\\_t get\\_uint](#) ([Dictionary](#) d, [String](#) key)
- [EmbReal get\\_real](#) ([Dictionary](#) d, [String](#) key)
- [String get\\_str](#) ([Dictionary](#) d, [String](#) key)
- [QString get\\_qstr](#) ([Dictionary](#) d, [String](#) key)
- [StringList get\\_str\\_list](#) ([Dictionary](#) d, [String](#) key)
- [StringList to\\_string\\_vector](#) ([QStringList](#) list)
- to\_string\_vector*
- [StringList tokenize](#) ([String](#) str, const char delim)
- tokenize*
- [QPointF to\\_QPointF](#) ([EmbVector](#) a)
- [EmbVector to\\_EmbVector](#) ([QPointF](#) a)
- [EmbVector operator+](#) ([EmbVector](#) a, [EmbVector](#) b)
- operator + Wrapper for embVector\_add to use the syntax a + b.*
- [EmbVector operator-](#) ([EmbVector](#) a, [EmbVector](#) b)
- operator - Wrapper for embVector\_subtract to use the syntax a - b.*
- [EmbVector operator\\*](#) ([EmbVector](#) v, [EmbReal](#) s)
- operator \**
- [EmbReal radians\\_\\_](#) ([EmbReal](#) degrees)
- radians\_\_*
- [EmbReal degrees\\_\\_](#) ([EmbReal](#) radian)
- degrees\_\_*
- [std::vector< QGraphicsItem \\* > to\\_vector](#) ([QList< QGraphicsItem \\* >](#) list)
- to\_vector*
- [QList< QGraphicsItem \\* > to\\_qlist](#) ([std::vector< QGraphicsItem \\* >](#) list)
- to\_qlist*
- void [debug\\_message](#) ([std::string](#) msg)
- debug\_message*
- [std::vector< float > get\\_n\\_reals](#) ([StringList](#) list, int n, int \*offset)
- [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)



### 18.9.1 Detailed Description

For wrappers to the Qt internals.

To help reduce reliance on Qt, only the functions wrap the Qt functions have a wrapper here. Ideally we could move some of the Qt headers here.

### 18.9.2 Function Documentation

**18.9.2.1 add\_to\_path()** QPainterPath add\_to\_path (   
     QPainterPath *path*,   
     EmbVector *scale*,   
     String *command* )

**18.9.2.2 debug\_message()** void debug\_message (   
     std::string *msg* )   
 debug\_message

Parameters

<i>msg</i>	
------------	--

**18.9.2.3 degrees\_\_()** EmbReal degrees\_\_ (   
     EmbReal *radian* )   
 degrees\_\_

Parameters

<i>radian</i>	
---------------	--

Returns

**18.9.2.4 get\_bool()** bool get\_bool (   
     Dictionary *d*,   
     String *key* )

**18.9.2.5 get\_int()** int get\_int (   
     Dictionary *d*,   
     String *key* )

**18.9.2.6 get\_n\_reals()** std::vector< float > get\_n\_reals (   
     StringList *list*,   
     int *n*,   
     int \* *offset* )

Utility function for add\_to\_path.

**18.9.2.7 get\_qstr()** `QString get_qstr (`  
    `Dictionary d,`  
    `String key )`

**18.9.2.8 get\_real()** `EmbReal get_real (`  
    `Dictionary d,`  
    `String key )`

**18.9.2.9 get\_str()** `String get_str (`  
    `Dictionary d,`  
    `String key )`

**18.9.2.10 get\_str\_list()** `StringList get_str_list (`  
    `Dictionary d,`  
    `String key )`

**18.9.2.11 get\_uint()** `uint32_t get_uint (`  
    `Dictionary d,`  
    `String key )`

**18.9.2.12 make\_checkbox()** `QCheckBox * make_checkbox (`  
    `QGroupBox * gb,`  
    `String dictionary,`  
    `const char * label,`  
    `const char * icon,`  
    `String key )`

**18.9.2.13 make\_spinbox()** `QDoubleSpinBox * make_spinbox (`  
    `QGroupBox * gb,`  
    `String dictionary,`  
    `QString object_name,`  
    `EmbReal single_step,`  
    `EmbReal lower,`  
    `EmbReal upper,`  
    `String key )`

**18.9.2.14 make\_ui\_element()** `void make_ui_element (`  
    `Dictionary description )`

**18.9.2.15 node\_bool()** `Node node_bool (`  
    `bool value )`

set\_node

Parameters

<i>node</i>	
<i>value</i>	

**18.9.2.16 node\_int()** `Node node_int (`  
                  `int32_t value )`

`create_node`

Parameters

<i>mode</i>	
-------------	--

Returns

**18.9.2.17 node\_qstr()** `Node node_qstr (`  
                  `QString value )`

`set_node`

Parameters

<i>node</i>	
<i>value</i>	

**18.9.2.18 node\_real()** `Node node_real (`  
                  `EmbReal value )`

`set_node`

Parameters

<i>node</i>	
<i>value</i>	

**18.9.2.19 node\_str()** `Node node_str (`  
                  `String value )`

`set_node`

Parameters

<i>node</i>	
<i>value</i>	

**18.9.2.20 node\_str\_list()** `Node node_str_list (`  
                  `StringList value )`

`set_node`

Parameters

<i>node</i>	
<i>value</i>	

**18.9.2.21 node\_uint()** `Node node_uint (`  
    `uint32_t value )`

`create_node`

Parameters

<i>mode</i>	
-------------	--

Returns

**18.9.2.22 operator\*()** `EmbVector operator* (`  
    `EmbVector v,`  
    `EmbReal s )`

`operator *`

Parameters

<i>v</i>	
<i>s</i>	

Returns

**18.9.2.23 operator+()** `EmbVector operator+ (`  
    `EmbVector a,`  
    `EmbVector b )`

`operator +` Wrapper for `embVector_add` to use the syntax `a + b`.

**18.9.2.24 operator-()** `EmbVector operator- (`  
    `EmbVector a,`  
    `EmbVector b )`

`operator -` Wrapper for `embVector_subtract` to use the syntax `a - b`.

**18.9.2.25 radians\_\_()** `EmbReal radians__ (`  
    `EmbReal degrees )`

`radians__`

Parameters

<i>degrees</i>	
----------------	--

Returns

**18.9.2.26 set\_enabled()** void set\_enabled (   
     QObject \* *parent*,   
     const char \* *key*,   
     bool *enabled* )

set\_enabled

Parameters

<i>parent</i>	
<i>key</i>	
<i>enabled</i>	

**Todo** error reporting.

**18.9.2.27 set\_visibility()** void set\_visibility (   
     QObject \* *parent*,   
     const char \* *key*,   
     bool *visibility* )

set\_visibility

Parameters

<i>parent</i>	
<i>key</i>	
<i>visibility</i>	

**Todo** error reporting.

**18.9.2.28 to\_EmbVector()** EmbVector to\_EmbVector (   
     QPointF *a* )

**18.9.2.29 to\_qlist()** QList< QGraphicsItem \* > to\_qlist (   
     std::vector< QGraphicsItem \* > *list* )

to\_qlist

Parameters

<i>list</i>	
-------------	--

Returns

**18.9.2.30 to\_QPointF()** QPointF to\_QPointF (   
     EmbVector *a* )

**18.9.2.31 to\_string\_vector()** StringList to\_string\_vector (   
     QStringList *list* )

to\_string\_vector

#### Parameters

<i>list</i>	
-------------	--

#### Returns

**18.9.2.32 to\_vector()** `std::vector< QGraphicsItem * > to_vector (`  
`QList< QGraphicsItem * > list )`

to\_vector

#### Parameters

<i>list</i>	
-------------	--

#### Returns

**18.9.2.33 tokenize()** `StringList tokenize (`  
`String str,`  
`const char delim )`

tokenize

#### Parameters

<i>str</i>	
<i>delim</i>	

#### Returns

**18.9.2.34 translate\_str()** `QString translate_str (`  
`const char * str )`

## 18.10 embroidermodder2/layer-manager.cpp File Reference

```
#include "embroidermodder.h"
```

### 18.10.1 Detailed Description

Embroidermodder 2.

Copyright 2013-2022 The Embroidermodder Team Embroidermodder 2 is Open Source Software. See LICENSE for licensing terms.

Use Python's PEP7 style guide. <https://peps.python.org/pep-0007/>

## 18.11 embroidermodder2/mainwindow-menus.cpp File Reference

```
#include "embroidermodder.h"
```

### Functions

- void [create\\_menu](#) (std::string menu, [StringList](#) def, bool topLevel)  
*create\_menu*

#### 18.11.1 Function Documentation

**18.11.1.1 [create\\_menu\(\)](#)** void [create\\_menu](#) (  
std::string *menu*,  
[StringList](#) *def*,  
bool *topLevel* )

[create\\_menu](#)

#### Parameters

<i>menu</i>	
<i>def</i>	
<i>topLevel</i>	

## 18.12 embroidermodder2/mainwindow-toolbars.cpp File Reference

```
#include "embroidermodder.h"
```

## 18.13 embroidermodder2/mainwindow.cpp File Reference

```
#include "embroidermodder.h"
#include <cerrno>
#include <iostream>
#include <fstream>
#include <string>
```

### Enumerations

- enum [OBJ\\_LTYPE\\_VALUES](#) {  
OBJ\_LTYPE\_CONT = 0 , OBJ\_LTYPE\_CENTER = 1 , OBJ\_LTYPE\_DOT = 2 , OBJ\_LTYPE\_HIDDEN = 3 ,  
OBJ\_LTYPE\_PHANTOM = 4 , OBJ\_LTYPE\_ZIGZAG = 5 , OBJ\_LTYPE\_RUNNING = 6 , OBJ\_LTYPE\_SATIN  
= 7 ,  
OBJ\_LTYPE\_FISHBONE = 8 }
- enum [OBJ\\_LWT\\_VALUES](#) {  
OBJ\_LWT\_BYLAYER = -2 , OBJ\_LWT\_BYBLOCK = -1 , OBJ\_LWT\_DEFAULT = 0 , OBJ\_LWT\_01 = 1 ,  
OBJ\_LWT\_02 = 2 , OBJ\_LWT\_03 = 3 , OBJ\_LWT\_04 = 4 , OBJ\_LWT\_05 = 5 ,  
OBJ\_LWT\_06 = 6 , OBJ\_LWT\_07 = 7 , OBJ\_LWT\_08 = 8 , OBJ\_LWT\_09 = 9 ,  
OBJ\_LWT\_10 = 10 , OBJ\_LWT\_11 = 11 , OBJ\_LWT\_12 = 12 , OBJ\_LWT\_13 = 13 ,  
OBJ\_LWT\_14 = 14 , OBJ\_LWT\_15 = 15 , OBJ\_LWT\_16 = 16 , OBJ\_LWT\_17 = 17 ,  
OBJ\_LWT\_18 = 18 , OBJ\_LWT\_19 = 19 , OBJ\_LWT\_20 = 20 , OBJ\_LWT\_21 = 21 ,  
OBJ\_LWT\_22 = 22 , OBJ\_LWT\_23 = 23 , OBJ\_LWT\_24 = 24 }
- enum [OBJ\\_SNAP\\_VALUES](#) {  
OBJ\_SNAP\_NULL = 0 , OBJ\_SNAP\_ENDPOINT = 1 , OBJ\_SNAP\_MIDPOINT = 2 , OBJ\_SNAP\_CENTER

```

= 3 ,
OBJ_SNAP_NODE = 4 , OBJ_SNAP_QUADRANT = 5 , OBJ_SNAP_INTERSECTION = 6 , OBJ_SNAP_EXTENSION
= 7 ,
OBJ_SNAP_INSERTION = 8 , OBJ_SNAP_PERPENDICULAR = 9 , OBJ_SNAP_TANGENT = 10 ,
OBJ_SNAP_NEAREST = 11 ,
OBJ_SNAP_APPINTERSECTION = 12 , OBJ_SNAP_PARALLEL = 13 }

```

## Functions

- static `String about_action (String args)`
- static `String add_arc_action (String args)`  
`add_arc_action`
- static `String add_circle_action (String args)`  
`add_circle_action`
- static `String add_dim_leader_action (String args)`
- static `String add_ellipse_action (String args)`  
`AddEllipse.`
- static `String add_geometry_action (String args)`  
`add_geometry_action`
- static `String add_horizontal_dimension_action (String args)`
- static `String add_image_action (String args)`
- static `String add_infinite_line_action (String args)`
- static `String add_line_action (String args)`
- static `String add_path_action (String args)`
- static `String add_point_action (String args)`  
`add_point_action`
- static `String add_polygon_action (String args)`  
`add_polygon_action`
- static `String add_polyline_action (String args)`
- static `String add_ray_action (String args)`
- static `String add_rectangle_action (String args)`  
`add_rectangle_action`
- static `String add_regular_polygon_action (String args)`  
`AddRegularPolygon.`
- static `String add_rounded_rectangle_action (String args)`  
`add_rounded_rectangle_action`
- static `String add_rubber_action (String args)`  
`add_rubber_action`
- static `String add_slot_action (String args)`  
`add_slot_action`
- static `String add_text_multi_action (String args)`  
`add_text_multi_action`
- static `String add_text_single_action (String args)`  
`add_text_single_action`
- static `String add_to_selection_action (String args)`  
`add_to_selection_action`
- static `String add_triangle_action (String args)`  
`add_triangle_action`
- static `String add_vertical_dimension_action (String args)`
- static `String alert_action (String args)`  
`alert_action`
- static `String allow_rubber_action (String args)`  
`AllowRubber.`



- static [String append\\_history\\_action](#) (String args)  
*append\_history\_action*
- static [String append\\_prompt\\_history\\_action](#) (String args)  
*AppendPromptHistory.*
- static [String calculate\\_angle\\_action](#) (String args)  
*calculate\_angle\_action*
- static [String calculate\\_distance\\_action](#) (String args)  
*calculate\_distance*
- static [String changelog\\_action](#) (String args)  
*changelog\_action*
- static [String clear\\_rubber\\_action](#) (String args)  
*ClearRubber.*
- static [String copy\\_action](#) (String args)  
*copy\_action*
- static [String copy\\_selected\\_action](#) (String args)  
*CopySelected x y.*
- static [String cut\\_action](#) (String args)  
*cut\_action*
- static [String cut\\_selected\\_action](#) (String args)  
*CutSelected x y.*
- static [String day\\_vision\\_action](#) (String args)  
*MainWindow::dayVision.*
- static [String delete\\_selected\\_action](#) (String args)  
*DeleteSelected.*
- static [String design\\_details\\_action](#) (String args)
- static [String do\\_nothing\\_action](#) (String args)  
*do\_nothing\_action This action intensionally does nothing.*
- static [String end\\_action](#) (String args)  
*end\_action*
- static [String error\\_action](#) (String args)  
*Error.*
- static [String help\\_action](#) (String args)  
*help\_action*
- static [String icon\\_action](#) (String command)  
*icon\_action*
- static [String init\\_action](#) (String args)  
*init\_action*
- static [String messagebox\\_action](#) (String args)  
*MessageBox type title text.*
- static [String mirror\\_selected\\_action](#) (String args)  
*MirrorSelected x1 y1 x2 y2.*
- static [String mouse\\_x\\_action](#) (String args)  
*MouseX.*
- static [String mouse\\_y\\_action](#) (String args)  
*MouseY.*
- static [String move\\_selected\\_action](#) (String args)  
*MoveSelected dx dy.*
- static [String new\\_action](#) (String args)  
*new\_action*
- static [String night\\_vision\\_action](#) (String args)  
*MainWindow::nightVision.*

- static [String num\\_selected\\_action](#) (String args)  
*NumSelected.*
- static [String open\\_action](#) (String args)  
*open\_action*
- static [String pan\\_action](#) (String mode)  
*pan\_action*
- static [String paste\\_action](#) (String args)  
*paste\_action*
- static [String paste\\_selected\\_action](#) (String args)  
*PasteSelected x y.*
- static [String perpendicular\\_distance\\_action](#) (String args)
- static [String platform\\_action](#) (String args)  
*platform\_action*
- static [String preview\\_off\\_action](#) (String args)  
*PreviewOff.*
- static [String preview\\_on\\_action](#) (String args)  
*preview\_on\_action*
- static [String print\\_action](#) (String args)  
*print\_action*
- static [String print\\_area\\_action](#) (String args)  
*PrintArea x y w h.*
- static [String qsnap\\_x\\_action](#) (String args)  
*QSnapX.*
- static [String qsnap\\_y\\_action](#) (String args)  
*QSnapY.*
- static [String quit\\_action](#) (String args)  
*quit\_action*
- static [String redo\\_action](#) (String args)  
*redo\_action*
- static [String rotate\\_selected\\_action](#) (String args)  
*RotateSelected x y rot.*
- static [String scale\\_selected\\_action](#) (String args)  
*ScaleSelected x y factor.*
- static [String select\\_all\\_action](#) (String args)  
*select\_all\_action*
- static [String set\\_background\\_color\\_action](#) (String args)  
*set\_background\_color\_action*
- static [String set\\_crosshair\\_color\\_action](#) (String args)
- static [String set\\_cursor\\_shape\\_action](#) (String args)
- static [String set\\_grid\\_color\\_action](#) (String args)
- static [String set\\_prompt\\_prefix\\_action](#) (String args)  
*set\_prompt\_prefix\_action*
- static [String set\\_rubber\\_filter\\_action](#) (String args)
- static [String set\\_rubber\\_mode\\_action](#) (String args)
- static [String set\\_rubber\\_point\\_action](#) (String args)
- static [String set\\_rubber\\_text\\_action](#) (String args)  
*set\_rubber\_text\_action*
- static [String settings\\_dialog\\_action](#) (String showTab)  
*settings\_dialog*
- static [String spare\\_rubber\\_action](#) (String args)  
*SpareRubber.*

- static [String tip\\_of\\_the\\_day\\_action](#) ([String](#) args)  
*tip\_of\_the\_day\_action*
- static [String todo\\_action](#) ([String](#) args)  
*Todo.*
- static [String undo\\_action](#) ([String](#) args)  
*undo\_action*
- static [String version\\_action](#) ([String](#) args)  
*version\_action*
- static [String whats\\_this\\_action](#) ([String](#) args)  
*whats\_this\_action*
- static [String window\\_action](#) ([String](#) args)  
*window\_action*
- static [String zoom\\_action](#) ([String](#) mode)  
*zoom\_action*
- void [no\\_argument\\_debug](#) ([String](#) function\_name, [String](#) args)  
*no\_argument\_debug*
- [String platformString](#) (void)  
*platformString*
- [View](#) \* [activeView](#) (void)  
*activeView*
- [QGraphicsScene](#) \* [activeScene](#) ()  
*MainWindow::activeScene.*
- [String make\\_layer\\_active\\_action](#) ([String](#) args)  
*MainWindow::makeLayerActive.*
- [String layer\\_manager\\_action](#) ([String](#) args)  
*layer\_manager\_action*
- [String layer\\_previous\\_action](#) ([String](#) args)  
*layer\_previous\_action*
- static [String set\\_crosshair\\_color\\_action](#) (uint8\_t r, uint8\_t g, uint8\_t b)  
*SetCrossHairColor.*
- static [String set\\_grid\\_color\\_action](#) (uint8\_t r, uint8\_t g, uint8\_t b)  
*set\_grid\_color*
- static [String preview\\_on\\_action](#) ([String](#) clone, [String](#) mode, [EmbReal](#) x, [EmbReal](#) y, [EmbReal](#) data)  
*PreviewOn.*
- static [String SetRubberText](#) (QString key, QString txt)
- static [String add\\_point\\_action](#) ([EmbReal](#) x, [EmbReal](#) y)  
*AddPoint.*
- [String construct\\_command](#) ([String](#) command, const char \*fmt,...)  
*construct\_command*
- [String read\\_string\\_setting](#) (toml\_table\_t \*table, const char \*key)
- std::vector< [String](#) > [read\\_string\\_list\\_setting](#) (toml\_table\_t \*table, const char \*key)
- int [read\\_configuration](#) (void)  
*Read the settings from file which aren't editable by the user. These files need to be placed in the install folder.*
- bool [validRGB](#) (int r, int g, int b)
- [String disable\\_action](#) ([String](#) variable)  
*disable\_action*
- [String run\\_script\\_file](#) ([String](#) fname)  
*MainWindow::run\_script\_file.*
- [String run\\_script](#) ([StringList](#) script)  
*A basic line-by-line script processor to allow for extensions to the program.*
- [String actuator](#) ([String](#) line)

*MainWindow::actuator.*

- static [String clear\\_selection\\_action](#) ([String](#) args)
- static [String debug\\_action](#) ([String](#) args)
- static [String vulcanize\\_action](#) ([String](#) args)
- static [String rubber\\_action](#) ([String](#) command)
- static [String blink\\_prompt\\_action](#) ([String](#) args)
- [String convert\\_args\\_to\\_type](#) ([String](#) label, [std::vector](#)< [String](#) > args, [const char](#) \*args\_template, [NodeList](#) a)

*Inspired by PyArg\_ParseTupleAndKeywords allowing a uniform argument parsing framework.*

- [String include\\_action](#) ([NodeList](#) a)

*Include.*

- [String is\\_int\\_action](#) ([String](#) args)
- [String SetTextAngle\\_action](#) ([String](#) args)
- [bool validFileFormat](#) ([String](#) fileName)

*MainWindow::validFileFormat.*

## Variables

- [MainWindow](#) \* [\\_mainWin](#) = 0
- [MdiArea](#) \* [mdiArea](#) = 0
- [CmdPrompt](#) \* [prompt](#) = 0
- [PropertyEditor](#) \* [dockPropEdit](#) = 0
- [UndoEditor](#) \* [dockUndoEdit](#) = 0
- [StatusBar](#) \* [statusbar](#) = 0
- [QWizard](#) \* [wizardTipOfTheDay](#)
- [QLabel](#) \* [labelTipOfTheDay](#)
- [QCheckBox](#) \* [checkBoxTipOfTheDay](#)
- [Dictionary settings](#)

*Settings System.*

- [Dictionary dialog](#)
- [Dictionary config](#)
- [std::unordered\\_map](#)< [String](#), [StringList](#) > [scripts](#)
- [std::unordered\\_map](#)< [String](#), [QGroupBox](#) \* > [groupBoxes](#)
- [std::unordered\\_map](#)< [String](#), [QCheckBox](#) \* > [checkBoxes](#)
- [std::unordered\\_map](#)< [String](#), [QSpinBox](#) \* > [spinBoxes](#)
- [std::unordered\\_map](#)< [String](#), [QDoubleSpinBox](#) \* > [doubleSpinBoxes](#)
- [std::unordered\\_map](#)< [String](#), [QLabel](#) \* > [labels](#)
- [std::unordered\\_map](#)< [String](#), [QComboBox](#) \* > [comboBoxes](#)
- [std::unordered\\_map](#)< [String](#), [QLineEdit](#) \* > [lineEdits](#)
- [std::unordered\\_map](#)< [String](#), [QToolButton](#) \* > [toolButtons](#)
- [std::unordered\\_map](#)< [String](#), [Dictionary](#) > [config\\_tables](#)
- [std::unordered\\_map](#)< [String](#), [QAction](#) \* > [actionHash](#)
- [std::unordered\\_map](#)< [String](#), [QToolBar](#) \* > [toolbarHash](#)
- [std::unordered\\_map](#)< [String](#), [QMenu](#) \* > [menuHash](#)
- [std::unordered\\_map](#)< [String](#), [QMenu](#) \* > [subMenuHash](#)
- [std::unordered\\_map](#)< [String](#), [Command](#) > [command\\_map](#)
- [StringList rubber\\_modes](#)

## 18.13.1 Enumeration Type Documentation

### 18.13.1.1 OBJ\_LTYPE\_VALUES `enum OBJ_LTYPE_VALUES`

## Enumerator

OBJ_LTYPE_CONT	
OBJ_LTYPE_CENTER	
OBJ_LTYPE_DOT	
OBJ_LTYPE_HIDDEN	
OBJ_LTYPE_PHANTOM	
OBJ_LTYPE_ZIGZAG	
OBJ_LTYPE_RUNNING	
OBJ_LTYPE_SATIN	
OBJ_LTYPE_FISHBONE	

**18.13.1.2 OBJ\_LWT\_VALUES** enum [OBJ\\_LWT\\_VALUES](#)

## Enumerator

OBJ_LWT_BYLAYER	
OBJ_LWT_BYBLOCK	
OBJ_LWT_DEFAULT	
OBJ_LWT_01	
OBJ_LWT_02	
OBJ_LWT_03	
OBJ_LWT_04	
OBJ_LWT_05	
OBJ_LWT_06	
OBJ_LWT_07	
OBJ_LWT_08	
OBJ_LWT_09	
OBJ_LWT_10	
OBJ_LWT_11	
OBJ_LWT_12	
OBJ_LWT_13	
OBJ_LWT_14	
OBJ_LWT_15	
OBJ_LWT_16	
OBJ_LWT_17	
OBJ_LWT_18	
OBJ_LWT_19	
OBJ_LWT_20	
OBJ_LWT_21	
OBJ_LWT_22	
OBJ_LWT_23	
OBJ_LWT_24	

**18.13.1.3 OBJ\_SNAP\_VALUES** enum [OBJ\\_SNAP\\_VALUES](#)

## Enumerator

OBJ_SNAP_NULL	
---------------	--

**Enumerator**

OBJ_SNAP_ENDPOINT	
OBJ_SNAP_MIDPOINT	
OBJ_SNAP_CENTER	
OBJ_SNAP_NODE	
OBJ_SNAP_QUADRANT	
OBJ_SNAP_INTERSECTION	
OBJ_SNAP_EXTENSION	
OBJ_SNAP_INSERTION	
OBJ_SNAP_PERPENDICULAR	
OBJ_SNAP_TANGENT	
OBJ_SNAP_NEAREST	
OBJ_SNAP_APPINTERSECTION	
OBJ_SNAP_PARALLEL	

**18.13.2 Function Documentation**

**18.13.2.1 about\_action()** `String` about\_action (   
 `String` args ) [static]

ACTIONS

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

**18.13.2.2 activeScene()** `QGraphicsScene *` activeScene ( )   
 MainWindow::activeScene.

Returns

**18.13.2.3 activeView()** `View *` activeView (   
 void )

activeView

Returns

**18.13.2.4 actuator()** `String` actuator (   
 `String` line )

MainWindow::actuator.

Parameters

<i>command</i>	
----------------	--

**18.13.2.5 RUN COMMAND** `QAction* act = qobject_cast<QAction*>(sender()); if (act) { prompt->endCommand(); prompt->setCurrentText(act->objectName()); prompt->processInput(); }`

**18.13.2.6 INIT** `QString fileName = "commands/" + cmd + "/" + cmd + ".js"; if (!getSettingsSelectionModePickFirst()) { actuator("clear-selection"); } TODO: Uncomment this line when post-selection is available engine->evaluate(cmd + "_main(void)", fileName);`

**18.13.2.7 PROMPT** `QString fileName = "commands/" + cmd + "/" + cmd + ".js"; NOTE: Replace any special characters that will cause a syntax error QString safeStr = str; safeStr.replace("\\", "\\"); safeStr.replace("'", "\'"); if (prompt->promptInput->rapidFireEnabled) { engine->evaluate(cmd + "_prompt(" + safeStr + ")", fileName); } else { engine->evaluate(cmd + "_prompt(" + safeStr.toUpper() + ")", fileName); }`

**18.13.2.8 add\_arc\_action()** `static String add_arc_action (String args) [static]`  
`add_arc_action`

Parameters

<code>args</code>	
-------------------	--

Returns

`EmbReal startX, EmbReal startY, EmbReal midX, EmbReal midY, EmbReal endX, EmbReal endY, String rubberMode`

**18.13.2.9 add\_circle\_action()** `static String add_circle_action (String args) [static]`  
`add_circle_action`

Returns

`EmbReal centerX, EmbReal centerY, EmbReal radius, bool fill, String rubberMode`

**18.13.2.10 add\_dim\_leader\_action()** `static String add_dim_leader_action (String args) [static]`  
`EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, EmbReal rot, String rubberMode`

**18.13.2.11 add\_ellipse\_action()** `static String add_ellipse_action (String args) [static]`  
`AddEllipse.`

Parameters

<code>args</code>	
-------------------	--

Returns

`EmbReal centerX, EmbReal centerY, EmbReal width, EmbReal height, EmbReal rot, bool fill, String rubberMode`

**18.13.2.12 add\_geometry\_action()** `static String add_geometry_action (String args) [static]`  
`add_geometry_action`

## Parameters

<i>args</i>	
-------------	--

## Returns

**18.13.2.13 add\_horizontal\_dimension\_action()** static *String* add\_horizontal\_dimension\_action ( *String* *args* ) [static]

EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, EmbReal legHeight

**18.13.2.14 add\_image\_action()** static *String* add\_image\_action ( *String* *args* ) [static]

QString img, EmbReal x, EmbReal y, EmbReal w, EmbReal h, EmbReal rot

**18.13.2.15 add\_infinite\_line\_action()** static *String* add\_infinite\_line\_action ( *String* *args* ) [static]

EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, EmbReal rot

**18.13.2.16 add\_line\_action()** static *String* add\_line\_action ( *String* *args* ) [static]

EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, EmbReal rot, String rubberMode

**18.13.2.17 add\_path\_action()** static *String* add\_path\_action ( *String* *args* ) [static]

## Note

This native is different than the rest in that the Y+ is down (scripters need not worry about this).

EmbReal startX, EmbReal startY, const QPainterPath& p, String rubberMode

**18.13.2.18 add\_point\_action()** [1/2] static *String* add\_point\_action ( *EmbReal* *x*, *EmbReal* *y* ) [static]

AddPoint.

## Parameters

<i>args</i>	
-------------	--

## Returns

**18.13.2.19 add\_point\_action()** [2/2] static *String* add\_point\_action ( *String* *args* ) [static]

add\_point\_action

## Parameters

<i>args</i>	
-------------	--



## Returns

**18.13.2.20 add\_polygon\_action()** static `String` add\_polygon\_action (   
`String args` ) [static]  
 add\_polygon\_action

## Parameters

<code>args</code>	
-------------------	--

## Returns

NOTE: This native is different than the rest in that the Y+ is down (scripters need not worry about this) Emb↔  
 Real startX, EmbReal startY, const QPainterPath& p, String rubberMode

**18.13.2.21 add\_polyline\_action()** static `String` add\_polyline\_action (   
`String args` ) [static]

**18.13.2.22 add\_ray\_action()** static `String` add\_ray\_action (   
`String args` ) [static]  
 EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, EmbReal rot

**18.13.2.23 add\_rectangle\_action()** static `String` add\_rectangle\_action (   
`String args` ) [static]  
 add\_rectangle\_action

## Parameters

<code>args</code>	
-------------------	--

## Returns

**18.13.2.24 add\_regular\_polygon\_action()** static `String` add\_regular\_polygon\_action (   
`String args` ) [static]  
 AddRegularPolygon.

## Returns

EmbReal centerX, EmbReal centerY, quint16 sides, uint8\_t mode, EmbReal rad, EmbReal rot, bool fill

**18.13.2.25 add\_rounded\_rectangle\_action()** static `String` add\_rounded\_rectangle\_action (   
`String args` ) [static]  
 add\_rounded\_rectangle\_action

**Parameters**

<i>args</i>	
-------------	--

**Returns**

EmbReal x, EmbReal y, EmbReal w, EmbReal h, EmbReal rad, EmbReal rot, bool fill

**18.13.2.26 add\_rubber\_action()** `String` add\_rubber\_action (  
    `String` args ) [static]  
add\_rubber\_action

**Parameters**

<i>args</i>	
-------------	--

**Returns**

**18.13.2.27 add\_slot\_action()** `static String` add\_slot\_action (  
    `String` args ) [static]  
add\_slot\_action

**Parameters**

<i>args</i>	
-------------	--

**Returns**

EmbReal centerX, EmbReal centerY, EmbReal diameter, EmbReal length, EmbReal rot, bool fill, String rubberMode

**18.13.2.28 add\_text\_multi\_action()** `static String` add\_text\_multi\_action (  
    `String` args ) [static]  
add\_text\_multi\_action

**Returns**

QString str, EmbReal x, EmbReal y, EmbReal rot, bool fill, String rubberMode

**18.13.2.29 add\_text\_single\_action()** `static String` add\_text\_single\_action (  
    `String` args ) [static]  
add\_text\_single\_action

**Returns**

QString str, EmbReal x, EmbReal y, EmbReal rot, bool fill, String rubberMode

**18.13.2.30 add\_to\_selection\_action()** static `String` add\_to\_selection\_action (   
 `String args` ) [static]  
add\_to\_selection\_action

**Parameters**

<i>args</i>	
-------------	--

**Returns**

**18.13.2.31 add\_triangle\_action()** static `String` add\_triangle\_action (   
 `String args` ) [static]  
add\_triangle\_action

**Parameters**

<i>args</i>	
-------------	--

**Returns**

**18.13.2.32 add\_vertical\_dimension\_action()** static `String` add\_vertical\_dimension\_action (   
 `String args` ) [static]  
EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, EmbReal legHeight

**18.13.2.33 alert\_action()** static `String` alert\_action (   
 `String args` ) [static]  
alert\_action

**Parameters**

<i>args</i>	
-------------	--

**Returns**

**18.13.2.34 allow\_rubber\_action()** static `String` allow\_rubber\_action (   
 `String args` ) [static]  
AllowRubber.

**Returns**

**18.13.2.35 append\_history\_action()** static `String` append\_history\_action (   
 `String args` ) [static]  
append\_history\_action

**Parameters**

<i>args</i>	
-------------	--

Returns

**18.13.2.36 append\_prompt\_history\_action()** `String append_prompt_history_action (String args ) [static]`

AppendPromptHistory.

Parameters

<i>a</i>	
----------	--

Returns

**18.13.2.37 blink\_prompt\_action()** `static String blink_prompt_action (String args ) [static]`

**18.13.2.38 calculate\_angle\_action()** `static String calculate_angle_action (String args ) [static]`

calculate\_angle\_action

Parameters

<i>args</i>	
-------------	--

Returns

**18.13.2.39 calculate\_distance\_action()** `static String calculate_distance_action (String args ) [static]`

calculate\_distance

Parameters

<i>args</i>	
-------------	--

Returns

**18.13.2.40 changelog\_action()** `static String changelog_action (String args ) [static]`

changelog\_action

Parameters

<i>args</i>	
-------------	--

## Returns

**18.13.2.41 clear\_rubber\_action()** static `String` clear\_rubber\_action (   
 `String` args ) [static]

ClearRubber.

**18.13.2.42 clear\_selection\_action()** static `String` clear\_selection\_action (   
 `String` args ) [static]

**18.13.2.43 construct\_command()** `String` construct\_command (   
 `String` command,   
 const char \* fmt,   
 ... )

construct\_command

## Parameters

<i>command</i>	
<i>fmt</i>	

## Returns

**18.13.2.44 convert\_args\_to\_type()** `String` convert\_args\_to\_type (   
 `String` label,   
 std::vector< `String` > args,   
 const char \* args\_template,   
 `NodeList` a )

Inspired by PyArg\_ParseTupleAndKeywords allowing a uniform argument parsing framework.

## Parameters

<i>label</i>	The caller's name.
<i>args</i>	The list of strings passed from the user.
<i>args_template</i>	The string of characters describing the types of the output.
<i>result</i>	The fixed length array of results.

## Returns

An error message if an error occurred or an empty string if it passes.

**18.13.2.45 copy\_action()** static `String` copy\_action (   
 `String` args ) [static]

copy\_action

## Parameters

<i>args</i>	
-------------	--

## Returns

**18.13.2.46 copy\_selected\_action()** static `String` copy\_selected\_action (   
 `String args` ) [static]  
CopySelected x y.

**18.13.2.47 cut\_action()** static `String` cut\_action (   
 `String args` ) [static]  
cut\_action

## Parameters

<i>args</i>	
-------------	--

## Returns

**18.13.2.48 cut\_selected\_action()** static `String` cut\_selected\_action (   
 `String args` ) [static]  
CutSelected x y.

**18.13.2.49 day\_vision\_action()** `String` day\_vision\_action (   
 `String args` ) [static]  
MainWindow::dayVision.

**Todo** Make day vision color settings.

**18.13.2.50 debug\_action()** static `String` debug\_action (   
 `String args` ) [static]

**18.13.2.51 delete\_selected\_action()** static `String` delete\_selected\_action (   
 `String args` ) [static]  
DeleteSelected.

**18.13.2.52 design\_details\_action()** `String` design\_details\_action (   
 `String args` ) [static]

**18.13.2.53 disable\_action()** `String` disable\_action (   
 `String variable` )  
disable\_action

**Parameters**

<i>variable</i>	
-----------------	--

**Returns**

**18.13.2.54 do\_nothing\_action()** `String do_nothing_action (`  
`String args ) [static]`  
do\_nothing\_action This action intensionally does nothing.

**Parameters**

<i>args</i>	This is ignored, it's present to make it a Command.
-------------	---

**Returns**

An empty string.

**18.13.2.55 end\_action()** `static String end_action (`  
`String args ) [static]`  
end\_action

**Parameters**

<i>args</i>	
-------------	--

**Returns**

**18.13.2.56 error\_action()** `String error_action (`  
`String args ) [static]`  
Error.

**Parameters**

<i>a</i>	
----------	--

**Returns**

**18.13.2.57 help\_action()** `static String help_action (`  
`String args ) [static]`  
help\_action



## Parameters

<i>args</i>	
-------------	--

## Returns

**18.13.2.58 icon\_action()** static `String` icon\_action (   
 `String` *command* ) [static]  
icon\_action

## Parameters

<i>command</i>	
----------------	--

## Returns

**18.13.2.59 include\_action()** `String` include\_action (   
 `NodeList` *a* )  
Include.

## Parameters

<i>a</i>	
----------	--

## Returns

**18.13.2.60 init\_action()** static `String` init\_action (   
 `String` *args* ) [static]  
init\_action

## Parameters

<i>args</i>	
-------------	--

## Returns

**18.13.2.61 is\_int\_action()** `String` is\_int\_action (   
 `String` *args* )  
argument string "i"

**18.13.2.62 layer\_manager\_action()** `String` layer\_manager\_action (   
 `String` *args* )

layer\_manager\_action

#### Parameters

<i>args</i>	
-------------	--

#### Returns

**18.13.2.63 layer\_previous\_action()** `String` layer\_previous\_action (   
 `String` *args* )

layer\_previous\_action

#### Parameters

<i>args</i>	
-------------	--

#### Returns

**18.13.2.64 make\_layer\_active\_action()** `String` make\_layer\_active\_action (   
 `String` *args* )

MainWindow::makeLayerActive.

#### Returns

**18.13.2.65 messagebox\_action()** `static String` messagebox\_action (   
 `String` *args* ) [static]

MessageBox *type title text*.

**18.13.2.66 mirror\_selected\_action()** `static String` mirror\_selected\_action (   
 `String` *args* ) [static]

MirrorSelected *x1 y1 x2 y2*.

**18.13.2.67 mouse\_x\_action()** `static String` mouse\_x\_action (   
 `String` *args* ) [static]

MouseX.

#### Returns

**18.13.2.68 mouse\_y\_action()** `static String` mouse\_y\_action (   
 `String` *args* ) [static]

MouseY.

Returns

**18.13.2.69 move\_selected\_action()** static `String` move\_selected\_action (   
 `String` args ) [static]

MoveSelected *dx dy*.

**18.13.2.70 new\_action()** static `String` new\_action (   
 `String` args ) [static]

new\_action

Parameters

<i>args</i>	
-------------	--

Returns

**18.13.2.71 night\_vision\_action()** `String` night\_vision\_action (   
 `String` args ) [static]

MainWindow::nightVision.

**Todo** Make night vision color settings.

**18.13.2.72 no\_argument\_debug()** void no\_argument\_debug (   
 `String` function\_name,   
 `String` args )

no\_argument\_debug

Parameters

<i>function_name</i>	
<i>args</i>	

**18.13.2.73 num\_selected\_action()** static `String` num\_selected\_action (   
 `String` args ) [static]

NumSelected.

Parameters

<i>args</i>	
-------------	--

Returns

**18.13.2.74 open\_action()** static `String` open\_action (   
 `String args` ) [static]  
open\_action

Parameters

<i>args</i>	
-------------	--

Returns

**18.13.2.75 pan\_action()** `String` pan\_action (   
 `String mode` ) [static]  
pan\_action

Parameters

<i>mode</i>	
-------------	--

Returns

**18.13.2.76 paste\_action()** static `String` paste\_action (   
 `String args` ) [static]  
paste\_action

Parameters

<i>args</i>	
-------------	--

Returns

**18.13.2.77 paste\_selected\_action()** static `String` paste\_selected\_action (   
 `String args` ) [static]  
PasteSelected x y.

**18.13.2.78 perpendicular\_distance\_action()** static `String` perpendicular\_distance\_action (   
 `String args` ) [static]

**18.13.2.79 platform\_action()** static `String` platform\_action (   
 `String args` ) [static]  
platform\_action

Parameters

<i>args</i>	
-------------	--

Returns

**18.13.2.80 platformString()** `String platformString (void )`

platformString

Returns

**18.13.2.81 preview\_off\_action()** `static String preview_off_action (String args ) [static]`

PreviewOff.

**18.13.2.82 preview\_on\_action()** [1/2] `String preview_on_action (String args ) [static]`

preview\_on\_action

Parameters

<i>args</i>	
-------------	--

Returns

**18.13.2.83 preview\_on\_action()** [2/2] `static String preview_on_action (String clone, String mode, EmbReal x, EmbReal y, EmbReal data ) [static]`

PreviewOn.

Parameters

<i>clone</i>	
<i>mode</i>	
<i>x</i>	
<i>y</i>	
<i>data</i>	

**18.13.2.84 print\_action()** `String print_action (String args ) [static]`

print\_action

**Parameters**

<i>args</i>	
-------------	--

**Returns**

**18.13.2.85 print\_area\_action()** static `String` print\_area\_action (   
 `String args` ) [static]

PrintArea x y w h.

EmbReal x, EmbReal y, EmbReal w, EmbReal h

**18.13.2.86 qsnap\_x\_action()** static `String` qsnap\_x\_action (   
 `String args` ) [static]

QSnapX.

**Returns**

**18.13.2.87 qsnap\_y\_action()** static `String` qsnap\_y\_action (   
 `String args` ) [static]

QSnapY.

**Returns**

**18.13.2.88 quit\_action()** static `String` quit\_action (   
 `String args` ) [static]

quit\_action

**Parameters**

<i>args</i>	
-------------	--

**Returns**

**18.13.2.89 read\_configuration()** int read\_configuration (   
 void )

Read the settings from file which aren't editable by the user. These files need to be placed in the install folder.

Expected Keys for actions String icon; The stub used for the icon and the basic command. String command; String tooltip; The label in the menus and the message that appears when you hover over an icon. String statustip; The message that appears at the bottom of the . String shortcut; The keyboard shortcut for this action. StringList aliases; A list of all alternative commands, if empty only the icon string will be . StringList script; If this is a compound action this will be a list of commands or it can allow for command line style command aliases. For example: icon16 would become the string list {"iconResize 16"}.

**18.13.2.90 read\_string\_list\_setting()** `std::vector< String > read_string_list_setting ( toml_table_t * table, const char * key )`

**18.13.2.91 read\_string\_setting()** `String read_string_setting ( toml_table_t * table, const char * key )`

**18.13.2.92 redo\_action()** `static String redo_action ( String args ) [static]`

redo\_action

Parameters

args	
------	--

Returns

**18.13.2.93 rotate\_selected\_action()** `static String rotate_selected_action ( String args ) [static]`

RotateSelected x y rot.

**18.13.2.94 rubber\_action()** `static String rubber_action ( String command ) [static]`

**18.13.2.95 run\_script()** `String run_script ( StringList script )`

A basic line-by-line script processor to allow for extensions to the program.

Since the actuator uses command line style parsing, a script is just a text file with each line a compatible command. It should be stressed that this has no control flow on purpose. We don't want this to be hacked into a full scripting language that could cause havoc on the user's system.

However, it may be useful to set and get variables and define macros: neither of these will allow for endless loops, stack overflow or other problems that third-party scripts could introduce.

example.sh

```
-----
# Save characters by defining functions.
# The syntax features
# Semi-colon ';' seperates out lines like in bash.
# The line ending is the end of the function, but the style
# is a shell function, so we need to write the end brace.
```

```
donut() { circle $1 $2 $3 $5 ; circle $1 $2 $4 $5 }
```

```
donut 10 20 20 black
donut 20 40 20 black
-----
```

**18.13.2.96 run\_script\_file()** `String run_script_file ( String fname )`

MainWindow::run\_script\_file.

## Parameters

<i>fname</i>	The path of the script to run.
--------------	--------------------------------

**18.13.2.97 scale\_selected\_action()** static `String` scale\_selected\_action (   
     `String args` ) [static]  
 ScaleSelected *x y factor*.

**18.13.2.98 select\_all\_action()** static `String` select\_all\_action (   
     `String args` ) [static]  
 select\_all\_action

## Parameters

<i>args</i>	
-------------	--

## Returns

**18.13.2.99 set\_background\_color\_action()** static `String` set\_background\_color\_action (   
     `String args` ) [static]  
 set\_background\_color\_action

## Parameters

<i>r</i>	
<i>g</i>	
<i>b</i>	

uint8\_t r, uint8\_t g, uint8\_t b

**18.13.2.100 set\_crosshair\_color\_action()** [1/2] static `String` set\_crosshair\_color\_action (   
     `String args` ) [static]  
 argument string "iii"

**18.13.2.101 set\_crosshair\_color\_action()** [2/2] static `String` set\_crosshair\_color\_action (   
     uint8\_t r,  
     uint8\_t g,  
     uint8\_t b ) [static]  
 SetCrossHairColor.

## Parameters

<i>r</i>	
<i>g</i>	
<i>b</i>	

**18.13.2.102 set\_cursor\_shape\_action()** static `String` set\_cursor\_shape\_action (



```
String str ) [static]
```

**18.13.2.103 set\_grid\_color\_action()** [1/2] `String set_grid_color_action (`  
`String args ) [static]`  
 argument string "iii"

**18.13.2.104 set\_grid\_color\_action()** [2/2] `static String set_grid_color_action (`  
`uint8_t r,`  
`uint8_t g,`  
`uint8_t b ) [static]`  
 set\_grid\_color

#### Parameters

<i>r</i>	
<i>g</i>	
<i>b</i>	

**18.13.2.105 set\_prompt\_prefix\_action()** `static String set_prompt_prefix_action (`  
`String args ) [static]`  
 set\_prompt\_prefix\_action

#### Parameters

<i>args</i>	
-------------	--

#### Returns

**18.13.2.106 set\_rubber\_filter\_action()** `static String set_rubber_filter_action (`  
`String args ) [static]`

**18.13.2.107 set\_rubber\_mode\_action()** `static String set_rubber_mode_action (`  
`String args ) [static]`

**18.13.2.108 set\_rubber\_point\_action()** `static String set_rubber_point_action (`  
`String args ) [static]`  
 QString key, EmbReal x, EmbReal y

**18.13.2.109 set\_rubber\_text\_action()** `String set_rubber_text_action (`  
`String args ) [static]`  
 set\_rubber\_text\_action

#### Parameters

<i>args</i>	
-------------	--

## Returns

**18.13.2.110 SetRubberText()** static `String` SetRubberText (   
 `QString key`,   
 `QString txt` ) [static]

**18.13.2.111 SetTextAngle\_action()** `String` SetTextAngle\_action (   
 `String args` )

**18.13.2.112 settings\_dialog\_action()** `String` settings\_dialog\_action (   
 `String showTab` ) [static]   
 settings\_dialog

## Parameters

<code>showTab</code>	
----------------------	--

**18.13.2.113 spare\_rubber\_action()** static `String` spare\_rubber\_action (   
 `String args` ) [static]   
 SpareRubber.

## Parameters

<code>qint64</code>	<code>id</code>
---------------------	-----------------

**18.13.2.114 tip\_of\_the\_day\_action()** `String` tip\_of\_the\_day\_action (   
 `String args` ) [static]   
 tip\_of\_the\_day\_action

## Parameters

<code>args</code>	
-------------------	--

## Returns

**18.13.2.115 todo\_action()** `String` todo\_action (   
 `String args` ) [static]   
 Todo.

## Parameters

<code>a</code>	
----------------	--

Returns

**18.13.2.116 undo\_action()** static `String` undo\_action (  
    `String` args ) [static]

undo\_action

Parameters

<i>args</i>	
-------------	--

Returns

**18.13.2.117 validFileFormat()** bool validFileFormat (  
    `String` fileName )

MainWindow::validFileFormat.

Parameters

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

Returns

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

**18.13.2.118 validRGB()** bool validRGB (  
    int r,  
    int g,  
    int b )

**18.13.2.119 version\_action()** static `String` version\_action (  
    `String` args ) [static]

version\_action

Parameters

<i>args</i>	
-------------	--

Returns

**18.13.2.120 vulcanize\_action()** static `String` vulcanize\_action (  
    `String` args ) [static]

**18.13.2.121** **whats\_this\_action()** `String` whats\_this\_action (  
    `String` args ) [static]  
whats\_this\_action

**Parameters**

<i>args</i>	
-------------	--

**Returns**

**18.13.2.122** **window\_action()** `static String` window\_action (  
    `String` args ) [static]  
window\_action

**Parameters**

<i>args</i>	
-------------	--

**Returns**

**18.13.2.123** **zoom\_action()** `String` zoom\_action (  
    `String` mode ) [static]  
zoom\_action

**Parameters**

<i>mode</i>	
-------------	--

**Returns**

### 18.13.3 Variable Documentation

**18.13.3.1** **\_mainWin** `MainWindow*` \_mainWin = 0

**18.13.3.2** **actionHash** `std::unordered_map<String, QAction*>` actionHash

**18.13.3.3** **checkBoxes** `std::unordered_map<String, QCheckBox *>` checkBoxes

**18.13.3.4** **checkBoxTipOfTheDay** `QCheckBox*` checkBoxTipOfTheDay

**18.13.3.5 comboBoxes** `std::unordered_map<String, QComboBox *> comboBoxes`

**18.13.3.6 command\_map** `std::unordered_map<String, Command> command_map`

**18.13.3.7 config** `Dictionary config`

**18.13.3.8 config\_tables** `std::unordered_map<String, Dictionary> config_tables`

**18.13.3.9 dialog** `Dictionary dialog`

**18.13.3.10 dockPropEdit** `PropertyEditor* dockPropEdit = 0`

**18.13.3.11 dockUndoEdit** `UndoEditor* dockUndoEdit = 0`

**18.13.3.12 doubleSpinBoxes** `std::unordered_map<String, QDoubleSpinBox *> doubleSpinBoxes`

**18.13.3.13 groupBoxes** `std::unordered_map<String, QGroupBox *> groupBoxes`

**18.13.3.14 labels** `std::unordered_map<String, QLabel *> labels`

**18.13.3.15 labelTipOfTheDay** `QLabel* labelTipOfTheDay`

**18.13.3.16 lineEdits** `std::unordered_map<String, QLineEdit *> lineEdits`

**18.13.3.17 mdiArea** `MdiArea* mdiArea = 0`

**18.13.3.18 menuHash** `std::unordered_map<String, QMenu*> menuHash`

**18.13.3.19 prompt** `CmdPrompt* prompt = 0`

**18.13.3.20 rubber\_modes** `StringList rubber_modes`

Initial value:

```
= {
    "CIRCLE_1P_RAD",
    "CIRCLE_1P_DIA",
    "CIRCLE_2P",
    "CIRCLE_3P",
    "CIRCLE_TTR",
    "CIRCLE_TTT",
    "DIMLEADER_LINE",
    "ELLIPSE_LINE",
    "ELLIPSE_MAJORDIAMETER_MINORRADIUS",
    "ELLIPSE_MAJORRADIUS_MINORRADIUS",
}
```

```
"ELLIPSE_ROTATION",
"LINE",
"POLYGON",
"POLYGON_INSCRIBE",
"POLYGON_CIRCUMSCRIBE",
"POLYLINE",
"RECTANGLE",
"TEXTSINGLE"
}
```

**18.13.3.21 scripts** `std::unordered_map<String, StringList> scripts`

**18.13.3.22 settings** `Dictionary settings`

Settings System.

Rather than pollute the global namespace, we collect together all the global settings into a structure that stores them. This also allows us to create a complete copy of the settings for the purpose of restoring them if the user cancels out of the Settings Dialog.

**18.13.3.23 spinBoxes** `std::unordered_map<String, QSpinBox *> spinBoxes`

**18.13.3.24 statusbar** `StatusBar* statusbar = 0`

**18.13.3.25 subMenuHash** `std::unordered_map<String, QMenu*> subMenuHash`

**18.13.3.26 toolbarHash** `std::unordered_map<String, QToolBar*> toolbarHash`

**18.13.3.27 toolButtons** `std::unordered_map<String, QToolButton *> toolButtons`

**18.13.3.28 wizardTipOfTheDay** `QWizard* wizardTipOfTheDay`

## 18.14 embroidermodder2/mdiarea.cpp File Reference

```
#include "embroidermodder.h"
```

## 18.15 embroidermodder2/mdiwindow.cpp File Reference

```
#include "embroidermodder.h"
```

### Functions

- `QString fileExtension (String fileName)`  
*MdiWindow::fileExtension.*

### 18.15.1 Function Documentation

**18.15.1.1 fileExtension()** `QString fileExtension (String fileName )`

*MdiWindow::fileExtension.*

## Parameters

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

## Returns

**18.16 embroidermodder2/objects.cpp File Reference**

```
#include "embroidermodder.h"
```

**Functions**

- QPointF [closest\\_point](#) (QPointF position, std::vector< QPointF > points)  
*mouse\_snap\_point*
- [EmbReal](#) [fourier\\_series](#) ([EmbReal](#) arg, std::vector< [EmbReal](#) > terms)  
*fourier\_series*
- void [add\\_polyline](#) (QPainterPath p, [String](#) rubberMode)  
*add\_polyline*
- [EmbVector](#) [rotate\\_vector](#) ([EmbVector](#) v, [EmbReal](#) alpha)

**18.16.1 Function Documentation**

**18.16.1.1 add\_polyline()** void add\_polyline (  
    QPainterPath *p*,  
    [String](#) *rubberMode* )  
add\_polyline

## Parameters

<i>p</i>	
<i>rubberMode</i>	

**18.16.1.2 closest\_point()** QPointF closest\_point (  
    QPointF *position*,  
    std::vector< QPointF > *points* )  
mouse\_snap\_point

## Parameters

<i>points</i>	
---------------	--

## Returns

**18.16.1.3 fourier\_series()** [EmbReal](#) fourier\_series (  
    [EmbReal](#) *arg*,  
    std::vector< [EmbReal](#) > *terms* )

fourier\_series

#### Parameters

<i>arg</i>	
<i>terms</i>	

**18.16.1.4 rotate\_vector()** `EmbVector rotate_vector (`  
    `EmbVector v,`  
    `EmbReal alpha )`

Returns

## 18.17 embroidermodder2/preview-dialog.cpp File Reference

```
#include "embroidermodder.h"
```

## 18.18 embroidermodder2/property-editor.cpp File Reference

```
#include "embroidermodder.h"
```

### Functions

- `std::vector< Dictionary > load_group_box_data_from_table (String key)`

### Variables

- `QString fieldOldText`
- `QString fieldNewText`
- `QString fieldVariesText`
- `QString fieldYesText`
- `QString fieldNoText`
- `QString fieldOnText`
- `QString fieldOffText`
- `QStringList object_names`
- `std::vector< std::pair< String, int > > group_box_types`
- `QFontComboBox * comboBoxTextSingleFont`
- `std::unordered_map< String, Dictionary > group_box_data`

### 18.18.1 Function Documentation

**18.18.1.1 load\_group\_box\_data\_from\_table()** `std::vector< Dictionary > load_group_box_data_↵`  
`from_table (`  
    `String key )`

### 18.18.2 Variable Documentation

**18.18.2.1 comboBoxTextSingleFont** `QFontComboBox* comboBoxTextSingleFont`



**18.18.2.2 fieldNewText** `QString fieldNewText`

**18.18.2.3 fieldNoText** `QString fieldNoText`

**18.18.2.4 fieldOffText** `QString fieldOffText`

**18.18.2.5 fieldOldText** `QString fieldOldText`

**18.18.2.6 fieldOnText** `QString fieldOnText`

**18.18.2.7 fieldVariesText** `QString fieldVariesText`

**18.18.2.8 fieldYesText** `QString fieldYesText`

**18.18.2.9 group\_box\_data** `std::unordered_map<String, Dictionary> group_box_data`

**18.18.2.10 group\_box\_types** `std::vector<std::pair<String, int> > group_box_types`

**18.18.2.11 object\_names** `StringList object_names`

**Initial value:**

```
= {
    "Base",
    "Arc",
    "Block",
    "Circle",
    "Aligned Dimension",
    "Angular Dimension",
    "Arclength Dimension",
    "Diameter Dimension",
    "Leader Dimension",
    "Linear Dimension",
    "Ordinate Dimension",
    "Radius Dimension",
    "Ellipse",
    "Image",
    "Infinite Line",
    "Line",
    "Path",
    "Point",
    "Polygon",
    "Polyline",
    "Ray",
    "Rectangle",
    "Multiline Text",
    "Text",
    "Unknown"
}
```

## 18.19 embroidermodder2/README.md File Reference

## 18.20 embroidermodder2/selectbox.cpp File Reference

```
#include "embroidermodder.h"
```

## 18.21 embroidermodder2/settings-dialog.cpp File Reference

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

### Functions

- void [make\\_editing\\_copy](#) ([StringList](#) props)
- void [read\\_settings](#) (void)  
*read\_settings*
- void [write\\_settings](#) (void)  
*MainWindow::writeSettings.*

### Variables

- [Dictionary](#) preview
- [Dictionary](#) accept\_
- [StringList](#) extensions
- [StringList](#) general\_props
- [StringList](#) display\_props
- [StringList](#) prompt\_props
- [StringList](#) quick\_snap\_props
- [StringList](#) opensave\_props

#### 18.21.1 Function Documentation

**18.21.1.1 [make\\_editing\\_copy\(\)](#)** void [make\\_editing\\_copy](#) (  
[StringList](#) props )

**18.21.1.2 [read\\_settings\(\)](#)** void [read\\_settings](#) (  
void )

[read\\_settings](#)

This file needs to be read from the users home directory to ensure it is writable.

**18.21.1.3 [write\\_settings\(\)](#)** void [write\\_settings](#) (  
void )

[MainWindow::writeSettings.](#)

This file needs to be read from the users home directory to ensure it is writable

#### 18.21.2 Variable Documentation

**18.21.2.1 [accept\\_](#)** [Dictionary](#) [accept\\_](#)

**18.21.2.2 [display\\_props](#)** [StringList](#) [display\\_props](#)

Initial value:

```
= {
    "display_use_opengl",
    "display_renderhint_aa",
    "display_renderhint_text_aa",
    "display_renderhint_smooth_pix",
    "display_renderhint_high_aa",
    "display_renderhint_noncosmetic",
    "display_show_scrollbars",
```

```

    "display_scrollbar_widget_num",
    "display_crosshair_color",
    "display_bg_color",
    "display_selectbox_left_color",
    "display_selectbox_left_fill",
    "display_selectbox_right_color",
    "display_selectbox_right_fill",
    "display_selectbox_alpha",
    "display_zoomscale_in",
    "display_zoomscale_out"
}

```

### 18.21.2.3 extensions `StringList` extensions

Initial value:

```

= {
    "100", "10o", "ART", "BMC", "BRO",
    "CND", "COL", "CSD", "CSV", "DAT",
    "DEM", "DSB", "DST", "DSZ", "DXF",
    "EDR", "EMD", "EXP", "EXY", "EYS",
    "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"
}

```

### 18.21.2.4 general\_props `StringList` general\_props

Initial value:

```

= {
    "general_icon_theme",
    "general_icon_size",
    "general_mdi_bg_use_logo",
    "general_mdi_bg_logo",
    "general_mdi_bg_use_texture",
    "general_mdi_bg_texture",
    "general_mdi_bg_use_color",
    "general_mdi_bg_color",
    "general_tip_of_the_day"
}

```

### 18.21.2.5 opensave\_props `StringList` opensave\_props

Initial value:

```

= {
    "opensave_custom_filter"
}

```

### 18.21.2.6 preview `Dictionary` preview

Embroidermodder 2

Copyright 2013-2022 The Embroidermodder Team Embroidermodder 2 is Open Source Software. See LICENSE for licensing terms.

Use Python's PEP7 style guide. <https://peps.python.org/pep-0007/> The actuator changes the program state via these global variables.

These copies of the settings struct are for restoring the state if the user doesn't want to accept their changes in the settings dialog.

### 18.21.2.7 prompt\_props `StringList` prompt\_props

Initial value:

```

= {
    "prompt_text_color",
    "prompt_background_color",
    "prompt_font_family",
    "prompt_font_size",
    "prompt_save_history",
    "prompt_save_history_as_html"
}

```

### 18.21.2.8 quick\_snap\_props `StringList` quick\_snap\_props

Initial value:

```
= {  
    "quicksnap_endpoint",  
    "quicksnap_midpoint",  
    "quicksnap_center",  
    "quicksnap_node",  
    "quicksnap_quadrant",  
    "quicksnap_intersection",  
    "quicksnap_extension",  
    "quicksnap_insertion",  
    "quicksnap_perpendicular",  
    "quicksnap_tangent",  
    "quicksnap_nearest",  
    "quicksnap_apparent",  
    "quicksnap_parallel",  
    "quicksnap_locator_color",  
    "quicksnap_locator_size",  
    "quicksnap_aperture_size"  
}
```

## 18.22 embroidermodder2/statusbar.cpp File Reference

```
#include "embroidermodder.h"
```

## 18.23 embroidermodder2/undo-commands.cpp File Reference

```
#include "embroidermodder.h"
```

## 18.24 embroidermodder2/undo-editor.cpp File Reference

```
#include "embroidermodder.h"
```

### 18.24.1 Detailed Description

Embroidermodder 2.

Copyright 2013-2022 The Embroidermodder Team Embroidermodder 2 is Open Source Software. See LICENSE for licensing terms.

Use Python's PEP7 style guide. <https://peps.python.org/pep-0007/>

---

## 18.25 embroidermodder2/view.cpp File Reference

```
#include "embroidermodder.h"  
#include <cassert>  
#include <QtOpenGL>
```

### Functions

- bool `contains` (`StringList` list, `String` entry)

### 18.25.1 Detailed Description

Embroidermodder 2.

Copyright 2013-2022 The Embroidermodder Team Embroidermodder 2 is Open Source Software. See LICENSE for licensing terms.

Use Python's PEP7 style guide. <https://peps.python.org/pep-0007/>

---

### 18.25.2 Function Documentation

**18.25.2.1 contains()** `bool contains (`  
     `StringList list,`  
     `String entry )`

## 18.26 extern/libembroidery/src/array.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "embroidery_internal.h"
```

### Functions

- `EmbArray * embArray_create` (int type)
- int `embArray_resize` (EmbArray \*a)
- void `embArray_copy` (EmbArray \*dst, EmbArray \*src)
- int `embArray_addArc` (EmbArray \*a, EmbArc b)
- int `embArray_addCircle` (EmbArray \*a, EmbCircle b)
- int `embArray_addEllipse` (EmbArray \*a, EmbEllipse b)
- int `embArray_addFlag` (EmbArray \*a, EmbFlag b)
- int `embArray_addLine` (EmbArray \*a, EmbLine b)
- int `embArray_addPath` (EmbArray \*a, EmbPath b)
- int `embArray_addPoint` (EmbArray \*a, EmbPoint b)
- int `embArray_addPolyline` (EmbArray \*a, EmbPolyline b)
- int `embArray_addPolygon` (EmbArray \*a, EmbPolygon b)
- int `embArray_addRect` (EmbArray \*a, EmbRect b)
- int `embArray_addStitch` (EmbArray \*a, EmbStitch b)
- int `embArray_addVector` (EmbArray \*a, EmbVector b)
- void `embArray_free` (EmbArray \*a)

### 18.26.1 Function Documentation

**18.26.1.1 embArray\_addArc()** `int embArray_addArc (`  
     `EmbArray * a,`  
     `EmbArc b )`

Add an arc *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.26.1.2 embArray\_addCircle()** `int embArray_addCircle (`  
     `EmbArray * a,`  
     `EmbCircle b )`

Add a circle *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.26.1.3 embArray\_addEllipse()** `int embArray_addEllipse (`  
     `EmbArray * a,`  
     `EmbEllipse b )`

Add an ellipse *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.26.1.4 embArray\_addFlag()** `int embArray_addFlag (`  
     `EmbArray * a,`  
     `EmbFlag b )`

Add a flag *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.26.1.5 embArray\_addLine()** `int embArray_addLine (`  
    `EmbArray * a,`  
    `EmbLine b )`

Add a line *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.26.1.6 embArray\_addPath()** `int embArray_addPath (`  
    `EmbArray * a,`  
    `EmbPath b )`

Add a path *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.26.1.7 embArray\_addPoint()** `int embArray_addPoint (`  
    `EmbArray * a,`  
    `EmbPoint b )`

Add a point *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.26.1.8 embArray\_addPolygon()** `int embArray_addPolygon (`  
    `EmbArray * a,`  
    `EmbPolygon b )`

Add a polygon *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.26.1.9 embArray\_addPolyline()** `int embArray_addPolyline (`  
    `EmbArray * a,`  
    `EmbPolyline b )`

Add a polyline *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.26.1.10 embArray\_addRect()** `int embArray_addRect (`  
    `EmbArray * a,`  
    `EmbRect b )`

Add a rectangle *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.26.1.11 embArray\_addStitch()** `int embArray_addStitch (`  
    `EmbArray * a,`  
    `EmbStitch b )`

Add a stitch *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.26.1.12 embArray\_addVector()** `int embArray_addVector (`  
    `EmbArray * a,`  
    `EmbVector b )`

Add a vector *b* to the EmbArray *a* and it returns if the element was successfully added.

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

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

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

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

**18.26.1.15 embArray\_free()** `void embArray_free (`  
    `EmbArray * a )`

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

**18.26.1.16 embArray\_resize()** `int embArray_resize (`  
    `EmbArray * a )`

Resizes the array *a* to be CHUNK\_SIZE entries longer if and only if the amount of room left is less than 3 entries.

## 18.27 extern/libembroidery/src/compress.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "embroidery_internal.h"
```

### Functions

- int [hus\\_compress](#) (char \*data, int length, char \*output, int \*output\_length)
- void [huffman\\_build\\_table](#) (huffman \*h)
- int \* [huffman\\_lookup](#) (huffman h, int byte\_lookup)
- void [compress\\_init](#) ()
- int [compress\\_get\\_bits](#) (compress \*c, int length)
- int [compress\\_pop](#) (compress \*c, int bit\_count)
- int [compress\\_peek](#) (compress \*c, int bit\_count)
- int [compress\\_read\\_variable\\_length](#) (compress \*c)
- void [compress\\_load\\_character\\_length\\_huffman](#) (compress \*c)
- void [compress\\_load\\_character\\_huffman](#) (compress \*c)
- void [compress\\_load\\_distance\\_huffman](#) (compress \*c)
- void [compress\\_load\\_block](#) (compress \*c)
- int [compress\\_get\\_token](#) (compress \*c)
- int [compress\\_get\\_position](#) (compress \*c)
- int [hus\\_decompress](#) (char \*data, int length, char \*output, int \*output\_length)

### Variables

- int [huffman\\_lookup\\_data](#) [2]

#### 18.27.1 Detailed Description

This is a work in progress.

Thanks to Jason Weiler for describing the binary formats of the HUS and VIP formats at:

<http://www.jasonweiler.com/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.

#### 18.27.2 Function Documentation

**18.27.2.1 [compress\\_get\\_bits\(\)](#)** int [compress\\_get\\_bits](#) (  
     compress \* c,  
     int length )

*c length* Returns .

**18.27.2.2 [compress\\_get\\_position\(\)](#)** int [compress\\_get\\_position](#) (  
     compress \* c )

*c* . Returns the position as an int.

**18.27.2.3 [compress\\_get\\_token\(\)](#)** int [compress\\_get\\_token](#) (  
     compress \* c )

*c* . Returns the token as an int.

**18.27.2.4 compress\_init()** void compress\_init ( )

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

c. Returns nothing.

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

Load character table to compress struct c. Returns nothing.

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

c. Returns.

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

c. Returns nothing.

**18.27.2.9 compress\_peek()** int compress\_peek (   
compress \* c,   
int bit\_count )

c bit\_count. Returns.

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

c bit\_count. Returns.

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

c. Returns.

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

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

**18.27.2.13 huffman\_lookup()** int \* huffman\_lookup (   
huffman h,   
int byte\_lookup )

Lookup *byte\_lookup* in huffman table *h* return result as two bytes using the memory *huffman\_lookup\_data*.

**18.27.2.14 hus\_compress()** int hus\_compress (   
char \* data,   
int length,   
char \* output,   
int \* output\_length )

*data* length output *output\_length*. Returns whether it was successful as an int.

This avoids the now unnecessary compression by placing a minimal header of 6 bytes and using only literals in the huffman compressed part (see the sources above).



**18.27.2.15 hus\_decompress()** `int hus_decompress (`  
     `char * data,`  
     `int length,`  
     `char * output,`  
     `int * output_length )`  
*data length output output\_length* . Returns whether the decompression was successful.

### 18.27.3 Variable Documentation

**18.27.3.1 huffman\_lookup\_data** `int huffman_lookup_data[2]`

## 18.28 extern/libembroidery/src/embroidery.h File Reference

### Classes

- struct [EmbColor\\_](#)
- struct [EmbVector\\_](#)
- struct [EmbImage\\_](#)
- struct [EmbBlock\\_](#)
- struct [EmbAlignedDim\\_](#)
- struct [EmbAngularDim\\_](#)
- struct [EmbArcLengthDim\\_](#)
- struct [EmbDiameterDim\\_](#)
- struct [EmbLeaderDim\\_](#)
- struct [EmbLinearDim\\_](#)
- struct [EmbOrdinateDim\\_](#)
- struct [EmbRadiusDim\\_](#)
- struct [EmbInfiniteLine\\_](#)
- struct [EmbRay\\_](#)
- struct [EmbTextMulti\\_](#)
- struct [EmbTextSingle\\_](#)
- struct [EmbTime\\_](#)
- struct [EmbPoint\\_](#)
- struct [EmbLine\\_](#)
- struct [EmbPath\\_](#)
- struct [EmbStitch\\_](#)
- struct [EmbThread\\_](#)
- struct [thread\\_color\\_](#)
- struct [EmbArc\\_](#)  
     *absolute position (not relative)*
- struct [EmbRect\\_](#)
- struct [EmbCircle\\_](#)
- struct [EmbSatinOutline\\_](#)
- struct [EmbEllipse\\_](#)
- struct [EmbBezier\\_](#)
- struct [EmbSpline\\_](#)
- struct [LSYSTEM](#)
- struct [EmbGeometry\\_](#)
- struct [EmbArray\\_](#)
- struct [EmbLayer\\_](#)
- struct [EmbPattern\\_](#)
- struct [EmbFormatList\\_](#)

## Macros

- #define [LIBEMBROIDERY\\_EMBEDDED\\_VERSION](#) 0
- #define [NORMAL](#) 0 /\*! stitch to (x, y) \*/
- #define [JUMP](#) 1 /\*! move to (x, y) \*/
- #define [TRIM](#) 2 /\*! trim + move to (x, y) \*/
- #define [STOP](#) 4 /\*! pause machine for thread change \*/
- #define [SEQUIN](#) 8 /\*! sequin \*/
- #define [END](#) 16 /\*! end of program \*/
- #define [EMB\\_FORMAT\\_100](#) 0
- #define [EMB\\_FORMAT\\_100](#) 1
- #define [EMB\\_FORMAT\\_ART](#) 2
- #define [EMB\\_FORMAT\\_BMC](#) 3
- #define [EMB\\_FORMAT\\_BRO](#) 4
- #define [EMB\\_FORMAT\\_CND](#) 5
- #define [EMB\\_FORMAT\\_COL](#) 6
- #define [EMB\\_FORMAT\\_CSD](#) 7
- #define [EMB\\_FORMAT\\_CSV](#) 8
- #define [EMB\\_FORMAT\\_DAT](#) 9
- #define [EMB\\_FORMAT\\_DEM](#) 10
- #define [EMB\\_FORMAT\\_DSB](#) 11
- #define [EMB\\_FORMAT\\_DST](#) 12
- #define [EMB\\_FORMAT\\_DSZ](#) 13
- #define [EMB\\_FORMAT\\_DXF](#) 14
- #define [EMB\\_FORMAT\\_EDR](#) 15
- #define [EMB\\_FORMAT\\_EMD](#) 16
- #define [EMB\\_FORMAT\\_EXP](#) 17
- #define [EMB\\_FORMAT\\_EXY](#) 18
- #define [EMB\\_FORMAT\\_EYS](#) 19
- #define [EMB\\_FORMAT\\_FXY](#) 20
- #define [EMB\\_FORMAT\\_GC](#) 21
- #define [EMB\\_FORMAT\\_GNC](#) 22
- #define [EMB\\_FORMAT\\_GT](#) 23
- #define [EMB\\_FORMAT\\_HUS](#) 24
- #define [EMB\\_FORMAT\\_INB](#) 25
- #define [EMB\\_FORMAT\\_INF](#) 26
- #define [EMB\\_FORMAT\\_JEF](#) 27
- #define [EMB\\_FORMAT\\_KSM](#) 28
- #define [EMB\\_FORMAT\\_MAX](#) 29
- #define [EMB\\_FORMAT\\_MIT](#) 30
- #define [EMB\\_FORMAT\\_NEW](#) 31
- #define [EMB\\_FORMAT\\_OFM](#) 32
- #define [EMB\\_FORMAT\\_PCD](#) 33
- #define [EMB\\_FORMAT\\_PCM](#) 34
- #define [EMB\\_FORMAT\\_PCQ](#) 35
- #define [EMB\\_FORMAT\\_PCS](#) 36
- #define [EMB\\_FORMAT\\_PEC](#) 37
- #define [EMB\\_FORMAT\\_PEL](#) 38
- #define [EMB\\_FORMAT\\_PEM](#) 39
- #define [EMB\\_FORMAT\\_PES](#) 40
- #define [EMB\\_FORMAT\\_PHB](#) 41
- #define [EMB\\_FORMAT\\_PHC](#) 42
- #define [EMB\\_FORMAT\\_PLT](#) 43
- #define [EMB\\_FORMAT\\_RGB](#) 44
- #define [EMB\\_FORMAT\\_SEW](#) 45

- `#define EMB_FORMAT_SHV` 46
- `#define EMB_FORMAT_SST` 47
- `#define EMB_FORMAT_STX` 48
- `#define EMB_FORMAT_SVG` 49
- `#define EMB_FORMAT_T01` 50
- `#define EMB_FORMAT_T09` 51
- `#define EMB_FORMAT_TAP` 52
- `#define EMB_FORMAT_THR` 53
- `#define EMB_FORMAT_TXT` 54
- `#define EMB_FORMAT_U00` 55
- `#define EMB_FORMAT_U01` 56
- `#define EMB_FORMAT_VIP` 57
- `#define EMB_FORMAT_VP3` 58
- `#define EMB_FORMAT_XXX` 59
- `#define EMB_FORMAT_ZSK` 60
- `#define Arc_Polyester` 0
- `#define Arc_Rayon` 1
- `#define CoatsAndClark_Rayon` 2
- `#define Exquisite_Polyester` 3
- `#define Fufu_Polyester` 4
- `#define Fufu_Rayon` 5
- `#define Hemingworth_Polyester` 6
- `#define Isacord_Polyester` 7
- `#define Isafil_Rayon` 8
- `#define Marathon_Polyester` 9
- `#define Marathon_Rayon` 10
- `#define Madeira_Polyester` 11
- `#define Madeira_Rayon` 12
- `#define Metro_Polyester` 13
- `#define Pantone` 14
- `#define RobisonAnton_Polyester` 15
- `#define RobisonAnton_Rayon` 16
- `#define Sigma_Polyester` 17
- `#define Sulky_Rayon` 18
- `#define ThreadArt_Rayon` 19
- `#define ThreadArt_Polyester` 20
- `#define ThreaDelight_Polyester` 21
- `#define Z102_Isacord_Polyester` 22
- `#define SVG_Colors` 23
- `#define hus_thread` 24
- `#define jef_thread` 25
- `#define pcm_thread` 26
- `#define pec_thread` 27
- `#define shv_thread` 28
- `#define dxf_color` 29
- `#define EMB_ARRAY` 0
- `#define EMB_ARC` 1
- `#define EMB_CIRCLE` 2
- `#define EMB_DIM_DIAMETER` 3
- `#define EMB_DIM_LEADER` 4
- `#define EMB_ELLIPSE` 5
- `#define EMB_FLAG` 6
- `#define EMB_LINE` 7
- `#define EMB_IMAGE` 8
- `#define EMB_PATH` 9

- #define `EMB_POINT` 10
- #define `EMB_POLYGON` 11
- #define `EMB_POLYLINE` 12
- #define `EMB_RECT` 13
- #define `EMB_SPLINE` 14
- #define `EMB_STITCH` 15
- #define `EMB_TEXT_SINGLE` 16
- #define `EMB_TEXT_MULTI` 17
- #define `EMB_VECTOR` 18
- #define `EMB_THREAD` 19
- #define `EMBFORMAT_UNSUPPORTED` 0
- #define `EMBFORMAT_STITCHONLY` 1
- #define `EMBFORMAT_OBJECTONLY` 2
- #define `EMBFORMAT_STCHANDOBJ` 3 /\* binary operation: 1+2=3 \*/
- #define `numberOfFormats` 61
- #define `CHUNK_SIZE` 128
- #define `EMB_MAX_LAYERS` 10
- #define `MAX_THREADS` 256
- #define `EMBFORMAT_MAXEXT` 3
- #define `EMBFORMAT_MAXDESC` 50
- #define `MAX_STITCHES` 1000000
- #define `EMB_PUBLIC`

## Typedefs

- typedef float `EmbReal`
- typedef struct `EmbColor_ EmbColor`
- typedef struct `EmbVector_ EmbVector`
- typedef struct `EmbArray_ EmbArray`
- typedef struct `EmbImage_ EmbImage`
- typedef struct `EmbBlock_ EmbBlock`
- typedef struct `EmbAlignedDim_ EmbAlignedDim`
- typedef struct `EmbAngularDim_ EmbAngularDim`
- typedef struct `EmbArcLengthDim_ EmbArcLengthDim`
- typedef struct `EmbDiameterDim_ EmbDiameterDim`
- typedef struct `EmbLeaderDim_ EmbLeaderDim`
- typedef struct `EmbLinearDim_ EmbLinearDim`
- typedef struct `EmbOrdinateDim_ EmbOrdinateDim`
- typedef struct `EmbRadiusDim_ EmbRadiusDim`
- typedef struct `EmbInfiniteLine_ EmbInfiniteLine`
- typedef struct `EmbRay_ EmbRay`
- typedef struct `EmbTextMulti_ EmbTextMulti`
- typedef struct `EmbTextSingle_ EmbTextSingle`
- typedef struct `EmbTime_ EmbTime`
- typedef struct `EmbPoint_ EmbPoint`
- typedef struct `EmbLine_ EmbLine`
- typedef struct `EmbPath_ EmbPath`
- typedef struct `EmbStitch_ EmbStitch`
- typedef struct `EmbThread_ EmbThread`
- typedef struct `thread_color_ thread_color`
- typedef struct `EmbArc_ EmbArc`  
*absolute position (not relative)*
- typedef struct `EmbRect_ EmbRect`
- typedef struct `EmbCircle_ EmbCircle`
- typedef `EmbPath EmbPolygon`

- typedef [EmbPath](#) [EmbPolyline](#)
- typedef int [EmbFlag](#)
- typedef struct [EmbSatinOutline\\_](#) [EmbSatinOutline](#)
- typedef struct [EmbEllipse\\_](#) [EmbEllipse](#)
- typedef struct [EmbBezier\\_](#) [EmbBezier](#)
- typedef struct [EmbSpline\\_](#) [EmbSpline](#)
- typedef struct [LSYSTEM](#) [L\\_system](#)
- typedef struct [EmbGeometry\\_](#) [EmbGeometry](#)
- typedef struct [EmbLayer\\_](#) [EmbLayer](#)
- typedef struct [EmbPattern\\_](#) [EmbPattern](#)
- typedef struct [EmbFormatList\\_](#) [EmbFormatList](#)

## Functions

- [EMB\\_PUBLIC](#) int [lindenmayer\\_system](#) ([L\\_system](#) L, char \*state, int iteration, int complete)
- [EMB\\_PUBLIC](#) int [hilbert\\_curve](#) ([EmbPattern](#) \*pattern, int iterations)
- [EMB\\_PUBLIC](#) int [emb\\_identify\\_format](#) (const char \*ending)  
*fileName*
- [EMB\\_PUBLIC](#) void [testMain](#) (int level)
- [EMB\\_PUBLIC](#) int [convert](#) (const char \*inf, const char \*outf)
- [EMB\\_PUBLIC](#) [EmbColor](#) [embColor\\_make](#) (unsigned char r, unsigned char g, unsigned char b)
- [EMB\\_PUBLIC](#) [EmbColor](#) \* [embColor\\_create](#) (unsigned char r, unsigned char g, unsigned char b)
- [EMB\\_PUBLIC](#) [EmbColor](#) [embColor\\_fromHexStr](#) (char \*val)  
*Converts a 6 digit hex string (I.E. "00FF00") into an EmbColor and returns it.*
- [EMB\\_PUBLIC](#) int [embColor\\_distance](#) ([EmbColor](#) a, [EmbColor](#) b)  
*a b*
- [EMB\\_PUBLIC](#) [EmbArray](#) \* [embArray\\_create](#) (int type)
- [EMB\\_PUBLIC](#) int [embArray\\_resize](#) ([EmbArray](#) \*g)
- [EMB\\_PUBLIC](#) void [embArray\\_copy](#) ([EmbArray](#) \*dst, [EmbArray](#) \*src)
- [EMB\\_PUBLIC](#) int [embArray\\_addArc](#) ([EmbArray](#) \*g, [EmbArc](#) arc)
- [EMB\\_PUBLIC](#) int [embArray\\_addCircle](#) ([EmbArray](#) \*g, [EmbCircle](#) circle)
- [EMB\\_PUBLIC](#) int [embArray\\_addEllipse](#) ([EmbArray](#) \*g, [EmbEllipse](#) ellipse)
- [EMB\\_PUBLIC](#) int [embArray\\_addFlag](#) ([EmbArray](#) \*g, int flag)
- [EMB\\_PUBLIC](#) int [embArray\\_addLine](#) ([EmbArray](#) \*g, [EmbLine](#) line)
- [EMB\\_PUBLIC](#) int [embArray\\_addRect](#) ([EmbArray](#) \*g, [EmbRect](#) rect)
- [EMB\\_PUBLIC](#) int [embArray\\_addPath](#) ([EmbArray](#) \*g, [EmbPath](#) p)
- [EMB\\_PUBLIC](#) int [embArray\\_addPoint](#) ([EmbArray](#) \*g, [EmbPoint](#) p)
- [EMB\\_PUBLIC](#) int [embArray\\_addPolygon](#) ([EmbArray](#) \*g, [EmbPolygon](#) p)
- [EMB\\_PUBLIC](#) int [embArray\\_addPolyline](#) ([EmbArray](#) \*g, [EmbPolyline](#) p)
- [EMB\\_PUBLIC](#) int [embArray\\_addStitch](#) ([EmbArray](#) \*g, [EmbStitch](#) st)
- [EMB\\_PUBLIC](#) int [embArray\\_addThread](#) ([EmbArray](#) \*g, [EmbThread](#) p)
- [EMB\\_PUBLIC](#) int [embArray\\_addVector](#) ([EmbArray](#) \*g, [EmbVector](#))
- [EMB\\_PUBLIC](#) void [embArray\\_free](#) ([EmbArray](#) \*p)
- [EMB\\_PUBLIC](#) [EmbLine](#) [embLine\\_make](#) ([EmbReal](#) x1, [EmbReal](#) y1, [EmbReal](#) x2, [EmbReal](#) y2)
- [EMB\\_PUBLIC](#) void [embLine\\_normalVector](#) ([EmbLine](#) line, [EmbVector](#) \*result, int clockwise)
- [EMB\\_PUBLIC](#) [EmbVector](#) [embLine\\_intersectionPoint](#) ([EmbLine](#) line1, [EmbLine](#) line2)
- [EMB\\_PUBLIC](#) int [embThread\\_findNearestColor](#) ([EmbColor](#) color, [EmbColor](#) \*colors, int n\_colors)
- [EMB\\_PUBLIC](#) int [embThread\\_findNearestThread](#) ([EmbColor](#) color, [EmbThread](#) \*threads, int n\_threads)  
*color thread\_list n\_threads*
- [EMB\\_PUBLIC](#) [EmbThread](#) [embThread\\_getRandom](#) (void)
- [EMB\\_PUBLIC](#) void [embVector\\_normalize](#) ([EmbVector](#) vector, [EmbVector](#) \*result)
- [EMB\\_PUBLIC](#) void [embVector\\_multiply](#) ([EmbVector](#) vector, [EmbReal](#) magnitude, [EmbVector](#) \*result)
- [EMB\\_PUBLIC](#) [EmbVector](#) [embVector\\_add](#) ([EmbVector](#) v1, [EmbVector](#) v2)
- [EMB\\_PUBLIC](#) [EmbVector](#) [embVector\\_average](#) ([EmbVector](#) v1, [EmbVector](#) v2)

- EMB\_PUBLIC EmbVector embVector\_subtract (EmbVector v1, EmbVector v2)
- EMB\_PUBLIC EmbReal embVector\_dot (EmbVector v1, EmbVector v2)
- EMB\_PUBLIC EmbReal embVector\_cross (EmbVector v1, EmbVector v2)
- The "cross product" as vectors a and b returned as a real value.*
- EMB\_PUBLIC void embVector\_transpose\_product (EmbVector v1, EmbVector v2, EmbVector \*result)
- EMB\_PUBLIC EmbReal embVector\_length (EmbVector vector)
- EMB\_PUBLIC EmbReal embVector\_relativeX (EmbVector a1, EmbVector a2, EmbVector a3)
- EMB\_PUBLIC EmbReal embVector\_relativeY (EmbVector a1, EmbVector a2, EmbVector a3)
- EMB\_PUBLIC EmbReal embVector\_angle (EmbVector v)
- EMB\_PUBLIC EmbReal embVector\_distance (EmbVector a, EmbVector b)
- EMB\_PUBLIC EmbVector embVector\_unit (EmbReal angle)
- EMB\_PUBLIC EmbArc embArc\_init (void)
- EMB\_PUBLIC char embArc\_clockwise (EmbArc arc)
- EMB\_PUBLIC void getArcCenter (EmbArc arc, EmbVector \*arcCenter)
- EMB\_PUBLIC char getArcDataFromBulge (EmbReal bulge, EmbArc \*arc, EmbReal \*arcCenterX, EmbReal \*arcCenterY, EmbReal \*radius, EmbReal \*diameter, EmbReal \*chord, EmbReal \*chordMidX, EmbReal \*chordMidY, EmbReal \*sagitta, EmbReal \*apothem, EmbReal \*incAngleInDegrees, char \*clockwise)
- EMB\_PUBLIC EmbCircle embCircle\_init (void)
- EMB\_PUBLIC int getCircleCircleIntersections (EmbCircle c0, EmbCircle c1, EmbVector \*v0, EmbVector \*v1)
- EMB\_PUBLIC int getCircleTangentPoints (EmbCircle c, EmbVector p, EmbVector \*v0, EmbVector \*v1)
- EMB\_PUBLIC EmbEllipse embEllipse\_init (void)
- EMB\_PUBLIC EmbEllipse embEllipse\_make (EmbReal cx, EmbReal cy, EmbReal rx, EmbReal ry)
- EMB\_PUBLIC EmbReal embEllipse\_diameterX (EmbEllipse ellipse)
- EMB\_PUBLIC EmbReal embEllipse\_diameterY (EmbEllipse ellipse)
- EMB\_PUBLIC EmbReal embEllipse\_width (EmbEllipse ellipse)
- EMB\_PUBLIC EmbReal embEllipse\_height (EmbEllipse ellipse)
- EMB\_PUBLIC EmbReal embEllipse\_area (EmbEllipse ellipse)
- EMB\_PUBLIC EmbReal embEllipse\_perimeter (EmbEllipse ellipse)
- EMB\_PUBLIC EmblImage emblImage\_create (int, int)
- EMB\_PUBLIC void emblImage\_read (EmblImage \*image, char \*fname)
- EMB\_PUBLIC int emblImage\_write (EmblImage \*image, char \*fname)
- EMB\_PUBLIC void emblImage\_free (EmblImage \*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.*

## 18.28.1 Macro Definition Documentation

**18.28.1.1 Arc\_Polyester** #define Arc\_Polyester 0

**18.28.1.2 Arc\_Rayon** #define Arc\_Rayon 1

**18.28.1.3 CHUNK\_SIZE** #define CHUNK\_SIZE 128

**18.28.1.4 CoatsAndClark\_Rayon** #define CoatsAndClark\_Rayon 2

**18.28.1.5 dxf\_color** #define dxf\_color 29

**18.28.1.6 EMB\_ARC** #define EMB\_ARC 1

**18.28.1.7 EMB\_ARRAY** #define EMB\_ARRAY 0



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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**18.28.1.24 EMB\_FORMAT\_DSB** `#define EMB_FORMAT_DSB 11`

**18.28.1.25 EMB\_FORMAT\_DST** `#define EMB_FORMAT_DST 12`

**18.28.1.26 EMB\_FORMAT\_DSZ** #define EMB\_FORMAT\_DSZ 13

**18.28.1.27 EMB\_FORMAT\_DXF** #define EMB\_FORMAT\_DXF 14

**18.28.1.28 EMB\_FORMAT\_EDR** #define EMB\_FORMAT\_EDR 15

**18.28.1.29 EMB\_FORMAT\_EMD** #define EMB\_FORMAT\_EMD 16

**18.28.1.30 EMB\_FORMAT\_EXP** #define EMB\_FORMAT\_EXP 17

**18.28.1.31 EMB\_FORMAT\_EXY** #define EMB\_FORMAT\_EXY 18

**18.28.1.32 EMB\_FORMAT\_EYS** #define EMB\_FORMAT\_EYS 19

**18.28.1.33 EMB\_FORMAT\_FXY** #define EMB\_FORMAT\_FXY 20

**18.28.1.34 EMB\_FORMAT\_GC** #define EMB\_FORMAT\_GC 21

**18.28.1.35 EMB\_FORMAT\_GNC** #define EMB\_FORMAT\_GNC 22

**18.28.1.36 EMB\_FORMAT\_GT** #define EMB\_FORMAT\_GT 23

**18.28.1.37 EMB\_FORMAT\_HUS** #define EMB\_FORMAT\_HUS 24

**18.28.1.38 EMB\_FORMAT\_INB** #define EMB\_FORMAT\_INB 25

**18.28.1.39 EMB\_FORMAT\_INF** #define EMB\_FORMAT\_INF 26

**18.28.1.40 EMB\_FORMAT\_JEF** #define EMB\_FORMAT\_JEF 27

**18.28.1.41 EMB\_FORMAT\_KSM** #define EMB\_FORMAT\_KSM 28

**18.28.1.42 EMB\_FORMAT\_MAX** #define EMB\_FORMAT\_MAX 29

**18.28.1.43 EMB\_FORMAT\_MIT** #define EMB\_FORMAT\_MIT 30

- 18.28.1.44 EMB\_FORMAT\_NEW** `#define EMB_FORMAT_NEW 31`
- 18.28.1.45 EMB\_FORMAT\_OFM** `#define EMB_FORMAT_OFM 32`
- 18.28.1.46 EMB\_FORMAT\_PCD** `#define EMB_FORMAT_PCD 33`
- 18.28.1.47 EMB\_FORMAT\_PCM** `#define EMB_FORMAT_PCM 34`
- 18.28.1.48 EMB\_FORMAT\_PCQ** `#define EMB_FORMAT_PCQ 35`
- 18.28.1.49 EMB\_FORMAT\_PCS** `#define EMB_FORMAT_PCS 36`
- 18.28.1.50 EMB\_FORMAT\_PEC** `#define EMB_FORMAT_PEC 37`
- 18.28.1.51 EMB\_FORMAT\_PEL** `#define EMB_FORMAT_PEL 38`
- 18.28.1.52 EMB\_FORMAT\_PEM** `#define EMB_FORMAT_PEM 39`
- 18.28.1.53 EMB\_FORMAT\_PES** `#define EMB_FORMAT_PES 40`
- 18.28.1.54 EMB\_FORMAT\_PHB** `#define EMB_FORMAT_PHB 41`
- 18.28.1.55 EMB\_FORMAT\_PHC** `#define EMB_FORMAT_PHC 42`
- 18.28.1.56 EMB\_FORMAT\_PLT** `#define EMB_FORMAT_PLT 43`
- 18.28.1.57 EMB\_FORMAT\_RGB** `#define EMB_FORMAT_RGB 44`
- 18.28.1.58 EMB\_FORMAT\_SEW** `#define EMB_FORMAT_SEW 45`
- 18.28.1.59 EMB\_FORMAT\_SHV** `#define EMB_FORMAT_SHV 46`
- 18.28.1.60 EMB\_FORMAT\_SST** `#define EMB_FORMAT_SST 47`
- 18.28.1.61 EMB\_FORMAT\_STX** `#define EMB_FORMAT_STX 48`

**18.28.1.62 EMB\_FORMAT\_SVG** #define EMB\_FORMAT\_SVG 49

**18.28.1.63 EMB\_FORMAT\_T01** #define EMB\_FORMAT\_T01 50

**18.28.1.64 EMB\_FORMAT\_T09** #define EMB\_FORMAT\_T09 51

**18.28.1.65 EMB\_FORMAT\_TAP** #define EMB\_FORMAT\_TAP 52

**18.28.1.66 EMB\_FORMAT\_THR** #define EMB\_FORMAT\_THR 53

**18.28.1.67 EMB\_FORMAT\_TXT** #define EMB\_FORMAT\_TXT 54

**18.28.1.68 EMB\_FORMAT\_U00** #define EMB\_FORMAT\_U00 55

**18.28.1.69 EMB\_FORMAT\_U01** #define EMB\_FORMAT\_U01 56

**18.28.1.70 EMB\_FORMAT\_VIP** #define EMB\_FORMAT\_VIP 57

**18.28.1.71 EMB\_FORMAT\_VP3** #define EMB\_FORMAT\_VP3 58

**18.28.1.72 EMB\_FORMAT\_XXX** #define EMB\_FORMAT\_XXX 59

**18.28.1.73 EMB\_FORMAT\_ZSK** #define EMB\_FORMAT\_ZSK 60

**18.28.1.74 EMB\_IMAGE** #define EMB\_IMAGE 8

**18.28.1.75 EMB\_LINE** #define EMB\_LINE 7

**18.28.1.76 EMB\_MAX\_LAYERS** #define EMB\_MAX\_LAYERS 10

**18.28.1.77 EMB\_PATH** #define EMB\_PATH 9

**18.28.1.78 EMB\_POINT** #define EMB\_POINT 10

**18.28.1.79 EMB\_POLYGON** #define EMB\_POLYGON 11

**18.28.1.80 EMB\_POLYLINE** `#define EMB_POLYLINE 12`

**18.28.1.81 EMB\_PUBLIC** `#define EMB_PUBLIC`

**18.28.1.82 EMB\_RECT** `#define EMB_RECT 13`

**18.28.1.83 EMB\_SPLINE** `#define EMB_SPLINE 14`

**18.28.1.84 EMB\_STITCH** `#define EMB_STITCH 15`

**18.28.1.85 EMB\_TEXT\_MULTI** `#define EMB_TEXT_MULTI 17`

**18.28.1.86 EMB\_TEXT\_SINGLE** `#define EMB_TEXT_SINGLE 16`

**18.28.1.87 EMB\_THREAD** `#define EMB_THREAD 19`

**18.28.1.88 EMB\_VECTOR** `#define EMB_VECTOR 18`

**18.28.1.89 EMBFORMAT\_MAXDESC** `#define EMBFORMAT_MAXDESC 50`

**18.28.1.90 EMBFORMAT\_MAXEXT** `#define EMBFORMAT_MAXEXT 3`

**18.28.1.91 EMBFORMAT\_OBJECTONLY** `#define EMBFORMAT_OBJECTONLY 2`

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

**18.28.1.93 EMBFORMAT\_STITCHONLY** `#define EMBFORMAT_STITCHONLY 1`

**18.28.1.94 EMBFORMAT\_UNSUPPORTED** `#define EMBFORMAT_UNSUPPORTED 0`

**18.28.1.95 END** `#define END 16 /*! end of program */`

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**18.28.1.114 numberOfFormats** `#define numberOfFormats 61`

**18.28.1.115 Pantone** `#define Pantone 14`

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## 18.28.2 Typedef Documentation

**18.28.2.1 EmbAlignedDim** `typedef struct EmbAlignedDim\_ EmbAlignedDim`

**18.28.2.2 EmbAngularDim** `typedef struct EmbAngularDim\_ EmbAngularDim`

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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



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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**18.28.2.37 EmbTime** typedef struct [EmbTime\\_](#) EmbTime

**18.28.2.38 EmbVector** typedef struct [EmbVector\\_](#) EmbVector

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

**18.28.2.39 L\_system** typedef struct LSYSTEM L\_system

**18.28.2.40 thread\_color** typedef struct thread\_color\_ thread\_color

### 18.28.3 Function Documentation

**18.28.3.1 convert()** EMB\_PUBLIC int convert (  
    const char \* *inf*,  
    const char \* *outf* )

**18.28.3.2 degrees()** EMB\_PUBLIC EmbReal degrees (  
    EmbReal *radian* )

**18.28.3.3 emb\_identify\_format()** EMB\_PUBLIC int emb\_identify\_format (  
    const char \* *fileName* )

*fileName*

Returns

int

**18.28.3.4 emb\_round()** EMB\_PUBLIC int emb\_round (  
    EmbReal *x* )

**18.28.3.5 embArc\_clockwise()** EMB\_PUBLIC char embArc\_clockwise (  
    EmbArc *arc* )

**18.28.3.6 embArc\_init()** EMB\_PUBLIC EmbArc embArc\_init (  
    void )

**18.28.3.7 embArray\_addArc()** EMB\_PUBLIC int embArray\_addArc (  
    EmbArray \* *a*,  
    EmbArc *b* )

Add an arc *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.28.3.8 embArray\_addCircle()** EMB\_PUBLIC int embArray\_addCircle (  
    EmbArray \* *a*,  
    EmbCircle *b* )

Add a circle *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.28.3.9 embArray\_addEllipse()** EMB\_PUBLIC int embArray\_addEllipse (  
    EmbArray \* *a*,  
    EmbEllipse *b* )

Add an ellipse *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.28.3.10** `embArray_addFlag()` `EMB_PUBLIC int embArray_addFlag (`  
    `EmbArray * a,`  
    `EmbFlag b )`

Add a flag *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.28.3.11** `embArray_addLine()` `EMB_PUBLIC int embArray_addLine (`  
    `EmbArray * a,`  
    `EmbLine b )`

Add a line *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.28.3.12** `embArray_addPath()` `EMB_PUBLIC int embArray_addPath (`  
    `EmbArray * a,`  
    `EmbPath b )`

Add a path *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.28.3.13** `embArray_addPoint()` `EMB_PUBLIC int embArray_addPoint (`  
    `EmbArray * a,`  
    `EmbPoint b )`

Add a point *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.28.3.14** `embArray_addPolygon()` `EMB_PUBLIC int embArray_addPolygon (`  
    `EmbArray * a,`  
    `EmbPolygon b )`

Add a polygon *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.28.3.15** `embArray_addPolyline()` `EMB_PUBLIC int embArray_addPolyline (`  
    `EmbArray * a,`  
    `EmbPolyline b )`

Add a polyline *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.28.3.16** `embArray_addRect()` `EMB_PUBLIC int embArray_addRect (`  
    `EmbArray * a,`  
    `EmbRect b )`

Add a rectangle *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.28.3.17** `embArray_addStitch()` `EMB_PUBLIC int embArray_addStitch (`  
    `EmbArray * a,`  
    `EmbStitch b )`

Add a stitch *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.28.3.18** `embArray_addThread()` `EMB_PUBLIC int embArray_addThread (`  
    `EmbArray * g,`  
    `EmbThread p )`

**18.28.3.19** `embArray_addVector()` `EMB_PUBLIC int embArray_addVector (`  
    `EmbArray * a,`  
    `EmbVector b )`

Add a vector *b* to the EmbArray *a* and it returns if the element was successfully added.

**18.28.3.20** `embArray_copy()` `EMB_PUBLIC void embArray_copy (`  
    `EmbArray * dst,`  
    `EmbArray * src )`

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

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

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

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

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

**18.28.3.23 embArray\_resize()** `EMB_PUBLIC int embArray_resize (`  
`EmbArray * a )`

Resizes the array *a* to be CHUNK\_SIZE entries longer if and only if the amount of room left is less than 3 entries.

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

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

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

*a b*

Returns

int

**18.28.3.27 embColor\_fromHexStr()** `EMB_PUBLIC EmbColor embColor_fromHexStr (`  
`char * val )`

Converts a 6 digit hex string (I.E. "00FF00") into an EmbColor and returns it.  
*val* 6 byte code describing the color as a hex string, doesn't require null termination.

Returns

EmbColor the same color as our internal type.

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

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

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

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

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

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

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

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

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

**18.28.3.37 embGeometry\_boundingRect()** `EMB_PUBLIC EmbRect embGeometry_boundingRect ( EmbGeometry * obj )`

Calculate the bounding box of geometry *obj* based on what kind of geometric object it is.  
*obj* A pointer to the geometry memory.

#### Returns

EmbRect The bounding box in the same scale as the input geometry.

In the case of a failure the bounding box returned is always the unit square with top left corner at (0, 0).

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

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

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

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

#### Returns

EmbGeometry\*

**18.28.3.40 embGeometry\_move()** `EMB_PUBLIC void embGeometry_move ( EmbGeometry * obj, EmbVector delta )`

Translate *obj* by the vector *delta*.  
*obj* A pointer to the geometry memory. *delta* A vector in the 0.1mm scale to offset the geometry by.

**18.28.3.41 embGeometry\_vulcanize()** `EMB_PUBLIC void embGeometry_vulcanize ( EmbGeometry * obj )`

Toggle the rubber mode of the object.  
*obj*

**Todo** Review. This could be controlled by a simple flag.

**18.28.3.42 embImage\_create()** `EMB_PUBLIC EmbImage embImage_create ( int , int )`

**18.28.3.43 embImage\_free()** `EMB_PUBLIC void embImage_free ( EmbImage * image )`

**18.28.3.44 embImage\_read()** `EMB_PUBLIC void embImage_read ( EmbImage * image, char * fname )`

**18.28.3.45 embImage\_write()** `EMB_PUBLIC int embImage_write ( EmbImage * image, char * fname )`

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

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

**18.28.3.48 embLine\_normalVector()** `EMB_PUBLIC void embLine_normalVector ( EmbLine line, EmbVector * result, int clockwise )`

Finds the normalized vector perpendicular (clockwise) to the line given by v1->v2 (normal to the line)

**18.28.3.49 embPattern\_addCircleAbs()** `EMB_PUBLIC void embPattern_addCircleAbs ( EmbPattern * p, EmbCircle circle )`

Adds a circle object to pattern (*p*) with its center at the absolute position (*cx,cy*) with a radius of (*r*). Positive y is up. Units are in millimeters.

**18.28.3.50 embPattern\_addEllipseAbs()** `EMB_PUBLIC void embPattern_addEllipseAbs ( EmbPattern * p, EmbEllipse ellipse )`

Adds an ellipse object to pattern (*p*) with its center at the absolute position (*cx,cy*) with radii of (*rx,ry*). Positive y is up. Units are in millimeters.

**18.28.3.51 embPattern\_addLineAbs()** `EMB_PUBLIC void embPattern_addLineAbs (`  
`EmbPattern * p,`  
`EmbLine line )`

Adds a line object to pattern (*p*) starting at the absolute position (*x1,y1*) and ending at the absolute position (*x2,y2*). Positive y is up. Units are in millimeters.

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

**18.28.3.53 embPattern\_addPointAbs()** `EMB_PUBLIC void embPattern_addPointAbs (`  
`EmbPattern * p,`  
`EmbPoint obj )`

Adds a point object to pattern (*p*) at the absolute position (*x,y*). Positive y is up. Units are in millimeters.

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

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

**18.28.3.56 embPattern\_addRectAbs()** `EMB_PUBLIC void embPattern_addRectAbs (`  
`EmbPattern * p,`  
`EmbRect rect )`

Adds a rectangle object to pattern (*p*) at the absolute position (*x,y*) with a width of (*w*) and a height of (*h*). Positive y is up. Units are in millimeters.

**18.28.3.57 embPattern\_addStitchAbs()** `EMB_PUBLIC void embPattern_addStitchAbs (`  
`EmbPattern * p,`  
`EmbReal x,`  
`EmbReal y,`  
`int flags,`  
`int isAutoColorIndex )`

Adds a stitch to the pattern (*p*) at the absolute position (*x,y*). Positive y is up. Units are in millimeters.

**18.28.3.58 embPattern\_addStitchRel()** `EMB_PUBLIC void embPattern_addStitchRel (`  
`EmbPattern * p,`  
`EmbReal dx,`  
`EmbReal dy,`  
`int flags,`  
`int isAutoColorIndex )`

Adds a stitch to the pattern (*p*) at the relative position (*dx,dy*) to the previous stitch. Positive y is up. Units are in millimeters.

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

*pattern thread*

Returns

int

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

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

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

Center the pattern *p*.

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

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

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

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

*p1 p2*

Returns

EmbPattern\*

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

*p*

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

*p*

**18.28.3.67 embPattern\_copyPolylinesToStitchList()** `EMB_PUBLIC void embPattern_copyPolylinesTo↔StitchList ( EmbPattern * pattern )`

**18.28.3.68 embPattern\_copyStitchListToPolylines()** `EMB_PUBLIC void embPattern_copyStitchListTo↔Polylines ( EmbPattern * pattern )`

**18.28.3.69 embPattern\_correctForMaxStitchLength()** `EMB_PUBLIC void embPattern_correctForMax↔StitchLength ( EmbPattern * p, EmbReal maxStitchLength, EmbReal maxJumpLength )`

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



**18.28.3.70 embPattern\_create()** `EMB_PUBLIC EmbPattern * embPattern_create ( void )`

Returns a pointer to an EmbPattern. It is created on the heap. The caller is responsible for freeing the allocated memory with `embPattern_free()`.

Returns

EmbPattern\*

**18.28.3.71 embPattern\_crossstitch()** `EMB_PUBLIC void embPattern_crossstitch ( EmbPattern * pattern, EmbImage * image, int threshold )`

*pattern image threshold*

Uses a threshold method to determine where to put crosses in the fill.

To improve this, we can remove the vertical stitches when two crosses neighbour. Currently the simple way to do this is to chain crosses that are neighbours exactly one ahead.

**18.28.3.72 embPattern\_designDetails()** `EMB_PUBLIC void embPattern_designDetails ( EmbPattern * p )`

**18.28.3.73 embPattern\_end()** `EMB_PUBLIC void embPattern_end ( EmbPattern * p )`

**18.28.3.74 embPattern\_fixColorCount()** `EMB_PUBLIC void embPattern_fixColorCount ( EmbPattern * p )`

*p*

**18.28.3.75 embPattern\_flip()** `EMB_PUBLIC void embPattern_flip ( EmbPattern * p, int horz, int vert )`

Flips the entire pattern (*p*) horizontally about the x-axis if (*horz*) is true. Flips the entire pattern (*p*) vertically about the y-axis if (*vert*) is true.

**18.28.3.76 embPattern\_flipHorizontal()** `EMB_PUBLIC void embPattern_flipHorizontal ( EmbPattern * p )`

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

**18.28.3.77 embPattern\_flipVertical()** `EMB_PUBLIC void embPattern_flipVertical ( EmbPattern * p )`

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

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

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

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

*p length*

**18.28.3.80 embPattern\_horizontal\_fill()** `EMB_PUBLIC void embPattern_horizontal_fill (`  
`EmbPattern * pattern,`  
`EmbImage * image,`  
`int threshold )`

*pattern image threshold*

Uses a threshold method to determine where to put lines in the fill.

Needs to pass a "donut test", i.e. an image with black pixels where:  $10 < x*x + y*y < 20$  over the area  $(-30, 30) \times (-30, 30)$ .

Use render then image difference to see how well it passes.

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

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

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

TODO: Description needed.

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

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

**18.28.3.86 embPattern\_movePolylinesToStitchList()** `EMB_PUBLIC void embPattern_movePolylinesTo↔`  
`StitchList (`  
`EmbPattern * pattern )`

**18.28.3.87 embPattern\_moveStitchListToPolylines()** `EMB_PUBLIC void embPattern_moveStitchListTo↔`  
`Polylines (`  
`EmbPattern * pattern )`

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

*pattern fileName format*

Returns

char

**18.28.3.89** `embPattern_readAuto()` `EMB_PUBLIC` `char embPattern_readAuto (`  
    `EmbPattern * pattern,`  
    `const char * fileName )`

*pattern fileName*

Returns

char

**18.28.3.90** `embPattern_realStitches()` `EMB_PUBLIC` `int embPattern_realStitches (`  
    `EmbPattern * pattern )`

**18.28.3.91** `embPattern_render()` `EMB_PUBLIC` `int embPattern_render (`  
    `EmbPattern * pattern,`  
    `char * fname )`

**18.28.3.92** `embPattern_scale()` `EMB_PUBLIC` `void embPattern_scale (`  
    `EmbPattern * p,`  
    `EmbReal scale )`

Very simple scaling of the x and y axis for every point. Doesn't insert or delete stitches to preserve density.

**18.28.3.93** `embPattern_simulate()` `EMB_PUBLIC` `int embPattern_simulate (`  
    `EmbPattern * pattern,`  
    `char * fname )`

**18.28.3.94** `embPattern_totalStitchLength()` `EMB_PUBLIC` `EmbReal embPattern_totalStitchLength (`  
    `EmbPattern * pattern )`

*pattern*

Returns

float

**18.28.3.95** `embPattern_trimStitches()` `EMB_PUBLIC` `int embPattern_trimStitches (`  
    `EmbPattern * pattern )`

**18.28.3.96** `embPattern_write()` `EMB_PUBLIC` `char embPattern_write (`  
    `EmbPattern * pattern,`  
    `const char * fileName,`  
    `int format )`

*pattern fileName format*

Returns

char

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

*pattern fileName*

Returns

char

**18.28.3.98 embRect\_area()** `EMB_PUBLIC EmbReal embRect_area (`  
`EmbRect rect )`

**18.28.3.99 embRect\_init()** `EMB_PUBLIC EmbRect embRect_init (`  
`void )`

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

*lines thickness result*

**18.28.3.101 embSatinOutline\_renderStitches()** `EMB_PUBLIC EmbArray * embSatinOutline_render↵`  
`Stitches (`  
`EmbSatinOutline * result,`  
`EmbReal density )`

*result density*

Returns

EmbArray\*

**18.28.3.102 embThread\_findNearestColor()** `EMB_PUBLIC int embThread_findNearestColor (`  
`EmbColor color,`  
`EmbColor * color_list,`  
`int n_colors )`

Returns the closest color to the required color based on a list of available threads. The algorithm is a simple least squares search against the list. If the (square of) Euclidean 3-dimensional distance between the points in (red, green, blue) space is smaller then the index is saved and the remaining index is returned to the caller. *color* The EmbColor color to match. *colors* The EmbThreadList pointer to start the search at. *mode* Is the argument an array of threads (0) or colors (1)?

Returns

closestIndex The entry in the ThreadList that matches.

**18.28.3.103 embThread\_findNearestThread()** `EMB_PUBLIC int embThread_findNearestThread (`  
`EmbColor color,`  
`EmbThread * thread_list,`  
`int n_threads )`

*color thread\_list n\_threads*

## Returns

int

**18.28.3.104 embThread\_getRandom()** `EMB_PUBLIC EmbThread embThread_getRandom ( void )`

Returns a random thread color, useful in filling in cases where the actual color of the thread doesn't matter but one needs to be declared to test or render a pattern.

## Returns

c The resulting color.

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

*t*

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

*t*

## Returns

EmbTime

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

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

Equivalent to:

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

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

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

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

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

Equivalent to:

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

**18.28.3.110 embVector\_cross()** `EMB_PUBLIC EmbReal embVector_cross (`  
`EmbVector a,`  
`EmbVector b )`

The "cross product" as vectors *a* and *b* returned as a real value.

Technically, this is the magnitude of the cross product when the embroidery is placed in the  $z=0$  plane (since the cross product is defined for 3-dimensional vectors). That is:

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

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

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

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

**18.28.3.112 embVector\_dot()** `EMB_PUBLIC EmbReal embVector_dot (`  
`EmbVector a,`  
`EmbVector b )`

The dot product as vectors *v1* and *v2* returned as a `EmbReal`.

Equivalent to:

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

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

The length or absolute value of the vector *vector*.

Equivalent to:

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

**18.28.3.114 embVector\_multiply()** `EMB_PUBLIC void embVector_multiply (`  
`EmbVector vector,`  
`EmbReal magnitude,`  
`EmbVector * result )`

The scalar multiple *magnitude* of a vector *vector*. Returned as *result*.

**Todo** make result return argument.

**18.28.3.115 embVector\_normalize()** `EMB_PUBLIC void embVector_normalize (`  
`EmbVector vector,`  
`EmbVector * result )`

Finds the unit length vector *result* in the same direction as *vector*.

Equivalent to:

$$\mathbf{u} = \frac{v}{|\mathbf{v}|}$$

**Todo** make result return argument.

**18.28.3.116** `embVector_relativeX()` `EMB_PUBLIC EmbReal embVector_relativeX (`  
`EmbVector a1,`  
`EmbVector a2,`  
`EmbVector a3 )`

The x-component of the vector

**18.28.3.117** `embVector_relativeY()` `EMB_PUBLIC EmbReal embVector_relativeY (`  
`EmbVector a1,`  
`EmbVector a2,`  
`EmbVector a3 )`

The y-component of the vector

**18.28.3.118** `embVector_subtract()` `EMB_PUBLIC EmbVector embVector_subtract (`  
`EmbVector v1,`  
`EmbVector v2 )`

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

Equivalent to:

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

**18.28.3.119** `embVector_transpose_product()` `EMB_PUBLIC void embVector_transpose_product (`  
`EmbVector v1,`  
`EmbVector v2,`  
`EmbVector * result )`

Since we aren't using full vector algebra here, all vectors are "vertical". so this is like the product  $\mathbf{v1}^T \mathbf{v2}$  for our vectors *v1* and *v2* so a "component-wise product". The result is stored at the pointer *result*.

That is  $\begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} xa \\ yb \end{pmatrix}$

**18.28.3.120** `embVector_unit()` `EMB_PUBLIC EmbVector embVector_unit (`  
`EmbReal alpha )`

The unit vector in the direction *angle*.

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

**18.28.3.121** `full_test_matrix()` `EMB_PUBLIC int full_test_matrix (`  
`char * fname )`

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

**18.28.3.123** `getArcDataFromBulge()` `EMB_PUBLIC char getArcDataFromBulge (`  
`EmbReal bulge,`  
`EmbArc * arc,`  
`EmbReal * arcCenterX,`  
`EmbReal * arcCenterY,`  
`EmbReal * radius,`  
`EmbReal * diameter,`  
`EmbReal * chord,`  
`EmbReal * chordMidX,`  
`EmbReal * chordMidY,`

```
EmbReal * sagitta,  
EmbReal * apothem,  
EmbReal * incAngleInDegrees,  
char * clockwise )
```

**18.28.3.124** **getCircleCircleIntersections()** `EMB_PUBLIC int getCircleCircleIntersections (`  
`EmbCircle c0,`  
`EmbCircle c1,`  
`EmbVector * v0,`  
`EmbVector * v1 )`

**18.28.3.125** **getCircleTangentPoints()** `EMB_PUBLIC int getCircleTangentPoints (`  
`EmbCircle c,`  
`EmbVector p,`  
`EmbVector * v0,`  
`EmbVector * v1 )`

**18.28.3.126** **hilbert\_curve()** `EMB_PUBLIC int hilbert_curve (`  
`EmbPattern * pattern,`  
`int iterations )`

*pattern iterations*

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

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

**18.28.3.127** **lindenmayer\_system()** `EMB_PUBLIC int lindenmayer_system (`  
`L_system L,`  
`char * state,`  
`int iterations,`  
`int complete )`

*L state iterations complete*

Returns

int

This is a slow generation algorithm.

**18.28.3.128** **radians()** `EMB_PUBLIC EmbReal radians (`  
`EmbReal degree )`

**18.28.3.129** **report()** `EMB_PUBLIC void report (`  
`int result,`  
`char * label )`

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

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



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

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

## 18.28.4 Variable Documentation

**18.28.4.1 \_dxfColorTable** `const unsigned char _dxfColorTable[][3] [extern]`

**18.28.4.2 black\_thread** `EmbThread black_thread [extern]`

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

**18.28.4.4 emb\_verbose** `int emb_verbose [extern]`  
 Verbosity level.

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

**18.28.4.6 formatTable** `EmbFormatList formatTable[numberOfFormats] [extern]`

This file is part of libembroidery.

Copyright 2018-2022 The Embroidermodder Team Licensed under the terms of the zlib license.

This file contains all the read and write functions for the library.

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

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

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

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

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

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

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

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

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

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

## 18.29 embroidery.h

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

```
1 #ifndef LIBEMBROIDERY_HEADER__
2 #define LIBEMBROIDERY_HEADER__
3
4 #ifdef __cplusplus
5 extern "C" {
6 #endif
7
16 #ifndef LIBEMBROIDERY_EMBEDDED_VERSION
17 #define LIBEMBROIDERY_EMBEDDED_VERSION 0
18 #endif
19
20 /* MACROS
21 *****/
22
26 #define NORMAL                0
27 #define JUMP                   1
28 #define TRIM                   2
29 #define STOP                   4
30 #define SEQUIN                 8
31 #define END                     16
36 #define EMB_FORMAT_100         0
37 #define EMB_FORMAT_100         1
38 #define EMB_FORMAT_ART         2
39 #define EMB_FORMAT_BMC         3
40 #define EMB_FORMAT_BRO         4
41 #define EMB_FORMAT_CND         5
42 #define EMB_FORMAT_COL         6
43 #define EMB_FORMAT_CSD         7
44 #define EMB_FORMAT_CSV         8
45 #define EMB_FORMAT_DAT         9
46 #define EMB_FORMAT_DEM        10
47 #define EMB_FORMAT_DSB        11
48 #define EMB_FORMAT_DST        12
49 #define EMB_FORMAT_DSZ        13
50 #define EMB_FORMAT_DXF        14
51 #define EMB_FORMAT_EDR        15
52 #define EMB_FORMAT_EMD        16
53 #define EMB_FORMAT_EXP        17
54 #define EMB_FORMAT_EXY        18
55 #define EMB_FORMAT_EYS        19
56 #define EMB_FORMAT_FXY        20
57 #define EMB_FORMAT_GC         21
58 #define EMB_FORMAT_GNC        22
59 #define EMB_FORMAT_GT         23
60 #define EMB_FORMAT_HUS        24
61 #define EMB_FORMAT_INB        25
62 #define EMB_FORMAT_INF        26
63 #define EMB_FORMAT_JEF        27
64 #define EMB_FORMAT_KSM        28
65 #define EMB_FORMAT_MAX        29
66 #define EMB_FORMAT_MIT        30
67 #define EMB_FORMAT_NEW        31
68 #define EMB_FORMAT_OFM        32
69 #define EMB_FORMAT_PCD        33
70 #define EMB_FORMAT_PCM        34
71 #define EMB_FORMAT_PCQ        35
72 #define EMB_FORMAT_PCS        36
73 #define EMB_FORMAT_PEC        37
74 #define EMB_FORMAT_PEL        38
75 #define EMB_FORMAT_PEM        39
76 #define EMB_FORMAT_PES        40
77 #define EMB_FORMAT_PHB        41
78 #define EMB_FORMAT_PHC        42
79 #define EMB_FORMAT_PLT        43
80 #define EMB_FORMAT_RGB        44
81 #define EMB_FORMAT_SEW        45
82 #define EMB_FORMAT_SHV        46
```

```

83 #define EMB_FORMAT_SST          47
84 #define EMB_FORMAT_STX          48
85 #define EMB_FORMAT_SVG          49
86 #define EMB_FORMAT_T01          50
87 #define EMB_FORMAT_T09          51
88 #define EMB_FORMAT_TAP          52
89 #define EMB_FORMAT_THR          53
90 #define EMB_FORMAT_TXT          54
91 #define EMB_FORMAT_U00          55
92 #define EMB_FORMAT_U01          56
93 #define EMB_FORMAT_VIP          57
94 #define EMB_FORMAT_VP3          58
95 #define EMB_FORMAT_XXX          59
96 #define EMB_FORMAT_ZSK          60
97
98 /* Thread color */
99 #define Arc_Polyester             0
100 #define Arc_Rayon                1
101 #define CoatsAndClark_Rayon      2
102 #define Exquisite_Polyester      3
103 #define Fufu_Polyester           4
104 #define Fufu_Rayon               5
105 #define Hemingworth_Polyester    6
106 #define Isacord_Polyester        7
107 #define Isafil_Rayon             8
108 #define Marathon_Polyester       9
109 #define Marathon_Rayon          10
110 #define Madeira_Polyester        11
111 #define Madeira_Rayon            12
112 #define Metro_Polyester          13
113 #define Pantone                  14
114 #define RobisonAnton_Polyester   15
115 #define RobisonAnton_Rayon       16
116 #define Sigma_Polyester          17
117 #define Sulky_Rayon              18
118 #define ThreadArt_Rayon          19
119 #define ThreadArt_Polyester       20
120 #define ThreaDelight_Polyester   21
121 #define Z102_Isacord_Polyester   22
122 #define SVG_Colors               23
123 #define hus_thread                24
124 #define jef_thread                25
125 #define pcm_thread                26
126 #define pec_thread                27
127 #define shv_thread                28
128 #define dxf_color                 29
129
130 #define EMB_ARRAY                 0
131 #define EMB_ARC                   1
132 #define EMB_CIRCLE                2
133 #define EMB_DIM_DIAMETER          3
134 #define EMB_DIM_LEADER            4
135 #define EMB_ELLIPSE               5
136 #define EMB_FLAG                  6
137 #define EMB_LINE                  7
138 #define EMB_IMAGE                 8
139 #define EMB_PATH                   9
140 #define EMB_POINT                 10
141 #define EMB_POLYGON               11
142 #define EMB_POLYLINE              12
143 #define EMB_RECT                  13
144 #define EMB_SPLINE                14
145 #define EMB_STITCH                15
146 #define EMB_TEXT_SINGLE           16
147 #define EMB_TEXT_MULTI            17
148 #define EMB_VECTOR                18
149 #define EMB_THREAD                19
150
151 #define EMBFORMAT_UNSUPPORTED      0
152 #define EMBFORMAT_STITCHONLY      1
153 #define EMBFORMAT_OBJECTONLY      2
154 #define EMBFORMAT_STCHANDOBJ      3 /* binary operation: 1+2=3 */
155
156 #define numberOfFormats            61
157
158 #define CHUNK_SIZE                 128
159
160 #define EMB_MAX_LAYERS             10
161 #define MAX_THREADS                256
162 #define EMBFORMAT_MAXEXT           3
163 /* maximum length of extension without dot */
164 #define EMBFORMAT_MAXDESC          50
165 /* the longest possible description string length */
166 #define MAX_STITCHES               1000000
167
168
169

```

```

170 #if defined(_WIN32) && !defined(WIN32)
171 #define WIN32
172 #endif
173
174 /* When building a shared library,
175 * use the proper export keyword depending on the compiler */
176 #define EMB_PUBLIC
177 #if defined(LIBEMBROIDERY_SHARED)
178 #undef EMB_PUBLIC
179 #if defined(__WIN32__) || defined(WIN32)
180 #define EMB_PUBLIC __declspec(dllexport)
181 #else
182 #define EMB_PUBLIC __attribute__((visibility("default")))
183 #endif
184 #endif
185
186 /* TYPEDEFS AND STRUCTS
187 *****/
188
189 typedef float EmbReal;
190
191 typedef struct EmbColor_
192 {
193     unsigned char r;
194     unsigned char g;
195     unsigned char b;
196 } EmbColor;
197
198 typedef struct EmbVector_
199 {
200     EmbReal x;
201     EmbReal y;
202 } EmbVector;
203
204 typedef struct EmbArray_ EmbArray;
205
206 typedef struct EmbImage_ {
207     EmbVector position;
208     EmbVector dimensions;
209     unsigned char* data;
210     int width;
211     int height;
212     char path[200];
213     char name[200];
214 } EmbImage;
215
216 typedef struct EmbBlock_ {
217     EmbVector position;
218 } EmbBlock;
219
220 typedef struct EmbAlignedDim_ {
221     EmbVector position;
222 } EmbAlignedDim;
223
224 typedef struct EmbAngularDim_ {
225     EmbVector position;
226 } EmbAngularDim;
227
228 typedef struct EmbArcLengthDim_ {
229     EmbVector position;
230 } EmbArcLengthDim;
231
232 typedef struct EmbDiameterDim_ {
233     EmbVector position;
234 } EmbDiameterDim;
235
236 typedef struct EmbLeaderDim_ {
237     EmbVector position;
238 } EmbLeaderDim;
239
240 typedef struct EmbLinearDim_ {
241     EmbVector position;
242 } EmbLinearDim;
243
244 typedef struct EmbOrdinateDim_ {
245     EmbVector position;
246 } EmbOrdinateDim;
247
248 typedef struct EmbRadiusDim_ {
249     EmbVector position;
250 } EmbRadiusDim;
251
252 typedef struct EmbInfiniteLine_ {
253     EmbVector position;
254 } EmbInfiniteLine;
255
256 typedef struct EmbRay_ {

```

```
312     EmbVector position;
313 } EmbRay;
314
315 typedef struct EmbTextMulti_ {
316     EmbVector position;
317     char text[200];
318 } EmbTextMulti;
319
320 typedef struct EmbTextSingle_ {
321     EmbVector position;
322     char text[200];
323 } EmbTextSingle;
324
325 typedef struct EmbTime_
326 {
327     unsigned int year;
328     unsigned int month;
329     unsigned int day;
330     unsigned int hour;
331     unsigned int minute;
332     unsigned int second;
333 } EmbTime;
334
335 typedef struct EmbPoint_
336 {
337     EmbVector position;
338     int lineType;
339     EmbColor color;
340 } EmbPoint;
341
342 typedef struct EmbLine_
343 {
344     EmbVector start;
345     EmbVector end;
346     int lineType;
347     EmbColor color;
348 } EmbLine;
349
350 typedef struct EmbPath_
351 {
352     EmbArray* pointList;
353     EmbArray* flagList;
354     int lineType;
355     EmbColor color;
356 } EmbPath;
357
358 typedef struct EmbStitch_
359 {
360     int flags;
361     EmbReal x;
362     EmbReal y;
363     int color;
364 } EmbStitch;
365
366 typedef struct EmbThread_
367 {
368     EmbColor color;
369     char description[50];
370     char catalogNumber[30];
371 } EmbThread;
372
373 typedef struct thread_color_ {
374     char name[22];
375     unsigned int hex_code;
376     int manufacturer_code;
377 } thread_color;
378
379 typedef struct EmbArc_
380 {
381     EmbVector start;
382     EmbVector mid;
383     EmbVector end;
384 } EmbArc;
385
386 typedef struct EmbRect_
387 {
388     EmbReal top;
389     EmbReal left;
390     EmbReal bottom;
391     EmbReal right;
392     EmbReal rotation;
393     EmbReal radius;
394 } EmbRect;
395
396 typedef struct EmbCircle_
397 {
398     EmbVector center;
```

```

448     EmbReal radius;
449 } EmbCircle;
450
455 typedef EmbPath EmbPolygon;
456
461 typedef EmbPath EmbPolyline;
462
467 typedef int EmbFlag;
468
473 typedef struct EmbSatinOutline_
474 {
475     int length;
476     EmbArray* side1;
477     EmbArray* side2;
478 } EmbSatinOutline;
479
484 typedef struct EmbEllipse_
485 {
486     EmbVector center;
487     EmbVector radius;
488     EmbReal rotation;
489 } EmbEllipse;
490
495 typedef struct EmbBezier_ {
496     EmbVector start;
497     EmbVector control1;
498     EmbVector control2;
499     EmbVector end;
500 } EmbBezier;
501
506 typedef struct EmbSpline_ {
507     EmbArray *beziers;
508 } EmbSpline;
509
514 typedef struct LSYSTEM {
515     char axiom;
516     char *alphabet;
517     char *constants;
518     char **rules;
519 } L_system;
520
525 typedef struct EmbGeometry_ {
526     union {
527         EmbArc arc;
528         EmbCircle circle;
529         EmbColor color;
530         EmbEllipse ellipse;
531         EmbLine line;
532         EmbPath path;
533         EmbPoint point;
534         EmbPolygon polygon;
535         EmbPolyline polyline;
536         EmbRect rect;
537         EmbSpline spline;
538         EmbVector vector;
539     } object;
540     EmbStitch stitch;
541     EmbThread thread;
542     int flag;
543     int type;
544     int lineType;
545     EmbColor color;
546 } EmbGeometry;
547
552 struct EmbArray_ {
553     EmbGeometry *geometry;
554     EmbStitch *stitch;
555     EmbThread *thread;
556     int count;
557     int length;
558     int type;
559 };
560
565 typedef struct EmbLayer_
566 {
567     char name[100];
568     EmbArray *geometry;
569 } EmbLayer;
570
575 typedef struct EmbPattern_
576 {
577     unsigned int dstJumpsPerTrim;
578     EmbVector home;
579     EmbReal hoop_width;
580     EmbReal hoop_height;
581     EmbArray *thread_list;
582     EmbArray *stitch_list;

```

```

583     EmbArray *geometry;
584     EmbLayer layer[EMB_MAX_LAYERS];
585     int currentColorIndex;
586 } EmbPattern;
587
588 typedef struct EmbFormatList_
589 {
590     char extension[2 + EMBFORMAT_MAXEXT];
591     char description[EMBFORMAT_MAXDESC];
592     char reader_state;
593     char writer_state;
594     int type;
595     int color_only;
596     int check_for_color_file;
597     int write_external_color_file;
598 } EmbFormatList;
599
600 /* Function Declarations
601 *****/
602 EMB_PUBLIC int lindenmayer_system(L_system L, char* state, int iteration, int complete);
603 EMB_PUBLIC int hilbert_curve(EmbPattern *pattern, int iterations);
604
605 EMB_PUBLIC int emb_identify_format(const char *ending);
606 EMB_PUBLIC void testMain(int level);
607 EMB_PUBLIC int convert(const char *inf, const char *outf);
608
609 EMB_PUBLIC EmbColor embColor_make(unsigned char r, unsigned char g, unsigned char b);
610 EMB_PUBLIC EmbColor* embColor_create(unsigned char r, unsigned char g, unsigned char b);
611 EMB_PUBLIC EmbColor embColor_fromHexStr(char* val);
612 EMB_PUBLIC int embColor_distance(EmbColor a, EmbColor b);
613
614 EMB_PUBLIC EmbArray* embArray_create(int type);
615 EMB_PUBLIC int embArray_resize(EmbArray *g);
616 EMB_PUBLIC void embArray_copy(EmbArray *dst, EmbArray *src);
617 EMB_PUBLIC int embArray_addArc(EmbArray* g, EmbArc arc);
618 EMB_PUBLIC int embArray_addCircle(EmbArray* g, EmbCircle circle);
619 EMB_PUBLIC int embArray_addEllipse(EmbArray* g, EmbEllipse ellipse);
620 EMB_PUBLIC int embArray_addFlag(EmbArray* g, int flag);
621 EMB_PUBLIC int embArray_addLine(EmbArray* g, EmbLine line);
622 EMB_PUBLIC int embArray_addRect(EmbArray* g, EmbRect rect);
623 EMB_PUBLIC int embArray_addPath(EmbArray* g, EmbPath p);
624 EMB_PUBLIC int embArray_addPoint(EmbArray* g, EmbPoint p);
625 EMB_PUBLIC int embArray_addPolygon(EmbArray* g, EmbPolygon p);
626 EMB_PUBLIC int embArray_addPolyline(EmbArray* g, EmbPolyline p);
627 EMB_PUBLIC int embArray_addSpline(EmbArray* g, EmbSpline p); /*
628 EMB_PUBLIC int embArray_addStitch(EmbArray* g, EmbStitch st);
629 EMB_PUBLIC int embArray_addThread(EmbArray* g, EmbThread p);
630 EMB_PUBLIC int embArray_addVector(EmbArray* g, EmbVector);
631 EMB_PUBLIC void embArray_free(EmbArray* p);
632
633 EMB_PUBLIC EmbLine embLine_make(EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2);
634
635 EMB_PUBLIC void embLine_normalVector(EmbLine line, EmbVector* result, int clockwise);
636 EMB_PUBLIC EmbVector embLine_intersectionPoint(EmbLine line1, EmbLine line2);
637
638 EMB_PUBLIC int embThread_findNearestColor(EmbColor color, EmbColor* colors, int n_colors);
639 EMB_PUBLIC int embThread_findNearestThread(EmbColor color, EmbThread* threads, int n_threads);
640 EMB_PUBLIC EmbThread embThread_getRandom(void);
641
642 EMB_PUBLIC void embVector_normalize(EmbVector vector, EmbVector* result);
643 EMB_PUBLIC void embVector_multiply(EmbVector vector, EmbReal magnitude, EmbVector* result);
644 EMB_PUBLIC EmbVector embVector_add(EmbVector v1, EmbVector v2);
645 EMB_PUBLIC EmbVector embVector_average(EmbVector v1, EmbVector v2);
646 EMB_PUBLIC EmbVector embVector_subtract(EmbVector v1, EmbVector v2);
647 EMB_PUBLIC EmbReal embVector_dot(EmbVector v1, EmbVector v2);
648 EMB_PUBLIC EmbReal embVector_cross(EmbVector v1, EmbVector v2);
649 EMB_PUBLIC void embVector_transpose_product(EmbVector v1, EmbVector v2, EmbVector* result);
650 EMB_PUBLIC EmbReal embVector_length(EmbVector vector);
651 EMB_PUBLIC EmbReal embVector_relativeX(EmbVector a1, EmbVector a2, EmbVector a3);
652 EMB_PUBLIC EmbReal embVector_relativeY(EmbVector a1, EmbVector a2, EmbVector a3);
653 EMB_PUBLIC EmbReal embVector_angle(EmbVector v);
654 EMB_PUBLIC EmbReal embVector_distance(EmbVector a, EmbVector b);
655 EMB_PUBLIC EmbVector embVector_unit(EmbReal angle);
656
657 EMB_PUBLIC EmbArc embArc_init(void);
658 EMB_PUBLIC char embArc_clockwise(EmbArc arc);
659
660 EMB_PUBLIC void getArcCenter(EmbArc arc, EmbVector *arcCenter);
661 EMB_PUBLIC char getArcDataFromBulge(EmbReal bulge,
662                                     EmbArc *arc,
663                                     EmbReal* arcCenterX,      EmbReal* arcCenterY,
664                                     EmbReal* radius,           EmbReal* diameter,
665                                     EmbReal* chord,            EmbReal* chordMidX,
666                                     EmbReal* chordMidY,         EmbReal* sagitta,
667                                     EmbReal* apothem,           EmbReal* incAngleInDegrees, char* clockwise);
668
669
670
671
672
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
751 EMB_PUBLIC void embPattern_addCircleAbs(EmbPattern* p, EmbCircle obj);
752 EMB_PUBLIC void embPattern_addEllipseAbs(EmbPattern* p, EmbEllipse obj);
753 EMB_PUBLIC void embPattern_addLineAbs(EmbPattern* p, EmbLine obj);
754 EMB_PUBLIC void embPattern_addPathAbs(EmbPattern* p, EmbPath obj);
755 EMB_PUBLIC void embPattern_addPointAbs(EmbPattern* p, EmbPoint obj);
756 EMB_PUBLIC void embPattern_addPolygonAbs(EmbPattern* p, EmbPolygon obj);
757 EMB_PUBLIC void embPattern_addPolylineAbs(EmbPattern* p, EmbPolyline obj);
758 EMB_PUBLIC void embPattern_addRectAbs(EmbPattern* p, EmbRect obj);
759
760 EMB_PUBLIC void embPattern_copyStitchListToPolylines(EmbPattern* pattern);

```



```

757 EMB_PUBLIC void embPattern_copyPolylinesToStitchList(EmbPattern* pattern);
758 EMB_PUBLIC void embPattern_moveStitchListToPolylines(EmbPattern* pattern);
759 EMB_PUBLIC void embPattern_movePolylinesToStitchList(EmbPattern* pattern);
760
761 EMB_PUBLIC char embPattern_read(EmbPattern *pattern, const char* fileName, int format);
762 EMB_PUBLIC char embPattern_write(EmbPattern *pattern, const char* fileName, int format);
763
764 EMB_PUBLIC char embPattern_readAuto(EmbPattern *pattern, const char* fileName);
765 EMB_PUBLIC char embPattern_writeAuto(EmbPattern *pattern, const char* fileName);
766
767 EMB_PUBLIC void report(int result, char *label);
768 EMB_PUBLIC int full_test_matrix(char *fname);
769
770 EMB_PUBLIC int emb_round(EmbReal x);
771 EMB_PUBLIC EmbReal radians(EmbReal degree);
772 EMB_PUBLIC EmbReal degrees(EmbReal radian);
773
774 /* NON-MACRO CONSTANTS
775 *****/
776
777 extern EmbFormatList formatTable[numberOfFormats];
778 extern const int pecThreadCount;
779 extern const int shvThreadCount;
780 extern const EmbReal embConstantPi;
781 extern const EmbThread husThreads[];
782 extern const EmbThread jefThreads[];
783 extern const EmbThread shvThreads[];
784 extern const EmbThread pcmThreads[];
785 extern const EmbThread pecThreads[];
786 extern const unsigned char _dxfColorTable[][3];
787 extern EmbThread black_thread;
788 extern const unsigned char vipDecodingTable[];
789
790 /* VARIABLES
791 *****/
792
793 extern int emb_error;
794
795 extern int emb_verbose;
796
797 #ifdef __cplusplus
798 }
799 #endif /* __cplusplus */
800
801 #endif /* LIBEMBROIDERY_HEADER__ */
802

```

## 18.30 extern/libembroidery/src/embroidery\_internal.h File Reference

```

#include "embroidery.h"
#include <stdio.h>

```

### Classes

- struct [\\_bcf\\_file\\_difat](#)
- struct [\\_bcf\\_file\\_fat](#)
- struct [\\_bcf\\_directory\\_entry](#)
- struct [\\_bcf\\_directory](#)
- struct [\\_bcf\\_file\\_header](#)
- struct [\\_bcf\\_file](#)
- struct [\\_vp3Hoop](#)
- struct [ThredHeader\\_](#)
- struct [ThredExtension\\_](#)
- struct [SubDescriptor\\_](#)
- struct [StxThread\\_](#)
- struct [VipHeader\\_](#)
- struct [SvgAttribute\\_](#)
- struct [Huffman](#)
- struct [Compress](#)

## Macros

- `#define CompoundFileSector_MaxRegSector 0xFFFFFFFFFA`
- `#define CompoundFileSector_DIFAT_Sector 0xFFFFFFFFFC`
- `#define CompoundFileSector_FAT_Sector 0xFFFFFFFFFD`
- `#define CompoundFileSector_EndOfChain 0xFFFFFFFFFE`
- `#define CompoundFileSector_FreeSector 0xFFFFFFFFFF`
- `#define ObjectTypeUnknown 0x00`
- `#define ObjectTypeStorage 0x01`
- `#define ObjectTypeStream 0x02`
- `#define ObjectTypeRootEntry 0x05`
- `#define CompoundFileStreamId_MaxRegularStreamId 0xFFFFFFFFFA`
- `#define CompoundFileStreamId_NoStream 0xFFFFFFFFFF`
- `#define ELEMENT_XML 0`
- `#define ELEMENT_A 1`
- `#define ELEMENT_ANIMATE 2`
- `#define ELEMENT_ANIMATECOLOR 3`
- `#define ELEMENT_ANIMATEMOTION 4`
- `#define ELEMENT_ANIMATETRANSFORM 5`
- `#define ELEMENT_ANIMATION 6`
- `#define ELEMENT_AUDIO 7`
- `#define ELEMENT_CIRCLE 8`
- `#define ELEMENT_DEFS 9`
- `#define ELEMENT_DESC 10`
- `#define ELEMENT_DISCARD 11`
- `#define ELEMENT_ELLIPSE 12`
- `#define ELEMENT_FONT 13`
- `#define ELEMENT_FONT_FACE 14`
- `#define ELEMENT_FONT_FACE_SRC 15`
- `#define ELEMENT_FONT_FACE_URI 16`
- `#define ELEMENT_FOREIGN_OBJECT 17`
- `#define ELEMENT_G 18`
- `#define ELEMENT_GLYPH 19`
- `#define ELEMENT_HANDLER 20`
- `#define ELEMENT_HKERN 21`
- `#define ELEMENT_IMAGE 22`
- `#define ELEMENT_LINE 23`
- `#define ELEMENT_LINEAR_GRADIENT 24`
- `#define ELEMENT_LISTENER 25`
- `#define ELEMENT_METADATA 26`
- `#define ELEMENT_MISSING_GLYPH 27`
- `#define ELEMENT_MPATH 28`
- `#define ELEMENT_PATH 29`
- `#define ELEMENT_POLYGON 30`
- `#define ELEMENT_POLYLINE 31`
- `#define ELEMENT_PREFETCH 32`
- `#define ELEMENT_RADIAL_GRADIENT 33`
- `#define ELEMENT_RECT 34`
- `#define ELEMENT_SCRIPT 35`
- `#define ELEMENT_SET 36`
- `#define ELEMENT_SOLID_COLOR 37`
- `#define ELEMENT_STOP 38`
- `#define ELEMENT_SVG 39`
- `#define ELEMENT_SWITCH 40`
- `#define ELEMENT_TBREAK 41`

- #define [ELEMENT\\_TEXT](#) 42
- #define [ELEMENT\\_TEXT\\_AREA](#) 43
- #define [ELEMENT\\_TITLE](#) 44
- #define [ELEMENT\\_TSPAN](#) 45
- #define [ELEMENT\\_USE](#) 46
- #define [ELEMENT\\_VIDEO](#) 47
- #define [RED\\_TERM\\_COLOR](#) "\x1B[0;31m"
- #define [GREEN\\_TERM\\_COLOR](#) "\x1B[0;32m"
- #define [YELLOW\\_TERM\\_COLOR](#) "\x1B[1;33m"
- #define [RESET\\_TERM\\_COLOR](#) "\033[0m"
- #define [HOOP\\_126X110](#) 0
- #define [HOOP\\_110X110](#) 1
- #define [HOOP\\_50X50](#) 2
- #define [HOOP\\_140X200](#) 3
- #define [HOOP\\_230X200](#) 4
- #define [EMB\\_MIN](#)(A, B) (((A) < (B)) ? (A) : (B))
- #define [EMB\\_MAX](#)(A, B) (((A) > (B)) ? (A) : (B))
- #define [EMB\\_BIG\\_ENDIAN](#) 0
- #define [EMB\\_LITTLE\\_ENDIAN](#) 1
- #define [ENDIAN\\_HOST](#) [EMB\\_LITTLE\\_ENDIAN](#)
- #define [EMB\\_INT16\\_BIG](#) 2
- #define [EMB\\_INT16\\_LITTLE](#) 3
- #define [EMB\\_INT32\\_BIG](#) 4
- #define [EMB\\_INT32\\_LITTLE](#) 5
- #define [PES0001](#) 0
- #define [PES0020](#) 1
- #define [PES0022](#) 2
- #define [PES0030](#) 3
- #define [PES0040](#) 4
- #define [PES0050](#) 5
- #define [PES0055](#) 6
- #define [PES0056](#) 7
- #define [PES0060](#) 8
- #define [PES0070](#) 9
- #define [PES0080](#) 10
- #define [PES0090](#) 11
- #define [PES0100](#) 12
- #define [N\\_PES\\_VERSIONS](#) 13
- #define [DXF\\_VERSION\\_R10](#) "AC1006"
- #define [DXF\\_VERSION\\_R11](#) "AC1009"
- #define [DXF\\_VERSION\\_R12](#) "AC1009"
- #define [DXF\\_VERSION\\_R13](#) "AC1012"
- #define [DXF\\_VERSION\\_R14](#) "AC1014"
- #define [DXF\\_VERSION\\_R15](#) "AC1015"
- #define [DXF\\_VERSION\\_R18](#) "AC1018"
- #define [DXF\\_VERSION\\_R21](#) "AC1021"
- #define [DXF\\_VERSION\\_R24](#) "AC1024"
- #define [DXF\\_VERSION\\_R27](#) "AC1027"
- #define [DXF\\_VERSION\\_2000](#) "AC1015"
- #define [DXF\\_VERSION\\_2002](#) "AC1015"
- #define [DXF\\_VERSION\\_2004](#) "AC1018"
- #define [DXF\\_VERSION\\_2006](#) "AC1018"
- #define [DXF\\_VERSION\\_2007](#) "AC1021"
- #define [DXF\\_VERSION\\_2009](#) "AC1021"
- #define [DXF\\_VERSION\\_2010](#) "AC1024"

- `#define DXF_VERSION_2013 "AC1027"`
- `#define SVG_CREATOR_NULL 0`
- `#define SVG_CREATOR_EMBROIDERMODDER 1`
- `#define SVG_CREATOR_ILLUSTRATOR 2`
- `#define SVG_CREATOR_INKSCAPE 3`
- `#define SVG_EXPECT_NULL 0`
- `#define SVG_EXPECT_ELEMENT 1`
- `#define SVG_EXPECT_ATTRIBUTE 2`
- `#define SVG_EXPECT_VALUE 3`
- `#define SVG_NULL 0`
- `#define SVG_ELEMENT 1`
- `#define SVG_PROPERTY 2`
- `#define SVG_MEDIA_PROPERTY 3`
- `#define SVG_ATTRIBUTE 4`
- `#define SVG_CATCH_ALL 5`
- `#define LINETO 0`
- `#define MOVETO 1`
- `#define BULGETOCONTROL 2`
- `#define BULGETOEND 4`
- `#define ELLIPSETORAD 8`
- `#define ELLIPSETOEND 16`
- `#define CUBICTOCONTROL1 32`
- `#define CUBICTOCONTROL2 64`
- `#define CUBICTOEND 128`
- `#define QUADTOCONTROL 256`
- `#define QUADTOEND 512`

## Typedefs

- `typedef struct _bcf_file_difat bcf_file_difat`
- `typedef struct _bcf_file_fat bcf_file_fat`
- `typedef struct _bcf_directory_entry bcf_directory_entry`
- `typedef struct _bcf_directory bcf_directory`
- `typedef struct _bcf_file_header bcf_file_header`
- `typedef struct _bcf_file bcf_file`
- `typedef struct _vp3Hoop vp3Hoop`
- `typedef struct ThredHeader_ ThredHeader`
- `typedef struct ThredExtension_ ThredExtension`
- `typedef struct SubDescriptor_ SubDescriptor`
- `typedef struct StxThread_ StxThread`
- `typedef struct VipHeader_ VipHeader`
- `typedef struct SvgAttribute_ SvgAttribute`
- `typedef struct Huffman huffman`
- `typedef struct Compress compress`

## Enumerations

- `enum CSV_EXPECT { CSV_EXPECT_NULL , CSV_EXPECT_QUOTE1 , CSV_EXPECT_QUOTE2 , CSV_EXPECT_COMMA }`
- `enum CSV_MODE { CSV_MODE_NULL , CSV_MODE_COMMENT , CSV_MODE_VARIABLE , CSV_MODE_THREAD , CSV_MODE_STITCH }`

## Functions

- void [huffman\\_build\\_table](#) ([huffman](#) \*h)
- int \* [huffman\\_table\\_lookup](#) ([huffman](#) \*h, int byte\_lookup, int \*lengths)
- int [compress\\_get\\_bits](#) ([compress](#) \*c, int length)
- int [compress\\_pop](#) ([compress](#) \*c, int bit\_count)
- int [compress\\_read\\_variable\\_length](#) ([compress](#) \*c)
- void [compress\\_load\\_character\\_length\\_huffman](#) ([compress](#) \*c)
- void [compress\\_load\\_character\\_huffman](#) ([compress](#) \*c)
- void [compress\\_load\\_distance\\_huffman](#) ([compress](#) \*c)
- void [compress\\_load\\_block](#) ([compress](#) \*c)
- int [compress\\_get\\_token](#) ([compress](#) \*c)
- int [compress\\_get\\_position](#) ([compress](#) \*c)
- void [readPecStitches](#) ([EmbPattern](#) \*pattern, FILE \*file)
- void [writePecStitches](#) ([EmbPattern](#) \*pattern, FILE \*file, const char \*filename)
- int [decodeNewStitch](#) (unsigned char value)
  - value*
- void [pfaffEncode](#) (FILE \*file, int x, int y, int flags)
- [EmbReal](#) [pfaffDecode](#) (unsigned char a1, unsigned char a2, unsigned char a3)
- unsigned char [mitEncodeStitch](#) ([EmbReal](#) value)
  - value*
- int [mitDecodeStitch](#) (unsigned char value)
  - value*
- int [encode\\_tajima\\_ternary](#) (unsigned char b[3], int x, int y)
- void [decode\\_tajima\\_ternary](#) (unsigned char b[3], int \*x, int \*y)
- void [encode\\_t01\\_record](#) (unsigned char b[3], int x, int y, int flags)
- int [decode\\_t01\\_record](#) (unsigned char b[3], int \*x, int \*y, int \*flags)
- void [readPESHeaderV5](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readPESHeaderV6](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readPESHeaderV7](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readPESHeaderV8](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readPESHeaderV9](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readPESHeaderV10](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readDescriptions](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readHoopName](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readImageString](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readProgrammableFills](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readMotifPatterns](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readFeatherPatterns](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readThreads](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [emblnt\\_read](#) (FILE \*f, char \*label, void \*b, int mode)
- void [emblnt\\_write](#) (FILE \*f, char \*label, void \*b, int mode)
- int [embl\\_readline](#) (FILE \*file, char \*line, int maxLength)
  - file line maxLength*
- int [bcfFile\\_read](#) (FILE \*file, [bcf\\_file](#) \*bcfFile)
  - file bcfFile*
- FILE \* [GetFile](#) ([bcf\\_file](#) \*bcfFile, FILE \*file, char \*fileToFind)
  - Get the File object.*
- void [bcf\\_file\\_free](#) ([bcf\\_file](#) \*bcfFile)
  - bcfFile*
- void [binaryReadString](#) (FILE \*file, char \*buffer, int maxLength)
  - file buffer maxLength*
- void [binaryReadUnicodeString](#) (FILE \*file, char \*buffer, const int stringLength)
  - file buffer stringLength*

- int [stringInArray](#) (const char \*s, const char \*\*array)
- void [fpad](#) (FILE \*f, char c, int n)
  - f*
- char \* [copy\\_trim](#) (char const \*s)
  - s*
- char \* [emb\\_optOut](#) (EmbReal num, char \*str)
  - Optimizes the number (num) for output to a text file and returns it as a string (str).*
- void [write\\_24bit](#) (FILE \*file, int)
  - file x*
- int [check\\_header\\_present](#) (FILE \*file, int minimum\_header\_length)
  - file minimum\_header\_length*
- unsigned short [fread\\_uint16](#) (FILE \*file)
  - f*
- short [fread\\_int16](#) (FILE \*f)
  - f*
- int [fread\\_int32\\_be](#) (FILE \*f)
  - f*
- void [safe\\_free](#) (void \*data)
  - data*
- void [binaryWriteUIntBE](#) (FILE \*f, unsigned int data)
  - f data*
- void [binaryWriteUInt](#) (FILE \*f, unsigned int data)
  - f data*
- void [binaryWriteIntBE](#) (FILE \*f, int data)
  - f data*
- void [binaryWriteInt](#) (FILE \*f, int data)
  - f data*
- void [binaryWriteUShort](#) (FILE \*f, unsigned short data)
  - f data*
- void [binaryWriteUShortBE](#) (FILE \*f, unsigned short data)
  - f data*
- void [binaryWriteShort](#) (FILE \*f, short data)
  - f data*
- [bcf\\_file\\_difat](#) \* [bcf\\_difat\\_create](#) (FILE \*file, unsigned int fatSectors, const unsigned int [sectorSize](#))
  - file fatSectors sectorSize*
- unsigned int [readFullSector](#) (FILE \*file, [bcf\\_file\\_difat](#) \*bcfFile, unsigned int \*numberOfDifatEntriesStillToRead)
  - file bcfFile difatEntriesToRead*
- unsigned int [numberOfEntriesInDifatSector](#) ([bcf\\_file\\_difat](#) \*fat)
- void [bcf\\_file\\_difat\\_free](#) ([bcf\\_file\\_difat](#) \*difat)
- unsigned int [entriesInDifatSector](#) ([bcf\\_file\\_difat](#) \*fat)
  - fat*
- [bcf\\_file\\_fat](#) \* [bcfFileFat\\_create](#) (const unsigned int [sectorSize](#))
  - sectorSize*
- void [loadFatFromSector](#) ([bcf\\_file\\_fat](#) \*fat, FILE \*file)
  - fat file*
- void [bcf\\_file\\_fat\\_free](#) ([bcf\\_file\\_fat](#) \*\*fat)
- [bcf\\_directory\\_entry](#) \* [CompoundFileDirectoryEntry](#) (FILE \*file)
  - file*
- [bcf\\_directory](#) \* [CompoundFileDirectory](#) (const unsigned int maxNumberOfDirectoryEntries)
  - maxNumberOfDirectoryEntries*

- void `readNextSector` (FILE \*file, `bcf_directory` \*dir)
  - file dir*
- void `bcf_directory_free` (`bcf_directory` \*\*dir)
  - dir*
- `bcf_file_header` `bcfFileHeader_read` (FILE \*file)
  - file*
- int `bcfFileHeader_isValid` (`bcf_file_header` header)
- int `hus_compress` (char \*input, int size, char \*output, int \*out\_size)
- int `hus_decompress` (char \*input, int size, char \*output, int \*out\_size)
- void `testTangentPoints` (`EmbCircle` c, `EmbVector` p, `EmbVector` \*t0, `EmbVector` \*t1)
- void `printArcResults` (`EmbReal` bulge, `EmbArc` arc, `EmbReal` centerX, `EmbReal` centerY, `EmbReal` radius, `EmbReal` diameter, `EmbReal` chord, `EmbReal` chordMidX, `EmbReal` chordMidY, `EmbReal` sagitta, `EmbReal` apothem, `EmbReal` incAngle, char clockwise)
- int `create_test_file_1` (const char \*outf)
- int `create_test_file_2` (const char \*outf)
- int `create_test_file_3` (const char \*outf)
- int `testEmbCircle` (void)
- int `testEmbCircle_2` (void)
- int `testGeomArc` (void)
- int `testThreadColor` (void)
- int `testEmbFormat` (void)
- void `embColor_read` (FILE \*f, `EmbColor` \*c, int toRead)
  - f c toRead*
- void `embColor_write` (FILE \*f, `EmbColor` c, int toWrite)
  - f c toWrite*
- char `read100` (`EmbPattern` \*pattern, FILE \*file)
- char `write100` (`EmbPattern` \*pattern, FILE \*file)
- char `read10o` (`EmbPattern` \*pattern, FILE \*file)
- char `write10o` (`EmbPattern` \*pattern, FILE \*file)
- char `readArt` (`EmbPattern` \*pattern, FILE \*file)
- char `writeArt` (`EmbPattern` \*pattern, FILE \*file)
- char `readBmc` (`EmbPattern` \*pattern, FILE \*file)
- char `writeBmc` (`EmbPattern` \*pattern, FILE \*file)
- char `readBro` (`EmbPattern` \*pattern, FILE \*file)
- char `writeBro` (`EmbPattern` \*pattern, FILE \*file)
- char `readCnd` (`EmbPattern` \*pattern, FILE \*file)
- char `writeCnd` (`EmbPattern` \*pattern, FILE \*file)
- char `readCol` (`EmbPattern` \*pattern, FILE \*file)
- char `writeCol` (`EmbPattern` \*pattern, FILE \*file)
- char `readCsd` (`EmbPattern` \*pattern, FILE \*file)
- char `writeCsd` (`EmbPattern` \*pattern, FILE \*file)
- char `readCsv` (`EmbPattern` \*pattern, FILE \*file)
- char `writeCsv` (`EmbPattern` \*pattern, FILE \*file)
- char `readDat` (`EmbPattern` \*pattern, FILE \*file)
- char `writeDat` (`EmbPattern` \*pattern, FILE \*file)
- char `readDem` (`EmbPattern` \*pattern, FILE \*file)
- char `writeDem` (`EmbPattern` \*pattern, FILE \*file)
- char `readDsb` (`EmbPattern` \*pattern, FILE \*file)
- char `writeDsb` (`EmbPattern` \*pattern, FILE \*file)
- char `readDst` (`EmbPattern` \*pattern, FILE \*file)
- char `writeDst` (`EmbPattern` \*pattern, FILE \*file)
- char `readDsz` (`EmbPattern` \*pattern, FILE \*file)
- char `writeDsz` (`EmbPattern` \*pattern, FILE \*file)
- char `readDxf` (`EmbPattern` \*pattern, FILE \*file)

- char [writeDxf](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readEdr](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeEdr](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readEmd](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeEmd](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readExp](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeExp](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readExy](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeExy](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readEys](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeEys](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readFxy](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeFxy](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readGc](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeGc](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readGnc](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeGnc](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readGt](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeGt](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readHus](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeHus](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readInb](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeInb](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readInf](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeInf](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readJef](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeJef](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readKsm](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeKsm](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readMax](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeMax](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readMit](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeMit](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readNew](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeNew](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readOfm](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeOfm](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readPcd](#) ([EmbPattern](#) \*pattern, const char \*fileName, FILE \*file)
- char [writePcd](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readPcm](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writePcm](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readPcq](#) ([EmbPattern](#) \*pattern, const char \*fileName, FILE \*file)
- char [writePcq](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readPcs](#) ([EmbPattern](#) \*pattern, const char \*fileName, FILE \*file)
- char [writePcs](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readPec](#) ([EmbPattern](#) \*pattern, const char \*fileName, FILE \*file)
- char [writePec](#) ([EmbPattern](#) \*pattern, const char \*fileName, FILE \*file)
- char [readPel](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writePel](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readPem](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writePem](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readPes](#) ([EmbPattern](#) \*pattern, const char \*fileName, FILE \*file)
- char [writePes](#) ([EmbPattern](#) \*pattern, const char \*fileName, FILE \*file)
- char [readPhb](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writePhb](#) ([EmbPattern](#) \*pattern, FILE \*file)



- char [readPhc](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writePhc](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readPlt](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writePlt](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readRgb](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeRgb](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readSew](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeSew](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readShv](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeShv](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readSst](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeSst](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readStx](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeStx](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readSvg](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeSvg](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readT01](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeT01](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readT09](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeT09](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readTap](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeTap](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readThr](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeThr](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readTxt](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeTxt](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readU00](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeU00](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readU01](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeU01](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readVip](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeVip](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readVp3](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeVp3](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readXxx](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeXxx](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [readZsk](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeZsk](#) ([EmbPattern](#) \*pattern, FILE \*file)

## Variables

- const char [imageWithFrame](#) [38][48]

## 18.30.1 Macro Definition Documentation

**18.30.1.1 BULGETOCONTROL** `#define BULGETOCONTROL 2`

**18.30.1.2 BULGETOEND** `#define BULGETOEND 4`

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

**18.30.1.4 CompoundFileSector\_EndOfChain** #define CompoundFileSector\_EndOfChain 0xFFFFFFFFFE

**18.30.1.5 CompoundFileSector\_FAT\_Sector** #define CompoundFileSector\_FAT\_Sector 0xFFFFFFFFFD

**18.30.1.6 CompoundFileSector\_FreeSector** #define CompoundFileSector\_FreeSector 0xFFFFFFFFF

**18.30.1.7 CompoundFileSector\_MaxRegSector** #define CompoundFileSector\_MaxRegSector 0xFFFFFFFFFA  
Type of sector

**18.30.1.8 CompoundFileStreamId\_MaxRegularStreamId** #define CompoundFileStreamId\_MaxRegular↔  
StreamId 0xFFFFFFFFFA  
Special values for Stream Identifiers All real stream Ids are less than this

**18.30.1.9 CompoundFileStreamId\_NoStream** #define CompoundFileStreamId\_NoStream 0xFFFFFFFFF  
There is no valid stream Id

**18.30.1.10 CUBICTOCONTROL1** #define CUBICTOCONTROL1 32

**18.30.1.11 CUBICTOCONTROL2** #define CUBICTOCONTROL2 64

**18.30.1.12 CUBICTOEND** #define CUBICTOEND 128

**18.30.1.13 DXF\_VERSION\_2000** #define DXF\_VERSION\_2000 "AC1015"

**18.30.1.14 DXF\_VERSION\_2002** #define DXF\_VERSION\_2002 "AC1015"

**18.30.1.15 DXF\_VERSION\_2004** #define DXF\_VERSION\_2004 "AC1018"

**18.30.1.16 DXF\_VERSION\_2006** #define DXF\_VERSION\_2006 "AC1018"

**18.30.1.17 DXF\_VERSION\_2007** #define DXF\_VERSION\_2007 "AC1021"

**18.30.1.18 DXF\_VERSION\_2009** #define DXF\_VERSION\_2009 "AC1021"

**18.30.1.19 DXF\_VERSION\_2010** #define DXF\_VERSION\_2010 "AC1024"

**18.30.1.20 DXF\_VERSION\_2013** #define DXF\_VERSION\_2013 "AC1027"

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**18.30.1.57 ELEMENT\_MISSING\_GLYPH** #define ELEMENT\_MISSING\_GLYPH 27

**18.30.1.58 ELEMENT\_MPATH** #define ELEMENT\_MPATH 28

**18.30.1.59 ELEMENT\_PATH** #define ELEMENT\_PATH 29

**18.30.1.60 ELEMENT\_POLYGON** #define ELEMENT\_POLYGON 30

**18.30.1.61 ELEMENT\_POLYLINE** #define ELEMENT\_POLYLINE 31

**18.30.1.62 ELEMENT\_PREFETCH** #define ELEMENT\_PREFETCH 32

**18.30.1.63 ELEMENT\_RADIAL\_GRADIENT** #define ELEMENT\_RADIAL\_GRADIENT 33

**18.30.1.64 ELEMENT\_RECT** #define ELEMENT\_RECT 34

**18.30.1.65 ELEMENT\_SCRIPT** #define ELEMENT\_SCRIPT 35

**18.30.1.66 ELEMENT\_SET** #define ELEMENT\_SET 36

**18.30.1.67 ELEMENT\_SOLID\_COLOR** #define ELEMENT\_SOLID\_COLOR 37

**18.30.1.68 ELEMENT\_STOP** #define ELEMENT\_STOP 38

**18.30.1.69 ELEMENT\_SVG** #define ELEMENT\_SVG 39

**18.30.1.70 ELEMENT\_SWITCH** #define ELEMENT\_SWITCH 40

**18.30.1.71 ELEMENT\_TBREAK** #define ELEMENT\_TBREAK 41

**18.30.1.72 ELEMENT\_TEXT** #define ELEMENT\_TEXT 42

**18.30.1.73 ELEMENT\_TEXT\_AREA** #define ELEMENT\_TEXT\_AREA 43

**18.30.1.74 ELEMENT\_TITLE** #define ELEMENT\_TITLE 44

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

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

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

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

**18.30.1.79 ELLIPSETOEND** `#define ELLIPSETOEND 16`

**18.30.1.80 ELLIPSETORAD** `#define ELLIPSETORAD 8`

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

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

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

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

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

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

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

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

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

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

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

**18.30.1.92 HOOP\_126X110** `#define HOOP_126X110 0`

**18.30.1.93 HOOP\_140X200** `#define HOOP_140X200 3`

**18.30.1.94 HOOP\_230X200** `#define HOOP_230X200 4`

**18.30.1.95 HOOP\_50X50** `#define HOOP_50X50 2`

**18.30.1.96 LINETO** `#define LINETO 0`

**18.30.1.97 MOVETO** `#define MOVETO 1`

**18.30.1.98 N\_PES\_VERSIONS** `#define N_PES_VERSIONS 13`

**18.30.1.99 ObjectTypeRootEntry** `#define ObjectTypeRootEntry 0x05`  
the root entry

**18.30.1.100 ObjectTypeStorage** `#define ObjectTypeStorage 0x01`  
a directory type object

**18.30.1.101 ObjectTypeStream** `#define ObjectTypeStream 0x02`  
a file type object

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

**18.30.1.103 PES0001** `#define PES0001 0`

**18.30.1.104 PES0020** `#define PES0020 1`

**18.30.1.105 PES0022** `#define PES0022 2`

**18.30.1.106 PES0030** `#define PES0030 3`

**18.30.1.107 PES0040** `#define PES0040 4`

**18.30.1.108 PES0050** `#define PES0050 5`

**18.30.1.109 PES0055** `#define PES0055 6`

**18.30.1.110 PES0056** `#define PES0056 7`

**18.30.1.111 PES0060** `#define PES0060 8`

**18.30.1.112 PES0070** `#define PES0070 9`

**18.30.1.113 PES0080** `#define PES0080 10`

**18.30.1.114 PES0090** `#define PES0090 11`

**18.30.1.115 PES0100** `#define PES0100 12`

**18.30.1.116 QUADTOCONTROL** `#define QUADTOCONTROL 256`

**18.30.1.117 QUADTOEND** `#define QUADTOEND 512`

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

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

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

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

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

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

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

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

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



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

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

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

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

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

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

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

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

## 18.30.2 Typedef Documentation

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

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

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

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

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

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

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

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

**18.30.2.7 compress** `typedef struct Compress compress`

**18.30.2.8 huffman** `typedef struct Huffman huffman`

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

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

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

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

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

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

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

### 18.30.3 Enumeration Type Documentation

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

Enumerator

CSV_EXPECT_NULL	
CSV_EXPECT_QUOTE1	
CSV_EXPECT_QUOTE2	
CSV_EXPECT_COMMA	

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

Enumerator

CSV_MODE_NULL	
CSV_MODE_COMMENT	
CSV_MODE_VARIABLE	
CSV_MODE_THREAD	
CSV_MODE_STITCH	

### 18.30.4 Function Documentation

**18.30.4.1 bcf\_difat\_create()** [bcf\\_file\\_difat](#) \* bcf\_difat\_create (   
FILE \* *file*,   
unsigned int *fatSectors*,   
const unsigned int *sectorSize* )   
*file* *fatSectors* *sectorSize*

## Returns

bcf\_file\_difat\*

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

*dir*

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

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

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

*bcfFile*

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

*file bcfFile*

## Returns

int

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

*sectorSize*

## Returns

bcf\_file\_fat\*

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

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

*file*

## Returns

bcf\_file\_header

**18.30.4.10 binaryReadString()** `void binaryReadString (`  
    `FILE * file,`  
    `char * buffer,`  
    `int maxLength )`  
*file buffer maxLength*

**18.30.4.11 binaryReadUnicodeString()** `void binaryReadUnicodeString (`  
    `FILE * file,`  
    `char * buffer,`  
    `const int stringLength )`  
*file buffer stringLength*

**18.30.4.12 binaryWriteInt()** `void binaryWriteInt (`  
    `FILE * f,`  
    `int data )`  
*f data*

**Todo** replace with `emblnt_read`

**18.30.4.13 binaryWriteIntBE()** `void binaryWriteIntBE (`  
    `FILE * f,`  
    `int data )`  
*f data*

**Todo** replace with `emblnt_read`

**18.30.4.14 binaryWriteShort()** `void binaryWriteShort (`  
    `FILE * f,`  
    `short data )`  
*f data*

**Todo** replace with `emblnt_read`

**18.30.4.15 binaryWriteUInt()** `void binaryWriteUInt (`  
    `FILE * f,`  
    `unsigned int data )`  
*f data*

**Todo** replace with `emblnt_read`

**18.30.4.16 binaryWriteUIntBE()** `void binaryWriteUIntBE (`  
    `FILE * f,`  
    `unsigned int data )`  
*f data*

**Todo** replace with `emblnt_read`

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

**Todo** replace with embInt\_read

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

**Todo** replace with embInt\_read

**18.30.4.19 check\_header\_present()** int check\_header\_present (   
 FILE \* *file*,  
 int *minimum\_header\_length* )  
*file minimum\_header\_length*

Returns

int

Checks that there are enough bytes to interpret the header, stops possible segfaults when reading in the header bytes.

Returns 0 if there aren't enough, or the length of the file if there are.

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

Returns

bcf\_directory\*

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

Returns

bcf\_directory\_entry\*

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

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

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

*c* . Returns the token as an int.

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

*c* . Returns nothing.

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

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

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

*c* . Returns.

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

*c* . Returns nothing.

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

*c bit\_count* . Returns.

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

*c*. Returns.

**18.30.4.31 copy\_trim()** `char * copy_trim (`  
    `char const * s )`

*s*

Returns

char\*

**Todo** decription

**18.30.4.32 create\_test\_file\_1()** `int create_test_file_1 (`  
    `const char * outf )`

**18.30.4.33 create\_test\_file\_2()** `int create_test_file_2 (`  
    `const char * outf )`

**18.30.4.34 create\_test\_file\_3()** `int create_test_file_3 (`  
    `const char * outf )`

**18.30.4.35 decode\_t01\_record()** int decode\_t01\_record (

```

    unsigned char b[3],
    int * x,
    int * y,
    int * flags )

```

*b x y flags* .

**Todo** remove the unused return argument.

**18.30.4.36 decode\_tajima\_ternary()** void decode\_tajima\_ternary (

```

    unsigned char b[3],
    int * x,
    int * y )

```

Decode the signed ternary of the tajima format from *b* to the position values *x* and *y*. There is no return argument.

**18.30.4.37 decodeNewStitch()** int decodeNewStitch (

```

    unsigned char value )

```

*value*

Returns

int

**18.30.4.38 emb\_optOut()** char \* emb\_optOut (

```

    EmbReal num,
    char * str )

```

Optimizes the number (*num*) for output to a text file and returns it as a string (*str*).

*num str*

Returns

char\*

**18.30.4.39 emb\_readline()** int emb\_readline (

```

    FILE * file,
    char * line,
    int maxLength )

```

*file line maxLength*

Returns

int

**18.30.4.40 embColor\_read()** void embColor\_read (

```

    FILE * f,
    EmbColor * c,
    int toRead )

```

*f c toRead*

**18.30.4.41 embColor\_write()** void embColor\_write (  
FILE \* *f*,  
EmbColor *c*,  
int *toWrite* )  
*f c toWrite*

**18.30.4.42 embInt\_read()** void embInt\_read (  
FILE \* *f*,  
char \* *label*,  
void \* *b*,  
int *mode* )  
*f label b mode*

Read and write system for multiple byte types.

The caller passes the function to read/write from, the memory location as a void pointer and a mode identifier that describes the type. This way we can abstract out the endianness of the system running the library and don't have to maintain many functions, just two.

**18.30.4.43 embInt\_write()** void embInt\_write (  
FILE \* *f*,  
char \* *label*,  
void \* *b*,  
int *mode* )  
*f label b mode*

**18.30.4.44 encode\_t01\_record()** void encode\_t01\_record (  
unsigned char *b*[3],  
int *x*,  
int *y*,  
int *flags* )

Encode into bytes *b* the values of the x-position *x*, y-position *y* and the *flags*.

**18.30.4.45 encode\_tajima\_ternary()** int encode\_tajima\_ternary (  
unsigned char *b*[3],  
int *x*,  
int *y* )

Encode the signed ternary of the tajima format into *b* the position values *x* and *y*.

If the values of *x* or *y* fall outside of the valid range of -121 and +121 then it returns 0 and 1.

**18.30.4.46 entriesInDifatSector()** unsigned int entriesInDifatSector (  
bcf\_file\_difat \* *fat* )  
*fat*

Returns

unsigned int

**18.30.4.47 fpad()** void fpad (  
FILE \* *file*,  
char *c*,  
int *n* )  
*f*

Returns

int



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

*f*

Returns

short

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

*f*

Returns

int

**Todo** replace with embInt\_read

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

*f*

Returns

unsigned short

**Todo** replace with embInt\_read

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

Get the File object.

*bcfFile file fileToFind*

Returns

FILE\*

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

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

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

**18.30.4.54 hus\_compress()** int hus\_compress (  
char \* *data*,  
int *length*,  
char \* *output*,  
int \* *output\_length* )

*data length output output\_length* . Returns whether it was successful as an int.

This avoids the now unnecessary compression by placing a minimal header of 6 bytes and using only literals in the huffman compressed part (see the sources above).

**18.30.4.55 hus\_decompress()** int hus\_decompress (   
char \* data,   
int length,   
char \* output,   
int \* output\_length )   
*data length output output\_length* . Returns whether the decompression was successful.

**18.30.4.56 loadFatFromSector()** void loadFatFromSector (   
bcf\_file\_fat \* fat,   
FILE \* file )   
*fat file*

**18.30.4.57 mitDecodeStitch()** int mitDecodeStitch (   
unsigned char value )   
*value*   
Returns   
int

**18.30.4.58 mitEncodeStitch()** unsigned char mitEncodeStitch (   
EmbReal value )   
*value*   
Returns   
unsigned char

**18.30.4.59 numberOfEntriesInDifatSector()** unsigned int numberOfEntriesInDifatSector (   
bcf\_file\_difat \* fat )

**18.30.4.60 pfaffDecode()** EmbReal pfaffDecode (   
unsigned char a1,   
unsigned char a2,   
unsigned char a3 )   
Decode the bytes *a1*, *a2* and *a3* . Returns the EmbReal floating-point value.

**18.30.4.61 pfaffEncode()** void pfaffEncode (   
FILE \* file,   
int dx,   
int dy,   
int flags )   
*file dx dy flags*

**18.30.4.62 printArcResults()** void printArcResults (   
EmbReal bulge,   
EmbArc arc,   
EmbReal centerX,   
EmbReal centerY,   
EmbReal radius,   
EmbReal diameter,   
EmbReal chord,   
EmbReal chordMidX,

```
EmbReal chordMidY,  
EmbReal sagitta,  
EmbReal apothem,  
EmbReal incAngle,  
char clockwise )
```

**18.30.4.63 read100()** char read100 (  
EmbPattern \* pattern,  
FILE \* file )

**18.30.4.64 read10o()** char read10o (  
EmbPattern \* pattern,  
FILE \* file )

**18.30.4.65 readArt()** char readArt (  
EmbPattern \* pattern,  
FILE \* file )

**18.30.4.66 readBmc()** char readBmc (  
EmbPattern \* pattern,  
FILE \* file )

**18.30.4.67 readBro()** char readBro (  
EmbPattern \* pattern,  
FILE \* file )

**18.30.4.68 readCnd()** char readCnd (  
EmbPattern \* pattern,  
FILE \* file )

**18.30.4.69 readCol()** char readCol (  
EmbPattern \* pattern,  
FILE \* file )

**18.30.4.70 readCsd()** char readCsd (  
EmbPattern \* pattern,  
FILE \* file )

**18.30.4.71 readCsv()** char readCsv (  
EmbPattern \* pattern,  
FILE \* file )

**18.30.4.72 readDat()** char readDat (  
EmbPattern \* pattern,  
FILE \* file )

**18.30.4.73 readDem()** `char readDem (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.30.4.74 readDescriptions()** `void readDescriptions (`  
    `FILE * file,`  
    `EmbPattern * pattern )`

**18.30.4.75 readDsb()** `char readDsb (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.30.4.76 readDst()** `char readDst (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.30.4.77 readDsz()** `char readDsz (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

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

**18.30.4.79 readDxf()** `char readDxf (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.30.4.80 readEdr()** `char readEdr (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

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

**18.30.4.82 readEmd()** `char readEmd (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.30.4.83 readExp()** `char readExp (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.30.4.84 readExy()** `char readExy (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.30.4.85 readEys()** `char readEys (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

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

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

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

Returns

unsigned int

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

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

**18.30.4.91 readGc()** char readGc (  
EmbPattern \* *pattern*,  
FILE \* *file* )

Smoothie G-Code

Main Reference: Machinery's Handbook Guide A Guide to Tables, Formulas, & More in the 31st Edition by John Milton Amiss, Franklin D. Jones and Henry Ryffel

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

*file dir*

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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



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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**18.30.4.157 readTxt()** `char readTxt (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.30.4.158 File (.txt)** The txt format is stitch-only and isn't associated with a specific company.

**18.30.4.159 readU00()** `char readU00 (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.30.4.160 Embroidery Format (.u00)** The Barudan u00 format is stitch-only.

**18.30.4.161 readU01()** `char readU01 (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.30.4.162 Embroidery Format (.u00)** The Barudan u01 format is stitch-only.

**18.30.4.163 readVip()** `char readVip (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.30.4.164 readVp3()** `char readVp3 (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.30.4.165 readXxx()** `char readXxx (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.30.4.166 readZsk()** `char readZsk (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.30.4.167** **safe\_free()** void safe\_free (  
void \* data )  
*data*

**18.30.4.168** **stringInArray()** int stringInArray (  
const char \* s,  
const char \*\* array )

Tests for the presence of a string *s* in the supplied *array*.  
The end of the array is marked by an empty string.

Returns

0 if not present 1 if present.

**18.30.4.169** **testEmbCircle()** int testEmbCircle (  
void )

**18.30.4.170** **testEmbCircle\_2()** int testEmbCircle\_2 (  
void )

**18.30.4.171** **testEmbFormat()** int testEmbFormat (  
void )

**18.30.4.172** **testGeomArc()** int testGeomArc (  
void )

**18.30.4.173** **testTangentPoints()** void testTangentPoints (  
EmbCircle c,  
EmbVector p,  
EmbVector \* t0,  
EmbVector \* t1 )

**18.30.4.174** **testThreadColor()** int testThreadColor (  
void )

**18.30.4.175** **write100()** char write100 (  
EmbPattern \* pattern,  
FILE \* file )

**18.30.4.176** **write10o()** char write10o (  
EmbPattern \* pattern,  
FILE \* file )

**18.30.4.177 write\_24bit()** void write\_24bit (   
FILE \* *file*,   
int *x* )   
*file x*

**18.30.4.178 writeArt()** char writeArt (   
EmbPattern \* *pattern*,   
FILE \* *file* )

**18.30.4.179 writeBmc()** char writeBmc (   
EmbPattern \* *pattern*,   
FILE \* *file* )

**18.30.4.180 writeBro()** char writeBro (   
EmbPattern \* *pattern*,   
FILE \* *file* )

**18.30.4.181 writeCnd()** char writeCnd (   
EmbPattern \* *pattern*,   
FILE \* *file* )

**18.30.4.182 writeCol()** char writeCol (   
EmbPattern \* *pattern*,   
FILE \* *file* )

**18.30.4.183 writeCsd()** char writeCsd (   
EmbPattern \* *pattern*,   
FILE \* *file* )

**18.30.4.184 writeCsv()** char writeCsv (   
EmbPattern \* *pattern*,   
FILE \* *file* )

**18.30.4.185 writeDat()** char writeDat (   
EmbPattern \* *pattern*,   
FILE \* *file* )

**18.30.4.186 writeDem()** char writeDem (   
EmbPattern \* *pattern*,   
FILE \* *file* )

**18.30.4.187 writeDsb()** char writeDsb (   
EmbPattern \* *pattern*,   
FILE \* *file* )

**18.30.4.188 writeDst()** char writeDst (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.189 writeDsz()** char writeDsz (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.190 writeDxf()** char writeDxf (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.191 writeEdr()** char writeEdr (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.192 writeEmd()** char writeEmd (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.193 writeExp()** char writeExp (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.194 writeExy()** char writeExy (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.195 writeEys()** char writeEys (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.196 writeFxy()** char writeFxy (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.197 writeGc()** char writeGc (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.198 writeGnc()** char writeGnc (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.199 writeGt()** char writeGt (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.200 writeHus()** char writeHus (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.201 writeInb()** char writeInb (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.202 writeInf()** char writeInf (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.203 writeJef()** char writeJef (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.204 writeKsm()** char writeKsm (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.205 writeMax()** char writeMax (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.206 writeMit()** char writeMit (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.207 writeNew()** char writeNew (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.208 writeOfm()** char writeOfm (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.209 writePcd()** char writePcd (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.210 writePcm()** char writePcm (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.211 writePcq()** char writePcq (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.212 writePcs()** char writePcs (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.213 writePec()** char writePec (  
    EmbPattern \* pattern,  
    const char \* fileName,  
    FILE \* file )

**18.30.4.214 writePecStitches()** void writePecStitches (  
    EmbPattern \* pattern,  
    FILE \* file,  
    const char \* filename )

**18.30.4.215 writePel()** char writePel (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.216 writePem()** char writePem (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.217 writePes()** char writePes (  
    EmbPattern \* pattern,  
    const char \* fileName,  
    FILE \* file )

**18.30.4.218 writePhb()** char writePhb (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.219 writePhc()** char writePhc (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.220 writePlt()** char writePlt (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.221 writeRgb()** char writeRgb (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.222 writeSew()** char writeSew (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.223 writeShv()** char writeShv (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.224 writeSst()** char writeSst (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.225 writeStx()** char writeStx (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.226 writeSvg()** char writeSvg (  
    EmbPattern \* pattern,  
    FILE \* file )

Writes the data from *pattern* to a file with the given *fileName*. Returns `true` if successful, otherwise returns `false`.

**18.30.4.227 writeT01()** char writeT01 (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.228 writeT09()** char writeT09 (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.229 writeTap()** char writeTap (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.30.4.230 writeThr()** char writeThr (  
    EmbPattern \* pattern,  
    FILE \* file )



**18.30.4.231 writeTxt()** char writeTxt (  
     EmbPattern \* pattern,  
     FILE \* file )

**18.30.4.232 writeU00()** char writeU00 (  
     EmbPattern \* pattern,  
     FILE \* file )

**18.30.4.233 writeU01()** char writeU01 (  
     EmbPattern \* pattern,  
     FILE \* file )

**18.30.4.234 writeVip()** char writeVip (  
     EmbPattern \* pattern,  
     FILE \* file )

**18.30.4.235 writeVp3()** char writeVp3 (  
     EmbPattern \* pattern,  
     FILE \* file )

**18.30.4.236 writeXxx()** char writeXxx (  
     EmbPattern \* pattern,  
     FILE \* file )

**18.30.4.237 writeZsk()** char writeZsk (  
     EmbPattern \* pattern,  
     FILE \* file )

## 18.30.5 Variable Documentation

**18.30.5.1 imageWithFrame** const char imageWithFrame[38][48] [extern]

## 18.31 embroidery\_internal.h

[Go to the documentation of this file.](#)

```
1 #ifndef LIBEMBROIDERY_INTERNAL_HEADER__
2 #define LIBEMBROIDERY_INTERNAL_HEADER__
3
4 #include "embroidery.h"
5
10 /* For FILE * */
11 #include <stdio.h>
12
16 #define CompoundFileSector_MaxRegSector 0xFFFFFFFFFA
17 #define CompoundFileSector_DIFAT_Sector 0xFFFFFFFFFC
18 #define CompoundFileSector_FAT_Sector 0xFFFFFFFFFD
19 #define CompoundFileSector_EndOfChain 0xFFFFFFFFFE
20 #define CompoundFileSector_FreeSector 0xFFFFFFFFFF
21
25 #define ObjectTypeUnknown 0x00
26 #define ObjectTypeStorage 0x01
27 #define ObjectTypeStream 0x02
28 #define ObjectTypeRootEntry 0x05
33 #define CompoundFileStreamId_MaxRegularStreamId 0xFFFFFFFFFA
34 #define CompoundFileStreamId_NoStream 0xFFFFFFFFFF
36 #define ELEMENT_XML 0
```

```

37 #define ELEMENT_A 1
38 #define ELEMENT_ANIMATE 2
39 #define ELEMENT_ANIMATECOLOR 3
40 #define ELEMENT_ANIMATEMOTION 4
41 #define ELEMENT_ANIMATETRANSFORM 5
42 #define ELEMENT_ANIMATION 6
43 #define ELEMENT_AUDIO 7
44 #define ELEMENT_CIRCLE 8
45 #define ELEMENT_DEFS 9
46 #define ELEMENT_DESC 10
47 #define ELEMENT_DISCARD 11
48 #define ELEMENT_ELLIPSE 12
49 #define ELEMENT_FONT 13
50 #define ELEMENT_FONT_FACE 14
51 #define ELEMENT_FONT_FACE_SRC 15
52 #define ELEMENT_FONT_FACE_URI 16
53 #define ELEMENT_FOREIGN_OBJECT 17
54 #define ELEMENT_G 18
55 #define ELEMENT_GLYPH 19
56 #define ELEMENT_HANDLER 20
57 #define ELEMENT_HKERN 21
58 #define ELEMENT_IMAGE 22
59 #define ELEMENT_LINE 23
60 #define ELEMENT_LINEAR_GRADIENT 24
61 #define ELEMENT_LISTENER 25
62 #define ELEMENT_METADATA 26
63 #define ELEMENT_MISSING_GLYPH 27
64 #define ELEMENT_MPATH 28
65 #define ELEMENT_PATH 29
66 #define ELEMENT_POLYGON 30
67 #define ELEMENT_POLYLINE 31
68 #define ELEMENT_PREFETCH 32
69 #define ELEMENT_RADIAL_GRADIENT 33
70 #define ELEMENT_RECT 34
71 #define ELEMENT_SCRIPT 35
72 #define ELEMENT_SET 36
73 #define ELEMENT_SOLID_COLOR 37
74 #define ELEMENT_STOP 38
75 #define ELEMENT_SVG 39
76 #define ELEMENT_SWITCH 40
77 #define ELEMENT_TBREAK 41
78 #define ELEMENT_TEXT 42
79 #define ELEMENT_TEXT_AREA 43
80 #define ELEMENT_TITLE 44
81 #define ELEMENT_TSPAN 45
82 #define ELEMENT_USE 46
83 #define ELEMENT_VIDEO 47
84
85 /* INTERNAL DEFINES */
86 #define RED_TERM_COLOR "\x1B[0;31m"
87 #define GREEN_TERM_COLOR "\x1B[0;32m"
88 #define YELLOW_TERM_COLOR "\x1B[1;33m"
89 #define RESET_TERM_COLOR "\033[0m"
90
91 #define HOOP_126X110 0
92 #define HOOP_110X110 1
93 #define HOOP_50X50 2
94 #define HOOP_140X200 3
95 #define HOOP_230X200 4
96
97 #define EMB_MIN(A, B) ((A) < (B)) ? (A) : (B)
98 #define EMB_MAX(A, B) ((A) > (B)) ? (A) : (B)
99
100 /* Libembroidery's handling of integer types.
101 */
102 #define EMB_BIG_ENDIAN 0
103 #define EMB_LITTLE_ENDIAN 1
104
105 #define ENDIAN_HOST EMB_LITTLE_ENDIAN
106
107 #define EMB_INT16_BIG 2
108 #define EMB_INT16_LITTLE 3
109 #define EMB_INT32_BIG 4
110 #define EMB_INT32_LITTLE 5
111
112 #define PES0001 0
113 #define PES0020 1
114 #define PES0022 2
115 #define PES0030 3
116 #define PES0040 4
117 #define PES0050 5
118 #define PES0055 6
119 #define PES0056 7
120 #define PES0060 8
121 #define PES0070 9
122 #define PES0080 10
123 #define PES0090 11

```

```

124 #define PES0100          12
125 #define N_PES_VERSIONS 13
126
127 /* DXF Version Identifiers */
128 #define DXF_VERSION_R10 "AC1006"
129 #define DXF_VERSION_R11 "AC1009"
130 #define DXF_VERSION_R12 "AC1009"
131 #define DXF_VERSION_R13 "AC1012"
132 #define DXF_VERSION_R14 "AC1014"
133 #define DXF_VERSION_R15 "AC1015"
134 #define DXF_VERSION_R18 "AC1018"
135 #define DXF_VERSION_R21 "AC1021"
136 #define DXF_VERSION_R24 "AC1024"
137 #define DXF_VERSION_R27 "AC1027"
138
139 #define DXF_VERSION_2000 "AC1015"
140 #define DXF_VERSION_2002 "AC1015"
141 #define DXF_VERSION_2004 "AC1018"
142 #define DXF_VERSION_2006 "AC1018"
143 #define DXF_VERSION_2007 "AC1021"
144 #define DXF_VERSION_2009 "AC1021"
145 #define DXF_VERSION_2010 "AC1024"
146 #define DXF_VERSION_2013 "AC1027"
147
148 #define SVG_CREATOR_NULL          0
149 #define SVG_CREATOR_EMBROIDERMODDER 1
150 #define SVG_CREATOR_ILLUSTRATOR   2
151 #define SVG_CREATOR_INKSCAPE      3
152
153 #define SVG_EXPECT_NULL          0
154 #define SVG_EXPECT_ELEMENT       1
155 #define SVG_EXPECT_ATTRIBUTE     2
156 #define SVG_EXPECT_VALUE         3
157
158 /* SVG_TYPES
159 * -----
160 */
161 #define SVG_NULL          0
162 #define SVG_ELEMENT       1
163 #define SVG_PROPERTY      2
164 #define SVG_MEDIA_PROPERTY 3
165 #define SVG_ATTRIBUTE      4
166 #define SVG_CATCH_ALL     5
167
168 /* path flag codes */
169 #define LINETO          0
170 #define MOVETO          1
171 #define BULGETOCONTROL  2
172 #define BULGETOEND      4
173 #define ELLIPSETORAD     8
174 #define ELLIPSETOEND    16
175 #define CUBICTOCONTROL1  32
176 #define CUBICTOCONTROL2  64
177 #define CUBICTOEND      128
178 #define QUADTOCONTROL    256
179 #define QUADTOEND        512
180
181 /* STRUCTS
182 *****/
183
184 /* double-indirection file allocation table references */
185
186 typedef struct _bcf_file_difat
187 {
188     unsigned int fatSectorCount;
189     unsigned int fatSectorEntries[109];
190     unsigned int sectorSize;
191 } bcf_file_difat;
192
193 typedef struct _bcf_file_fat
194 {
195     int fatEntryCount;
196     unsigned int fatEntries[255]; /* maybe make this dynamic */
197     unsigned int numberOfEntriesInFatSector;
198 } bcf_file_fat;
199
200 typedef struct _bcf_directory_entry
201 {
202     char directoryEntryName[32];
203     unsigned short directoryEntryNameLength;
204     unsigned char objectType;
205     unsigned char colorFlag;
206     unsigned int leftSiblingId;
207     unsigned int rightSiblingId;
208     unsigned int childId;
209     unsigned char CLSID[16];
210     unsigned int stateBits;

```

```

223     EmbTime                creationTime;
224     EmbTime                modifiedTime;
225     unsigned int           startingSectorLocation;
226     unsigned long          streamSize; /* should be long long but in our case we shouldn't need
    it, and hard to support on c89 cross platform */
227     unsigned int           streamSizeHigh; /* store the high int of streamsize */
228     struct _bcf_directory_entry* next;
229 } bcf_directory_entry;
230
231 typedef struct _bcf_directory
232 {
233     bcf_directory_entry* dirEntries;
234     unsigned int         maxNumberOfDirectoryEntries;
235 } bcf_directory;
236
237 typedef struct _bcf_file_header
238 {
239     unsigned char    signature[8];
240     unsigned char    CLSID[16];
241     unsigned short   minorVersion;
242     unsigned short   majorVersion;
243     unsigned short   byteOrder;
244     unsigned short   sectorShift;
245     unsigned short   miniSectorShift;
246     unsigned short   reserved1;
247     unsigned int      reserved2;
248     unsigned int      numberOfDirectorySectors;
249     unsigned int      numberOfFATSectors;
250     unsigned int      firstDirectorySectorLocation;
251     unsigned int      transactionSignatureNumber;
252     unsigned int      miniStreamCutoffSize;
253     unsigned int      firstMiniFATSectorLocation;
254     unsigned int      numberOfMiniFatSectors;
255     unsigned int      firstDifatSectorLocation;
256     unsigned int      numberOfDifatSectors;
257 } bcf_file_header;
258
259 typedef struct _bcf_file
260 {
261     bcf_file_header header;
262     bcf_file_difat* difat;
263     bcf_file_fat* fat;
264     bcf_directory* directory;
265 } bcf_file;
266
267 typedef struct _vp3Hoop
268 {
269     int    right;
270     int    bottom;
271     int    left;
272     int    top;
273     int    threadLength;
274     char    unknown2;
275     unsigned char    numberOfColors;
276     unsigned short    unknown3;
277     int    unknown4;
278     int    numberOfBytesRemaining;
279
280     int    xOffset;
281     int    yOffset;
282
283     unsigned char    byte1;
284     unsigned char    byte2;
285     unsigned char    byte3;
286
287     /* Centered hoop dimensions */
288     int    right2;
289     int    left2;
290     int    bottom2;
291     int    top2;
292
293     int    width;
294     int    height;
295 } vp3Hoop;
296
297 typedef struct ThredHeader_ /* thred file header */
298 {
299     unsigned int    sigVersion; /* signature and version */
300     unsigned int    length; /* length of ThredHeader + length of stitch data */
301     unsigned short    numStiches; /* number of stitches */
302     unsigned short    hoopSize; /* size of hoop */
303     unsigned short    reserved[7]; /* reserved for expansion */
304 } ThredHeader;
305
306 typedef struct ThredExtension_ /* thred v1.0 file header extension */
307 {
308     float    hoopX; /* hoop size x dimension in 1/6 mm units */

```

```

334     float hoopY;                /* hoop size y dimension in 1/6 mm units */
335     float stitchGranularity;    /* stitches per millimeter--not implemented */
336     char creatorName[50];       /* name of the file creator */
337     char modifierName[50];      /* name of last file modifier */
338     char auxFormat;             /* auxiliary file format, 0=PCS,1=DST,2=PES */
339     char reserved[31];         /* reserved for expansion */
340 } ThredExtension;
341
342 typedef struct SubDescriptor_
343 {
344     int someNum;
345     int someInt;
346     int someOtherInt;
347     char* colorCode;
348     char* colorName;
349 } SubDescriptor;
350
351 typedef struct StxThread_
352 {
353     char* colorCode;
354     char* colorName;
355     char* sectionName;
356     SubDescriptor* subDescriptors;
357     EmbColor stxColor;
358 } StxThread;
359
360 typedef struct VipHeader_ {
361     int magicCode;
362     int numberOfStitches;
363     int numberOfColors;
364     short positiveXHoopSize;
365     short positiveYHoopSize;
366     short negativeXHoopSize;
367     short negativeYHoopSize;
368     int attributeOffset;
369     int xOffset;
370     int yOffset;
371     unsigned char stringVal[8];
372     short unknown;
373     int colorLength;
374 } VipHeader;
375
376 typedef enum
377 {
378     CSV_EXPECT_NULL,
379     CSV_EXPECT_QUOTE1,
380     CSV_EXPECT_QUOTE2,
381     CSV_EXPECT_COMMA
382 } CSV_EXPECT;
383
384 typedef enum
385 {
386     CSV_MODE_NULL,
387     CSV_MODE_COMMENT,
388     CSV_MODE_VARIABLE,
389     CSV_MODE_THREAD,
390     CSV_MODE_STITCH
391 } CSV_MODE;
392
393 typedef struct SvgAttribute_
394 {
395     char* name;
396     char* value;
397 } SvgAttribute;
398
399 typedef struct Huffman {
400     int default_value;
401     int lengths[1000];
402     int nlengths;
403     int table[1000];
404     int table_width;
405     int ntable;
406 } Huffman;
407
408 typedef struct Compress {
409     int bit_position;
410     char *input_data;
411     int input_length;
412     int bits_total;
413     int block_elements;
414     Huffman character_length_huffman;
415     Huffman character_huffman;
416     Huffman distance_huffman;
417 } Compress;
418
419 /* Function Declarations
420 *****/

```

```

453 void huffman_build_table(huffman *h);
454 int *huffman_table_lookup(huffman *h, int byte_lookup, int *lengths);
455
456 int compress_get_bits(compress *c, int length);
457 int compress_pop(compress *c, int bit_count);
458 int compress_read_variable_length(compress *c);
459 void compress_load_character_length_huffman(compress *c);
460 void compress_load_character_huffman(compress *c);
461 void compress_load_distance_huffman(compress *c);
462 void compress_load_block(compress *c);
463 int compress_get_token(compress *c);
464 int compress_get_position(compress *c);
465
466 void readPecStitches(EmbPattern* pattern, FILE* file);
467 void writePecStitches(EmbPattern* pattern, FILE* file, const char* filename);
468
469 int decodeNewStitch(unsigned char value);
470
471 void pfaffEncode(FILE* file, int x, int y, int flags);
472 EmbReal pfaffDecode(unsigned char a1, unsigned char a2, unsigned char a3);
473
474 unsigned char mitEncodeStitch(EmbReal value);
475 int mitDecodeStitch(unsigned char value);
476
477 int encode_tajima_ternary(unsigned char b[3], int x, int y);
478 void decode_tajima_ternary(unsigned char b[3], int *x, int *y);
479
480 void encode_t01_record(unsigned char b[3], int x, int y, int flags);
481 int decode_t01_record(unsigned char b[3], int *x, int *y, int *flags);
482 void readPESHeaderV5(FILE* file, EmbPattern* pattern);
483 void readPESHeaderV6(FILE* file, EmbPattern* pattern);
484 void readPESHeaderV7(FILE* file, EmbPattern* pattern);
485 void readPESHeaderV8(FILE* file, EmbPattern* pattern);
486 void readPESHeaderV9(FILE* file, EmbPattern* pattern);
487 void readPESHeaderV10(FILE* file, EmbPattern* pattern);
488
489 void readDescriptions(FILE* file, EmbPattern* pattern);
490 void readHoopName(FILE* file, EmbPattern* pattern);
491 void readImageString(FILE* file, EmbPattern* pattern);
492 void readProgrammableFills(FILE* file, EmbPattern* pattern);
493 void readMotifPatterns(FILE* file, EmbPattern* pattern);
494 void readFeatherPatterns(FILE* file, EmbPattern* pattern);
495 void readThreads(FILE* file, EmbPattern* pattern);
496
497 void embInt_read(FILE* f, char *label, void *b, int mode);
498 void embInt_write(FILE* f, char *label, void *b, int mode);
499 int emb_readline(FILE* file, char *line, int maxLength);
500
501 int bcfFile_read(FILE* file, bcf_file* bcfFile);
502 FILE* GetFile(bcf_file* bcfFile, FILE* file, char* fileToFind);
503 void bcf_file_free(bcf_file* bcfFile);
504
505 void binaryReadString(FILE* file, char *buffer, int maxLength);
506 void binaryReadUnicodeString(FILE* file, char *buffer, const int stringLength);
507
508 int stringInArray(const char *s, const char **array);
509 void fpad(FILE *f, char c, int n);
510 char *copy_trim(char const *s);
511 char* emb_optOut(EmbReal num, char* str);
512
513 void write_24bit(FILE* file, int);
514 int check_header_present(FILE* file, int minimum_header_length);
515
516 unsigned short fread_uint16(FILE *file);
517 short fread_int16(FILE* f);
518 int fread_int32_be(FILE* f);
519 void safe_free(void *data);
520 void embInt_read(FILE* f, char *label, void *b, int mode);
521
522 void binaryWriteUIntBE(FILE* f, unsigned int data);
523 void binaryWriteUInt(FILE* f, unsigned int data);
524 void binaryWriteIntBE(FILE* f, int data);
525 void binaryWriteInt(FILE* f, int data);
526 void binaryWriteUShort(FILE* f, unsigned short data);
527 void binaryWriteUShortBE(FILE* f, unsigned short data);
528 void binaryWriteShort(FILE* f, short data);
529
530 bcf_file_difat* bcf_difat_create(FILE* file, unsigned int fatSectors, const unsigned int sectorSize);
531 unsigned int readFullSector(FILE* file, bcf_file_difat* bcfFile, unsigned int*
    numberOfDifatEntriesStillToRead);
532 unsigned int numberOfEntriesInDifatSector(bcf_file_difat* fat);
533 void bcf_file_difat_free(bcf_file_difat* difat);
534
535 unsigned int entriesInDifatSector(bcf_file_difat* fat);
536 bcf_file_fat* bcfFileFat_create(const unsigned int sectorSize);
537 void loadFatFromSector(bcf_file_fat* fat, FILE* file);
538 void bcf_file_fat_free(bcf_file_fat** fat);

```

```

539
540 bcf_directory_entry* CompoundFileDirectoryEntry(FILE* file);
541 bcf_directory* CompoundFileDirectory(const unsigned int maxNumberOfDirectoryEntries);
542 void readNextSector(FILE* file, bcf_directory* dir);
543 void bcf_directory_free(bcf_directory** dir);
544
545 bcf_file_header bcfFileHeader_read(FILE* file);
546 int bcfFileHeader_isValid(bcf_file_header header);
547
548 int hus_compress(char* input, int size, char* output, int *out_size);
549 int hus_decompress(char* input, int size, char* output, int *out_size);
550
551 int encode_tajima_ternary(unsigned char b[3], int x, int y);
552 void decode_tajima_ternary(unsigned char b[3], int *x, int *y);
553 void testTangentPoints(EmbCircle c, EmbVector p, EmbVector *t0, EmbVector *t1);
554 void printArcResults(EmbReal bulge, EmbArc arc,
555                     EmbReal centerX, EmbReal centerY,
556                     EmbReal radius, EmbReal diameter,
557                     EmbReal chord,
558                     EmbReal chordMidX, EmbReal chordMidY,
559                     EmbReal sagitta, EmbReal apothem,
560                     EmbReal incAngle, char clockwise);
561 int create_test_file_1(const char* outf);
562 int create_test_file_2(const char* outf);
563 int create_test_file_3(const char* outf);
564 int testEmbCircle(void);
565 int testEmbCircle_2(void);
566 int testGeomArc(void);
567 int testThreadColor(void);
568 int testEmbFormat(void);
569
570 void embColor_read(FILE *f, EmbColor *c, int toRead);
571 void embColor_write(FILE *f, EmbColor c, int toWrite);
572
573 char read100(EmbPattern *pattern, FILE* file);
574 char write100(EmbPattern *pattern, FILE* file);
575 char read10o(EmbPattern *pattern, FILE* file);
576 char write10o(EmbPattern *pattern, FILE* file);
577 char readArt(EmbPattern *pattern, FILE* file);
578 char writeArt(EmbPattern *pattern, FILE* file);
579 char readBmc(EmbPattern *pattern, FILE* file);
580 char writeBmc(EmbPattern *pattern, FILE* file);
581 char readBro(EmbPattern *pattern, FILE* file);
582 char writeBro(EmbPattern *pattern, FILE* file);
583 char readCnd(EmbPattern *pattern, FILE* file);
584 char writeCnd(EmbPattern *pattern, FILE* file);
585 char readCol(EmbPattern *pattern, FILE* file);
586 char writeCol(EmbPattern *pattern, FILE* file);
587 char readCsd(EmbPattern *pattern, FILE* file);
588 char writeCsd(EmbPattern *pattern, FILE* file);
589 char readCsv(EmbPattern *pattern, FILE* file);
590 char writeCsv(EmbPattern *pattern, FILE* file);
591 char readDat(EmbPattern *pattern, FILE* file);
592 char writeDat(EmbPattern *pattern, FILE* file);
593 char readDem(EmbPattern *pattern, FILE* file);
594 char writeDem(EmbPattern *pattern, FILE* file);
595 char readDsb(EmbPattern *pattern, FILE* file);
596 char writeDsb(EmbPattern *pattern, FILE* file);
597 char readDst(EmbPattern *pattern, FILE* file);
598 char writeDst(EmbPattern *pattern, FILE* file);
599 char readDsz(EmbPattern *pattern, FILE* file);
600 char writeDsz(EmbPattern *pattern, FILE* file);
601 char readDxf(EmbPattern *pattern, FILE* file);
602 char writeDxf(EmbPattern *pattern, FILE* file);
603 char readEdr(EmbPattern *pattern, FILE* file);
604 char writeEdr(EmbPattern *pattern, FILE* file);
605 char readEmd(EmbPattern *pattern, FILE* file);
606 char writeEmd(EmbPattern *pattern, FILE* file);
607 char readExp(EmbPattern *pattern, FILE* file);
608 char writeExp(EmbPattern *pattern, FILE* file);
609 char readExy(EmbPattern *pattern, FILE* file);
610 char writeExy(EmbPattern *pattern, FILE* file);
611 char readEys(EmbPattern *pattern, FILE* file);
612 char writeEys(EmbPattern *pattern, FILE* file);
613 char readFxy(EmbPattern *pattern, FILE* file);
614 char writeFxy(EmbPattern *pattern, FILE* file);
615 char readGc(EmbPattern *pattern, FILE* file);
616 char writeGc(EmbPattern *pattern, FILE* file);
617 char readGnc(EmbPattern *pattern, FILE* file);
618 char writeGnc(EmbPattern *pattern, FILE* file);
619 char readGt(EmbPattern *pattern, FILE* file);
620 char writeGt(EmbPattern *pattern, FILE* file);
621 char readHus(EmbPattern *pattern, FILE* file);
622 char writeHus(EmbPattern *pattern, FILE* file);
623 char readInb(EmbPattern *pattern, FILE* file);
624 char writeInb(EmbPattern *pattern, FILE* file);
625 char readInf(EmbPattern *pattern, FILE* file);

```

```

626 char writeInf(EmbPattern *pattern, FILE* file);
627 char readJef(EmbPattern *pattern, FILE* file);
628 char writeJef(EmbPattern *pattern, FILE* file);
629 char readKsm(EmbPattern *pattern, FILE* file);
630 char writeKsm(EmbPattern *pattern, FILE* file);
631 char readMax(EmbPattern *pattern, FILE* file);
632 char writeMax(EmbPattern *pattern, FILE* file);
633 char readMit(EmbPattern *pattern, FILE* file);
634 char writeMit(EmbPattern *pattern, FILE* file);
635 char readNew(EmbPattern *pattern, FILE* file);
636 char writeNew(EmbPattern *pattern, FILE* file);
637 char readOfm(EmbPattern *pattern, FILE* file);
638 char writeOfm(EmbPattern *pattern, FILE* file);
639 char readPcd(EmbPattern *pattern, const char *fileName, FILE* file);
640 char writePcd(EmbPattern *pattern, FILE* file);
641 char readPcm(EmbPattern *pattern, FILE* file);
642 char writePcm(EmbPattern *pattern, FILE* file);
643 char readPcq(EmbPattern *pattern, const char *fileName, FILE* file);
644 char writePcq(EmbPattern *pattern, FILE* file);
645 char readPcs(EmbPattern *pattern, const char *fileName, FILE* file);
646 char writePcs(EmbPattern *pattern, FILE* file);
647 char readPec(EmbPattern *pattern, const char *fileName, FILE* file);
648 char writePec(EmbPattern *pattern, const char *fileName, FILE* file);
649 char readPel(EmbPattern *pattern, FILE *file);
650 char writePel(EmbPattern *pattern, FILE *file);
651 char readPem(EmbPattern *pattern, FILE *file);
652 char writePem(EmbPattern *pattern, FILE *file);
653 char readPes(EmbPattern *pattern, const char *fileName, FILE* file);
654 char writePes(EmbPattern *pattern, const char *fileName, FILE* file);
655 char readPhb(EmbPattern *pattern, FILE* file);
656 char writePhb(EmbPattern *pattern, FILE *file);
657 char readPhc(EmbPattern *pattern, FILE* file);
658 char writePhc(EmbPattern *pattern, FILE *file);
659 char readPlt(EmbPattern *pattern, FILE* file);
660 char writePlt(EmbPattern *pattern, FILE* file);
661 char readRgb(EmbPattern *pattern, FILE* file);
662 char writeRgb(EmbPattern *pattern, FILE* file);
663 char readSew(EmbPattern *pattern, FILE* file);
664 char writeSew(EmbPattern *pattern, FILE* file);
665 char readShv(EmbPattern *pattern, FILE* file);
666 char writeShv(EmbPattern *pattern, FILE *file);
667 char readSst(EmbPattern *pattern, FILE* file);
668 char writeSst(EmbPattern *pattern, FILE *file);
669 char readStx(EmbPattern *pattern, FILE* file);
670 char writeStx(EmbPattern *pattern, FILE *file);
671 char readSvg(EmbPattern *pattern, FILE* file);
672 char writeSvg(EmbPattern *pattern, FILE* file);
673 char readT01(EmbPattern *pattern, FILE* file);
674 char writeT01(EmbPattern *pattern, FILE* file);
675 char readT09(EmbPattern *pattern, FILE* file);
676 char writeT09(EmbPattern *pattern, FILE* file);
677 char readTap(EmbPattern *pattern, FILE* file);
678 char writeTap(EmbPattern *pattern, FILE* file);
679 char readThr(EmbPattern *pattern, FILE* file);
680 char writeThr(EmbPattern *pattern, FILE* file);
681 char readTxt(EmbPattern *pattern, FILE* file);
682 char writeTxt(EmbPattern *pattern, FILE* file);
683 char readU00(EmbPattern *pattern, FILE* file);
684 char writeU00(EmbPattern *pattern, FILE *file);
685 char readU01(EmbPattern *pattern, FILE* file);
686 char writeU01(EmbPattern *pattern, FILE *file);
687 char readVip(EmbPattern *pattern, FILE* file);
688 char writeVip(EmbPattern *pattern, FILE* file);
689 char readVp3(EmbPattern *pattern, FILE* file);
690 char writeVp3(EmbPattern *pattern, FILE* file);
691 char readXxx(EmbPattern *pattern, FILE* file);
692 char writeXxx(EmbPattern *pattern, FILE* file);
693 char readZsk(EmbPattern *pattern, FILE* file);
694 char writeZsk(EmbPattern *pattern, FILE* file);
695
696 extern const char imageWithFrame[38][48];
697
698 #endif

```

## 18.32 extern/libembroidery/src/encoding.c File Reference

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "embroidery_internal.h"

```



## Functions

- void `write_24bit` (FILE \*file, int)
  - file x*
- `EmbColor embColor_fromHexStr` (char \*val)
  - Converts a 6 digit hex string (I.E. "00FF00") into an EmbColor and returns it.*
- void `reverse_byte_order` (void \*b, int bytes)
- int `decode_t01_record` (unsigned char b[3], int \*x, int \*y, int \*flags)
- void `encode_t01_record` (unsigned char b[3], int x, int y, int flags)
- int `encode_tajima_ternary` (unsigned char b[3], int x, int y)
- void `decode_tajima_ternary` (unsigned char b[3], int \*x, int \*y)
- void `pfaffEncode` (FILE \*file, int dx, int dy, int flags)
- `EmbReal pfaffDecode` (unsigned char a1, unsigned char a2, unsigned char a3)
- unsigned char `mitEncodeStitch` (`EmbReal` value)
  - value*
- int `mitDecodeStitch` (unsigned char value)
  - value*
- int `decodeNewStitch` (unsigned char value)
  - value*
- void `emblnt_read` (FILE \*f, char \*label, void \*b, int mode)
- void `emblnt_write` (FILE \*f, char \*label, void \*b, int mode)

### 18.32.1 Detailed Description

The functions in this file are grouped together to aid the developer's understanding of the similarities between the file formats. This also helps reduce errors between reimplementations of the same idea.

For example: the Tajima ternary encoding of positions is used by at least 4 formats and the only part that changes is the flag encoding.

### 18.32.2 Function Documentation

**18.32.2.1 `decode_t01_record()`** int decode\_t01\_record (
 unsigned char b[3],
 int \* x,
 int \* y,
 int \* flags )

*b x y flags .*

**Todo** remove the unused return argument.

**18.32.2.2 `decode_tajima_ternary()`** void decode\_tajima\_ternary (
 unsigned char b[3],
 int \* x,
 int \* y )

Decode the signed ternary of the tajima format from *b* to the position values *x* and *y*. There is no return argument.

**18.32.2.3 `decodeNewStitch()`** int decodeNewStitch (
 unsigned char value )

*value*

Returns

int

**18.32.2.4 embColor\_fromHexStr()** `EmbColor embColor_fromHexStr (`  
`char * val )`

Converts a 6 digit hex string (I.E. "00FF00") into an EmbColor and returns it.  
*val* 6 byte code describing the color as a hex string, doesn't require null termination.

Returns

EmbColor the same color as our internal type.

**18.32.2.5 embInt\_read()** `void embInt_read (`  
`FILE * f,`  
`char * label,`  
`void * b,`  
`int mode )`

*f label b mode*

Read and write system for multiple byte types.

The caller passes the function to read/write from, the memory location as a void pointer and a mode identifier that describes the type. This way we can abstract out the endianness of the system running the library and don't have to maintain many functions, just two.

**18.32.2.6 embInt\_write()** `void embInt_write (`  
`FILE * f,`  
`char * label,`  
`void * b,`  
`int mode )`

*f label b mode*

**18.32.2.7 encode\_t01\_record()** `void encode_t01_record (`  
`unsigned char b[3],`  
`int x,`  
`int y,`  
`int flags )`

Encode into bytes *b* the values of the x-position *x*, y-position *y* and the *flags*.

**18.32.2.8 encode\_tajima\_ternary()** `int encode_tajima_ternary (`  
`unsigned char b[3],`  
`int x,`  
`int y )`

Encode the signed ternary of the tajima format into *b* the position values *x* and *y*.

If the values of *x* or *y* fall outside of the valid range of -121 and +121 then it returns 0 and 1.

**18.32.2.9 mitDecodeStitch()** `int mitDecodeStitch (`  
`unsigned char value )`

*value*

Returns

int

**18.32.2.10 mitEncodeStitch()** `unsigned char mitEncodeStitch (`  
`EmbReal value )`

*value*

Returns

unsigned char

**18.32.2.11 pfaffDecode()** `EmbReal pfaffDecode (`  
     unsigned char *a1*,  
     unsigned char *a2*,  
     unsigned char *a3* )

Decode the bytes *a1*, *a2* and *a3*. Returns the EmbReal floating-point value.

**18.32.2.12 pfaffEncode()** `void pfaffEncode (`  
     FILE \* *file*,  
     int *dx*,  
     int *dy*,  
     int *flags* )

*file dx dy flags*

**18.32.2.13 reverse\_byte\_order()** `void reverse_byte_order (`  
     void \* *b*,  
     int *bytes* )

Reverses the byte order of *bytes* number of bytes at memory location *b*. Only works for 2 or 4 byte arrays.

**18.32.2.14 write\_24bit()** `void write_24bit (`  
     FILE \* *file*,  
     int *x* )

*file x*

## 18.33 extern/libembroidery/src/fill.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "embroidery_internal.h"
```

### Functions

- int [lindenmayer\\_system](#) (L\_system *L*, char \**state*, int *iterations*, int *complete*)
- static void [join\\_short\\_stitches](#) (int \**points*, int *n\_points*, int *width*, int *tolerance*)
- static int \* [threshold\\_method](#) (EmbImage \**image*, int *n\_points*, int *subsample\_width*, int *subsample\_height*, int *threshold*)
- static void [greedy\\_algorithm](#) (int \**points*, int *n\_points*, int *width*, EmbReal *bias*)
- static void [save\\_points\\_to\\_pattern](#) (EmbPattern \**pattern*, int \**points*, int *n\_points*, EmbReal *scale*, int *width*, int *height*)
- void [embPattern\\_horizontal\\_fill](#) (EmbPattern \**pattern*, EmbImage \**image*, int *threshold*)
- void [embPattern\\_crossstitch](#) (EmbPattern \**pattern*, EmbImage \**image*, int *threshold*)
- int [hilbert\\_curve](#) (EmbPattern \**pattern*, int *iterations*)
- void [generate\\_dragon\\_curve](#) (char \**state*, int *iterations*)
- int [dragon\\_curve](#) (int *iterations*)
- void [embPolygon\\_reduceByDistance](#) (EmbArray \**vertices*, EmbArray \**simplified*, float *distance*)
- void [embPolygon\\_reduceByNth](#) (EmbArray \**vertices*, EmbArray \**out*, int *nth*)
- EmbPattern \* [embPattern\\_combine](#) (EmbPattern \**p1*, EmbPattern \**p2*)
- void [embPattern\\_stitchArc](#) (EmbPattern \**p*, EmbArc *arc*, int *thread\_index*, int *style*)
- void [embPattern\\_stitchCircle](#) (EmbPattern \**p*, EmbCircle *circle*, int *thread\_index*, int *style*)
- void [embPattern\\_stitchEllipse](#) (EmbPattern \**p*, EmbEllipse *ellipse*, int *thread\_index*, int *style*)
- void [embPattern\\_stitchPath](#) (EmbPattern \**p*, EmbPath *path*, int *thread\_index*, int *style*)
- void [embPattern\\_stitchPolygon](#) (EmbPattern \**p*, EmbPolygon *polygon*, int *thread\_index*, int *style*)
- void [embPattern\\_stitchPolyline](#) (EmbPattern \**p*, EmbPolyline *polyline*, int *thread\_index*, int *style*)
- void [embPattern\\_stitchRect](#) (EmbPattern \**p*, EmbRect *rect*, int *thread\_index*, int *style*)
- void [embPattern\\_stitchText](#) (EmbPattern \**p*, EmbRect *rect*, int *thread\_index*, int *style*)
- void [embPattern\\_convertGeometry](#) (EmbPattern \**p*)

## Variables

- `const char * rules [] = {"+BF-AFA-FB+", "-AF+BFB+FA-"}`
- `L_system hilbert_curve_l_system`

### 18.33.1 Function Documentation

**18.33.1.1 dragon\_curve()** `int dragon_curve (`  
     `int iterations )`

Create the dragon curve for *iterations*.

Returns 0 if the number of iterations is greater than 10 and 1 otherwise.

**18.33.1.2 embPattern\_combine()** `EmbPattern * embPattern_combine (`  
     `EmbPattern * p1,`  
     `EmbPattern * p2 )`

*p1 p2*

Returns

`EmbPattern*`

**18.33.1.3 embPattern\_convertGeometry()** `void embPattern_convertGeometry (`  
     `EmbPattern * p )`

*p*

**18.33.1.4 embPattern\_crossstitch()** `void embPattern_crossstitch (`  
     `EmbPattern * pattern,`  
     `EmbImage * image,`  
     `int threshold )`

*pattern image threshold*

Uses a threshold method to determine where to put crosses in the fill.

To improve this, we can remove the vertical stitches when two crosses neighbour. Currently the simple way to do this is to chain crosses that are neighbours exactly one ahead.

**18.33.1.5 embPattern\_horizontal\_fill()** `void embPattern_horizontal_fill (`  
     `EmbPattern * pattern,`  
     `EmbImage * image,`  
     `int threshold )`

*pattern image threshold*

Uses a threshold method to determine where to put lines in the fill.

Needs to pass a "donut test", i.e. an image with black pixels where:  $10 < x*x + y*y < 20$  over the area  $(-30, 30) \times (-30, 30)$ .

Use render then image difference to see how well it passes.

**18.33.1.6 embPattern\_stitchArc()** `void embPattern_stitchArc (`  
     `EmbPattern * p,`  
     `EmbArc arc,`  
     `int thread_index,`  
     `int style )`

*p arc thread\_index style*

**18.33.1.7 embPattern\_stitchCircle()** void embPattern\_stitchCircle (  
     EmbPattern \* p,  
     EmbCircle circle,  
     int thread\_index,  
     int style )

*p circle thread\_index style*

style determines: stitch density fill pattern outline or fill

For now it's a straight fill of 1000 stitches of the whole object by default.

Consider the intersection of a line in direction "d" that passes through the disc with center "c", radius "r". The start and end points are:

$(c - r(d/|d|), c + r(d/|d|))$

Lines that are above and below this with an even separation  $ss$  can be found by taking the point on the line to be  $c + sn$  where the  $n$  is the unit normal vector to  $d$  and the vector to be  $d$  again. The intersection points are therefore a right angled triangle, with one side  $r$ , another  $s$  and the third the length to be solved, by Pythagoras we have:

$(c + sn - \sqrt{r^2 - s^2}(d/|d|), c + sn + \sqrt{r^2 - s^2}(d/|d|))$

repeating this process gives us all the end points and the fill only alters these lines by splitting the ones longer than some tolerance.

**18.33.1.8 embPattern\_stitchEllipse()** void embPattern\_stitchEllipse (  
     EmbPattern \* p,  
     EmbEllipse ellipse,  
     int thread\_index,  
     int style )

*p ellipse thread\_index style*

**Todo** finish stitchEllipse

**18.33.1.9 embPattern\_stitchPath()** void embPattern\_stitchPath (  
     EmbPattern \* p,  
     EmbPath path,  
     int thread\_index,  
     int style )

*p rect thread\_index style*

**Todo** finish stitch path

**18.33.1.10 embPattern\_stitchPolygon()** void embPattern\_stitchPolygon (  
     EmbPattern \* p,  
     EmbPolygon polygon,  
     int thread\_index,  
     int style )

*p rect thread\_index style*

**Todo** finish stitch polygon

**18.33.1.11 embPattern\_stitchPolyline()** void embPattern\_stitchPolyline (  
     EmbPattern \* p,  
     EmbPolyline polyline,  
     int thread\_index,  
     int style )

*p rect thread\_index style*

**Todo** finish stitch polyline

**18.33.1.12 embPattern\_stitchRect()** void embPattern\_stitchRect (   
     EmbPattern \* p,   
     EmbRect rect,   
     int thread\_index,   
     int style )

*p rect thread\_index style*

Here we just stitch the rectangle in the direction of it's longer side.

**18.33.1.13 embPattern\_stitchText()** void embPattern\_stitchText (   
     EmbPattern \* p,   
     EmbRect rect,   
     int thread\_index,   
     int style )

*p rect thread\_index style*

**18.33.1.14 embPolygon\_reduceByDistance()** void embPolygon\_reduceByDistance (   
     EmbArray \* vertices,   
     EmbArray \* simplified,   
     float distance )

*vertices simplified distance*

Reduces the polygon by distance.

This is a non-destructive function, so the caller is responsible for freeing "vertices" if they choose to keep "simplified".

**18.33.1.15 embPolygon\_reduceByNth()** void embPolygon\_reduceByNth (   
     EmbArray \* vertices,   
     EmbArray \* out,   
     int nth )

*vertices out nth*

Reduces the polygon by removing the Nth vertex in the vertices list. This is a non-destructive function, so the caller is responsible for freeing vertices if they choose to keep out.

**18.33.1.16 generate\_dragon\_curve()** void generate\_dragon\_curve (   
     char \* state,   
     int iterations )

*state iterations*

using the "paper folding" method

**Todo** find citation for paper folding method

**18.33.1.17 greedy\_algorithm()** static void greedy\_algorithm (   
     int \* points,   
     int n\_points,   
     int width,   
     EmbReal bias ) [static]

*points n\_points width bias*

**18.33.1.18 Greedy Algorithm** For each point in the list find the shortest distance to any possible neighbour, then perform a swap to make that neighbour the next item in the list.

To make the stitches lie more on one axis than the other bias the distance operator to prefer horizontal direction.

**18.33.1.19 hilbert\_curve()** `int hilbert_curve (`  
`EmbPattern * pattern,`  
`int iterations )`

*pattern iterations*

[https://en.wikipedia.org/wiki/Hilbert\\_curve](https://en.wikipedia.org/wiki/Hilbert_curve)

Using the Lindenmayer System, so we can save work across different functions.

**18.33.1.20 join\_short\_stitches()** `static void join_short_stitches (`  
`int * points,`  
`int * n_points,`  
`int width,`  
`int tolerance ) [static]`

*points n\_points width tolerance*

Remove points that lie in the middle of two short stitches that could be one longer stitch. Repeat until none are found.

**18.33.1.21 lindenmayer\_system()** `int lindenmayer_system (`  
`L_system L,`  
`char * state,`  
`int iterations,`  
`int complete )`

*L state iterations complete*

Returns

int

This is a slow generation algorithm.

**18.33.1.22 save\_points\_to\_pattern()** `static void save_points_to_pattern (`  
`EmbPattern * pattern,`  
`int * points,`  
`int n_points,`  
`EmbReal scale,`  
`int width,`  
`int height ) [static]`

*pattern points n\_points scale width height*

**18.33.1.23 threshold\_method()** `static int * threshold_method (`  
`EmbImage * image,`  
`int * n_points,`  
`int subsample_width,`  
`int subsample_height,`  
`int threshold ) [static]`

*image n\_points subsample\_width subsample\_height threshold*

Returns

int\*

Identify darker pixels to put stitches in.

## 18.33.2 Variable Documentation

**18.33.2.1 hilbert\_curve\_l\_system** `L_system hilbert_curve_l_system`

Initial value:

```
= {
    'A', "AB", "F+-", (char**)rules
}
```

**18.33.2.2 rules** `const char* rules[] = {"+BF-AFA-FB+", "-AF+BFB+FA+"}`

## 18.34 extern/libembroidery/src/formats.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <ctype.h>
#include "embroidery_internal.h"
```

### Functions

- void [safe\\_free](#) (void \*data)  
*data*
- int [embFormat\\_getExtension](#) (const char \*fileName, char \*ending)  
*fileName ending*
- int [emb\\_identify\\_format](#) (const char \*fileName)  
*fileName*
- short [fread\\_int16](#) (FILE \*f)  
*f*
- unsigned short [fread\\_uint16](#) (FILE \*f)  
*f*
- int [fread\\_int32\\_be](#) (FILE \*f)  
*f*
- void [fpad](#) (FILE \*file, char c, int n)  
*f*
- void [binaryWriteShort](#) (FILE \*f, short data)  
*f data*
- void [binaryWriteUShort](#) (FILE \*f, unsigned short data)  
*f data*
- void [binaryWriteUShortBE](#) (FILE \*f, unsigned short data)  
*f data*
- void [binaryWriteInt](#) (FILE \*f, int data)  
*f data*
- void [binaryWriteIntBE](#) (FILE \*f, int data)  
*f data*
- void [binaryWriteUInt](#) (FILE \*f, unsigned int data)  
*f data*
- void [binaryWriteUIntBE](#) (FILE \*f, unsigned int data)  
*f data*
- char [embPattern\\_read](#) ([EmbPattern](#) \*pattern, const char \*fileName, int format)  
*pattern fileName format*
- char [embPattern\\_write](#) ([EmbPattern](#) \*pattern, const char \*fileName, int format)  
*pattern fileName format*
- char [embPattern\\_readAuto](#) ([EmbPattern](#) \*pattern, const char \*fileName)  
*pattern fileName*
- char [embPattern\\_writeAuto](#) ([EmbPattern](#) \*pattern, const char \*fileName)  
*pattern fileName*



## Variables

- [EmbFormatList](#) `formatTable` [`numberOfFormats`]
- `const char` [imageWithFrame](#) [`38`][`48`]

### 18.34.1 Function Documentation

**18.34.1.1 `binaryWriteInt()`** `void binaryWriteInt (`  
    `FILE * f,`  
    `int data )`

*f data*

**Todo** replace with `emblnt_read`

**18.34.1.2 `binaryWriteIntBE()`** `void binaryWriteIntBE (`  
    `FILE * f,`  
    `int data )`

*f data*

**Todo** replace with `emblnt_read`

**18.34.1.3 `binaryWriteShort()`** `void binaryWriteShort (`  
    `FILE * f,`  
    `short data )`

*f data*

**Todo** replace with `emblnt_read`

**18.34.1.4 `binaryWriteUInt()`** `void binaryWriteUInt (`  
    `FILE * f,`  
    `unsigned int data )`

*f data*

**Todo** replace with `emblnt_read`

**18.34.1.5 `binaryWriteUIntBE()`** `void binaryWriteUIntBE (`  
    `FILE * f,`  
    `unsigned int data )`

*f data*

**Todo** replace with `emblnt_read`

**18.34.1.6 `binaryWriteUShort()`** `void binaryWriteUShort (`  
    `FILE * f,`  
    `unsigned short data )`

*f data*

**Todo** replace with `emblnt_read`

**18.34.1.7 binaryWriteUShortBE()** void binaryWriteUShortBE (   
FILE \* *f*,  
unsigned short *data* )  
*f data*

**Todo** replace with embInt\_read

**18.34.1.8 emb\_identify\_format()** int emb\_identify\_format (   
const char \* *fileName* )  
*fileName*

Returns

int

**18.34.1.9 embFormat\_getExtension()** int embFormat\_getExtension (   
const char \* *fileName*,  
char \* *ending* )  
*fileName ending*

Returns

int

**18.34.1.10 embPattern\_read()** char embPattern\_read (   
EmbPattern \* *pattern*,  
const char \* *fileName*,  
int *format* )  
*pattern fileName format*

Returns

char

**18.34.1.11 embPattern\_readAuto()** char embPattern\_readAuto (   
EmbPattern \* *pattern*,  
const char \* *fileName* )  
*pattern fileName*

Returns

char

**18.34.1.12 embPattern\_write()** char embPattern\_write (   
EmbPattern \* *pattern*,  
const char \* *fileName*,  
int *format* )  
*pattern fileName format*

Returns

char

**18.34.1.13 embPattern\_writeAuto()** char embPattern\_writeAuto (  
    EmbPattern \* pattern,  
    const char \* fileName )  
*pattern fileName*

Returns

char

**18.34.1.14 fpad()** void fpad (  
    FILE \* file,  
    char c,  
    int n )  
*f*

Returns

int

**18.34.1.15 fread\_int16()** short fread\_int16 (  
    FILE \* f )  
*f*

Returns

short

**18.34.1.16 fread\_int32\_be()** int fread\_int32\_be (  
    FILE \* f )  
*f*

Returns

int

**Todo** replace with embInt\_read

**18.34.1.17 fread\_uint16()** unsigned short fread\_uint16 (  
    FILE \* f )  
*f*

Returns

unsigned short

**Todo** replace with embInt\_read

**18.34.1.18 safe\_free()** void safe\_free (  
    void \* data )  
*data*

## 18.34.2 Variable Documentation

#### 18.34.2.1 `formatTable` `EmbFormatList` `formatTable[numberOfFormats]`

This file is part of libembroidery.

Copyright 2018-2022 The Embroidermodder Team Licensed under the terms of the zlib license.

This file contains all the read and write functions for the library.

**Todo** This list needs reviewed in case some stitch formats also can contain object data (EMBFORMAT\_↵ STCHANDOBJ). \*

#### 18.34.2.2 `imageWithFrame` `const char imageWithFrame[38][48]`

### 18.35 `extern/libembroidery/src/formats/format_100.c` File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

#### Functions

- char `read100` (`EmbPattern` \*pattern, FILE \*file)
- char `write100` (`EmbPattern` \*pattern, FILE \*file)

#### 18.35.1 Detailed Description

The Toyota Embroidery Format (.10o)

The Toyota 10o format is a stitch-only format that uses an external color file.

The stitch encoding is in 3 byte chunks.

#### 18.35.2 Function Documentation

**18.35.2.1 `read100()`** `char read100 (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.35.2.2 `write100()`** `char write100 (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

### 18.36 `extern/libembroidery/src/formats/format_10o.c` File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

#### Functions

- char `read10o` (`EmbPattern` \*pattern, FILE \*file)
- char `write10o` (`EmbPattern` \*pattern, FILE \*file)

#### 18.36.1 Detailed Description

The Toyota Embroidery Format (.100)

The Toyota 100 format is a stitch-only format that uses an external color file.

The stitch encoding is in 4 byte chunks.

## 18.36.2 Function Documentation

**18.36.2.1 read10o()** char read10o (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.36.2.2 write10o()** char write10o (  
    EmbPattern \* pattern,  
    FILE \* file )

## 18.37 extern/libembroidery/src/formats/format\_art.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char readArt (EmbPattern \*pattern, FILE \*file)
- char writeArt (EmbPattern \*pattern, FILE \*file)

### 18.37.1 Detailed Description

The Bernina Embroidery Format (.art)  
We don't know much about this format.

**Todo** Find a source.

## 18.37.2 Function Documentation

**18.37.2.1 readArt()** char readArt (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.37.2.2 writeArt()** char writeArt (  
    EmbPattern \* pattern,  
    FILE \* file )

## 18.38 extern/libembroidery/src/formats/format\_bmc.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char readBmc (EmbPattern \*pattern, FILE \*file)
- char writeBmc (EmbPattern \*pattern, FILE \*file)

### 18.38.1 Detailed Description

The Bitmap Cache Embroidery Format (.bmc)  
We don't know much about this format.

**Todo** Find a source.

### 18.38.2 Function Documentation

**18.38.2.1 readBmc()** `char readBmc (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.38.2.2 writeBmc()** `char writeBmc (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.39 extern/libembroidery/src/formats/format\_bro.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readBro](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeBro](#) ([EmbPattern](#) \*pattern, FILE \*file)

### 18.39.1 Detailed Description

The Bits and Volts Embroidery Format (.bro)  
The Bits and Volts bro format is a stitch-only format that uses an external color file.  
The header is 256 bytes. There's a series of unknown variables in the header.  
The stitch list uses a variable length encoding which is 2 bytes for any stitch.

### 18.39.2 Function Documentation

**18.39.2.1 readBro()** `char readBro (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.39.2.2 writeBro()** `char writeBro (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.40 extern/libembroidery/src/formats/format\_cnd.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

## Functions

- char [readCnd](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeCnd](#) ([EmbPattern](#) \*pattern, FILE \*file)

### 18.40.1 Detailed Description

The Melco Embroidery Format (.cnd)

The Melco cnd format is a stitch-only format.

We don't know much about this format.

**Todo** Find a source.

### 18.40.2 Function Documentation

**18.40.2.1 readCnd()** char readCnd (  
[EmbPattern](#) \* pattern,  
 FILE \* file )

**18.40.2.2 writeCnd()** char writeCnd (  
[EmbPattern](#) \* pattern,  
 FILE \* file )

## 18.41 extern/libembroidery/src/formats/format\_col.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

## Functions

- char [readCol](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeCol](#) ([EmbPattern](#) \*pattern, FILE \*file)

### 18.41.1 Detailed Description

The Embroidery Thread Color Format (.col)

An external color file format for formats that do not record their own colors.

It is a human-readable format that has a header that is a single line containing only the number of threads in decimal followed by the windows line break `\r\n`.

Then the rest of the file is a comma separated value list of all threads with 4 values per line: the index of the thread then the red, green and blue channels of the color in that order.

**18.41.1.0.1 Example** If we had a pattern called "example" with four colors: black, red, magenta and cyan in that order then the file is (with the white space written out):

```
example.col
4\r
0,0,0,0\r
1,255,0,0\r
2,0,255,0\r
3,0,0,255\r
```

## 18.41.2 Function Documentation

**18.41.2.1 readCol()** `char readCol (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.41.2.2 writeCol()** `char writeCol (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.42 extern/libembroidery/src/formats/format\_csd.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Macros

- `#define CsdSubMaskSize` 479
- `#define CsdXorMaskSize` 501

### Functions

- void `BuildDecryptionTable` (int seed)
- unsigned char `DecodeCsdByte` (long fileOffset, unsigned char val, int type)
- char `readCsd` (`EmbPattern *pattern`, `FILE *file`)
- char `writeCsd` (`EmbPattern *pattern`, `FILE *file`)

### Variables

- char `_subMask` [`CsdSubMaskSize`]
- char `_xorMask` [`CsdXorMaskSize`]
- const unsigned char `csd_decryptArray` []

### 18.42.1 Detailed Description

The Singer Embroidery Format (.csd)  
Stitch Only Format.

### 18.42.2 Macro Definition Documentation

**18.42.2.1 CsdSubMaskSize** `#define CsdSubMaskSize` 479

**18.42.2.2 CsdXorMaskSize** `#define CsdXorMaskSize` 501

### 18.42.3 Function Documentation



**18.42.3.1 BuildDecryptionTable()** void BuildDecryptionTable (  
     int seed )

**18.42.3.2 DecodeCsdByte()** unsigned char DecodeCsdByte (  
     long fileOffset,  
     unsigned char val,  
     int type )

**18.42.3.3 readCsd()** char readCsd (  
     EmbPattern \* pattern,  
     FILE \* file )

**18.42.3.4 writeCsd()** char writeCsd (  
     EmbPattern \* pattern,  
     FILE \* file )

## 18.42.4 Variable Documentation

**18.42.4.1 \_subMask** char \_subMask[CsdSubMaskSize]

**18.42.4.2 \_xorMask** char \_xorMask[CsdXorMaskSize]

**18.42.4.3 csd\_decryptArray** const unsigned char csd\_decryptArray[]

Initial value:

```
= {
    0x43, 0x6E, 0x72, 0x7A, 0x76, 0x6C, 0x61, 0x6F, 0x7C, 0x29, 0x5D, 0x62, 0x60, 0x6E, 0x61, 0x62,
    0x20, 0x41, 0x66, 0x6A, 0x3A, 0x35, 0x5A, 0x63, 0x7C, 0x37, 0x3A, 0x2A, 0x25, 0x24, 0x2A, 0x33,
    0x00, 0x10, 0x14, 0x03, 0x72, 0x4C, 0x48, 0x42, 0x08, 0x7A, 0x5E, 0x0B, 0x6F, 0x45, 0x47, 0x5F,
    0x40, 0x54, 0x5C, 0x57, 0x55, 0x59, 0x53, 0x3A, 0x32, 0x6F, 0x53, 0x54, 0x50, 0x5C, 0x4A, 0x56,
    0x2F, 0x2F, 0x62, 0x2C, 0x22, 0x65, 0x25, 0x28, 0x38, 0x30, 0x38, 0x22, 0x2B, 0x25, 0x3A, 0x6F,
    0x27, 0x38, 0x3E, 0x3F, 0x74, 0x37, 0x33, 0x77, 0x2E, 0x30, 0x3D, 0x34, 0x2E, 0x32, 0x2B, 0x2C,
    0x0C, 0x18, 0x42, 0x13, 0x16, 0x0A, 0x15, 0x02, 0x0B, 0x1C, 0x1E, 0x0E, 0x08, 0x60, 0x64, 0x0D,
    0x09, 0x51, 0x25, 0x1A, 0x18, 0x16, 0x19, 0x1A, 0x58, 0x10, 0x14, 0x5B, 0x08, 0x15, 0x1B, 0x5F,
    0xD5, 0xD2, 0xAE, 0xA3, 0xC1, 0xF0, 0xF4, 0xE8, 0xF8, 0xEC, 0xA6, 0xAB, 0xCD, 0xF8, 0xFD, 0xFB,
    0xE2, 0xF0, 0xFE, 0xFA, 0xF5, 0xB5, 0xF7, 0xF9, 0xFC, 0xB9, 0xF5, 0xEF, 0xF4, 0xF8, 0xEC, 0xBF,
    0xC3, 0xCE, 0xD7, 0xCD, 0xD0, 0xD7, 0xCF, 0xC2, 0xDB, 0xA4, 0xA0, 0xB0, 0xAF, 0xBE, 0x98, 0xE2,
    0xC2, 0x91, 0xE5, 0xDC, 0xDA, 0xD2, 0x96, 0xC4, 0x98, 0xF8, 0xC9, 0xD2, 0xDD, 0xD3, 0x9E, 0xDE,
    0xAE, 0xA5, 0xE2, 0x8C, 0xB6, 0xAC, 0xA3, 0xA9, 0xBC, 0xA8, 0xA6, 0xEB, 0x8B, 0xBF, 0xA1, 0xAC,
    0xB5, 0xA3, 0xBB, 0xB6, 0xA7, 0xD8, 0xDC, 0x9A, 0xAA, 0xF9, 0x82, 0xFB, 0x9D, 0xB9, 0xAB, 0xB3,
    0x94, 0xC1, 0xA0, 0x8C, 0x8B, 0x8E, 0x95, 0x8F, 0x87, 0x99, 0xE7, 0xE1, 0xA3, 0x83, 0x8B, 0xCF,
    0xA3, 0x85, 0x9D, 0x83, 0xD4, 0xB7, 0x83, 0x84, 0x91, 0x97, 0x9F, 0x88, 0x8F, 0xDD, 0xAD, 0x90
}
```

## 18.43 extern/libembroidery/src/formats/format\_csv.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char \* [csvStitchFlagToStr](#) (int flags)
- int [csvStrToStitchFlag](#) (const char \*str)

- char `readCsv` (`EmbPattern` \*pattern, FILE \*file)
- char `writeCsv` (`EmbPattern` \*pattern, FILE \*file)

### 18.43.1 Detailed Description

Comma Separated Values (.csv)

Comma Separated Values files aren't a universal system, here we aim to offer a broad support. The dialect is detected based on the opening lines, as each manufacturer should label their CSV files there.

**18.43.1.0.1 Embroidermodder 2.0 CSV Dialect** Our own version has the identifier comment line:

| Control Symbol | Type | Description | |—| | # | COMMENT | | > | VARIABLE | To store records of a pattern's width, height etc. This means that data stored in the header of say a .dst file is preserved. | | \$ | THREAD | | \* | STITCH | | \* | JUMP | | \* | COLOR | To change a color: used for trim as well | | \* | END | To end a pattern. | | \* | UNKNOWN | For any feature that we can't identify.

### 18.43.1.0.2 EmBird CSV Dialect

## 18.43.2 Function Documentation

**18.43.2.1 csvStitchFlagToStr()** char \* csvStitchFlagToStr (   
int flags )

**18.43.2.2 csvStrToStitchFlag()** int csvStrToStitchFlag (   
const char \* str )

**18.43.2.3 readCsv()** char readCsv (   
EmbPattern \* pattern,   
FILE \* file )

**18.43.2.4 writeCsv()** char writeCsv (   
EmbPattern \* pattern,   
FILE \* file )

## 18.44 extern/libembroidery/src/formats/format\_dat.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char `readDat` (`EmbPattern` \*pattern, FILE \*file)
- char `writeDat` (`EmbPattern` \*pattern, FILE \*file)

### 18.44.1 Function Documentation

**18.44.1.1 readDat()** char readDat (   
EmbPattern \* pattern,   
FILE \* file )

**18.44.1.2 writeDat()** `char writeDat (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.45 extern/libembroidery/src/formats/format\_dem.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readDem](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeDem](#) ([EmbPattern](#) \*pattern, FILE \*file)

#### 18.45.1 Detailed Description

The Melco Embroidery Format (.dem)  
Stitch Only Format

#### 18.45.2 Function Documentation

**18.45.2.1 readDem()** `char readDem (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.45.2.2 writeDem()** `char writeDem (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.46 extern/libembroidery/src/formats/format\_dsb.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readDsb](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeDsb](#) ([EmbPattern](#) \*pattern, FILE \*file)

#### 18.46.1 Detailed Description

The Barudan Embroidery Format (.dsb)

- Stitch Only Format.
- [X] Basic Read Support
- [o] Basic Write Support
- [o] Well Tested Read
- [o] Well Tested Write

## 18.46.2 Function Documentation

**18.46.2.1 readDsb()** `char readDsb (`  
     `EmbPattern * pattern,`  
     `FILE * file )`

**18.46.2.2 writeDsb()** `char writeDsb (`  
     `EmbPattern * pattern,`  
     `FILE * file )`

## 18.47 extern/libembroidery/src/formats/format\_dst.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Macros

- `#define cci(c1, c2) (c1*256+c2)`

### Functions

- `int decode_record_flags` (unsigned char b2)
- `void encode_record` (FILE \*file, int x, int y, int flags)
- `void set_dst_variable` (EmbPattern \*pattern, char \*var, char \*val)
- `char readDst` (EmbPattern \*pattern, FILE \*file)
- `char writeDst` (EmbPattern \*pattern, FILE \*file)

### 18.47.1 Detailed Description

.DST (Tajima) embroidery file read/write routines Format comments are thanks to [tspilman@dalcoathletic.com](mailto:tspilman@dalcoathletic.com) who's notes appeared at <http://www.wotsit.org> under Tajima Format.

**18.47.1.1 Tajima Embroidery Format (.dst)** Stitch Only Format. [X] Basic Read Support [X] Basic Write Support [ ] Well Tested Read [ ] Well Tested Write

.DST (Tajima) embroidery file read/write routines Format comments are thanks to [tspilman@dalcoathletic.com](mailto:tspilman@dalcoathletic.com) who's notes appeared at <http://www.wotsit.org> under Tajima Format.

Other references: [2], [1].

**18.47.1.1.1 Header** The header contains general information about the design. It is in lines of ASCII, so if you open a DST file as a text file, it's the only part that's easy to read. The line ending symbol is ``0x0D`. The header is necessary for the file to be read by most softwares and hardwares.

The header is 125 bytes of data followed by padding spaces to make it 512 bytes in total.

The lines are as follows.

Label	Size	Description	Example
LA:	17	The design name with no path or extension. The space reserved is 16 characters, but the name must not be longer than 8 and be padded to 16 with spaces (0x20).	"LA:Star "
ST:	8	The stitch count. An integer in the format <code>%07d</code> , that is: a 7 digit number padded by leading zeros. This is the total accross all possible stitch flags.	

Label	Size	Description	Example
CO:	4	The number of color changes (not to be confused with thread count, an all black design we would have the record <code>\textbf{000}</code> ). An integer in the format <code>%03d</code> , 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 <code>%05d</code> , 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 <code>%05d</code> , 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 <code>%05d</code> , 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 <code>%05d</code> , 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 <code>%05d</code> , 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 <code>%05d</code> , that is: a 5 digit integer padded by leading zeros.	
MX:	7	The x co-ordinate of the last point in the previous file should the design span multiple files. Like AX, it is the sign, followed by a 5 digit integer. If we have a one file design set it to zero.	
MY:	7	The y co-ordinate of the last point in the previous file should the design span multiple files. Like AY, it is the sign, followed by a 5 digit integer. If we have a one file design set it to zero.	
PD:	10	Information about multivolume designs.	

**18.47.1.1.2 Stitch Data** Uses 3 byte per stitch encoding with the format as follows:

Bit	7	6	5	4	3	2	1	0
Byte 0	y+1	y-1	y+9	y-9	x-9	x+9	x-1	x+1
Byte 1	y+3	y-3	y+27	y-27	x-27	x+27	x-3	x+3
Byte 2	jump	color change	y+81	y-81	x-81	x+81	set	set

T01 and Tap appear to use Tajima Ternary.

Where the stitch type is determined as:

Normal Stitch `0b00000011 0x03` Jump Stitch `0b10000011 0x83` Stop/Change Color `0b11000011 0x↵`  
C3 End Design `0b11110011 0xF3`

Inclusive or'ed with the last byte.

Note that the max stitch length is the largest sum of  $1+3+9+27+81=121$  where the unit length is 0.1mm so 12.↵  
1mm. The coordinate system is right handed.

## 18.47.2 Macro Definition Documentation

**18.47.2.1 cci** `#define cci(  
c1,  
c2 ) (c1*256+c2)`

## 18.47.3 Function Documentation

**18.47.3.1 decode\_record\_flags()** `int decode_record_flags (  
unsigned char b2 )`

**18.47.3.2 encode\_record()** void encode\_record (

```
FILE * file,
int x,
int y,
int flags )
```

**18.47.3.3 readDst()** char readDst (

```
EmbPattern * pattern,
FILE * file )
```

**18.47.3.4 set\_dst\_variable()** void set\_dst\_variable (

```
EmbPattern * pattern,
char * var,
char * val )
```

**18.47.3.5 writeDst()** char writeDst (

```
EmbPattern * pattern,
FILE * file )
```

## 18.48 extern/libembroidery/src/formats/format\_dsz.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readDsz](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeDsz](#) ([EmbPattern](#) \*pattern, FILE \*file)

### 18.48.1 Function Documentation

**18.48.1.1 readDsz()** char readDsz (

```
EmbPattern * pattern,
FILE * file )
```

**18.48.1.2 ZSK USA Embroidery Format (.dsz)** The ZSK USA dsz format is stitch-only.

**18.48.1.3 writeDsz()** char writeDsz (

```
EmbPattern * pattern,
FILE * file )
```

## 18.49 extern/libembroidery/src/formats/format\_dxf.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
```

```
#include "../embroidery_internal.h"
```

## Functions

- void [readLine](#) (FILE \*file, char \*str)
- char [readDxf](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeDxf](#) ([EmbPattern](#) \*pattern, FILE \*file)

### 18.49.1 Function Documentation

**18.49.1.1 readDxf()** char readDxf (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.49.1.2 readLine()** void readLine (  
    FILE \* file,  
    char \* str )

**18.49.1.3 Drawing Exchange Format (.dxf)** Graphics format for drawing files designed and used by AutoDesk for their AutoCAD program. [[dxf\\_reference](#)]

**18.49.1.4 writeDxf()** char writeDxf (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.50 extern/libembroidery/src/formats/format\_edr.c File Reference

```
#include <stdio.h>  
#include <math.h>  
#include "../embroidery_internal.h"
```

## Functions

- char [readEdr](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeEdr](#) ([EmbPattern](#) \*pattern, FILE \*file)

### 18.50.1 Function Documentation

**18.50.1.1 readEdr()** char readEdr (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.50.1.2 Embird Embroidery Format (.edr)** Stitch Only Format

**18.50.1.3 writeEdr()** char writeEdr (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.51 extern/libembroidery/src/formats/format\_emd.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [emdDecode](#) (unsigned char inputByte)
- char [readEmd](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeEmd](#) ([EmbPattern](#) \*pattern, FILE \*file)

#### 18.51.1 Detailed Description

The Elna Embroidery Format (.emd)  
Stitch Only Format.

#### 18.51.2 Function Documentation

**18.51.2.1 [emdDecode\(\)](#)** char emdDecode (  
    unsigned char *inputByte* )

**18.51.2.2 [readEmd\(\)](#)** char readEmd (  
    [EmbPattern](#) \* *pattern*,  
    FILE \* *file* )

**18.51.2.3 [writeEmd\(\)](#)** char writeEmd (  
    [EmbPattern](#) \* *pattern*,  
    FILE \* *file* )

## 18.52 extern/libembroidery/src/formats/format\_exp.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [expDecode](#) (unsigned char a1)
- char [readExp](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeExp](#) ([EmbPattern](#) \*pattern, FILE \*file)

#### 18.52.1 Function Documentation

**18.52.1.1 [expDecode\(\)](#)** char expDecode (  
    unsigned char *a1* )

**18.52.1.2 Melco Embroidery Format (.exp)** Stitch Only Format.



**18.52.1.3 readExp()** char readExp (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.52.1.4 writeExp()** char writeExp (  
    EmbPattern \* pattern,  
    FILE \* file )

## 18.53 extern/libembroidery/src/formats/format\_exy.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- int [decode\\_exy\\_flags](#) (unsigned char b2)
- char [readExy](#) (EmbPattern \*pattern, FILE \*file)
- char [writeExy](#) (EmbPattern \*pattern, FILE \*file)

#### 18.53.1 Function Documentation

**18.53.1.1 decode\_exy\_flags()** int decode\_exy\_flags (  
    unsigned char b2 )

**18.53.1.2 Eltac Embroidery Format (.exy)** Stitch Only Format.

**18.53.1.3 readExy()** char readExy (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.53.1.4 writeExy()** char writeExy (  
    EmbPattern \* pattern,  
    FILE \* file )

## 18.54 extern/libembroidery/src/formats/format\_eyc.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readEys](#) (EmbPattern \*pattern, FILE \*file)
- char [writeEys](#) (EmbPattern \*pattern, FILE \*file)

#### 18.54.1 Function Documentation

**18.54.1.1 readEys()** `char readEys (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.54.1.2 Sierra Expanded Embroidery Format (.eys)** Stitch Only Format.  
Smoothie G-Code Embroidery Format (.fxy)?

**18.54.1.3 writeEys()** `char writeEys (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.55 extern/libembroidery/src/formats/format\_fxy.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readFxy](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeFxy](#) ([EmbPattern](#) \*pattern, FILE \*file)

#### 18.55.1 Function Documentation

**18.55.1.1 readFxy()** `char readFxy (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.55.1.2 Embroidery Format (.fxy)** Stitch Only Format.

**18.55.1.3 writeFxy()** `char writeFxy (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.56 extern/libembroidery/src/formats/format\_gc.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readGc](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeGc](#) ([EmbPattern](#) \*pattern, FILE \*file)

#### 18.56.1 Function Documentation

**18.56.1.1 readGc()** `char readGc (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

Smoothie G-Code

Main Reference: Machinery's Handbook Guide A Guide to Tables, Formulas, & More in the 31st Edition by John Milton Amiss, Franklin D. Jones and Henry Ryffel

**18.56.1.2 writeGc()** `char writeGc (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.57 extern/libembroidery/src/formats/format\_gnc.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char `readGnc` (`EmbPattern *pattern`, `FILE *file`)
- char `writeGnc` (`EmbPattern *pattern`, `FILE *file`)

#### 18.57.1 Function Documentation

**18.57.1.1 readGnc()** `char readGnc (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.57.1.2 Great Notions Embroidery Format (.gnc)** Stitch Only Format.

**18.57.1.3 writeGnc()** `char writeGnc (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.58 extern/libembroidery/src/formats/format\_gt.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char `readGt` (`EmbPattern *pattern`, `FILE *file`)
- char `writeGt` (`EmbPattern *pattern`, `FILE *file`)

#### 18.58.1 Function Documentation

**18.58.1.1 readGt()** `char readGt (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.58.1.2 Gold Thread Embroidery Format (.gt) Stitch Only Format.**

**18.58.1.3 writeGt()** `char writeGt (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.59 extern/libembroidery/src/formats/format\_hus.c File Reference**

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery_internal.h"
```

**Functions**

- int [husDecodeStitchType](#) (unsigned char b)
- unsigned char \* [husDecompressData](#) (unsigned char \*input, int compressedInputLength, int decompressedContentLength)
- unsigned char \* [husCompressData](#) (unsigned char \*input, int decompressedInputSize, int \*compressedSize)
- int [husDecodeByte](#) (unsigned char b)
- unsigned char [husEncodeByte](#) (EmbReal f)
- unsigned char [husEncodeStitchType](#) (int st)
- char [readHus](#) (EmbPattern \*pattern, FILE \*file)
- char [writeHus](#) (EmbPattern \*pattern, FILE \*file)

**18.59.1 Function Documentation**

**18.59.1.1 husCompressData()** `unsigned char * husCompressData (`  
    `unsigned char * input,`  
    `int decompressedInputSize,`  
    `int * compressedSize )`

**18.59.1.2 husDecodeByte()** `int husDecodeByte (`  
    `unsigned char b )`

**18.59.1.3 husDecodeStitchType()** `int husDecodeStitchType (`  
    `unsigned char b )`

**18.59.1.4 Husqvarna Viking Embroidery Format (.hus) Stitch Only Format.**

**18.59.1.5 husDecompressData()** `unsigned char * husDecompressData (`  
    `unsigned char * input,`  
    `int compressedInputLength,`  
    `int decompressedContentLength )`

**18.59.1.6 husEncodeByte()** `unsigned char husEncodeByte (`  
    `EmbReal f )`

**18.59.1.7 husEncodeStitchType()** unsigned char husEncodeStitchType (  
int *st* )

**18.59.1.8 readHus()** char readHus (  
EmbPattern \* *pattern*,  
FILE \* *file* )

**18.59.1.9 writeHus()** char writeHus (  
EmbPattern \* *pattern*,  
FILE \* *file* )

## 18.60 extern/libembroidery/src/formats/format\_inb.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char readInb (EmbPattern \*pattern, FILE \*file)
- char writeInb (EmbPattern \*pattern, FILE \*file)

#### 18.60.1 Function Documentation

**18.60.1.1 readInb()** char readInb (  
EmbPattern \* *pattern*,  
FILE \* *file* )

**18.60.1.2 Inbro Embroidery Format (.inb)** Stitch Only Format.

**18.60.1.3 writeInb()** char writeInb (  
EmbPattern \* *pattern*,  
FILE \* *file* )

## 18.61 extern/libembroidery/src/formats/format\_inf.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char readInf (EmbPattern \*pattern, FILE \*file)
- char writeInf (EmbPattern \*pattern, FILE \*file)

#### 18.61.1 Function Documentation

**18.61.1.1 readInf()** `char readInf (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.61.1.2 Embroidery Color Format (.inf)** Stitch Only Format.

**18.61.1.3 writeInf()** `char writeInf (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.62 extern/libembroidery/src/formats/format\_jef.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Classes

- struct [hoop\\_padding](#)

### Functions

- int [jefGetHoopSize](#) (int width, int height)
- char [jefDecode](#) (unsigned char inputByte)
- void [jefSetHoopFromId](#) ([EmbPattern](#) \*pattern, int hoopCode)
- void [read\\_hoop](#) (FILE \*file, struct [hoop\\_padding](#) \*hoop, char \*label)
- char [readJef](#) ([EmbPattern](#) \*pattern, FILE \*file)
- void [jefEncode](#) (unsigned char \*b, char dx, char dy, int flags)
- char [writeJef](#) ([EmbPattern](#) \*pattern, FILE \*file)

### 18.62.1 Function Documentation

**18.62.1.1 jefDecode()** `char jefDecode (`  
    unsigned char *inputByte* )

**18.62.1.2 jefEncode()** `void jefEncode (`  
    unsigned char \* *b*,  
    char *dx*,  
    char *dy*,  
    int *flags* )

**18.62.1.3 jefGetHoopSize()** `int jefGetHoopSize (`  
    int *width*,  
    int *height* )

**18.62.1.4 Janome Embroidery Format (.jef)** Stitch Only Format.

**18.62.1.5 jefSetHoopFromId()** `void jefSetHoopFromId (`  
    [EmbPattern](#) \* *pattern*,  
    int *hoopCode* )

```
18.62.1.6 read_hoop() void read_hoop (
    FILE * file,
    struct hoop_padding * hoop,
    char * label )
```

```
18.62.1.7 readJef() char readJef (
    EmbPattern * pattern,
    FILE * file )
```

```
18.62.1.8 writeJef() char writeJef (
    EmbPattern * pattern,
    FILE * file )
```

## 18.63 extern/libembroidery/src/formats/format\_ksm.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- void [ksmEncode](#) (unsigned char \**b*, char *dx*, char *dy*, int *flags*)
- char [readKsm](#) (EmbPattern \**pattern*, FILE \**file*)
- char [writeKsm](#) (EmbPattern \**pattern*, FILE \**file*)

### 18.63.1 Function Documentation

```
18.63.1.1 ksmEncode() void ksmEncode (
    unsigned char * b,
    char dx,
    char dy,
    int flags )
```

**18.63.1.2 Pfaff professional Design format (.ksm)** Stitch Only Format.

```
18.63.1.3 readKsm() char readKsm (
    EmbPattern * pattern,
    FILE * file )
```

```
18.63.1.4 writeKsm() char writeKsm (
    EmbPattern * pattern,
    FILE * file )
```

## 18.64 extern/libembroidery/src/formats/format\_max.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

## Functions

- char [readMax](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeMax](#) ([EmbPattern](#) \*pattern, FILE \*file)

## Variables

- const unsigned char [max\\_header](#) []

### 18.64.1 Function Documentation

**18.64.1.1 readMax()** char readMax (  
[EmbPattern](#) \* pattern,  
 FILE \* file )

**18.64.1.2 writeMax()** char writeMax (  
[EmbPattern](#) \* pattern,  
 FILE \* file )

### 18.64.2 Variable Documentation

**18.64.2.1 max\_header** const unsigned char max\_header[]

Initial value:

```
= {
    0x56, 0x43, 0x53, 0x4D, 0xFC, 0x03, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00,
    0xF6, 0x25, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 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, 0x31, 0x33,
    0x37, 0x38, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 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
}
```

**18.64.2.2 Pfaff Embroidery Format (.max)** Stitch Only Format.

## 18.65 extern/libembroidery/src/formats/format\_mit.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

## Functions

- char [readMit](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeMit](#) ([EmbPattern](#) \*pattern, FILE \*file)

### 18.65.1 Function Documentation



**18.65.1.1 readMit()** `char readMit (`  
     `EmbPattern * pattern,`  
     `FILE * file )`

**18.65.1.2 Mitsubishi Embroidery Format (.mit)** Stitch Only Format.

**18.65.1.3 writeMit()** `char writeMit (`  
     `EmbPattern * pattern,`  
     `FILE * file )`

## 18.66 extern/libembroidery/src/formats/format\_new.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readNew](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeNew](#) ([EmbPattern](#) \*pattern, FILE \*file)

### 18.66.1 Function Documentation

**18.66.1.1 readNew()** `char readNew (`  
     `EmbPattern * pattern,`  
     `FILE * file )`

**18.66.1.2 Ameco Embroidery Format (.new)** Stitch Only Format.

**18.66.1.3 writeNew()** `char writeNew (`  
     `EmbPattern * pattern,`  
     `FILE * file )`

## 18.67 extern/libembroidery/src/formats/format\_ofm.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char \* [ofmReadLibrary](#) (FILE \*file)
- static int [ofmReadClass](#) (FILE \*file)
- void [ofmReadBlockHeader](#) (FILE \*file)
- void [ofmReadColorChange](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [ofmReadThreads](#) (FILE \*file, [EmbPattern](#) \*p)
- [EmbReal](#) [ofmDecode](#) (unsigned char b1, unsigned char b2)
- void [ofmReadExpanded](#) (FILE \*file, [EmbPattern](#) \*p)
- char [readOfm](#) ([EmbPattern](#) \*pattern, FILE \*fileCompound)
- char [writeOfm](#) ([EmbPattern](#) \*pattern, FILE \*file)

### 18.67.1 Function Documentation

**18.67.1.1 ofmDecode()** `EmbReal ofmDecode (`  
    `unsigned char b1,`  
    `unsigned char b2 )`

**18.67.1.2 ofmReadBlockHeader()** `void ofmReadBlockHeader (`  
    `FILE * file )`

**18.67.1.3 ofmReadClass()** `static int ofmReadClass (`  
    `FILE * file ) [static]`

**18.67.1.4 ofmReadColorChange()** `void ofmReadColorChange (`  
    `FILE * file,`  
    `EmbPattern * pattern )`

**18.67.1.5 ofmReadExpanded()** `void ofmReadExpanded (`  
    `FILE * file,`  
    `EmbPattern * p )`

**18.67.1.6 ofmReadLibrary()** `char * ofmReadLibrary (`  
    `FILE * file )`

**18.67.1.7 Melco Embroidery Format (.ofm)** Stitch Only Format.

**18.67.1.8 ofmReadThreads()** `void ofmReadThreads (`  
    `FILE * file,`  
    `EmbPattern * p )`

**18.67.1.9 readOfm()** `char readOfm (`  
    `EmbPattern * pattern,`  
    `FILE * fileCompound )`

**18.67.1.10 writeOfm()** `char writeOfm (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

### 18.68 extern/libembroidery/src/formats/format\_pcd.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

## Functions

- char [readPcd](#) ([EmbPattern](#) \*pattern, const char \*fileName, FILE \*file)
- char [writePcd](#) ([EmbPattern](#) \*pattern, FILE \*file)

### 18.68.1 Function Documentation

**18.68.1.1 readPcd()** char readPcd (  
[EmbPattern](#) \* pattern,  
 const char \* fileName,  
 FILE \* file )

#### 18.68.1.2 Pfaff PCD File Format (.pcd) Stitch Only Format.

The format uses a signed 3 byte-length number type.

See the description here ([5](5)) for the overview of the format.

For an example of the format see ([11](11)).

**18.68.1.3 writePcd()** char writePcd (  
[EmbPattern](#) \* pattern,  
 FILE \* file )

## 18.69 extern/libembroidery/src/formats/format\_pcm.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

## Functions

- char [readPcm](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writePcm](#) ([EmbPattern](#) \*pattern, FILE \*file)

### 18.69.1 Function Documentation

**18.69.1.1 readPcm()** char readPcm (  
[EmbPattern](#) \* pattern,  
 FILE \* file )

#### 18.69.1.2 Pfaff Embroidery Format (.pcm) The Pfaff pcm format is stitch-only.

**18.69.1.3 writePcm()** char writePcm (  
[EmbPattern](#) \* pattern,  
 FILE \* file )

## 18.70 extern/libembroidery/src/formats/format\_pcq.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

## Functions

- char [readPcq](#) ([EmbPattern](#) \*pattern, const char \*fileName, FILE \*file)
- char [writePcq](#) ([EmbPattern](#) \*pattern, FILE \*file)

### 18.70.1 Function Documentation

**18.70.1.1 readPcq()** char readPcq (  
    [EmbPattern](#) \* pattern,  
    const char \* fileName,  
    FILE \* file )

**18.70.1.2 Embroidery Format (.pcq)** The Pfaff pcq format is stitch-only.

**18.70.1.3 writePcq()** char writePcq (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.71 extern/libembroidery/src/formats/format\_pcs.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

## Functions

- char [readPcs](#) ([EmbPattern](#) \*pattern, const char \*fileName, FILE \*file)
- char [writePcs](#) ([EmbPattern](#) \*pattern, FILE \*file)

### 18.71.1 Function Documentation

**18.71.1.1 readPcs()** char readPcs (  
    [EmbPattern](#) \* pattern,  
    const char \* fileName,  
    FILE \* file )

**18.71.1.2 Embroidery Format (.pcq)** The Pfaff pcs format is stitch-only.

**18.71.1.3 writePcs()** char writePcs (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.72 extern/libembroidery/src/formats/format\_pec.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

## Functions

- void [readPecStitches](#) ([EmbPattern](#) \*pattern, FILE \*file)
- void [pecEncodeJump](#) (FILE \*file, int x, int types)
- void [pecEncodeStop](#) (FILE \*file, unsigned char val)
- char [readPec](#) ([EmbPattern](#) \*pattern, const char \*fileName, FILE \*file)
- void [pecEncode](#) (FILE \*file, [EmbPattern](#) \*p)
- void [writeImage](#) (FILE \*file, unsigned char image[ ][48])
- void [writePecStitches](#) ([EmbPattern](#) \*pattern, FILE \*file, const char \*fileName)
- char [writePec](#) ([EmbPattern](#) \*pattern, const char \*fileName, FILE \*file)

### 18.72.1 Function Documentation

**18.72.1.1 [pecEncode\(\)](#)** void [pecEncode](#) (  
FILE \* *file*,  
[EmbPattern](#) \* *p* )

**18.72.1.2 [pecEncodeJump\(\)](#)** void [pecEncodeJump](#) (  
FILE \* *file*,  
int *x*,  
int *types* )

**18.72.1.3 [pecEncodeStop\(\)](#)** void [pecEncodeStop](#) (  
FILE \* *file*,  
unsigned char *val* )

**18.72.1.4 [readPec\(\)](#)** char [readPec](#) (  
[EmbPattern](#) \* *pattern*,  
const char \* *fileName*,  
FILE \* *file* )

**18.72.1.5 [readPecStitches\(\)](#)** void [readPecStitches](#) (  
[EmbPattern](#) \* *pattern*,  
FILE \* *file* )

**18.72.1.6 [Embroidery Format \(.pec\)](#)** The Brother pec format is stitch-only.

**18.72.1.7 [writeImage\(\)](#)** void [writeImage](#) (  
FILE \* *file*,  
unsigned char *image*[ ][48] )

Write a PES embedded *image* to the given *file* pointer.

**18.72.1.8 [writePec\(\)](#)** char [writePec](#) (  
[EmbPattern](#) \* *pattern*,  
const char \* *fileName*,  
FILE \* *file* )

**18.72.1.9 writePecStitches()** void writePecStitches (  
    EmbPattern \* pattern,  
    FILE \* file,  
    const char \* fileName )

## 18.73 extern/libembroidery/src/formats/format\_pel.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char readPel (EmbPattern \*pattern, FILE \*file)
- char writePel (EmbPattern \*pattern, FILE \*file)

#### 18.73.1 Function Documentation

**18.73.1.1 readPel()** char readPel (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.73.1.2 Embroidery Format (.pec)** The Brother pel format is stitch-only.

**18.73.1.3 writePel()** char writePel (  
    EmbPattern \* pattern,  
    FILE \* file )

## 18.74 extern/libembroidery/src/formats/format\_pem.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char readPem (EmbPattern \*pattern, FILE \*file)
- char writePem (EmbPattern \*pattern, FILE \*file)

#### 18.74.1 Function Documentation

**18.74.1.1 readPem()** char readPem (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.74.1.2 Embroidery Format (.pec)** The Brother pem format is stitch-only.

**18.74.1.3 writePem()** char writePem (  
    EmbPattern \* pattern,  
    FILE \* file )

## 18.75 extern/libembroidery/src/formats/format\_pes.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readPes](#) ([EmbPattern](#) \*pattern, const char \*fileName, FILE \*file)
- void [readDescriptions](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readPESHeaderV5](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readPESHeaderV6](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readPESHeaderV7](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readPESHeaderV8](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readPESHeaderV9](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readPESHeaderV10](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readHoopName](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readImageString](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readProgrammableFills](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readMotifPatterns](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readFeatherPatterns](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [readThreads](#) (FILE \*file, [EmbPattern](#) \*pattern)
- void [pesWriteSewSegSection](#) ([EmbPattern](#) \*pattern, FILE \*file)
- void [pesWriteEmbOneSection](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writePes](#) ([EmbPattern](#) \*pattern, const char \*fileName, FILE \*file)

### Variables

- const char \* [pes\\_version\\_strings](#) []
- int [pes\\_version](#) = PES0001

### 18.75.1 Function Documentation

**18.75.1.1 [pesWriteEmbOneSection\(\)](#)** void [pesWriteEmbOneSection](#) (  
[EmbPattern](#) \* pattern,  
FILE \* file )

**18.75.1.2 [pesWriteSewSegSection\(\)](#)** void [pesWriteSewSegSection](#) (  
[EmbPattern](#) \* pattern,  
FILE \* file )

**18.75.1.3 [readDescriptions\(\)](#)** void [readDescriptions](#) (  
FILE \* file,  
[EmbPattern](#) \* pattern )

**18.75.1.4 [readFeatherPatterns\(\)](#)** void [readFeatherPatterns](#) (  
FILE \* file,  
[EmbPattern](#) \* pattern )

**18.75.1.5 readHoopName()** void readHoopName (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.75.1.6 readImageString()** void readImageString (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.75.1.7 readMotifPatterns()** void readMotifPatterns (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.75.1.8 readPes()** char readPes (  
EmbPattern \* *pattern*,  
const char \* *fileName*,  
FILE \* *file* )

**18.75.1.9 readPESHeaderV10()** void readPESHeaderV10 (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.75.1.10 readPESHeaderV5()** void readPESHeaderV5 (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.75.1.11 readPESHeaderV6()** void readPESHeaderV6 (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.75.1.12 readPESHeaderV7()** void readPESHeaderV7 (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.75.1.13 readPESHeaderV8()** void readPESHeaderV8 (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.75.1.14 readPESHeaderV9()** void readPESHeaderV9 (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.75.1.15 readProgrammableFills()** void readProgrammableFills (  
FILE \* *file*,  
EmbPattern \* *pattern* )



**18.75.1.16 readThreads()** void readThreads (  
FILE \* *file*,  
EmbPattern \* *pattern* )

**18.75.1.17 writePes()** char writePes (  
EmbPattern \* *pattern*,  
const char \* *fileName*,  
FILE \* *file* )

## 18.75.2 Variable Documentation

**18.75.2.1 pes\_version** int pes\_version = PES0001

**18.75.2.2 pes\_version\_strings** const char\* pes\_version\_strings[]  
Initial value:

```
= {
    "#PES0001",
    "#PES0020",
    "#PES0022",
    "#PES0030",
    "#PES0040",
    "#PES0050",
    "#PES0055",
    "#PES0056",
    "#PES0060",
    "#PES0070",
    "#PES0080",
    "#PES0090",
    "#PES0100",
}
```

**18.75.2.3 Embroidery Format (.pec)** The Brother pes format is stitch-only.

## 18.76 extern/libembroidery/src/formats/format\_phb.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char readPhb (EmbPattern \*pattern, FILE \*file)
- char writePhb (EmbPattern \*pattern, FILE \*file)

### 18.76.1 Function Documentation

**18.76.1.1 readPhb()** char readPhb (  
EmbPattern \* *pattern*,  
FILE \* *file* )

**18.76.1.2 Embroidery Format (.pec)** The Brother phb format is stitch-only.

**18.76.1.3 writePhb()** char writePhb (  
EmbPattern \* *pattern*,  
FILE \* *file* )

## 18.77 extern/libembroidery/src/formats/format\_phc.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readPhc](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writePhc](#) ([EmbPattern](#) \*pattern, FILE \*file)

#### 18.77.1 Function Documentation

**18.77.1.1 readPhc()** char readPhc (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.77.1.2 Embroidery Format (.pec)** The Brother phc format is stitch-only.

**18.77.1.3 writePhc()** char writePhc (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.78 extern/libembroidery/src/formats/format\_plt.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readPlt](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writePlt](#) ([EmbPattern](#) \*pattern, FILE \*file)

#### 18.78.1 Function Documentation

**18.78.1.1 readPlt()** char readPlt (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.78.1.2 Embroidery Format (.plt)** The AutoCAD plt format is stitch-only.

**18.78.1.3 writePlt()** char writePlt (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.79 extern/libembroidery/src/formats/format\_rgb.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readRgb](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeRgb](#) ([EmbPattern](#) \*pattern, FILE \*file)

#### 18.79.1 Function Documentation

**18.79.1.1 readRgb()** char readRgb (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.79.1.2 Color File (.rgb)** The RGB format is a color-only format to act as an external color file for other formats.

**18.79.1.3 writeRgb()** char writeRgb (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.80 extern/libembroidery/src/formats/format\_sew.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [sewDecode](#) (unsigned char inputByte)
- char [readSew](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeSew](#) ([EmbPattern](#) \*pattern, FILE \*file)

#### 18.80.1 Function Documentation

**18.80.1.1 readSew()** char readSew (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.80.1.2 sewDecode()** char sewDecode (  
    unsigned char inputByte )

**18.80.1.3 Embroidery Format (.sew)** The Janome sew format is stitch-only.

**18.80.1.4 writeSew()** char writeSew (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.81 extern/libembroidery/src/formats/format\_shv.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [shvDecode](#) (unsigned char inputByte)
- short [shvDecodeShort](#) (unsigned short inputByte)
- char [readShv](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeShv](#) ([EmbPattern](#) \*pattern, FILE \*file)

### 18.81.1 Function Documentation

**18.81.1.1 readShv()** char readShv (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.81.1.2 shvDecode()** char shvDecode (  
    unsigned char inputByte )

**18.81.1.3 Viking Embroidery Format (.shv)** The Husqvarna Viking shv format is stitch-only.

**18.81.1.4 shvDecodeShort()** short shvDecodeShort (  
    unsigned short inputByte )

**18.81.1.5 writeShv()** char writeShv (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.82 extern/libembroidery/src/formats/format\_sst.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readSst](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeSst](#) ([EmbPattern](#) \*pattern, FILE \*file)

### 18.82.1 Function Documentation

**18.82.1.1 readSst()** char readSst (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.82.1.2 Embroidery Format (.sst)** The Sunstar sst format is stitch-only.

**18.82.1.3 writeSst()** char writeSst (  
    EmbPattern \* pattern,  
    FILE \* file )

## 18.83 extern/libembroidery/src/formats/format\_stx.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- int [stxReadThread](#) (StxThread \*thread, FILE \*file)
- char [readStx](#) (EmbPattern \*pattern, FILE \*file)
- char [writeStx](#) (EmbPattern \*pattern, FILE \*file)

#### 18.83.1 Function Documentation

**18.83.1.1 readStx()** char readStx (  
    EmbPattern \* pattern,  
    FILE \* file )

**18.83.1.2 stxReadThread()** int stxReadThread (  
    StxThread \* thread,  
    FILE \* file )

**18.83.1.3 Stitch Embroidery Format (.stx)** The Data Stitch stx format is stitch-only.

**18.83.1.4 writeStx()** char writeStx (  
    EmbPattern \* pattern,  
    FILE \* file )

## 18.84 extern/libembroidery/src/formats/format\_svg.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readSvg](#) (EmbPattern \*pattern, FILE \*file)
- char [writeSvg](#) (EmbPattern \*pattern, FILE \*file)

## Variables

- int `svgCreator`
- int `svgExpect`
- int `svgMultiValue`
- int `current_element_id`
- `SvgAttribute` `attributeList` [1000]
- int `n_attributes` = 0
- char `currentAttribute` [1000]
- char `currentValue` [1000]

## 18.84.1 Function Documentation

**18.84.1.1 `readSvg()`** `char readSvg (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.84.1.2 `writeSvg()`** `char writeSvg (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

Writes the data from *pattern* to a file with the given *fileName*. Returns `true` if successful, otherwise returns `false`.

## 18.84.2 Variable Documentation

**18.84.2.1 `attributeList`** `SvgAttribute attributeList`[1000]

**18.84.2.2 `current_element_id`** `int current_element_id`

**18.84.2.3 `currentAttribute`** `char currentAttribute`[1000]

**18.84.2.4 `currentValue`** `char currentValue`[1000]

**18.84.2.5 `n_attributes`** `int n_attributes` = 0

**18.84.2.6 `svgCreator`** `int svgCreator`

**18.84.2.7 `Vector Graphics (.svg)`** The scalable vector graphics (SVG) format is a graphics format maintained by ...

**18.84.2.8 `svgExpect`** `int svgExpect`

**18.84.2.9 `svgMultiValue`** `int svgMultiValue`

## 18.85 extern/libembroidery/src/formats/format\_t01.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readT01](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeT01](#) ([EmbPattern](#) \*pattern, FILE \*file)

#### 18.85.1 Function Documentation

**18.85.1.1 readT01()** char readT01 (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.85.1.2 Embroidery Format (.pcq)** The Pfaff t01 format is stitch-only.

**18.85.1.3 writeT01()** char writeT01 (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.86 extern/libembroidery/src/formats/format\_t09.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readT09](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeT09](#) ([EmbPattern](#) \*pattern, FILE \*file)

#### 18.86.1 Function Documentation

**18.86.1.1 readT09()** char readT09 (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

**18.86.1.1.1 Embroidery Format (.pcq)** The Pfaff t09 format is stitch-only.

**18.86.1.2 writeT09()** char writeT09 (  
    [EmbPattern](#) \* pattern,  
    FILE \* file )

## 18.87 extern/libembroidery/src/formats/format\_tap.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../embroidery_internal.h"
```

### Functions

- void [encode\\_tap\\_record](#) (FILE \*file, int x, int y, int flags)
- int [decode\\_tap\\_record\\_flags](#) (unsigned char b2)
- char [readTap](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeTap](#) ([EmbPattern](#) \*pattern, FILE \*file)

### 18.87.1 Function Documentation

**18.87.1.1 [decode\\_tap\\_record\\_flags\(\)](#)** int [decode\\_tap\\_record\\_flags](#) (  
    unsigned char *b2* )

**18.87.1.2 [encode\\_tap\\_record\(\)](#)** void [encode\\_tap\\_record](#) (  
    FILE \* *file*,  
    int *x*,  
    int *y*,  
    int *flags* )

**18.87.1.3 Embroidery Format (.tap)** The Happy tap format is stitch-only.

**18.87.1.4 [readTap\(\)](#)** char [readTap](#) (  
    [EmbPattern](#) \* *pattern*,  
    FILE \* *file* )

**18.87.1.5 [writeTap\(\)](#)** char [writeTap](#) (  
    [EmbPattern](#) \* *pattern*,  
    FILE \* *file* )

## 18.88 extern/libembroidery/src/formats/format\_thr.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readThr](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeThr](#) ([EmbPattern](#) \*pattern, FILE \*file)

### 18.88.1 Function Documentation



**18.88.1.1 readThr()** `char readThr (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.88.1.2 Embroidery Format (.thr)** The ThreadWorks thr format is stitch-only.

**18.88.1.3 writeThr()** `char writeThr (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.89 extern/libembroidery/src/formats/format\_txt.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readTxt](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeTxt](#) ([EmbPattern](#) \*pattern, FILE \*file)

#### 18.89.1 Function Documentation

**18.89.1.1 readTxt()** `char readTxt (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.89.1.2 File (.txt)** The txt format is stitch-only and isn't associated with a specific company.

**18.89.1.3 writeTxt()** `char writeTxt (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.90 extern/libembroidery/src/formats/format\_u00.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readU00](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeU00](#) ([EmbPattern](#) \*pattern, FILE \*file)

#### 18.90.1 Function Documentation

**18.90.1.1 readU00()** `char readU00 (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.90.1.2 Embroidery Format (.u00)** The Barudan u00 format is stitch-only.

**18.90.1.3 writeU00()** `char writeU00 (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.91 extern/libembroidery/src/formats/format\_u01.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- char [readU01](#) ([EmbPattern](#) \*pattern, FILE \*file)
- char [writeU01](#) ([EmbPattern](#) \*pattern, FILE \*file)

#### 18.91.1 Function Documentation

**18.91.1.1 readU01()** `char readU01 (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.91.1.2 Embroidery Format (.u00)** The Barudan u01 format is stitch-only.

**18.91.1.3 writeU01()** `char writeU01 (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.92 extern/libembroidery/src/formats/format\_vip.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- int [vipDecodeByte](#) (unsigned char b)
- int [vipDecodeStitchType](#) (unsigned char b)
- unsigned char \* [vipDecompressData](#) (unsigned char \*input, int compressedInputLength, int decompressedContentLength)
- char [readVip](#) ([EmbPattern](#) \*pattern, FILE \*file)
- unsigned char \* [vipCompressData](#) (unsigned char \*input, int decompressedInputSize, int \*compressedSize)
- unsigned char [vipEncodeByte](#) ([EmbReal](#) f)
- unsigned char [vipEncodeStitchType](#) (int st)
- char [writeVip](#) ([EmbPattern](#) \*pattern, FILE \*file)

### Variables

- const unsigned char [vipDecodingTable](#) []

## 18.92.1 Function Documentation

**18.92.1.1 readVip()** char readVip (  
     EmbPattern \* pattern,  
     FILE \* file )

**18.92.1.2 vipCompressData()** unsigned char \* vipCompressData (  
     unsigned char \* input,  
     int decompressedInputSize,  
     int \* compressedSize )

**18.92.1.3 vipDecodeByte()** int vipDecodeByte (  
     unsigned char b )

**18.92.1.4 vipDecodeStitchType()** int vipDecodeStitchType (  
     unsigned char b )

**18.92.1.5 vipDecompressData()** unsigned char \* vipDecompressData (  
     unsigned char \* input,  
     int compressedInputLength,  
     int decompressedContentLength )

**18.92.1.6 vipEncodeByte()** unsigned char vipEncodeByte (  
     EmbReal f )

**18.92.1.7 vipEncodeStitchType()** unsigned char vipEncodeStitchType (  
     int st )

**18.92.1.8 writeVip()** char writeVip (  
     EmbPattern \* pattern,  
     FILE \* file )

## 18.92.2 Variable Documentation

**18.92.2.1 vipDecodingTable** const unsigned char vipDecodingTable[]

Initial value:

```
= {
    0x2E, 0x82, 0xE4, 0x6F, 0x38, 0xA9, 0xDC, 0xC6, 0x7B, 0xB6, 0x28, 0xAC, 0xFD, 0xAA, 0x8A, 0x4E,
    0x76, 0x2E, 0xF0, 0xE4, 0x25, 0x1B, 0x8A, 0x68, 0x4E, 0x92, 0xB9, 0xB4, 0x95, 0xF0, 0x3E, 0xEF,
    0xF7, 0x40, 0x24, 0x18, 0x39, 0x31, 0xBB, 0xE1, 0x53, 0xA8, 0x1F, 0xB1, 0x3A, 0x07, 0xFB, 0xCB,
    0xE6, 0x00, 0x81, 0x50, 0x0E, 0x40, 0xE1, 0x2C, 0x73, 0x50, 0x0D, 0x91, 0xD6, 0x0A, 0x5D, 0xD6,
    0x8B, 0xB8, 0x62, 0xAE, 0x47, 0x00, 0x53, 0x5A, 0xB7, 0x80, 0xAA, 0x28, 0xF7, 0x5D, 0x70, 0x5E,
    0x2C, 0x0B, 0x98, 0xE3, 0xA0, 0x98, 0x60, 0x47, 0x89, 0x9B, 0x82, 0xFB, 0x40, 0xC9, 0xB4, 0x00,
    0x0E, 0x68, 0x6A, 0x1E, 0x09, 0x85, 0xC0, 0x53, 0x81, 0xD1, 0x98, 0x89, 0xAF, 0xE8, 0x85, 0x4F,
    0xE3, 0x69, 0x89, 0x03, 0xA1, 0x2E, 0x8F, 0xCF, 0xED, 0x91, 0x9F, 0x58, 0x1E, 0xD6, 0x84, 0x3C,
    0x09, 0x27, 0xBD, 0xF4, 0xC3, 0x90, 0xC0, 0x51, 0x1B, 0x2B, 0x63, 0xBC, 0xB9, 0x3D, 0x40, 0x4D,
    0x62, 0x6F, 0xE0, 0x8C, 0xF5, 0x5D, 0x08, 0xFD, 0x3D, 0x50, 0x36, 0xD7, 0xC9, 0xC9, 0x43, 0xE4,
    0x2D, 0xCB, 0x95, 0xB6, 0xF4, 0x0D, 0xEA, 0xC2, 0xFD, 0x66, 0x3F, 0x5E, 0xBD, 0x69, 0x06, 0x2A,
    0x03, 0x19, 0x47, 0x2B, 0xDF, 0x38, 0xEA, 0x4F, 0x80, 0x49, 0x95, 0xB2, 0xD6, 0xF9, 0x9A, 0x75,
    0xF4, 0xD8, 0x9B, 0x1D, 0xB0, 0xA4, 0x69, 0xDB, 0xA9, 0x21, 0x79, 0x6F, 0xD8, 0xDE, 0x33, 0xFE,
```

```

0x9F, 0x04, 0xE5, 0x9A, 0x6B, 0x9B, 0x73, 0x83, 0x62, 0x7C, 0xB9, 0x66, 0x76, 0xF2, 0x5B, 0xC9,
0x5E, 0xFC, 0x74, 0xAA, 0x6C, 0xF1, 0xCD, 0x93, 0xCE, 0xE9, 0x80, 0x53, 0x03, 0x3B, 0x97, 0x4B,
0x39, 0x76, 0xC2, 0xC1, 0x56, 0xCB, 0x70, 0xFD, 0x3B, 0x3E, 0x52, 0x57, 0x81, 0x5D, 0x56, 0x8D,
0x51, 0x90, 0xD4, 0x76, 0xD7, 0xD5, 0x16, 0x02, 0x6D, 0xF2, 0x4D, 0xE1, 0x0E, 0x96, 0x4F, 0xA1,
0x3A, 0xA0, 0x60, 0x59, 0x64, 0x04, 0x1A, 0xE4, 0x67, 0xB6, 0xED, 0x3F, 0x74, 0x20, 0x55, 0x1F,
0xFB, 0x23, 0x92, 0x91, 0x53, 0xC8, 0x65, 0xAB, 0x9D, 0x51, 0xD6, 0x73, 0xDE, 0x01, 0xB1, 0x80,
0xB7, 0xC0, 0xD6, 0x80, 0x1C, 0x2E, 0x3C, 0x83, 0x63, 0xEE, 0xBC, 0x33, 0x25, 0xE2, 0x0E, 0x7A,
0x67, 0xDE, 0x3F, 0x71, 0x14, 0x49, 0x9C, 0x92, 0x93, 0x0D, 0x26, 0x9A, 0x0E, 0xDA, 0xED, 0x6F,
0xA4, 0x89, 0x0C, 0x1B, 0xF0, 0xA1, 0xDF, 0xE1, 0x9E, 0x3C, 0x04, 0x78, 0xE4, 0xAB, 0x6D, 0xFF,
0x9C, 0xAF, 0xCA, 0xC7, 0x88, 0x17, 0x9C, 0xE5, 0xB7, 0x33, 0x6D, 0xDC, 0xED, 0x8F, 0x6C, 0x18,
0x1D, 0x71, 0x06, 0xB1, 0xC5, 0xE2, 0xCF, 0x13, 0x77, 0x81, 0xC5, 0xB7, 0x0A, 0x14, 0x0A, 0x6B,
0x40, 0x26, 0xA0, 0x88, 0xD1, 0x62, 0x6A, 0xB3, 0x50, 0x12, 0xB9, 0x9B, 0xB5, 0x83, 0x9B, 0x37
}

```

**18.92.2.2 Embroidery Format (.pcq)** The Pfaff vip format is stitch-only.

## 18.93 extern/libembroidery/src/formats/format\_vp3.c File Reference

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"

```

### Functions

- unsigned char \* [vp3ReadString](#) (FILE \*file)
- int [vp3Decode](#) (unsigned char inputByte)
- short [vp3DecodeInt16](#) (unsigned short inputByte)
- [vp3Hoop vp3ReadHoopSection](#) (FILE \*file)
- char [readVp3](#) ([EmbPattern](#) \*pattern, FILE \*file)
- void [vp3WriteStringLen](#) (FILE \*file, const char \*str, int len)
- void [vp3WriteString](#) (FILE \*file, const char \*str)
- void [vp3PatchByteCount](#) (FILE \*file, int offset, int adjustment)
- char [writeVp3](#) ([EmbPattern](#) \*pattern, FILE \*file)

### 18.93.1 Function Documentation

**18.93.1.1 readVp3()** char readVp3 (  
[EmbPattern](#) \* pattern,  
FILE \* file )

**18.93.1.2 vp3Decode()** int vp3Decode (  
unsigned char inputByte )

**18.93.1.3 vp3DecodeInt16()** short vp3DecodeInt16 (  
unsigned short inputByte )

**18.93.1.4 vp3PatchByteCount()** void vp3PatchByteCount (  
FILE \* file,  
int offset,  
int adjustment )

**18.93.1.5** `vp3ReadHoopSection()` `vp3Hoop` `vp3ReadHoopSection` (  
`FILE * file` )

**18.93.1.6** `vp3ReadString()` `unsigned char *` `vp3ReadString` (  
`FILE * file` )

**18.93.1.7** **Embroidery Format (.pcq)** The Pfaff vp3 format is stitch-only.

**18.93.1.8** `vp3WriteString()` `void` `vp3WriteString` (  
`FILE * file`,  
`const char * str` )

**18.93.1.9** `vp3WriteStringLen()` `void` `vp3WriteStringLen` (  
`FILE * file`,  
`const char * str`,  
`int len` )

**18.93.1.10** `writeVp3()` `char` `writeVp3` (  
`EmbPattern * pattern`,  
`FILE * file` )

## 18.94 extern/libembroidery/src/formats/format\_xxx.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

### Functions

- `char` `xxxDecodeByte` (`unsigned char` `inputByte`)
- `char` `readXxx` (`EmbPattern * pattern`, `FILE * file`)
- `void` `xxxEncodeStop` (`FILE * file`, `EmbStitch s`)
- `void` `xxxEncodeStitch` (`FILE * file`, `EmbReal` `deltaX`, `EmbReal` `deltaY`, `int` `flags`)
- `void` `xxxEncodeDesign` (`FILE * file`, `EmbPattern * p`)
- `char` `writeXxx` (`EmbPattern * pattern`, `FILE * file`)

### 18.94.1 Function Documentation

**18.94.1.1** `readXxx()` `char` `readXxx` (  
`EmbPattern * pattern`,  
`FILE * file` )

**18.94.1.2** `writeXxx()` `char` `writeXxx` (  
`EmbPattern * pattern`,  
`FILE * file` )

**18.94.1.3** `xxxDecodeByte()` `char` `xxxDecodeByte` (  
`unsigned char` `inputByte` )

**18.94.1.4 Embroidery Format (.xxx)** The Singer xxx format is stitch-only.

**18.94.1.5 xxxEncodeDesign()** `void xxxEncodeDesign (`  
    `FILE * file,`  
    `EmbPattern * p )`

**18.94.1.6 xxxEncodeStitch()** `void xxxEncodeStitch (`  
    `FILE * file,`  
    `EmbReal deltaX,`  
    `EmbReal deltaY,`  
    `int flags )`

**18.94.1.7 xxxEncodeStop()** `void xxxEncodeStop (`  
    `FILE * file,`  
    `EmbStitch s )`

## 18.95 extern/libembroidery/src/formats/format\_zsk.c File Reference

```
#include <stdio.h>
#include <string.h>
#include "../embroidery_internal.h"
```

### Functions

- char `readZsk` (`EmbPattern *pattern`, `FILE *file`)
- char `writeZsk` (`EmbPattern *pattern`, `FILE *file`)

#### 18.95.1 Detailed Description

The ZSK USA Embroidery Format (.zsk)  
The ZSK USA zsk format is stitch-only.

#### 18.95.2 Function Documentation

**18.95.2.1 readZsk()** `char readZsk (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

**18.95.2.2 writeZsk()** `char writeZsk (`  
    `EmbPattern * pattern,`  
    `FILE * file )`

## 18.96 extern/libembroidery/src/geometry.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "embroidery.h"
```

## Functions

- `EmbGeometry * embGeometry_init (int type_in)`  
*Our generic object interface backends to each individual type.*
- `void embGeometry_free (EmbGeometry *obj)`  
*Free the memory occupied by a non-stitch geometry object.*
- `void embGeometry_move (EmbGeometry *obj, EmbVector delta)`  
*Translate obj by the vector delta.*
- `EmbRect embGeometry_boundingRect (EmbGeometry *obj)`  
*Calculate the bounding box of geometry obj based on what kind of geometric object it is.*
- `void embGeometry_vulcanize (EmbGeometry *obj)`  
*Toggle the rubber mode of the object.*

### 18.96.1 Function Documentation

**18.96.1.1 embGeometry\_boundingRect()** `EmbRect embGeometry_boundingRect ( EmbGeometry * obj )`

Calculate the bounding box of geometry *obj* based on what kind of geometric object it is.  
*obj* A pointer to the geometry memory.

#### Returns

EmbRect The bounding box in the same scale as the input geometry.

In the case of a failure the bounding box returned is always the unit square with top left corner at (0, 0).

**18.96.1.2 embGeometry\_free()** `void embGeometry_free ( EmbGeometry * obj )`

Free the memory occupied by a non-stitch geometry object.  
*obj* Pointer to geometry memory.

**18.96.1.3 embGeometry\_init()** `EmbGeometry * embGeometry_init ( int type_in )`

Our generic object interface backends to each individual type.  
*type\_in*

#### Returns

EmbGeometry\*

**18.96.1.4 embGeometry\_move()** `void embGeometry_move ( EmbGeometry * obj, EmbVector delta )`

Translate *obj* by the vector *delta*.

*obj* A pointer to the geometry memory. *delta* A vector in the 0.1mm scale to offset the geometry by.

**18.96.1.5 embGeometry\_vulcanize()** `void embGeometry_vulcanize ( EmbGeometry * obj )`

Toggle the rubber mode of the object.  
*obj*

**Todo** Review. This could be controlled by a simple flag.

## 18.97 extern/libembroidery/src/geometry/arc.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

### Functions

- [EmbArc embArc\\_init](#) (void)
- char [embArc\\_clockwise](#) (EmbArc arc)
- void [getArcCenter](#) (EmbArc arc, EmbVector \*arcCenter)
- char [getArcDataFromBulge](#) (EmbReal bulge, EmbArc \*arc, EmbReal \*arcCenterX, EmbReal \*arcCenterY, EmbReal \*radius, EmbReal \*diameter, EmbReal \*chord, EmbReal \*chordMidX, EmbReal \*chordMidY, EmbReal \*sagitta, EmbReal \*apothem, EmbReal \*incAngleInDegrees, char \*clockwise)
- char [clockwise](#) (EmbGeometry \*obj)
- void [embArc\\_setCenter](#) (EmbArc \*arc, EmbVector point)
- void [embArc\\_setRadius](#) (EmbArc \*arc, float radius)
- void [embArc\\_setStartAngle](#) (EmbArc \*arc, float angle)
- void [embArc\\_setEndAngle](#) (EmbArc \*arc, float angle)
- float [embArc\\_startAngle](#) (EmbArc arc)
- float [embArc\\_endAngle](#) (EmbArc arc)
- float [embArc\\_area](#) (EmbArc arc)
- float [embArc\\_arcLength](#) (EmbArc arc)
- float [embArc\\_chord](#) (EmbArc arc)
- float [embArc\\_includedAngle](#) (EmbArc arc)
- char [Arc\\_clockwise](#) ()
- void [embArc\\_updatePath](#) (EmbArc arc)
- void [embArc\\_paint](#) (void)
- void [embArc\\_updateRubber](#) (EmbArc arc, int pattern, int layer, int index)
- EmbVector [embArc\\_mouseSnapPoint](#) (EmbArc arc, EmbVector mousePoint)
- void [embArc\\_gripEdit](#) (EmbArc \*arc, EmbVector before, EmbVector after)
- void [set\\_object\\_color](#) (EmbGeometry \*obj, EmbColor color)
- void [embBase\\_setColorRGB](#) (EmbGeometry \*obj, unsigned int rgb)
- void [Base\\_setLineType](#) (EmbGeometry \*obj, int lineType)
- void [Base\\_setLineWeight](#) (EmbGeometry \*obj, float lineWeight)
- EmbVector [Base\\_objectRubberPoint](#) (EmbGeometry \*obj, const char \*key)
- const char \* [Base\\_objectRubberText](#) (EmbGeometry \*obj, const char \*key)
- void [embCircle\\_prompt](#) (const char \*str)
- void [embCircle\\_setArea](#) (EmbCircle \*circle, float area)
- void [embCircle\\_setCircumference](#) (EmbCircle \*circle, float circumference)
- void [embEllipse\\_main](#) ()
- void [embEllipse\\_click](#) (float x, float y)
- EmbVector [embRect\\_bottomLeft](#) (EmbRect rect)
- EmbVector [embRect\\_bottomRight](#) (EmbRect rect)

### 18.97.1 Function Documentation

#### 18.97.1.1 [Arc\\_clockwise\(\)](#) char Arc\_clockwise ( )



**18.97.1.2 Base\_objectRubberPoint()** `EmbVector Base_objectRubberPoint (`  
    `EmbGeometry * obj,`  
    `const char * key )`

**18.97.1.3 Base\_objectRubberText()** `const char * Base_objectRubberText (`  
    `EmbGeometry * obj,`  
    `const char * key )`

**18.97.1.4 Base\_setLineType()** `void Base_setLineType (`  
    `EmbGeometry * obj,`  
    `int lineType )`

**18.97.1.5 Base\_setLineWeight()** `void Base_setLineWeight (`  
    `EmbGeometry * obj,`  
    `float lineWeight )`

**18.97.1.6 clockwise()** `char clockwise (`  
    `EmbGeometry * obj )`

**18.97.1.7 embArc\_arcLength()** `float embArc_arcLength (`  
    `EmbArc arc )`

**18.97.1.8 embArc\_area()** `float embArc_area (`  
    `EmbArc arc )`

**18.97.1.9 embArc\_chord()** `float embArc_chord (`  
    `EmbArc arc )`

**18.97.1.10 embArc\_clockwise()** `char embArc_clockwise (`  
    `EmbArc arc )`

**18.97.1.11 embArc\_endAngle()** `float embArc_endAngle (`  
    `EmbArc arc )`

**18.97.1.12 embArc\_gripEdit()** `void embArc_gripEdit (`  
    `EmbArc * arc,`  
    `EmbVector before,`  
    `EmbVector after )`

**18.97.1.13 embArc\_includedAngle()** `float embArc_includedAngle (`  
    `EmbArc arc )`

**18.97.1.14 embArc\_init()** `EmbArc embArc_init (`  
`void )`

**18.97.1.15 embArc\_mouseSnapPoint()** `EmbVector embArc_mouseSnapPoint (`  
`EmbArc arc,`  
`EmbVector mousePoint )`

**18.97.1.16 embArc\_paint()** `void embArc_paint (`  
`void )`

**18.97.1.17 embArc\_setCenter()** `void embArc_setCenter (`  
`EmbArc * arc,`  
`EmbVector point )`

**18.97.1.18 embArc\_setEndAngle()** `void embArc_setEndAngle (`  
`EmbArc * arc,`  
`float angle )`

**18.97.1.19 embArc\_setRadius()** `void embArc_setRadius (`  
`EmbArc * arc,`  
`float radius )`

**18.97.1.20 embArc\_setStartAngle()** `void embArc_setStartAngle (`  
`EmbArc * arc,`  
`float angle )`

**18.97.1.21 embArc\_startAngle()** `float embArc_startAngle (`  
`EmbArc arc )`

**18.97.1.22 embArc\_updatePath()** `void embArc_updatePath (`  
`EmbArc arc )`

**18.97.1.23 embArc\_updateRubber()** `void embArc_updateRubber (`  
`EmbArc arc,`  
`int pattern,`  
`int layer,`  
`int index )`

**18.97.1.24 embBase\_setColorRGB()** `void embBase_setColorRGB (`  
`EmbGeometry * obj,`  
`unsigned int rgb )`

**18.97.1.25 embCircle\_prompt()** `void embCircle_prompt (`  
`const char * str )`

- 18.97.1.26 embCircle\_setArea()** void embCircle\_setArea (   
     EmbCircle \* circle,   
     float area )
- 18.97.1.27 embCircle\_setCircumference()** void embCircle\_setCircumference (   
     EmbCircle \* circle,   
     float circumference )
- 18.97.1.28 embEllipse\_click()** void embEllipse\_click (   
     float x,   
     float y )
- 18.97.1.29 embEllipse\_main()** void embEllipse\_main ( )
- 18.97.1.30 embRect\_bottomLeft()** EmbVector embRect\_bottomLeft (   
     EmbRect rect )
- 18.97.1.31 embRect\_bottomRight()** EmbVector embRect\_bottomRight (   
     EmbRect rect )
- 18.97.1.32 getArcCenter()** void getArcCenter (   
     EmbArc arc,   
     EmbVector \* arcCenter )
- 18.97.1.33 getArcDataFromBulge()** char getArcDataFromBulge (   
     EmbReal bulge,   
     EmbArc \* arc,   
     EmbReal \* arcCenterX,   
     EmbReal \* arcCenterY,   
     EmbReal \* radius,   
     EmbReal \* diameter,   
     EmbReal \* chord,   
     EmbReal \* chordMidX,   
     EmbReal \* chordMidY,   
     EmbReal \* sagitta,   
     EmbReal \* apothem,   
     EmbReal \* incAngleInDegrees,   
     char \* clockwise )
- 18.97.1.34 set\_object\_color()** void set\_object\_color (   
     EmbGeometry \* obj,   
     EmbColor color )

## 18.98 extern/libembroidery/src/geometry/circle.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

```
#include "../embroidery.h"
```

## Functions

- [EmbCircle embCircle\\_init](#) (void)
- [EmbReal embCircle\\_area](#) (EmbCircle circle)
- [EmbReal embCircle\\_circumference](#) (EmbCircle circle)
- int [getCircleCircleIntersections](#) (EmbCircle c0, EmbCircle c1, EmbVector \*p0, EmbVector \*p1)
- int [getCircleTangentPoints](#) (EmbCircle c, EmbVector point, EmbVector \*t0, EmbVector \*t1)

### 18.98.1 Function Documentation

**18.98.1.1 [embCircle\\_area\(\)](#)** [EmbReal](#) embCircle\_area (  
[EmbCircle](#) circle )

**18.98.1.2 [embCircle\\_circumference\(\)](#)** [EmbReal](#) embCircle\_circumference (  
[EmbCircle](#) circle )

**18.98.1.3 [embCircle\\_init\(\)](#)** [EmbCircle](#) embCircle\_init (  
void )

**18.98.1.4 [getCircleCircleIntersections\(\)](#)** int getCircleCircleIntersections (  
[EmbCircle](#) c0,  
[EmbCircle](#) c1,  
[EmbVector](#) \* p0,  
[EmbVector](#) \* p1 )

**18.98.1.5 [getCircleTangentPoints\(\)](#)** int getCircleTangentPoints (  
[EmbCircle](#) c,  
[EmbVector](#) point,  
[EmbVector](#) \* t0,  
[EmbVector](#) \* t1 )

## 18.99 extern/libembroidery/src/geometry/ellipse.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

## Functions

- [EmbEllipse embEllipse\\_init](#) (void)
- [EmbReal embEllipse\\_area](#) (EmbEllipse ellipse)
- [EmbReal embEllipse\\_perimeter](#) (EmbEllipse ellipse)
- [EmbReal embEllipse\\_diameterX](#) (EmbEllipse ellipse)
- [EmbReal embEllipse\\_diameterY](#) (EmbEllipse ellipse)
- [EmbReal embEllipse\\_width](#) (EmbEllipse ellipse)
- [EmbReal embEllipse\\_height](#) (EmbEllipse ellipse)

- void `embEllipse_setSize` (float width, float height)
- void `embEllipse_setRadiusMajor` (float radius)
- void `embEllipse_setRadiusMinor` (float radius)
- void `embEllipse_setDiameterMajor` (`EmbEllipse` \*ellipse, float diameter)
- void `embEllipse_setDiameterMinor` (`EmbEllipse` \*ellipse, float diameter)
- `EmbVector` `ellipse_objectQuadrant0` (`EmbEllipse` \*ellipse)
- `EmbVector` `ellipse_objectQuadrant90` (`EmbEllipse` \*ellipse)
- `EmbVector` `ellipse_objectQuadrant180` (`EmbEllipse` \*ellipse)
- `EmbVector` `ellipse_objectQuadrant270` (`EmbEllipse` \*ellipse)
- void `embEllipse_updatePath` ()

### 18.99.1 Function Documentation

**18.99.1.1** `ellipse_objectQuadrant0()` `EmbVector` `ellipse_objectQuadrant0` (  
    `EmbEllipse` \* *ellipse* )

**18.99.1.2** `ellipse_objectQuadrant180()` `EmbVector` `ellipse_objectQuadrant180` (  
    `EmbEllipse` \* *ellipse* )

**18.99.1.3** `ellipse_objectQuadrant270()` `EmbVector` `ellipse_objectQuadrant270` (  
    `EmbEllipse` \* *ellipse* )

**18.99.1.4** `ellipse_objectQuadrant90()` `EmbVector` `ellipse_objectQuadrant90` (  
    `EmbEllipse` \* *ellipse* )

**18.99.1.5** `embEllipse_area()` `EmbReal` `embEllipse_area` (  
    `EmbEllipse` *ellipse* )

**18.99.1.6** `embEllipse_diameterX()` `EmbReal` `embEllipse_diameterX` (  
    `EmbEllipse` *ellipse* )

**18.99.1.7** `embEllipse_diameterY()` `EmbReal` `embEllipse_diameterY` (  
    `EmbEllipse` *ellipse* )

**18.99.1.8** `embEllipse_height()` `EmbReal` `embEllipse_height` (  
    `EmbEllipse` *ellipse* )

**18.99.1.9** `embEllipse_init()` `EmbEllipse` `embEllipse_init` (  
    void )

**18.99.1.10** `embEllipse_perimeter()` `EmbReal` `embEllipse_perimeter` (  
    `EmbEllipse` *ellipse* )

**18.99.1.11 embEllipse\_setDiameterMajor()** void embEllipse\_setDiameterMajor (   
    EmbEllipse \* ellipse,   
    float diameter )

**18.99.1.12 embEllipse\_setDiameterMinor()** void embEllipse\_setDiameterMinor (   
    EmbEllipse \* ellipse,   
    float diameter )

**18.99.1.13 embEllipse\_setRadiusMajor()** void embEllipse\_setRadiusMajor (   
    float radius )

**18.99.1.14 embEllipse\_setRadiusMinor()** void embEllipse\_setRadiusMinor (   
    float radius )

**18.99.1.15 embEllipse\_setSize()** void embEllipse\_setSize (   
    float width,   
    float height )

**18.99.1.16 embEllipse\_updatePath()** void embEllipse\_updatePath ( )

**18.99.1.17 embEllipse\_width()** EmbReal embEllipse\_width (   
    EmbEllipse ellipse )

## 18.100 extern/libembroidery/src/geometry/functions.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

### Functions

- int [emb\\_round](#) (EmbReal x)
- [EmbReal radians](#) (EmbReal degree)
- [EmbReal degrees](#) (EmbReal radian)

#### 18.100.1 Function Documentation

**18.100.1.1 degrees()** EmbReal degrees (   
    EmbReal radian )

**18.100.1.2 emb\_round()** int emb\_round (   
    EmbReal x )

**18.100.1.3 radians()** `EmbReal radians (`  
`EmbReal degree )`

## 18.101 extern/libembroidery/src/geometry/line.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

### Functions

- void `embLine_normalVector` (`EmbLine` line, `EmbVector` \*result, int clockwise)
- `EmbVector` `embLine_toVector` (`EmbLine` line)
- `EmbVector` `embLine_intersectionPoint` (`EmbLine` line1, `EmbLine` line2)

### 18.101.1 Function Documentation

**18.101.1.1 embLine\_intersectionPoint()** `EmbVector embLine_intersectionPoint (`  
`EmbLine line1,`  
`EmbLine line2 )`

**18.101.1.2 embLine\_normalVector()** `void embLine_normalVector (`  
`EmbLine line,`  
`EmbVector * result,`  
`int clockwise )`

Finds the normalized vector perpendicular (clockwise) to the line given by v1->v2 (normal to the line)

**18.101.1.3 embLine\_toVector()** `EmbVector embLine_toVector (`  
`EmbLine line )`

## 18.102 extern/libembroidery/src/geometry/path.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

## 18.103 extern/libembroidery/src/geometry/polygon.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

## 18.104 extern/libembroidery/src/geometry/polyline.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

## 18.105 extern/libembroidery/src/geometry/rect.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

### Functions

- [EmbRect embRect\\_init](#) (void)
- [EmbReal embRect\\_area](#) (EmbRect rect)

#### 18.105.1 Function Documentation

**18.105.1.1 [embRect\\_area\(\)](#)** [EmbReal](#) embRect\_area (  
    [EmbRect](#) rect )

**18.105.1.2 [embRect\\_init\(\)](#)** [EmbRect](#) embRect\_init (  
    void )

## 18.106 extern/libembroidery/src/geometry/text.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

### Functions

- void [textSingle\\_setTextFont](#) (const char \*font)
- void [textSingle\\_setJustify](#) (const char \*justify)
- void [textSingle\\_setTextSize](#) (float size)
- void [textSingle\\_setTextStyle](#) (char bold, char italic, char under, char strike, char over)
- void [textSingle\\_setTextBold](#) (char val)
- void [textSingle\\_setTextItalic](#) (char val)
- void [textSingle\\_setTextUnderline](#) (char val)
- void [textSingle\\_setTextStrikeOut](#) (char val)
- void [textSingle\\_setTextOverline](#) (char val)
- void [textSingle\\_setTextBackward](#) (char val)
- void [textSingle\\_setTextUpsideDown](#) (char val)
- void [textSingle\\_paint](#) ()
- void [textSingle\\_updateRubber](#) ()
- [EmbVector](#) [textSingle\\_mouseSnapPoint](#) ([EmbVector](#) mousePoint)
- void [textSingle\\_gripEdit](#) ([EmbVector](#) before, [EmbVector](#) after)

#### 18.106.1 Function Documentation

**18.106.1.1 [textSingle\\_gripEdit\(\)](#)** void textSingle\_gripEdit (  
    [EmbVector](#) before,  
    [EmbVector](#) after )



- 18.106.1.2 textSingle\_mouseSnapPoint()** `EmbVector textSingle_mouseSnapPoint ( EmbVector mousePoint )`
- 18.106.1.3 textSingle\_paint()** `void textSingle_paint ( )`
- 18.106.1.4 textSingle\_setJustify()** `void textSingle_setJustify ( const char * justify )`
- 18.106.1.5 textSingle\_setTextBackward()** `void textSingle_setTextBackward ( char val )`
- 18.106.1.6 textSingle\_setTextBold()** `void textSingle_setTextBold ( char val )`
- 18.106.1.7 textSingle\_setTextFont()** `void textSingle_setTextFont ( const char * font )`
- 18.106.1.8 textSingle\_setTextItalic()** `void textSingle_setTextItalic ( char val )`
- 18.106.1.9 textSingle\_setTextOverline()** `void textSingle_setTextOverline ( char val )`
- 18.106.1.10 textSingle\_setTextSize()** `void textSingle_setTextSize ( float size )`
- 18.106.1.11 textSingle\_setTextStrikeOut()** `void textSingle_setTextStrikeOut ( char val )`
- 18.106.1.12 textSingle\_setTextStyle()** `void textSingle_setTextStyle ( char bold, char italic, char under, char strike, char over )`
- 18.106.1.13 textSingle\_setTextUnderline()** `void textSingle_setTextUnderline ( char val )`
- 18.106.1.14 textSingle\_setTextUpsideDown()** `void textSingle_setTextUpsideDown ( char val )`

**18.106.1.15 textSingle\_updateRubber()** void textSingle\_updateRubber ( )

## 18.107 extern/libembroidery/src/geometry/vector.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery.h"
```

### Functions

- void `embVector_normalize` (EmbVector vector, EmbVector \*result)
- void `embVector_multiply` (EmbVector vector, EmbReal magnitude, EmbVector \*result)
- EmbVector `embVector_add` (EmbVector a, EmbVector b)
- EmbVector `embVector_average` (EmbVector a, EmbVector b)
- EmbVector `embVector_subtract` (EmbVector v1, EmbVector v2)
- EmbReal `embVector_dot` (EmbVector a, EmbVector b)
- EmbReal `embVector_cross` (EmbVector a, EmbVector b)

*The "cross product" as vectors a and b returned as a real value.*

- void `embVector_transpose_product` (EmbVector v1, EmbVector v2, EmbVector \*result)
- EmbReal `embVector_length` (EmbVector vector)
- EmbReal `embVector_relativeX` (EmbVector a1, EmbVector a2, EmbVector a3)
- EmbReal `embVector_relativeY` (EmbVector a1, EmbVector a2, EmbVector a3)
- EmbReal `embVector_angle` (EmbVector v)
- EmbVector `embVector_unit` (EmbReal alpha)
- EmbReal `embVector_distance` (EmbVector a, EmbVector b)

### 18.107.1 Function Documentation

**18.107.1.1 embVector\_add()** EmbVector embVector\_add (

EmbVector a,

EmbVector b )

The sum of vectors *a* and *b* returned as a vector.

Equivalent to:

$$\mathbf{c} = \mathbf{a} + \mathbf{b} = \begin{pmatrix} a_x + b_x \\ a_y + b_y \end{pmatrix}$$

**18.107.1.2 embVector\_angle()** EmbReal embVector\_angle (

EmbVector v )

The angle, measured anti-clockwise from the x-axis, of a vector *v*.

**18.107.1.3 embVector\_average()** EmbVector embVector\_average (

EmbVector a,

EmbVector b )

The average of vectors *v1* and *v2* returned as a vector.

Equivalent to:

$$\mathbf{c} = \frac{\mathbf{a} + \mathbf{b}}{2} = \begin{pmatrix} \frac{a_x + b_x}{2} \\ \frac{a_y + b_y}{2} \end{pmatrix}$$

**18.107.1.4 embVector\_cross()** `EmbReal embVector_cross (`  
`EmbVector a,`  
`EmbVector b )`

The "cross product" as vectors *a* and *b* returned as a real value.

Technically, this is the magnitude of the cross product when the embroidery is placed in the z=0 plane (since the cross product is defined for 3-dimensional vectors). That is:

$$|c| = \left| \begin{pmatrix} a_x \\ a_y \\ 0 \end{pmatrix} \times \begin{pmatrix} b_x \\ b_y \\ 0 \end{pmatrix} \right| = \left| \begin{pmatrix} 0 \\ 0 \\ a_x b_y - a_y b_x \end{pmatrix} \right| = a_x b_y - a_y b_x$$

**18.107.1.5 embVector\_distance()** `EmbReal embVector_distance (`  
`EmbVector a,`  
`EmbVector b )`

The distance between *a* and *b* returned as a real value.

$$d = |\mathbf{a} - \mathbf{b}| = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}$$

**18.107.1.6 embVector\_dot()** `EmbReal embVector_dot (`  
`EmbVector a,`  
`EmbVector b )`

The dot product as vectors *v1* and *v2* returned as a `EmbReal`.

Equivalent to:

$$c = \mathbf{a} \cdot \mathbf{b} = a_x b_x + a_y b_y$$

**18.107.1.7 embVector\_length()** `EmbReal embVector_length (`  
`EmbVector vector )`

The length or absolute value of the vector *vector*.

Equivalent to:

$$|v| = \sqrt{v_x^2 + v_y^2}$$

**18.107.1.8 embVector\_multiply()** `void embVector_multiply (`  
`EmbVector vector,`  
`EmbReal magnitude,`  
`EmbVector * result )`

The scalar multiple *magnitude* of a vector *vector*. Returned as *result*.

**Todo** make result return argument.

**18.107.1.9 embVector\_normalize()** `void embVector_normalize (`  
`EmbVector vector,`  
`EmbVector * result )`

Finds the unit length vector *result* in the same direction as *vector*.

Equivalent to:

$$\mathbf{u} = \frac{v}{|\mathbf{v}|}$$

**Todo** make result return argument.

**18.107.1.10** `embVector_relativeX()` `EmbReal` `embVector_relativeX` (

```

    EmbVector a1,
    EmbVector a2,
    EmbVector a3 )

```

The x-component of the vector

**18.107.1.11** `embVector_relativeY()` `EmbReal` `embVector_relativeY` (

```

    EmbVector a1,
    EmbVector a2,
    EmbVector a3 )

```

The y-component of the vector

**18.107.1.12** `embVector_subtract()` `EmbVector` `embVector_subtract` (

```

    EmbVector v1,
    EmbVector v2 )

```

The difference between vectors `v1` and `v2` returned as *result*.

Equivalent to:

$$\mathbf{c} = \mathbf{a} - \mathbf{b} = \begin{pmatrix} a_x - b_x \\ a_y - b_y \end{pmatrix}$$

**18.107.1.13** `embVector_transpose_product()` `void` `embVector_transpose_product` (

```

    EmbVector v1,
    EmbVector v2,
    EmbVector * result )

```

Since we aren't using full vector algebra here, all vectors are "vertical". so this is like the product  $\mathbf{v1}^T \mathbf{v2}$  for our vectors `v1` and `v2` so a "component-wise product". The result is stored at the pointer *result*.

That is  $\begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} xa & xb \end{pmatrix}$

**18.107.1.14** `embVector_unit()` `EmbVector` `embVector_unit` (

```

    EmbReal alpha )

```

The unit vector in the direction *angle*.

$$\mathbf{a}_\alpha = \begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \end{pmatrix}$$

## 18.108 extern/libembroidery/src/image.c File Reference

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "embroidery_internal.h"

```

### Functions

- void `writelnImage` (FILE \*file, unsigned char image[ ][48])
- float `image_diff` (unsigned char \*a, unsigned char \*b, int size)

### 18.108.1 Detailed Description

This backends to the stb libraries and nanosvg library.  
Use Python PEP7 for coding style.

### 18.108.2 Function Documentation

**18.108.2.1 image\_diff()** float image\_diff (

```

    unsigned char * a,
    unsigned char * b,
    int size )

```

The distance between the arrays *a* and *b* of length *size*.

**18.108.2.2 writeImage()** void writeImage (

```

    FILE * file,
    unsigned char image[][48] )

```

Write a PES embedded *image* to the given *file* pointer.

## 18.109 extern/libembroidery/src/main.c File Reference

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <time.h>
#include "embroidery_internal.h"

```

### Macros

- #define [FLAG\\_TO](#) 0
- #define [FLAG\\_TO\\_SHORT](#) 1
- #define [FLAG\\_HELP](#) 2
- #define [FLAG\\_HELP\\_SHORT](#) 3
- #define [FLAG\\_FORMATS](#) 4
- #define [FLAG\\_FORMATS\\_SHORT](#) 5
- #define [FLAG\\_QUIET](#) 6
- #define [FLAG\\_QUIET\\_SHORT](#) 7
- #define [FLAG\\_VERBOSE](#) 8
- #define [FLAG\\_VERBOSE\\_SHORT](#) 9
- #define [FLAG\\_VERSION](#) 10
- #define [FLAG\\_VERSION\\_SHORT](#) 11
- #define [FLAG\\_CIRCLE](#) 12
- #define [FLAG\\_CIRCLE\\_SHORT](#) 13
- #define [FLAG\\_ELLIPSE](#) 14
- #define [FLAG\\_ELLIPSE\\_SHORT](#) 15
- #define [FLAG\\_LINE](#) 16
- #define [FLAG\\_LINE\\_SHORT](#) 17
- #define [FLAG\\_POLYGON](#) 18
- #define [FLAG\\_POLYGON\\_SHORT](#) 19
- #define [FLAG\\_POLYLINE](#) 20
- #define [FLAG\\_POLYLINE\\_SHORT](#) 21
- #define [FLAG\\_RENDER](#) 22
- #define [FLAG\\_RENDER\\_SHORT](#) 23
- #define [FLAG\\_SATIN](#) 24
- #define [FLAG\\_SATIN\\_SHORT](#) 25
- #define [FLAG\\_STITCH](#) 26
- #define [FLAG\\_STITCH\\_SHORT](#) 27
- #define [FLAG\\_TEST](#) 28
- #define [FLAG\\_FULL\\_TEST\\_SUITE](#) 29
- #define [FLAG\\_HILBERT\\_CURVE](#) 30
- #define [FLAG\\_SIERPINSKI\\_TRIANGLE](#) 31
- #define [FLAG\\_FILL](#) 32

- `#define FLAG_FILL_SHORT` 33
- `#define FLAG_SIMULATE` 34
- `#define FLAG_COMBINE` 35
- `#define FLAG_CROSS_STITCH` 36
- `#define NUM_FLAGS` 37

## Functions

- void `embVector_print` (`EmbVector` v, char \*label)  
*v label*
- void `embArc_print` (`EmbArc` arc)  
*arc*
- int `check_header_present` (FILE \*file, int minimum\_header\_length)  
*file minimum\_header\_length*
- unsigned int `sectorSize` (`bcf_file` \*bcfFile)  
*bcfFile*
- int `haveExtraDIFATsectors` (`bcf_file` \*file)  
*file*
- int `seekToSector` (`bcf_file` \*bcfFile, FILE \*file, const unsigned int sector)  
*bcfFile file sector*
- void `parseDIFATsectors` (FILE \*file, `bcf_file` \*bcfFile)  
*file bcfFile*
- int `bcfFile_read` (FILE \*file, `bcf_file` \*bcfFile)  
*file bcfFile*
- FILE \* `GetFile` (`bcf_file` \*bcfFile, FILE \*file, char \*fileToFind)  
*Get the File object.*
- void `bcf_file_free` (`bcf_file` \*bcfFile)  
*bcfFile*
- `bcf_file_difat` \* `bcf_difat_create` (FILE \*file, unsigned int fatSectors, const unsigned int `sectorSize`)  
*file fatSectors sectorSize*
- unsigned int `entriesInDifatSector` (`bcf_file_difat` \*fat)  
*fat*
- unsigned int `readFullSector` (FILE \*file, `bcf_file_difat` \*bcfFile, unsigned int \*difatEntriesToRead)  
*file bcfFile difatEntriesToRead*
- void `parseDirectoryEntryName` (FILE \*file, `bcf_directory_entry` \*dir)  
*file dir*
- `bcf_directory` \* `CompoundFileDirectory` (const unsigned int maxNumberOfDirectoryEntries)  
*maxNumberOfDirectoryEntries*
- `EmbTime` `parseTime` (FILE \*file)  
*file*
- `bcf_directory_entry` \* `CompoundFileDirectoryEntry` (FILE \*file)  
*file*
- void `readNextSector` (FILE \*file, `bcf_directory` \*dir)  
*file dir*
- void `bcf_directory_free` (`bcf_directory` \*\*dir)  
*dir*
- `bcf_file_fat` \* `bcfFileFat_create` (const unsigned int `sectorSize`)  
*sectorSize*
- void `loadFatFromSector` (`bcf_file_fat` \*fat, FILE \*file)  
*fat file*
- `bcf_file_header` `bcfFileHeader_read` (FILE \*file)

- file*
- void `embSatinOutline_generateSatinOutline` (`EmbArray` \*lines, `EmbReal` thickness, `EmbSatinOutline` \*result)
- lines thickness result*
- `EmbArray` \* `embSatinOutline_renderStitches` (`EmbSatinOutline` \*result, `EmbReal` density)
- result density*
- void `write_24bit` (`FILE` \*file, int x)
- file x*
- int `embColor_distance` (`EmbColor` a, `EmbColor` b)
- a b*
- void `embColor_read` (`FILE` \*f, `EmbColor` \*c, int toRead)
- f c toRead*
- void `embColor_write` (`FILE` \*f, `EmbColor` c, int toWrite)
- f c toWrite*
- int `embThread_findNearestColor` (`EmbColor` color, `EmbColor` \*color\_list, int n\_colors)
- int `embThread_findNearestThread` (`EmbColor` color, `EmbThread` \*thread\_list, int n\_threads)
- color thread\_list n\_threads*
- `EmbThread` `embThread_getRandom` (void)
- void `binaryReadString` (`FILE` \*file, char \*buffer, int maxLength)
- file buffer maxLength*
- void `binaryReadUnicodeString` (`FILE` \*file, char \*buffer, const int stringLength)
- file buffer stringLength*
- int `stringInArray` (const char \*s, const char \*\*array)
- int `emb_readline` (`FILE` \*file, char \*line, int maxLength)
- file line maxLength*
- void `get_trim_bounds` (char const \*s, char const \*\*firstWord, char const \*\*trailingSpace)
- Get the trim bounds object.*
- char \* `copy_trim` (char const \*s)
- s*
- char \* `emb_optOut` (`EmbReal` num, char \*str)
- Optimizes the number (num) for output to a text file and returns it as a string (str).*
- void `embTime_initNow` (`EmbTime` \*t)
- t*
- `EmbTime` `embTime_time` (`EmbTime` \*t)
- t*

## Variables

- `EmbThread` `black_thread` = { { 0, 0, 0 }, "Black", "Black" }
- int `emb_verbose` = 0
- Verbosity level.*
- int `emb_error` = 0
- Error code storage for optional control flow blocking.*
- const `EmbReal` `embConstantPi` = 3.1415926535
- const unsigned int `difatEntriesInHeader` = 109
- const unsigned int `sizeOfFatEntry` = sizeof(unsigned int)
- const unsigned int `sizeOfDifatEntry` = 4
- const unsigned int `sizeOfChainingEntryAtEndOfDifatSector` = 4
- const unsigned int `sizeOfDirectoryEntry` = 128
- char const `WHITESPACE` [] = "\t\n\r"

### 18.109.1 Macro Definition Documentation

**18.109.1.1 FLAG\_CIRCLE** `#define FLAG_CIRCLE 12`

**18.109.1.2 FLAG\_CIRCLE\_SHORT** `#define FLAG_CIRCLE_SHORT 13`

**18.109.1.3 FLAG\_COMBINE** `#define FLAG_COMBINE 35`

**18.109.1.4 FLAG\_CROSS\_STITCH** `#define FLAG_CROSS_STITCH 36`

**18.109.1.5 FLAG\_ELLIPSE** `#define FLAG_ELLIPSE 14`

**18.109.1.6 FLAG\_ELLIPSE\_SHORT** `#define FLAG_ELLIPSE_SHORT 15`

**18.109.1.7 FLAG\_FILL** `#define FLAG_FILL 32`

**18.109.1.8 FLAG\_FILL\_SHORT** `#define FLAG_FILL_SHORT 33`

**18.109.1.9 FLAG\_FORMATS** `#define FLAG_FORMATS 4`

**18.109.1.10 FLAG\_FORMATS\_SHORT** `#define FLAG_FORMATS_SHORT 5`

**18.109.1.11 FLAG\_FULL\_TEST\_SUITE** `#define FLAG_FULL_TEST_SUITE 29`

**18.109.1.12 FLAG\_HELP** `#define FLAG_HELP 2`

**18.109.1.13 FLAG\_HELP\_SHORT** `#define FLAG_HELP_SHORT 3`

**18.109.1.14 FLAG\_HILBERT\_CURVE** `#define FLAG_HILBERT_CURVE 30`

**18.109.1.15 FLAG\_LINE** `#define FLAG_LINE 16`

**18.109.1.16 FLAG\_LINE\_SHORT** `#define FLAG_LINE_SHORT 17`

**18.109.1.17 FLAG\_POLYGON** `#define FLAG_POLYGON 18`

**18.109.1.18 FLAG\_POLYGON\_SHORT** `#define FLAG_POLYGON_SHORT 19`



**18.109.1.19 FLAG\_POLYLINE** `#define FLAG_POLYLINE 20`

**18.109.1.20 FLAG\_POLYLINE\_SHORT** `#define FLAG_POLYLINE_SHORT 21`

**18.109.1.21 FLAG\_QUIET** `#define FLAG_QUIET 6`

**18.109.1.22 FLAG\_QUIET\_SHORT** `#define FLAG_QUIET_SHORT 7`

**18.109.1.23 FLAG\_RENDER** `#define FLAG_RENDER 22`

**18.109.1.24 FLAG\_RENDER\_SHORT** `#define FLAG_RENDER_SHORT 23`

**18.109.1.25 FLAG\_SATIN** `#define FLAG_SATIN 24`

**18.109.1.26 FLAG\_SATIN\_SHORT** `#define FLAG_SATIN_SHORT 25`

**18.109.1.27 FLAG\_SIERPINSKI\_TRIANGLE** `#define FLAG_SIERPINSKI_TRIANGLE 31`

**18.109.1.28 FLAG\_SIMULATE** `#define FLAG_SIMULATE 34`

**18.109.1.29 FLAG\_STITCH** `#define FLAG_STITCH 26`

**18.109.1.30 FLAG\_STITCH\_SHORT** `#define FLAG_STITCH_SHORT 27`

**18.109.1.31 FLAG\_TEST** `#define FLAG_TEST 28`

**18.109.1.32 FLAG\_TO** `#define FLAG_TO 0`

**18.109.1.33 FLAG\_TO\_SHORT** `#define FLAG_TO_SHORT 1`

**18.109.1.34 FLAG\_VERBOSE** `#define FLAG_VERBOSE 8`

**18.109.1.35 FLAG\_VERBOSE\_SHORT** `#define FLAG_VERBOSE_SHORT 9`

**18.109.1.36 FLAG\_VERSION** `#define FLAG_VERSION 10`

**18.109.1.37 FLAG\_VERSION\_SHORT** `#define FLAG_VERSION_SHORT 11`

**18.109.1.38 NUM\_FLAGS** `#define NUM_FLAGS 37`

## 18.109.2 Function Documentation

**18.109.2.1 bcf\_diffat\_create()** `bcf_file_difat * bcf_difat_create (`  
    `FILE * file,`  
    `unsigned int fatSectors,`  
    `const unsigned int sectorSize )`  
*file fatSectors sectorSize*

Returns

`bcf_file_difat*`

**18.109.2.2 bcf\_directory\_free()** `void bcf_directory_free (`  
    `bcf_directory ** dir )`  
*dir*

**18.109.2.3 bcf\_file\_free()** `void bcf_file_free (`  
    `bcf_file * bcfFile )`  
*bcfFile*

**18.109.2.4 bcfFile\_read()** `int bcfFile_read (`  
    `FILE * file,`  
    `bcf_file * bcfFile )`  
*file bcfFile*

Returns

`int`

**18.109.2.5 bcfFileFat\_create()** `bcf_file_fat * bcfFileFat_create (`  
    `const unsigned int sectorSize )`  
*sectorSize*

Returns

`bcf_file_fat*`

**18.109.2.6 bcfFileHeader\_read()** `bcf_file_header bcfFileHeader_read (`  
    `FILE * file )`  
*file*

Returns

`bcf_file_header`

**18.109.2.7 binaryReadString()** void binaryReadString (

```
FILE * file,
char * buffer,
int maxLength )
```

*file buffer maxLength*

**18.109.2.8 binaryReadUnicodeString()** void binaryReadUnicodeString (

```
FILE * file,
char * buffer,
const int stringLength )
```

*file buffer stringLength*

**18.109.2.9 check\_header\_present()** int check\_header\_present (

```
FILE * file,
int minimum_header_length )
```

*file minimum\_header\_length*

#### Returns

int

Checks that there are enough bytes to interpret the header, stops possible segfaults when reading in the header bytes.

Returns 0 if there aren't enough, or the length of the file if there are.

**18.109.2.10 CompoundFileDirectory()** bcf\_directory \* CompoundFileDirectory (

```
const unsigned int maxNumberOfDirectoryEntries )
```

*maxNumberOfDirectoryEntries*

#### Returns

bcf\_directory\*

**18.109.2.11 CompoundFileDirectoryEntry()** bcf\_directory\_entry \* CompoundFileDirectoryEntry (

```
FILE * file )
```

*file*

#### Returns

bcf\_directory\_entry\*

**18.109.2.12 copy\_trim()** char \* copy\_trim (

```
char const * s )
```

*s*

#### Returns

char\*

**Todo** decription

**18.109.2.13 emb\_optOut()** char \* emb\_optOut (   
     EmbReal num,   
     char \* str )

Optimizes the number (*num*) for output to a text file and returns it as a string (*str*).  
*num str*

Returns

char\*

**18.109.2.14 emb\_readline()** int emb\_readline (   
     FILE \* file,   
     char \* line,   
     int maxLength )

*file line maxLength*

Returns

int

**18.109.2.15 embArc\_print()** void embArc\_print (   
     EmbArc arc )

*arc*

**Todo** move to [arc.c](#)

**18.109.2.16 embColor\_distance()** int embColor\_distance (   
     EmbColor a,   
     EmbColor b )

*a b*

Returns

int

**18.109.2.17 embColor\_read()** void embColor\_read (   
     FILE \* f,   
     EmbColor \* c,   
     int toRead )

*f c toRead*

**18.109.2.18 embColor\_write()** void embColor\_write (   
     FILE \* f,   
     EmbColor c,   
     int toWrite )

*f c toWrite*

**18.109.2.19 embSatinOutline\_generateSatinOutline()** void embSatinOutline\_generateSatinOutline (   
     EmbArray \* lines,   
     EmbReal thickness,   
     EmbSatinOutline \* result )

*lines thickness result*

**18.109.2.20 embSatinOutline\_renderStitches()** `EmbArray * embSatinOutline_renderStitches (`  
`EmbSatinOutline * result,`  
`EmbReal density )`

*result density*

Returns

EmbArray\*

**18.109.2.21 embThread\_findNearestColor()** `int embThread_findNearestColor (`  
`EmbColor color,`  
`EmbColor * color_list,`  
`int n_colors )`

Returns the closest color to the required color based on a list of available threads. The algorithm is a simple least squares search against the list. If the (square of) Euclidean 3-dimensional distance between the points in (red, green, blue) space is smaller then the index is saved and the remaining index is returned to the caller.

*color* The EmbColor color to match. *colors* The EmbThreadList pointer to start the search at. *mode* Is the argument an array of threads (0) or colors (1)?

Returns

closestIndex The entry in the ThreadList that matches.

**18.109.2.22 embThread\_findNearestThread()** `int embThread_findNearestThread (`  
`EmbColor color,`  
`EmbThread * thread_list,`  
`int n_threads )`

*color thread\_list n\_threads*

Returns

int

**18.109.2.23 embThread\_getRandom()** `EmbThread embThread_getRandom (`  
`void )`

Returns a random thread color, useful in filling in cases where the actual color of the thread doesn't matter but one needs to be declared to test or render a pattern.

Returns

c The resulting color.

**18.109.2.24 embTime\_initNow()** `void embTime_initNow (`  
`EmbTime * t )`

*t*

**18.109.2.25 embTime\_time()** `EmbTime embTime_time (`  
`EmbTime * t )`

*t*

Returns

EmbTime

**18.109.2.26 embVector\_print()** void embVector\_print (  
    EmbVector v,  
    char \* label )

*v label*

move to [vector.c](#)

**18.109.2.27 entriesInDifatSector()** unsigned int entriesInDifatSector (  
    bcf\_file\_difat \* fat )

*fat*

Returns

unsigned int

**18.109.2.28 get\_trim\_bounds()** void get\_trim\_bounds (  
    char const \* s,  
    char const \*\* firstWord,  
    char const \*\* trailingSpace )

Get the trim bounds object.

*s firstWord trailingSpace*

**18.109.2.29 GetFile()** FILE \* GetFile (  
    bcf\_file \* bcfFile,  
    FILE \* file,  
    char \* fileToFind )

Get the File object.

*bcfFile file fileToFind*

Returns

FILE\*

**18.109.2.30 haveExtraDIFATSectors()** int haveExtraDIFATSectors (  
    bcf\_file \* file )

*file*

Returns

int

**18.109.2.31 loadFatFromSector()** void loadFatFromSector (  
    bcf\_file\_fat \* fat,  
    FILE \* file )

*fat file*

**18.109.2.32 parseDIFATSectors()** void parseDIFATSectors (  
    FILE \* file,  
    bcf\_file \* bcfFile )

*file bcfFile*

**18.109.2.33 parseDirectoryEntryName()** void parseDirectoryEntryName (  
    FILE \* file,  
    bcf\_directory\_entry \* dir )

*file dir*

**18.109.2.34 parseTime()** `EmbTime parseTime (`  
    `FILE * file )`

*file*

Returns

EmbTime

**18.109.2.35 readFullSector()** `unsigned int readFullSector (`  
    `FILE * file,`  
    `bcf_file_difat * bcfFile,`  
    `unsigned int * difatEntriesToRead )`

*file bcfFile difatEntriesToRead*

Returns

unsigned int

**18.109.2.36 readNextSector()** `void readNextSector (`  
    `FILE * file,`  
    `bcf_directory * dir )`

*file dir*

**18.109.2.37 sectorSize()** `unsigned int sectorSize (`  
    `bcf_file * bcfFile )`

*bcfFile*

Returns

unsigned int

**18.109.2.38 seekToSector()** `int seekToSector (`  
    `bcf_file * bcfFile,`  
    `FILE * file,`  
    `const unsigned int sector )`

*bcfFile file sector*

Returns

int

**18.109.2.39 stringInArray()** `int stringInArray (`  
    `const char * s,`  
    `const char ** array )`

Tests for the presence of a string *s* in the supplied *array*.  
The end of the array is marked by an empty string.

Returns

0 if not present 1 if present.

**18.109.2.40 write\_24bit()** void write\_24bit (   
FILE \* *file*,   
int *x* )   
*file x*

### 18.109.3 Variable Documentation

**18.109.3.1 black\_thread** EmbThread black\_thread = { { 0, 0, 0 }, "Black", "Black" }

**18.109.3.2 difatEntriesInHeader** const unsigned int difatEntriesInHeader = 109

**18.109.3.3 emb\_error** int emb\_error = 0   
Error code storage for optional control flow blocking.

**18.109.3.4 emb\_verbose** int emb\_verbose = 0   
Verbosity level.

**18.109.3.5 embConstantPi** const EmbReal embConstantPi = 3.1415926535

**18.109.3.6 sizeOfChainingEntryAtEndOfDifatSector** const unsigned int sizeOfChainingEntryAtEndOfDifatSector = 4

**18.109.3.7 sizeOfDifatEntry** const unsigned int sizeOfDifatEntry = 4

**18.109.3.8 sizeOfDirectoryEntry** const unsigned int sizeOfDirectoryEntry = 128

**18.109.3.9 sizeOfFatEntry** const unsigned int sizeOfFatEntry = sizeof(unsigned int)

**18.109.3.10 WHITESPACE** char const WHITESPACE[] = " \t\n\r"

## 18.110 extern/libembroidery/src/pattern.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "embroidery_internal.h"
```

### Functions

- EmbPattern \* embPattern\_create (void)
- void embPattern\_hideStitchesOverLength (EmbPattern \*p, int length)
- int embPattern\_addThread (EmbPattern \*pattern, EmbThread thread)
- void embPattern\_fixColorCount (EmbPattern \*p)



- void `embPattern_copystitch_listToPolylines` (`EmbPattern *p`)
- void `embPattern_copyPolylinesToStitch_list` (`EmbPattern *p`)
- void `embPattern_movestitch_listToPolylines` (`EmbPattern *p`)
- void `embPattern_movePolylinesToStitch_list` (`EmbPattern *p`)
- void `embPattern_addStitchAbs` (`EmbPattern *p`, `EmbReal x`, `EmbReal y`, `int flags`, `int isAutoColorIndex`)
- void `embPattern_addStitchRel` (`EmbPattern *p`, `EmbReal dx`, `EmbReal dy`, `int flags`, `int isAutoColorIndex`)
- void `embPattern_changeColor` (`EmbPattern *p`, `int index`)
- void `embPattern_scale` (`EmbPattern *p`, `EmbReal scale`)
- `EmbRect` `embPattern_calcBoundingBox` (`EmbPattern *p`)
- void `embPattern_flipHorizontal` (`EmbPattern *p`)
- void `embPattern_flipVertical` (`EmbPattern *p`)
- void `embPattern_flip` (`EmbPattern *p`, `int horz`, `int vert`)
- void `embPattern_combineJumpStitches` (`EmbPattern *p`)
- void `embPattern_correctForMaxStitchLength` (`EmbPattern *p`, `EmbReal maxStitchLength`, `EmbReal maxJumpLength`)
- void `embPattern_center` (`EmbPattern *p`)
- void `embPattern_loadExternalColorFile` (`EmbPattern *p`, `const char *fileName`)
- void `embPattern_free` (`EmbPattern *p`)
- void `embPattern_addCircleAbs` (`EmbPattern *p`, `EmbCircle circle`)
- void `embPattern_addEllipseAbs` (`EmbPattern *p`, `EmbEllipse ellipse`)
- void `embPattern_addLineAbs` (`EmbPattern *p`, `EmbLine line`)
- void `embPattern_addPathAbs` (`EmbPattern *p`, `EmbPath obj`)
- void `embPattern_addPointAbs` (`EmbPattern *p`, `EmbPoint obj`)
- void `embPattern_addPolygonAbs` (`EmbPattern *p`, `EmbPolygon obj`)
- void `embPattern_addPolylineObjectAbs` (`EmbPattern *p`, `EmbPolyline obj`)
- void `embPattern_addRectAbs` (`EmbPattern *p`, `EmbRect rect`)
- void `embPattern_end` (`EmbPattern *p`)
- int `embPattern_color_count` (`EmbPattern *pattern`, `EmbColor startColor`)
- void `embPattern_designDetails` (`EmbPattern *pattern`)
- int `convert` (`const char *inf`, `const char *outf`)
- float `embPattern_totalStitchLength` (`EmbPattern *pattern`)
- float `embPattern_minimumStitchLength` (`EmbPattern *pattern`)
- float `embPattern_maximumStitchLength` (`EmbPattern *pattern`)
- void `embPattern_lengthHistogram` (`EmbPattern *pattern`, `int *bin`, `int NUMBINS`)
- int `embPattern_realStitches` (`EmbPattern *pattern`)
- int `embPattern_jumpStitches` (`EmbPattern *pattern`)
- int `embPattern_trimStitches` (`EmbPattern *pattern`)

### 18.110.1 Detailed Description

The file is for the management of the main struct: `EmbPattern`.

### 18.110.2 Function Documentation

**18.110.2.1 `convert()`** `int convert (`  
`const char * inf,`  
`const char * outf )`

**18.110.2.2 `embPattern_addCircleAbs()`** `void embPattern_addCircleAbs (`  
`EmbPattern * p,`  
`EmbCircle circle )`

Adds a circle object to pattern (`p`) with its center at the absolute position (`cx,cy`) with a radius of (`r`). Positive `y` is up. Units are in millimeters.

**18.110.2.3 embPattern\_addEllipseAbs()** void embPattern\_addEllipseAbs (   
     EmbPattern \* p,   
     EmbEllipse ellipse )

Adds an ellipse object to pattern (*p*) with its center at the absolute position (*cx,cy*) with radii of (*rx,ry*). Positive y is up. Units are in millimeters.

**18.110.2.4 embPattern\_addLineAbs()** void embPattern\_addLineAbs (   
     EmbPattern \* p,   
     EmbLine line )

Adds a line object to pattern (*p*) starting at the absolute position (*x1,y1*) and ending at the absolute position (*x2,y2*). Positive y is up. Units are in millimeters.

**18.110.2.5 embPattern\_addPathAbs()** void embPattern\_addPathAbs (   
     EmbPattern \* p,   
     EmbPath obj )

**18.110.2.6 embPattern\_addPointAbs()** void embPattern\_addPointAbs (   
     EmbPattern \* p,   
     EmbPoint obj )

Adds a point object to pattern (*p*) at the absolute position (*x,y*). Positive y is up. Units are in millimeters.

**18.110.2.7 embPattern\_addPolygonAbs()** void embPattern\_addPolygonAbs (   
     EmbPattern \* p,   
     EmbPolygon obj )

**18.110.2.8 embPattern\_addPolylineObjectAbs()** void embPattern\_addPolylineObjectAbs (   
     EmbPattern \* p,   
     EmbPolyline obj )

**18.110.2.9 embPattern\_addRectAbs()** void embPattern\_addRectAbs (   
     EmbPattern \* p,   
     EmbRect rect )

Adds a rectangle object to pattern (*p*) at the absolute position (*x,y*) with a width of (*w*) and a height of (*h*). Positive y is up. Units are in millimeters.

**18.110.2.10 embPattern\_addStitchAbs()** void embPattern\_addStitchAbs (   
     EmbPattern \* p,   
     EmbReal x,   
     EmbReal y,   
     int flags,   
     int isAutoColorIndex )

Adds a stitch to the pattern (*p*) at the absolute position (*x,y*). Positive y is up. Units are in millimeters.

**18.110.2.11 embPattern\_addStitchRel()** void embPattern\_addStitchRel (   
     EmbPattern \* p,   
     EmbReal dx,   
     EmbReal dy,   
     int flags,   
     int isAutoColorIndex )

Adds a stitch to the pattern (*p*) at the relative position (*dx,dy*) to the previous stitch. Positive y is up. Units are in millimeters.

**18.110.2.12 `embPattern_addThread()`** `int embPattern_addThread (`  
     `EmbPattern * pattern,`  
     `EmbThread thread )`

*pattern thread*

Returns

int

**18.110.2.13 `embPattern_calcBoundingBox()`** `EmbRect embPattern_calcBoundingBox (`  
     `EmbPattern * p )`

Returns an EmbRect that encapsulates all stitches and objects in the pattern (*p*).

**18.110.2.14 `embPattern_center()`** `void embPattern_center (`  
     `EmbPattern * p )`

Center the pattern *p*.

**18.110.2.15 `embPattern_changeColor()`** `void embPattern_changeColor (`  
     `EmbPattern * p,`  
     `int index )`

Change the currentColorIndex of pattern *p* to *index*.

**18.110.2.16 `embPattern_color_count()`** `int embPattern_color_count (`  
     `EmbPattern * pattern,`  
     `EmbColor startColor )`

**18.110.2.17 `embPattern_combineJumpStitches()`** `void embPattern_combineJumpStitches (`  
     `EmbPattern * p )`

*p*

**18.110.2.18 `embPattern_copyPolylinesToStitch_list()`** `void embPattern_copyPolylinesToStitch_list (`  
     `EmbPattern * p )`

Copies all of the EmbPolylineObjectList data to Embstitch\_list data for pattern (*p*).

**18.110.2.19 `embPattern_copystitch_listToPolylines()`** `void embPattern_copystitch_listToPolylines (`  
     `EmbPattern * p )`

Copies all of the Embstitch\_list data to EmbPolylineObjectList data for pattern (*p*).

**18.110.2.20 `embPattern_correctForMaxStitchLength()`** `void embPattern_correctForMaxStitchLength (`  
     `EmbPattern * p,`  
     `EmbReal maxStitchLength,`  
     `EmbReal maxJumpLength )`

**Todo** The params determine the max XY movement rather than the length. They need renamed or clarified further.

**18.110.2.21 `embPattern_create()`** `EmbPattern * embPattern_create (`  
     `void )`

Returns a pointer to an EmbPattern. It is created on the heap. The caller is responsible for freeing the allocated memory with `embPattern_free()`.

Returns

EmbPattern\*

**18.110.2.22 embPattern\_designDetails()** void embPattern\_designDetails (   
     EmbPattern \* pattern )

**18.110.2.23 embPattern\_end()** void embPattern\_end (   
     EmbPattern \* p )

**18.110.2.24 embPattern\_fixColorCount()** void embPattern\_fixColorCount (   
     EmbPattern \* p )

*p*

**18.110.2.25 embPattern\_flip()** void embPattern\_flip (   
     EmbPattern \* p,   
     int horz,   
     int vert )

Flips the entire pattern (*p*) horizontally about the x-axis if (*horz*) is true. Flips the entire pattern (*p*) vertically about the y-axis if (*vert*) is true.

**18.110.2.26 embPattern\_flipHorizontal()** void embPattern\_flipHorizontal (   
     EmbPattern \* p )

Flips the entire pattern (*p*) horizontally about the y-axis.

**18.110.2.27 embPattern\_flipVertical()** void embPattern\_flipVertical (   
     EmbPattern \* p )

Flips the entire pattern (*p*) vertically about the x-axis.

**18.110.2.28 embPattern\_free()** void embPattern\_free (   
     EmbPattern \* p )

Frees all memory allocated in the pattern (*p*).

**18.110.2.29 embPattern\_hideStitchesOverLength()** void embPattern\_hideStitchesOverLength (   
     EmbPattern \* p,   
     int length )

*p length*

**18.110.2.30 embPattern\_jumpStitches()** int embPattern\_jumpStitches (   
     EmbPattern \* pattern )

**18.110.2.31 embPattern\_lengthHistogram()** void embPattern\_lengthHistogram (   
     EmbPattern \* pattern,   
     int \* bin,   
     int NUMBINS )

**18.110.2.32 embPattern\_loadExternalColorFile()** void embPattern\_loadExternalColorFile (   
     EmbPattern \* p,   
     const char \* fileName )

TODO: Description needed.

**18.110.2.33 embPattern\_maximumStitchLength()** float embPattern\_maximumStitchLength (   
     EmbPattern \* pattern )

**18.110.2.34** `embPattern_minimumStitchLength()` `float embPattern_minimumStitchLength ( EmbPattern * pattern )`

**18.110.2.35** `embPattern_movePolylinesTostitch_list()` `void embPattern_movePolylinesTostitch_list ( EmbPattern * p )`

Moves all of the EmbPolylineObjectList data to Embstitch\_list data for pattern (*p*).

**18.110.2.36** `embPattern_movestitch_listToPolylines()` `void embPattern_movestitch_listToPolylines ( EmbPattern * p )`

Moves all of the Embstitch\_list data to EmbPolylineObjectList data for pattern (*p*).

**18.110.2.37** `embPattern_realStitches()` `int embPattern_realStitches ( EmbPattern * pattern )`

**18.110.2.38** `embPattern_scale()` `void embPattern_scale ( EmbPattern * p, EmbReal scale )`

Very simple scaling of the x and y axis for every point. Doesn't insert or delete stitches to preserve density.

**18.110.2.39** `embPattern_totalStitchLength()` `float embPattern_totalStitchLength ( EmbPattern * pattern )`

*pattern*

Returns

float

**18.110.2.40** `embPattern_trimStitches()` `int embPattern_trimStitches ( EmbPattern * pattern )`

## 18.111 extern/libembroidery/src/thread-color.c File Reference

```
#include <stdio.h>
#include <string.h>
#include "embroidery_internal.h"
```

### Functions

- int `threadColor` (const char \*name, int brand)
- int `threadColorNum` (unsigned int color, int brand)
- const char \* `threadColorName` (unsigned int color, int brand)

### Variables

- const unsigned char `_dxfColorTable` [][3] = {{ 0, 0, 0 }}
- const `EmbThread husThreads` [] = {{{ 0, 0, 0 }, "END", "END"}}
- const `EmbThread jefThreads` [] = {{{ 0, 0, 0 }, "END", "END"}}
- const `EmbThread shvThreads` [] = {{{ 0, 0, 0 }, "END", "END"}}
- const `EmbThread pcmThreads` [] = {{{ 0, 0, 0 }, "END", "END"}}
- const `EmbThread pecThreads` [] = {{{ 0, 0, 0 }, "END", "END"}}
- const int `shvThreadCount` = 42
- const int `pecThreadCount` = 65
- `thread_color` \* `brand_codes` []
- const char \* `brand_codes_files` []

### 18.111.1 Function Documentation

**18.111.1.1 threadColor()** `int threadColor (`  
    `const char * name,`  
    `int brand )`

**18.111.1.2 threadColorName()** `const char * threadColorName (`  
    `unsigned int color,`  
    `int brand )`

**18.111.1.3 threadColorNum()** `int threadColorNum (`  
    `unsigned int color,`  
    `int brand )`

### 18.111.2 Variable Documentation

**18.111.2.1 \_dxfColorTable** `const unsigned char _dxfColorTable[][3] = {{ 0, 0, 0 }}`

**18.111.2.2 brand\_codes** `thread_color* brand_codes[]`

**18.111.2.3 brand\_codes\_files** `const char* brand_codes_files[]`

Initial value:

```
= {  
    "arc_polyester_colors.csv",  
    "arc_rayon_colors.csv",  
    "coats_and_clark_rayon_colors.csv",  
    "exquisite_polyester_colors.csv",  
    "fufu_Polyester_colors.csv",  
    "fufu_Rayon_colors.csv",  
    "Hemingworth_Polyester_colors.csv",  
    "Isacord_Polyester_colors.csv",  
    "Isafil_Rayon_colors.csv",  
    "Marathon_Polyester_colors.csv",  
    "Marathon_Rayon_colors.csv",  
    "Madeira_Polyester_colors.csv",  
    "Madeira_Rayon_colors.csv",  
    "Metro_Polyester_colors.csv",  
    "Pantone_colors.csv",  
    "RobisonAnton_Polyester_colors.csv",  
    "RobisonAnton_Rayon_colors.csv",  
    "Sigma_Polyester_colors.csv",  
    "Sulky_Rayon_colors.csv",  
    "ThreadArt_Rayon_colors.csv",  
    "ThreadArt_Polyester_colors.csv",  
    "ThreaDelight_Polyester_colors.csv",  
    "Z102_Isacord_Polyester_colors.csv",  
    "svg_color_colors.csv"  
}
```

**18.111.2.4 husThreads** `const EmbThread husThreads[] = {{{ 0, 0, 0 }, "END", "END"}}`

**18.111.2.5 jefThreads** `const EmbThread jefThreads[] = {{{ 0, 0, 0 }, "END", "END"}}`

**18.111.2.6 pcmThreads** `const EmbThread pcmThreads[] = {{{ 0, 0, 0 }, "END", "END"}}`

**18.111.2.7 pecThreadCount** `const int pecThreadCount = 65`

**18.111.2.8 pecThreads** `const EmbThread pecThreads[] = {{{ 0, 0, 0 }, "END", "END"}}`

**18.111.2.9 shvThreadCount** `const int shvThreadCount = 42`

**18.111.2.10 shvThreads** `const EmbThread shvThreads[] = {{{ 0, 0, 0 }, "END", "END"}}`

## 18.112 privacy\_policy.md File Reference

## References

- [1] acatina. Technical info. [422](#)
- [2] KDE Community. Projects/liberty/file formats/tajima ternary - kde community wiki. [422](#)
- [3] G. van Rossum and B. Warsaw. Python pep 7. [12](#)



## Index

- `__mainWin`
    - Application, [51](#)
- `_appVer_`
  - embroidermodder.cpp, [231](#)
- `_bcf_directory,` [40](#)
  - dirEntries, [40](#)
  - maxNumberOfDirectoryEntries, [40](#)
- `_bcf_directory_entry,` [41](#)
  - childId, [41](#)
  - CLSID, [41](#)
  - colorFlag, [41](#)
  - creationTime, [41](#)
  - directoryEntryName, [41](#)
  - directoryEntryNameLength, [41](#)
  - leftSiblingId, [42](#)
  - modifiedTime, [42](#)
  - next, [42](#)
  - objectType, [42](#)
  - rightSiblingId, [42](#)
  - startingSectorLocation, [42](#)
  - stateBits, [42](#)
  - streamSize, [42](#)
  - streamSizeHigh, [42](#)
- `_bcf_file,` [42](#)
  - difat, [43](#)
  - directory, [43](#)
  - fat, [43](#)
  - header, [43](#)
- `_bcf_file_difat,` [43](#)
  - fatSectorCount, [43](#)
  - fatSectorEntries, [44](#)
  - sectorSize, [44](#)
- `_bcf_file_fat,` [44](#)
  - fatEntries, [44](#)
  - fatEntryCount, [44](#)
  - numberOfEntriesInFatSector, [44](#)
- `_bcf_file_header,` [44](#)
  - byteOrder, [45](#)
  - CLSID, [45](#)
  - firstDifatSectorLocation, [45](#)
  - firstDirectorySectorLocation, [45](#)
  - firstMiniFATSectorLocation, [45](#)
  - majorVersion, [45](#)
  - miniSectorShift, [46](#)
  - miniStreamCutoffSize, [46](#)
  - minorVersion, [46](#)
  - numberOfDifatSectors, [46](#)
  - numberOfDirectorySectors, [46](#)
  - numberOfFAT Sectors, [46](#)
  - numberOfMiniFatSectors, [46](#)
  - reserved1, [46](#)
  - reserved2, [46](#)
  - sectorShift, [46](#)
  - signature, [46](#)
  - transactionSignatureNumber, [47](#)
- `_dxfColorTable`
  - embroidery.h, [347](#)
  - thread-color.c, [488](#)
- `_mainWin`
  - embroidermodder.h, [246](#)
  - mainwindow.cpp, [302](#)
- `_subMask`
  - format\_csd.c, [419](#)
- `_vp3Hoop,` [47](#)
  - bottom, [47](#)
  - bottom2, [47](#)
  - byte1, [47](#)
  - byte2, [48](#)
  - byte3, [48](#)
  - height, [48](#)
  - left, [48](#)
  - left2, [48](#)
  - numberOfBytesRemaining, [48](#)
  - numberOfColors, [48](#)
  - right, [48](#)
  - right2, [48](#)
  - threadLength, [48](#)
  - top, [48](#)
  - top2, [49](#)
  - unknown2, [49](#)
  - unknown3, [49](#)
  - unknown4, [49](#)
  - width, [49](#)
  - xOffset, [49](#)
  - yOffset, [49](#)
- `_xorMask`
  - format\_csd.c, [419](#)
- `~CmdPrompt`
  - CmdPrompt, [53](#)
- `~CmdPromptHandle`
  - CmdPromptHandle, [61](#)
- `~CmdPromptHistory`
  - CmdPromptHistory, [64](#)
- `~CmdPromptInput`
  - CmdPromptInput, [68](#)
- `~CmdPromptSplitter`
  - CmdPromptSplitter, [75](#)
- `~EmbDetailsDialog`
  - EmbDetailsDialog, [83](#)
- `~Geometry`
  - Geometry, [112](#)
- `~ImageWidget`
  - ImageWidget, [140](#)
- `~LayerManager`
  - LayerManager, [142](#)
- `~MainWindow`
  - MainWindow, [148](#)
- `~MdiArea`
  - MdiArea, [164](#)
- `~MdiWindow`

- MdiWindow, 169
- ~PreviewDialog
  - PreviewDialog, 178
- ~PropertyEditor
  - PropertyEditor, 180
- ~SaveObject
  - SaveObject, 186
- ~Settings\_Dialog
  - Settings\_Dialog, 199
- ~UndoEditor
  - UndoEditor, 214
- ~View
  - View, 218
- 10o, 9, 414
- 100, 9, 414
- about
  - MainWindow, 148
- about\_action
  - mainwindow.cpp, 280
- accept\_
  - settings-dialog.cpp, 308
- acceptChanges
  - Settings\_Dialog, 200
- actionHash
  - embroidermodder.h, 246
  - mainwindow.cpp, 302
- activeCommand
  - MainWindow, 148
- activeMdiWindow
  - MainWindow, 148
- activeScene
  - embroidermodder.h, 237
  - mainwindow.cpp, 280
- activeUndoStack
  - MainWindow, 148
- activeView
  - embroidermodder.h, 237
  - mainwindow.cpp, 280
- actuator
  - embroidermodder.h, 237
  - mainwindow.cpp, 280
- add\_arc\_action
  - mainwindow.cpp, 281
- add\_circle\_action
  - mainwindow.cpp, 281
- add\_dim\_leader\_action
  - mainwindow.cpp, 281
- add\_ellipse\_action
  - mainwindow.cpp, 281
- add\_geometry\_action
  - mainwindow.cpp, 281
- add\_horizontal\_dimension\_action
  - mainwindow.cpp, 282
- add\_image\_action
  - mainwindow.cpp, 282
- add\_infinite\_line\_action
  - mainwindow.cpp, 282
- add\_line\_action
  - mainwindow.cpp, 282
- add\_path\_action
  - mainwindow.cpp, 282
- add\_point\_action
  - mainwindow.cpp, 282
- add\_polygon\_action
  - mainwindow.cpp, 283
- add\_polyline
  - embroidermodder.h, 238
  - objects.cpp, 305
- add\_polyline\_action
  - mainwindow.cpp, 283
- add\_ray\_action
  - mainwindow.cpp, 283
- add\_rectangle\_action
  - mainwindow.cpp, 283
- add\_regular\_polygon\_action
  - mainwindow.cpp, 283
- add\_rounded\_rectangle\_action
  - mainwindow.cpp, 283
- add\_rubber\_action
  - mainwindow.cpp, 284
- add\_slot\_action
  - mainwindow.cpp, 284
- add\_text\_multi\_action
  - mainwindow.cpp, 284
- add\_text\_single\_action
  - mainwindow.cpp, 284
- add\_to\_path
  - embroidermodder.h, 238
  - interface.cpp, 267
- add\_to\_selection\_action
  - mainwindow.cpp, 284
- add\_triangle\_action
  - mainwindow.cpp, 286
- add\_vertical\_dimension\_action
  - mainwindow.cpp, 286
- addArc
  - SaveObject, 187
- addBlock
  - SaveObject, 187
- addCircle
  - SaveObject, 187
- addColorsToComboBox
  - Settings\_Dialog, 200
- addDimAligned
  - SaveObject, 187
- addDimAngular
  - SaveObject, 188
- addDimArcLength
  - SaveObject, 188
- addDimDiameter
  - SaveObject, 188
- addDimLeader
  - SaveObject, 188
- addDimLinear
  - SaveObject, 189

- addDimOrdinate
  - SaveObject, [189](#)
- addDimRadius
  - SaveObject, [189](#)
- addEllipse
  - SaveObject, [190](#)
- addEllipseArc
  - SaveObject, [190](#)
- addGrid
  - SaveObject, [190](#)
- addHatch
  - SaveObject, [190](#)
- addImage
  - SaveObject, [191](#)
- addInfiniteLine
  - SaveObject, [191](#)
- addLayer
  - LayerManager, [142](#)
- addLine
  - SaveObject, [191](#)
- addObject
  - View, [218](#)
- addPath
  - SaveObject, [191](#)
- addPoint
  - SaveObject, [192](#)
- addPolygon
  - SaveObject, [192](#)
- addPolyline
  - SaveObject, [192](#)
- addRay
  - SaveObject, [193](#)
- addRectangle
  - SaveObject, [193](#)
- addSlot
  - SaveObject, [193](#)
- addSpline
  - SaveObject, [193](#)
- addStack
  - UndoEditor, [214](#)
- addTextMulti
  - SaveObject, [194](#)
- addTextSingle
  - SaveObject, [194](#)
- addToRubberRoom
  - View, [218](#)
- after
  - UndoableCommand, [212](#)
- alert
  - CmdPrompt, [53](#)
- alert\_action
  - mainwindow.cpp, [286](#)
- alignScenePointWithViewPoint
  - View, [218](#)
- allGripPoints
  - Geometry, [113](#)
- allow\_rubber\_action
  - mainwindow.cpp, [286](#)
- allowRubber
  - View, [219](#)
- allowZoomIn
  - View, [219](#)
- allowZoomOut
  - View, [219](#)
- alpha
  - SelectBox, [197](#)
- alphabet
  - LSYSTEM, [143](#)
- Ameco, [384](#), [435](#)
- angle
  - UndoableCommand, [212](#)
- append\_history\_action
  - mainwindow.cpp, [286](#)
- append\_prompt\_history\_action
  - mainwindow.cpp, [287](#)
- appendHistory
  - CmdPrompt, [53](#)
  - CmdPromptHistory, [64](#)
  - CmdPromptInput, [68](#)
- appendTheHistory
  - CmdPrompt, [53](#)
- Application, [49](#)
  - \_\_mainWin, [51](#)
  - Application, [50](#)
  - event, [50](#)
  - setMainWin, [51](#)
- applyFormatting
  - CmdPromptHistory, [64](#)
  - CmdPromptInput, [68](#)
- arc
  - EmbGeometry\_, [87](#)
- arc.c
  - Arc\_clockwise, [458](#)
  - Base\_objectRubberPoint, [458](#)
  - Base\_objectRubberText, [459](#)
  - Base\_setLineType, [459](#)
  - Base\_setLineWeight, [459](#)
  - clockwise, [459](#)
  - embArc\_arcLength, [459](#)
  - embArc\_area, [459](#)
  - embArc\_chord, [459](#)
  - embArc\_clockwise, [459](#)
  - embArc\_endAngle, [459](#)
  - embArc\_gripEdit, [459](#)
  - embArc\_includedAngle, [459](#)
  - embArc\_init, [459](#)
  - embArc\_mouseSnapPoint, [460](#)
  - embArc\_paint, [460](#)
  - embArc\_setCenter, [460](#)
  - embArc\_setEndAngle, [460](#)
  - embArc\_setRadius, [460](#)
  - embArc\_setStartAngle, [460](#)
  - embArc\_startAngle, [460](#)
  - embArc\_updatePath, [460](#)
  - embArc\_updateRubber, [460](#)
  - embBase\_setColorRGB, [460](#)

- embCircle\_prompt, 460
- embCircle\_setArea, 460
- embCircle\_setCircumference, 461
- embEllipse\_click, 461
- embEllipse\_main, 461
- embRect\_bottomLeft, 461
- embRect\_bottomRight, 461
- getArcCenter, 461
- getArcDataFromBulge, 461
- set\_object\_color, 461
- Arc\_clockwise
  - arc.c, 458
- Arc\_Polyester
  - embroidery.h, 322
- Arc\_Rayon
  - embroidery.h, 322
- arcEndPoint
  - Geometry, 135
- arcMidPoint
  - Geometry, 135
- arcStartPoint
  - Geometry, 135
- array.c
  - embArray\_addArc, 311
  - embArray\_addCircle, 311
  - embArray\_addEllipse, 311
  - embArray\_addFlag, 311
  - embArray\_addLine, 311
  - embArray\_addPath, 312
  - embArray\_addPoint, 312
  - embArray\_addPolygon, 312
  - embArray\_addPolyline, 312
  - embArray\_addRect, 312
  - embArray\_addStitch, 312
  - embArray\_addVector, 312
  - embArray\_copy, 312
  - embArray\_create, 312
  - embArray\_free, 312
  - embArray\_resize, 312
- ArrowStyle
  - Geometry, 108
- arrowStyleAngle
  - Geometry, 135
- arrowStyleLength
  - Geometry, 135
- arrowStylePath
  - Geometry, 135
- art, 9, 415
- attributeList
  - format\_svg.c, 448
- attributeOffset
  - VipHeader\_, 229
- AutoCAD, 387, 425, 444
- AutoDesk, 425
- auxFormat
  - ThredExtension\_, 209
- axiom
  - LSYSTEM, 144
- b
  - EmbColor\_, 82
  - Node\_, 177
  - Barudan, 388, 421, 452
  - Base\_objectRubberPoint
    - arc.c, 458
  - Base\_objectRubberText
    - arc.c, 459
  - Base\_setLineType
    - arc.c, 459
  - Base\_setLineWeight
    - arc.c, 459
  - bcf\_difat\_create
    - embroidery\_internal.h, 372
    - main.c, 476
  - bcf\_directory
    - embroidery\_internal.h, 371
  - bcf\_directory\_entry
    - embroidery\_internal.h, 371
  - bcf\_directory\_free
    - embroidery\_internal.h, 373
    - main.c, 476
  - bcf\_file
    - embroidery\_internal.h, 371
  - bcf\_file\_difat
    - embroidery\_internal.h, 371
  - bcf\_file\_difat\_free
    - embroidery\_internal.h, 373
  - bcf\_file\_fat
    - embroidery\_internal.h, 371
  - bcf\_file\_fat\_free
    - embroidery\_internal.h, 373
  - bcf\_file\_free
    - embroidery\_internal.h, 373
    - main.c, 476
  - bcf\_file\_header
    - embroidery\_internal.h, 371
  - bcfFile\_read
    - embroidery\_internal.h, 373
    - main.c, 476
  - bcfFileFat\_create
    - embroidery\_internal.h, 373
    - main.c, 476
  - bcfFileHeader\_isValid
    - embroidery\_internal.h, 373
  - bcfFileHeader\_read
    - embroidery\_internal.h, 373
    - main.c, 476
  - before
    - UndoableCommand, 212
  - Bernina, 415
  - beziers
    - EmbSpline\_, 98
  - bgColor
    - MdiArea, 166
  - bgLogo
    - MdiArea, 166
  - bgTexture

- MdiArea, 166
- binaryReadString
  - embroidery\_internal.h, 373
  - main.c, 476
- binaryReadUnicodeString
  - embroidery\_internal.h, 374
  - main.c, 477
- binaryWriteInt
  - embroidery\_internal.h, 374
  - formats.c, 411
- binaryWriteIntBE
  - embroidery\_internal.h, 374
  - formats.c, 411
- binaryWriteShort
  - embroidery\_internal.h, 374
  - formats.c, 411
- binaryWriteUInt
  - embroidery\_internal.h, 374
  - formats.c, 411
- binaryWriteUIntBE
  - embroidery\_internal.h, 374
  - formats.c, 411
- binaryWriteUShort
  - embroidery\_internal.h, 374
  - formats.c, 411
- binaryWriteUShortBE
  - embroidery\_internal.h, 375
  - formats.c, 411
- bit\_position
  - Compress, 76
- Bitmap Cache, 416
- Bits and Volts, 416
- bits\_total
  - Compress, 76
- black\_thread
  - embroidery.h, 347
  - main.c, 482
- blink
  - CmdPrompt, 54
- blink\_prompt\_action
  - mainwindow.cpp, 287
- blinkState
  - CmdPrompt, 59
- blinkTimer
  - CmdPrompt, 59
- block\_elements
  - Compress, 76
- bmc, 416
- BOOL\_TYPE
  - embroidermodder.h, 235
- bottom
  - \_vp3Hoop, 47
  - EmbRect\_, 97
  - hoop\_padding, 138
- bottom2
  - \_vp3Hoop, 47
- boundingRect
  - EmbDetailsDialog, 84
- Geometry, 113
- Box
  - Geometry, 109
- boxDir
  - SelectBox, 197
- brand\_codes
  - thread-color.c, 488
- brand\_codes\_files
  - thread-color.c, 488
- bro, 9, 416
- Brother, 385, 386, 439, 440, 443, 444
- BuildDecryptionTable
  - format\_csd.c, 418
- BULGETOCONTROL
  - embroidery\_internal.h, 363
- BULGETOEND
  - embroidery\_internal.h, 363
- buttonBox
  - EmbDetailsDialog, 84
  - Settings\_Dialog, 205
- buttonCustomFilterClearAll
  - Settings\_Dialog, 200
- buttonCustomFilterClearAllClicked
  - Settings\_Dialog, 200
- buttonCustomFilterSelectAll
  - Settings\_Dialog, 200
- buttonCustomFilterSelectAllClicked
  - Settings\_Dialog, 200
- buttonQSnapClearAll
  - Settings\_Dialog, 200
- buttonQSnapClearAllClicked
  - Settings\_Dialog, 200
- buttonQSnapSelectAll
  - Settings\_Dialog, 200
- buttonQSnapSelectAllClicked
  - Settings\_Dialog, 200
- buttons
  - StatusBar, 207
- buttonTipOfTheDayClicked
  - MainWindow, 149
- byte1
  - \_vp3Hoop, 47
- byte2
  - \_vp3Hoop, 48
- byte3
  - \_vp3Hoop, 48
- byteOrder
  - \_bcf\_file\_header, 45
- calculate\_angle\_action
  - mainwindow.cpp, 287
- calculate\_distance\_action
  - mainwindow.cpp, 287
- calculateArcData
  - Geometry, 113
- canRedo
  - UndoEditor, 214
- canUndo
  - UndoEditor, 214

- cascade
  - MdiArea, 164
- catalogNumber
  - EmbThread\_, 101
- cci
  - format\_dst.c, 423
- center
  - EmbCircle\_, 81
  - EmbEllipse\_, 85
  - View, 219
- centerAt
  - View, 219
- changeFormatting
  - CmdPromptInput, 68
- changelog\_action
  - mainwindow.cpp, 287
- character\_huffman
  - Compress, 76
- character\_length\_huffman
  - Compress, 76
- check\_for\_color\_file
  - EmbFormatList\_, 86
- check\_header\_present
  - embroidery\_internal.h, 375
  - main.c, 477
- checkBoxCustomFilterStateChanged
  - Settings\_Dialog, 200
- checkBoxes
  - embroidermodder.h, 246
  - mainwindow.cpp, 302
- checkBoxGeneralMdiBGUseColorStateChanged
  - Settings\_Dialog, 200
- checkBoxGeneralMdiBGUseLogoStateChanged
  - Settings\_Dialog, 200
- checkBoxGeneralMdiBGUseTextureStateChanged
  - Settings\_Dialog, 200
- checkBoxGridCenterOnOriginStateChanged
  - Settings\_Dialog, 201
- checkBoxGridColorMatchCrossHairStateChanged
  - Settings\_Dialog, 201
- checkBoxGridLoadFromFileStateChanged
  - Settings\_Dialog, 201
- checkBoxLwtRealRenderStateChanged
  - Settings\_Dialog, 201
- checkBoxLwtShowLwtStateChanged
  - Settings\_Dialog, 201
- checkBoxPromptSaveHistoryAsHtmlStateChanged
  - Settings\_Dialog, 201
- checkBoxRulerShowOnLoadStateChanged
  - Settings\_Dialog, 201
- checkBoxShowScrollBarsStateChanged
  - Settings\_Dialog, 201
- checkBoxTipOfTheDay
  - mainwindow.cpp, 302
- checkChangedText
  - CmdPromptInput, 68
- checkCursorPosition
  - CmdPromptInput, 68
- checkEditedText
  - CmdPromptInput, 69
- checkForUpdates
  - MainWindow, 149
- checkSelection
  - CmdPromptInput, 69
- childId
  - \_bcf\_directory\_entry, 41
- chooseDisplayBackgroundColor
  - Settings\_Dialog, 201
- chooseDisplayCrossHairColor
  - Settings\_Dialog, 201
- chooseDisplaySelectBoxLeftColor
  - Settings\_Dialog, 201
- chooseDisplaySelectBoxLeftFill
  - Settings\_Dialog, 201
- chooseDisplaySelectBoxRightColor
  - Settings\_Dialog, 202
- chooseDisplaySelectBoxRightFill
  - Settings\_Dialog, 202
- chooseGeneralMdiBackgroundColor
  - Settings\_Dialog, 202
- chooseGeneralMdiBackgroundLogo
  - Settings\_Dialog, 202
- chooseGeneralMdiBackgroundTexture
  - Settings\_Dialog, 202
- chooseGridColor
  - Settings\_Dialog, 202
- choosePromptBackgroundColor
  - Settings\_Dialog, 202
- choosePromptTextColor
  - Settings\_Dialog, 202
- chooseRulerColor
  - Settings\_Dialog, 202
- CHUNK\_SIZE
  - embroidery.h, 322
- circle
  - EmbGeometry\_, 87
- circle.c
  - embCircle\_area, 462
  - embCircle\_circumference, 462
  - embCircle\_init, 462
  - getCircleCircleIntersections, 462
  - getCircleTangentPoints, 462
- circle\_click
  - Geometry, 113
- clear\_rubber\_action
  - mainwindow.cpp, 288
- clear\_selection\_action
  - mainwindow.cpp, 288
- clearAllFields
  - PropertyEditor, 180
- clearFormatting
  - CmdPromptInput, 69
- clearRubberRoom
  - View, 219
- clearSelection
  - View, 219

- clockwise
  - arc.c, [459](#)
- Closed
  - Geometry, [109](#)
- closeEvent
  - MainWindow, [149](#)
  - MdiWindow, [169](#)
- closest\_point
  - objects.cpp, [305](#)
- closeToolBar
  - MainWindow, [149](#)
- CLSID
  - \_bcf\_directory\_entry, [41](#)
  - \_bcf\_file\_header, [45](#)
- cmdActive
  - CmdPromptInput, [73](#)
- CmdPrompt, [51](#)
  - ~CmdPrompt, [53](#)
  - alert, [53](#)
  - appendHistory, [53](#)
  - appendTheHistory, [53](#)
  - blink, [54](#)
  - blinkState, [59](#)
  - blinkTimer, [59](#)
  - CmdPrompt, [53](#)
  - copyPressed, [54](#)
  - cutPressed, [54](#)
  - deletePressed, [54](#)
  - downPressed, [54](#)
  - escapePressed, [54](#)
  - F10Pressed, [54](#)
  - F11Pressed, [54](#)
  - F12Pressed, [54](#)
  - F1Pressed, [54](#)
  - F2Pressed, [54](#)
  - F3Pressed, [55](#)
  - F4Pressed, [55](#)
  - F5Pressed, [55](#)
  - F6Pressed, [55](#)
  - F7Pressed, [55](#)
  - F8Pressed, [55](#)
  - F9Pressed, [55](#)
  - floatingChanged, [55](#)
  - historyAppended, [55](#)
  - pastePressed, [56](#)
  - promptDivider, [59](#)
  - promptHistory, [59](#)
  - promptInput, [59](#)
  - promptSplitter, [59](#)
  - promptVBoxLayout, [59](#)
  - redoPressed, [56](#)
  - runCommand, [56](#)
  - saveHistory, [56](#)
  - selectAllPressed, [56](#)
  - setCurrentText, [56](#)
  - setHistory, [56](#)
  - setPrefix, [56](#)
  - setPromptBackgroundColor, [57](#)
  - setPromptFontFamily, [57](#)
  - setPromptFontSize, [57](#)
  - setPromptFontStyle, [57](#)
  - setPromptTextColor, [58](#)
  - shiftPressed, [58](#)
  - shiftReleased, [58](#)
  - showSettings, [58](#)
  - startBlinking, [58](#)
  - startCommand, [58](#)
  - stopBlinking, [58](#)
  - styleHash, [59](#)
  - tabPressed, [58](#)
  - undoPressed, [58](#)
  - updateStyle, [59](#)
  - upPressed, [59](#)
- CmdPromptHandle, [60](#)
  - ~CmdPromptHandle, [61](#)
  - CmdPromptHandle, [60](#)
  - handleMoved, [61](#)
  - handlePressed, [61](#)
  - handleReleased, [61](#)
  - mouseMoveEvent, [61](#)
  - mousePressEvent, [61](#)
  - mouseReleaseEvent, [62](#)
  - moveY, [62](#)
  - pressY, [62](#)
  - releaseY, [62](#)
- CmdPromptHistory, [62](#)
  - ~CmdPromptHistory, [64](#)
  - appendHistory, [64](#)
  - applyFormatting, [64](#)
  - CmdPromptHistory, [63](#)
  - contextMenuEvent, [64](#)
  - historyAppended, [65](#)
  - resizeHistory, [65](#)
  - startResizeHistory, [65](#)
  - stopResizeHistory, [65](#)
  - tmpHeight, [65](#)
- CmdPromptInput, [66](#)
  - ~CmdPromptInput, [68](#)
  - appendHistory, [68](#)
  - applyFormatting, [68](#)
  - changeFormatting, [68](#)
  - checkChangedText, [68](#)
  - checkCursorPosition, [68](#)
  - checkEditedText, [69](#)
  - checkSelection, [69](#)
  - clearFormatting, [69](#)
  - cmdActive, [73](#)
  - CmdPromptInput, [67](#)
  - contextMenuEvent, [69](#)
  - copyClip, [69](#)
  - copyPressed, [70](#)
  - curCmd, [73](#)
  - curText, [73](#)
  - cutPressed, [70](#)
  - defaultPrefix, [73](#)
  - deletePressed, [70](#)

- downPressed, [70](#)
- endCommand, [70](#)
- escapePressed, [70](#)
- eventFilter, [70](#)
- F10Pressed, [70](#)
- F11Pressed, [71](#)
- F12Pressed, [71](#)
- F1Pressed, [71](#)
- F2Pressed, [71](#)
- F3Pressed, [71](#)
- F4Pressed, [71](#)
- F5Pressed, [71](#)
- F6Pressed, [71](#)
- F7Pressed, [71](#)
- F8Pressed, [71](#)
- F9Pressed, [71](#)
- isBlinking, [74](#)
- lastCmd, [74](#)
- pasteClip, [72](#)
- pastePressed, [72](#)
- prefix, [74](#)
- processInput, [72](#)
- rapidFireEnabled, [74](#)
- redoPressed, [72](#)
- runCommand, [72](#)
- selectAllPressed, [72](#)
- shiftPressed, [72](#)
- shiftReleased, [72](#)
- showSettings, [72](#)
- startCommand, [72](#)
- stopBlinking, [73](#)
- tabPressed, [73](#)
- undoPressed, [73](#)
- updateCurrentText, [73](#)
- upPressed, [73](#)
- CmdPromptSplitter, [74](#)
  - ~CmdPromptSplitter, [75](#)
  - CmdPromptSplitter, [74](#)
  - createHandle, [75](#)
  - moveResizeHistory, [75](#)
  - pressResizeHistory, [75](#)
  - releaseResizeHistory, [75](#)
- cnd, [9](#), [417](#)
- CoatsAndClark\_Rayon
  - embroidery.h, [322](#)
- CODE\_OF\_CONDUCT.md, [230](#)
- col, [9](#), [417](#)
- color
  - EmbGeometry\_, [87](#)
  - EmbLine\_, [92](#)
  - EmbPath\_, [93](#)
  - EmbPoint\_, [95](#)
  - EmbStitch\_, [99](#)
  - EmbThread\_, [101](#)
- color\_only
  - EmbFormatList\_, [86](#)
- colorChanges
  - EmbDetailsDialog, [84](#)
- colorCode
  - StxThread\_, [207](#)
  - SubDescriptor\_, [208](#)
- colorFlag
  - \_bcf\_directory\_entry, [41](#)
- colorLength
  - VipHeader\_, [229](#)
- colorName
  - StxThread\_, [207](#)
  - SubDescriptor\_, [208](#)
- colorSelector
  - MainWindow, [161](#)
- colorSelectorIndexChanged
  - MainWindow, [149](#)
- colorTotal
  - EmbDetailsDialog, [84](#)
- comboBoxes
  - embroidermodder.h, [246](#)
  - mainwindow.cpp, [302](#)
- comboBoxGridTypeCurrentIndexChanged
  - Settings\_Dialog, [202](#)
- comboBoxIconSizeCurrentIndexChanged
  - Settings\_Dialog, [202](#)
- comboBoxPromptFontFamilyCurrentIndexChanged
  - Settings\_Dialog, [203](#)
- comboBoxPromptFontStyleCurrentIndexChanged
  - Settings\_Dialog, [203](#)
- comboBoxQSnapLocatorColorCurrentIndexChanged
  - Settings\_Dialog, [203](#)
- comboBoxRulerMetricCurrentIndexChanged
  - Settings\_Dialog, [203](#)
- comboBoxScrollBarWidgetCurrentIndexChanged
  - Settings\_Dialog, [203](#)
- comboBoxSelected
  - PropertyEditor, [183](#)
- comboBoxSelectionCoolGripColorCurrentIndexChanged
  - Settings\_Dialog, [203](#)
- comboBoxSelectionHotGripColorCurrentIndexChanged
  - Settings\_Dialog, [203](#)
- comboBoxTextSingleFont
  - property-editor.cpp, [306](#)
- Command
  - embroidermodder.h, [236](#)
- command
  - UndoableCommand, [212](#)
- command\_map
  - mainwindow.cpp, [303](#)
- CompoundFileDirectory
  - embroidery\_internal.h, [375](#)
  - main.c, [477](#)
- CompoundFileDirectoryEntry
  - embroidery\_internal.h, [375](#)
  - main.c, [477](#)
- CompoundFileSector\_DIFAT\_Sector
  - embroidery\_internal.h, [363](#)
- CompoundFileSector\_EndOfChain
  - embroidery\_internal.h, [363](#)
- CompoundFileSector\_FAT\_Sector



- embroidery\_internal.h, [364](#)
- CompoundFileSector\_FreeSector
  - embroidery\_internal.h, [364](#)
- CompoundFileSector\_MaxRegSector
  - embroidery\_internal.h, [364](#)
- CompoundFileStreamId\_MaxRegularStreamId
  - embroidery\_internal.h, [364](#)
- CompoundFileStreamId\_NoStream
  - embroidery\_internal.h, [364](#)
- Compress, [75](#)
  - bit\_position, [76](#)
  - bits\_total, [76](#)
  - block\_elements, [76](#)
  - character\_huffman, [76](#)
  - character\_length\_huffman, [76](#)
  - distance\_huffman, [76](#)
  - input\_data, [76](#)
  - input\_length, [76](#)
- compress
  - embroidery\_internal.h, [371](#)
- compress.c
  - compress\_get\_bits, [313](#)
  - compress\_get\_position, [313](#)
  - compress\_get\_token, [313](#)
  - compress\_init, [313](#)
  - compress\_load\_block, [314](#)
  - compress\_load\_character\_huffman, [314](#)
  - compress\_load\_character\_length\_huffman, [314](#)
  - compress\_load\_distance\_huffman, [314](#)
  - compress\_peek, [314](#)
  - compress\_pop, [314](#)
  - compress\_read\_variable\_length, [314](#)
  - huffman\_build\_table, [314](#)
  - huffman\_lookup, [314](#)
  - huffman\_lookup\_data, [315](#)
  - hus\_compress, [314](#)
  - hus\_decompress, [314](#)
- compress\_get\_bits
  - compress.c, [313](#)
  - embroidery\_internal.h, [375](#)
- compress\_get\_position
  - compress.c, [313](#)
  - embroidery\_internal.h, [375](#)
- compress\_get\_token
  - compress.c, [313](#)
  - embroidery\_internal.h, [375](#)
- compress\_init
  - compress.c, [313](#)
- compress\_load\_block
  - compress.c, [314](#)
  - embroidery\_internal.h, [376](#)
- compress\_load\_character\_huffman
  - compress.c, [314](#)
  - embroidery\_internal.h, [376](#)
- compress\_load\_character\_length\_huffman
  - compress.c, [314](#)
  - embroidery\_internal.h, [376](#)
- compress\_load\_distance\_huffman
  - compress.c, [314](#)
  - embroidery\_internal.h, [376](#)
- compress\_peek
  - compress.c, [314](#)
- compress\_pop
  - compress.c, [314](#)
  - embroidery\_internal.h, [376](#)
- compress\_read\_variable\_length
  - compress.c, [314](#)
  - embroidery\_internal.h, [376](#)
- config
  - embroidermodder.h, [246](#)
  - mainwindow.cpp, [303](#)
- config\_tables
  - embroidermodder.h, [246](#)
  - mainwindow.cpp, [303](#)
- constants
  - LSYSTEM, [144](#)
- construct\_command
  - embroidermodder.h, [238](#)
  - mainwindow.cpp, [288](#)
- contains
  - embroidermodder.h, [238](#)
  - view.cpp, [310](#)
- context\_menu\_action
  - StatusBar, [206](#)
- context\_menu\_event
  - StatusBar, [206](#)
- contextMenuEvent
  - CmdPromptHistory, [64](#)
  - CmdPromptInput, [69](#)
  - View, [219](#)
- control1
  - EmbBezier\_, [80](#)
- control2
  - EmbBezier\_, [80](#)
- convert
  - embroidery.h, [332](#)
  - pattern.c, [483](#)
- convert\_args\_to\_type
  - embroidermodder.h, [239](#)
  - mainwindow.cpp, [288](#)
- copy
  - View, [219](#)
- copy\_action
  - mainwindow.cpp, [288](#)
- copy\_selected\_action
  - mainwindow.cpp, [289](#)
- copy\_trim
  - embroidery\_internal.h, [376](#)
  - main.c, [477](#)
- copyClip
  - CmdPromptInput, [69](#)
- copyPressed
  - CmdPrompt, [54](#)
  - CmdPromptInput, [70](#)
- copySelected
  - View, [219](#)

- cornerButtonClicked
  - View, [219](#)
- count
  - EmbArray\_, [79](#)
- create\_checkbox
  - Settings\_Dialog, [203](#)
- create\_float\_spinbox
  - Settings\_Dialog, [203](#)
- create\_icon
  - MainWindow, [150](#)
- create\_menu
  - embroidermodder.h, [239](#)
  - mainwindow-menus.cpp, [273](#)
- create\_test\_file\_1
  - embroidery\_internal.h, [376](#)
- create\_test\_file\_2
  - embroidery\_internal.h, [376](#)
- create\_test\_file\_3
  - embroidery\_internal.h, [376](#)
- create\_toolbar
  - MainWindow, [150](#)
- createAllActions
  - MainWindow, [150](#)
- createAllMenus
  - MainWindow, [150](#)
- createAllToolbars
  - MainWindow, [151](#)
- createComboBoxSelected
  - PropertyEditor, [180](#)
- createGrid
  - View, [219](#)
- createGridIso
  - View, [219](#)
- createGridPolar
  - View, [219](#)
- createGridRect
  - View, [219](#)
- createGroupBox
  - PropertyEditor, [181](#)
- createHandle
  - CmdPromptSplitter, [75](#)
- createHistogram
  - EmbDetailsDialog, [83](#)
- createLineEdit
  - PropertyEditor, [181](#)
- createMainWidget
  - EmbDetailsDialog, [83](#)
- createObjectList
  - View, [219](#)
- createOrigin
  - View, [220](#)
- createRulerTextPath
  - View, [220](#)
- createTabDisplay
  - Settings\_Dialog, [203](#)
- createTabFilesPaths
  - Settings\_Dialog, [203](#)
- createTabGeneral
  - Settings\_Dialog, [204](#)
- createTabGridRuler
  - Settings\_Dialog, [204](#)
- createTabLineWeight
  - Settings\_Dialog, [204](#)
- createTabOpenSave
  - Settings\_Dialog, [204](#)
- createTabOrthoPolar
  - Settings\_Dialog, [204](#)
- createTabPrinting
  - Settings\_Dialog, [204](#)
- createTabPrompt
  - Settings\_Dialog, [204](#)
- createTabQuickSnap
  - Settings\_Dialog, [204](#)
- createTabQuickTrack
  - Settings\_Dialog, [204](#)
- createTabSelection
  - Settings\_Dialog, [204](#)
- createTabSnap
  - Settings\_Dialog, [204](#)
- createToolButton
  - PropertyEditor, [181](#)
- createToolButtonPickAdd
  - PropertyEditor, [181](#)
- createToolButtonQSelect
  - PropertyEditor, [181](#)
- creationTime
  - \_bcf\_directory\_entry, [41](#)
- creatorName
  - ThredExtension\_, [209](#)
- crosshairColor
  - View, [225](#)
- crosshairSize
  - View, [225](#)
- csd, [9](#), [418](#)
- csd\_decryptArray
  - format\_csd.c, [419](#)
- CsdSubMaskSize
  - format\_csd.c, [418](#)
- CsdXorMaskSize
  - format\_csd.c, [418](#)
- csv, [420](#)
- CSV\_EXPECT
  - embroidery\_internal.h, [372](#)
- CSV\_EXPECT\_COMMA
  - embroidery\_internal.h, [372](#)
- CSV\_EXPECT\_NULL
  - embroidery\_internal.h, [372](#)
- CSV\_EXPECT\_QUOTE1
  - embroidery\_internal.h, [372](#)
- CSV\_EXPECT\_QUOTE2
  - embroidery\_internal.h, [372](#)
- CSV\_MODE
  - embroidery\_internal.h, [372](#)
- CSV\_MODE\_COMMENT
  - embroidery\_internal.h, [372](#)
- CSV\_MODE\_NULL

- embroidery\_internal.h, 372
- CSV\_MODE\_STITCH
  - embroidery\_internal.h, 372
- CSV\_MODE\_THREAD
  - embroidery\_internal.h, 372
- CSV\_MODE\_VARIABLE
  - embroidery\_internal.h, 372
- csvStitchFlagToStr
  - format\_csv.c, 420
- csvStrToStitchFlag
  - format\_csv.c, 420
- CUBICTOCONTROL1
  - embroidery\_internal.h, 364
- CUBICTOCONTROL2
  - embroidery\_internal.h, 364
- CUBICTOEND
  - embroidery\_internal.h, 364
- curCmd
  - CmdPromptInput, 73
- currentColor
  - MdiWindow, 175
- curFile
  - MdiWindow, 175
- curLayer
  - MdiWindow, 175
- curLineType
  - MdiWindow, 176
- curLineWeight
  - MdiWindow, 176
- current\_element\_id
  - format\_svg.c, 448
- currentAttribute
  - format\_svg.c, 448
- currentColorChanged
  - MdiWindow, 169
- currentColorIndex
  - EmbPattern\_, 94
- currentDisplayBackgroundColorChanged
  - Settings\_Dialog, 204
- currentDisplayCrossHairColorChanged
  - Settings\_Dialog, 204
- currentDisplaySelectBoxLeftColorChanged
  - Settings\_Dialog, 204
- currentDisplaySelectBoxLeftFillChanged
  - Settings\_Dialog, 204
- currentDisplaySelectBoxRightColorChanged
  - Settings\_Dialog, 204
- currentDisplaySelectBoxRightFillChanged
  - Settings\_Dialog, 205
- currentGeneralMdiBackgroundColorChanged
  - Settings\_Dialog, 205
- currentGridColorChanged
  - Settings\_Dialog, 205
- currentLayerChanged
  - MdiWindow, 169
- currentLinetypeChanged
  - MdiWindow, 169
- currentLineweightChanged
  - MdiWindow, 171
- currentPromptBackgroundColorChanged
  - Settings\_Dialog, 205
- currentPromptTextColorChanged
  - Settings\_Dialog, 205
- currentRulerColorChanged
  - Settings\_Dialog, 205
- currentValue
  - format\_svg.c, 448
- curText
  - CmdPromptInput, 73
- curved
  - Geometry, 136
- cut
  - View, 220
- cut\_action
  - mainwindow.cpp, 289
- cut\_selected\_action
  - mainwindow.cpp, 289
- cutCopyMousePoint
  - View, 225
- cutCopyObjectList
  - MainWindow, 161
- cutPressed
  - CmdPrompt, 54
  - CmdPromptInput, 70
- d
  - em2\_dev\_script, 39
- dat, 9
- data
  - EmblImage\_, 89
- day
  - EmbTime\_, 101
- day\_vision\_action
  - mainwindow.cpp, 289
- debug\_action
  - mainwindow.cpp, 289
- debug\_message
  - embroidermodder.h, 239
  - interface.cpp, 267
- decode\_exy\_flags
  - format\_exy.c, 427
- decode\_record\_flags
  - format\_dst.c, 423
- decode\_t01\_record
  - embroidery\_internal.h, 376
  - encoding.c, 403
- decode\_tajima\_ternary
  - embroidery\_internal.h, 377
  - encoding.c, 403
- decode\_tap\_record\_flags
  - format\_tap.c, 450
- DecodeCsdByte
  - format\_csd.c, 419
- decodeNewStitch
  - embroidery\_internal.h, 377
  - encoding.c, 403
- default\_value

- Huffman, 139
- defaultPrefix
  - CmdPromptInput, 73
- degrees
  - embroidery.h, 332
  - functions.c, 464
- degrees\_\_\_
  - embroidermodder.h, 239
  - interface.cpp, 267
- delete\_selected\_action
  - mainwindow.cpp, 289
- deleteObject
  - View, 220
- deletePressed
  - CmdPrompt, 54
  - CmdPromptInput, 70
  - MainWindow, 151
  - MdiWindow, 171
  - View, 220
- deleteSelected
  - View, 220
- delta
  - UndoableCommand, 213
- dem, 9, 421
- description
  - EmbFormatList\_, 86
  - EmbThread\_, 101
- design\_details\_action
  - mainwindow.cpp, 289
- designDetails
  - MdiWindow, 171
- dialog
  - embroidermodder.h, 246
  - mainwindow.cpp, 303
- Dictionary
  - embroidermodder.h, 236
- difat
  - \_bcf\_file, 43
- difatEntriesInHeader
  - main.c, 482
- dimensions
  - EmblImage\_, 89
- dirBrush
  - SelectBox, 197
- directory
  - \_bcf\_file, 43
- directoryEntryName
  - \_bcf\_directory\_entry, 41
- directoryEntryNameLength
  - \_bcf\_directory\_entry, 41
- dirEntries
  - \_bcf\_directory, 40
- dirPen
  - SelectBox, 197
- disable\_action
  - mainwindow.cpp, 289
- display\_props
  - settings-dialog.cpp, 308
- distance\_huffman
  - Compress, 76
- do\_nothing\_action
  - mainwindow.cpp, 290
- docIndex
  - MainWindow, 161
- dockPropEdit
  - embroidermodder.h, 246
  - mainwindow.cpp, 303
- dockUndoEdit
  - embroidermodder.h, 246
  - mainwindow.cpp, 303
- done
  - UndoableCommand, 213
- Dot
  - Geometry, 109
- doubleSpinBoxes
  - embroidermodder.h, 246
  - mainwindow.cpp, 303
- downPressed
  - CmdPrompt, 54
  - CmdPromptInput, 70
- dragon\_curve
  - fill.c, 406
- drawBackground
  - View, 220
- drawForeground
  - View, 220
- drawRubberLine
  - Geometry, 113
- dsb, 9, 421
- dst, 9, 422
- dstJumpsPerTrim
  - EmbPattern\_, 94
- dsz, 9, 382, 424
- dx, 9, 425
- dx\_color
  - embroidery.h, 322
- DXF\_VERSION\_2000
  - embroidery\_internal.h, 364
- DXF\_VERSION\_2002
  - embroidery\_internal.h, 364
- DXF\_VERSION\_2004
  - embroidery\_internal.h, 364
- DXF\_VERSION\_2006
  - embroidery\_internal.h, 364
- DXF\_VERSION\_2007
  - embroidery\_internal.h, 364
- DXF\_VERSION\_2009
  - embroidery\_internal.h, 364
- DXF\_VERSION\_2010
  - embroidery\_internal.h, 364
- DXF\_VERSION\_2013
  - embroidery\_internal.h, 364
- DXF\_VERSION\_R10
  - embroidery\_internal.h, 364
- DXF\_VERSION\_R11
  - embroidery\_internal.h, 365

DXF\_VERSION\_R12  
    embroidery\_internal.h, [365](#)  
DXF\_VERSION\_R13  
    embroidery\_internal.h, [365](#)  
DXF\_VERSION\_R14  
    embroidery\_internal.h, [365](#)  
DXF\_VERSION\_R15  
    embroidery\_internal.h, [365](#)  
DXF\_VERSION\_R18  
    embroidery\_internal.h, [365](#)  
DXF\_VERSION\_R21  
    embroidery\_internal.h, [365](#)  
DXF\_VERSION\_R24  
    embroidery\_internal.h, [365](#)  
DXF\_VERSION\_R27  
    embroidery\_internal.h, [365](#)  
  
edr, [9](#), [382](#), [425](#)  
ELEMENT\_A  
    embroidery\_internal.h, [365](#)  
ELEMENT\_ANIMATE  
    embroidery\_internal.h, [365](#)  
ELEMENT\_ANIMATECOLOR  
    embroidery\_internal.h, [365](#)  
ELEMENT\_ANIMATEMOTION  
    embroidery\_internal.h, [365](#)  
ELEMENT\_ANIMATETRANSFORM  
    embroidery\_internal.h, [365](#)  
ELEMENT\_ANIMATION  
    embroidery\_internal.h, [365](#)  
ELEMENT\_AUDIO  
    embroidery\_internal.h, [365](#)  
ELEMENT\_CIRCLE  
    embroidery\_internal.h, [365](#)  
ELEMENT\_DEFS  
    embroidery\_internal.h, [365](#)  
ELEMENT\_DESC  
    embroidery\_internal.h, [366](#)  
ELEMENT\_DISCARD  
    embroidery\_internal.h, [366](#)  
ELEMENT\_ELLIPSE  
    embroidery\_internal.h, [366](#)  
ELEMENT\_FONT  
    embroidery\_internal.h, [366](#)  
ELEMENT\_FONT\_FACE  
    embroidery\_internal.h, [366](#)  
ELEMENT\_FONT\_FACE\_SRC  
    embroidery\_internal.h, [366](#)  
ELEMENT\_FONT\_FACE\_URI  
    embroidery\_internal.h, [366](#)  
ELEMENT\_FOREIGN\_OBJECT  
    embroidery\_internal.h, [366](#)  
ELEMENT\_G  
    embroidery\_internal.h, [366](#)  
ELEMENT\_GLYPH  
    embroidery\_internal.h, [366](#)  
ELEMENT\_HANDLER  
    embroidery\_internal.h, [366](#)  
ELEMENT\_HKERN  
    embroidery\_internal.h, [366](#)  
ELEMENT\_IMAGE  
    embroidery\_internal.h, [366](#)  
ELEMENT\_LINE  
    embroidery\_internal.h, [366](#)  
ELEMENT\_LINEAR\_GRADIENT  
    embroidery\_internal.h, [366](#)  
ELEMENT\_LISTENER  
    embroidery\_internal.h, [366](#)  
ELEMENT\_METADATA  
    embroidery\_internal.h, [366](#)  
ELEMENT\_MISSING\_GLYPH  
    embroidery\_internal.h, [366](#)  
ELEMENT\_MPATH  
    embroidery\_internal.h, [367](#)  
ELEMENT\_PATH  
    embroidery\_internal.h, [367](#)  
ELEMENT\_POLYGON  
    embroidery\_internal.h, [367](#)  
ELEMENT\_POLYLINE  
    embroidery\_internal.h, [367](#)  
ELEMENT\_PREFETCH  
    embroidery\_internal.h, [367](#)  
ELEMENT\_RADIAL\_GRADIENT  
    embroidery\_internal.h, [367](#)  
ELEMENT\_RECT  
    embroidery\_internal.h, [367](#)  
ELEMENT\_SCRIPT  
    embroidery\_internal.h, [367](#)  
ELEMENT\_SET  
    embroidery\_internal.h, [367](#)  
ELEMENT\_SOLID\_COLOR  
    embroidery\_internal.h, [367](#)  
ELEMENT\_STOP  
    embroidery\_internal.h, [367](#)  
ELEMENT\_SVG  
    embroidery\_internal.h, [367](#)  
ELEMENT\_SWITCH  
    embroidery\_internal.h, [367](#)  
ELEMENT\_TBREAK  
    embroidery\_internal.h, [367](#)  
ELEMENT\_TEXT  
    embroidery\_internal.h, [367](#)  
ELEMENT\_TEXT\_AREA  
    embroidery\_internal.h, [367](#)  
ELEMENT\_TITLE  
    embroidery\_internal.h, [367](#)  
ELEMENT\_TSPAN  
    embroidery\_internal.h, [367](#)  
ELEMENT\_USE  
    embroidery\_internal.h, [368](#)  
ELEMENT\_VIDEO  
    embroidery\_internal.h, [368](#)  
ELEMENT\_XML  
    embroidery\_internal.h, [368](#)  
ellipse  
    EmbGeometry\_, [87](#)  
ellipse.c

- ellipse\_objectQuadrant0, [463](#)
- ellipse\_objectQuadrant180, [463](#)
- ellipse\_objectQuadrant270, [463](#)
- ellipse\_objectQuadrant90, [463](#)
- embEllipse\_area, [463](#)
- embEllipse\_diameterX, [463](#)
- embEllipse\_diameterY, [463](#)
- embEllipse\_height, [463](#)
- embEllipse\_init, [463](#)
- embEllipse\_perimeter, [463](#)
- embEllipse\_setDiameterMajor, [463](#)
- embEllipse\_setDiameterMinor, [464](#)
- embEllipse\_setRadiusMajor, [464](#)
- embEllipse\_setRadiusMinor, [464](#)
- embEllipse\_setSize, [464](#)
- embEllipse\_updatePath, [464](#)
- embEllipse\_width, [464](#)
- ellipse\_objectQuadrant0
  - ellipse.c, [463](#)
- ellipse\_objectQuadrant180
  - ellipse.c, [463](#)
- ellipse\_objectQuadrant270
  - ellipse.c, [463](#)
- ellipse\_objectQuadrant90
  - ellipse.c, [463](#)
- ELLIPSETOEND
  - embroidery\_internal.h, [368](#)
- ELLIPSETORAD
  - embroidery\_internal.h, [368](#)
- Elna, [426](#)
- Eltac, [427](#)
- em2\_dev\_script, [39](#)
  - d, [39](#)
  - header, [39](#)
  - s, [40](#)
- EMB\_ARC
  - embroidery.h, [322](#)
- EMB\_ARRAY
  - embroidery.h, [322](#)
- EMB\_BIG\_ENDIAN
  - embroidery\_internal.h, [368](#)
- EMB\_CIRCLE
  - embroidery.h, [322](#)
- emb\_constant\_pi
  - embroidermodder.h, [246](#)
- EMB\_DIM\_DIAMETER
  - embroidery.h, [323](#)
- EMB\_DIM\_LEADER
  - embroidery.h, [323](#)
- EMB\_ELLIPSE
  - embroidery.h, [323](#)
- emb\_error
  - embroidery.h, [347](#)
  - main.c, [482](#)
- EMB\_FLAG
  - embroidery.h, [323](#)
- EMB\_FORMAT\_100
  - embroidery.h, [323](#)
- EMB\_FORMAT\_ART
  - embroidery.h, [323](#)
- EMB\_FORMAT\_BMC
  - embroidery.h, [323](#)
- EMB\_FORMAT\_BRO
  - embroidery.h, [323](#)
- EMB\_FORMAT\_CND
  - embroidery.h, [323](#)
- EMB\_FORMAT\_COL
  - embroidery.h, [323](#)
- EMB\_FORMAT\_CSD
  - embroidery.h, [323](#)
- EMB\_FORMAT\_CSV
  - embroidery.h, [323](#)
- EMB\_FORMAT\_DAT
  - embroidery.h, [323](#)
- EMB\_FORMAT\_DEM
  - embroidery.h, [323](#)
- EMB\_FORMAT\_DSB
  - embroidery.h, [323](#)
- EMB\_FORMAT\_DST
  - embroidery.h, [323](#)
- EMB\_FORMAT\_DSZ
  - embroidery.h, [323](#)
- EMB\_FORMAT\_DXF
  - embroidery.h, [324](#)
- EMB\_FORMAT\_EDR
  - embroidery.h, [324](#)
- EMB\_FORMAT\_EMD
  - embroidery.h, [324](#)
- EMB\_FORMAT\_EXP
  - embroidery.h, [324](#)
- EMB\_FORMAT\_EXY
  - embroidery.h, [324](#)
- EMB\_FORMAT\_EYS
  - embroidery.h, [324](#)
- EMB\_FORMAT\_FXY
  - embroidery.h, [324](#)
- EMB\_FORMAT\_GC
  - embroidery.h, [324](#)
- EMB\_FORMAT\_GNC
  - embroidery.h, [324](#)
- EMB\_FORMAT\_GT
  - embroidery.h, [324](#)
- EMB\_FORMAT\_HUS
  - embroidery.h, [324](#)
- EMB\_FORMAT\_INB
  - embroidery.h, [324](#)
- EMB\_FORMAT\_INF
  - embroidery.h, [324](#)
- EMB\_FORMAT\_JEF
  - embroidery.h, [324](#)
- EMB\_FORMAT\_KSM
  - embroidery.h, [324](#)
- EMB\_FORMAT\_MAX
  - embroidery.h, [324](#)

EMB\_FORMAT\_MIT  
embroidery.h, [324](#)

EMB\_FORMAT\_NEW  
embroidery.h, [324](#)

EMB\_FORMAT\_OFM  
embroidery.h, [325](#)

EMB\_FORMAT\_PCD  
embroidery.h, [325](#)

EMB\_FORMAT\_PCM  
embroidery.h, [325](#)

EMB\_FORMAT\_PCQ  
embroidery.h, [325](#)

EMB\_FORMAT\_PCS  
embroidery.h, [325](#)

EMB\_FORMAT\_PEC  
embroidery.h, [325](#)

EMB\_FORMAT\_PEL  
embroidery.h, [325](#)

EMB\_FORMAT\_PEM  
embroidery.h, [325](#)

EMB\_FORMAT\_PES  
embroidery.h, [325](#)

EMB\_FORMAT\_PHB  
embroidery.h, [325](#)

EMB\_FORMAT\_PHC  
embroidery.h, [325](#)

EMB\_FORMAT\_PLT  
embroidery.h, [325](#)

EMB\_FORMAT\_RGB  
embroidery.h, [325](#)

EMB\_FORMAT\_SEW  
embroidery.h, [325](#)

EMB\_FORMAT\_SHV  
embroidery.h, [325](#)

EMB\_FORMAT\_SST  
embroidery.h, [325](#)

EMB\_FORMAT\_STX  
embroidery.h, [325](#)

EMB\_FORMAT\_SVG  
embroidery.h, [325](#)

EMB\_FORMAT\_T01  
embroidery.h, [326](#)

EMB\_FORMAT\_T09  
embroidery.h, [326](#)

EMB\_FORMAT\_TAP  
embroidery.h, [326](#)

EMB\_FORMAT\_THR  
embroidery.h, [326](#)

EMB\_FORMAT\_TXT  
embroidery.h, [326](#)

EMB\_FORMAT\_U00  
embroidery.h, [326](#)

EMB\_FORMAT\_U01  
embroidery.h, [326](#)

EMB\_FORMAT\_VIP  
embroidery.h, [326](#)

EMB\_FORMAT\_VP3  
embroidery.h, [326](#)

EMB\_FORMAT\_XXX  
embroidery.h, [326](#)

EMB\_FORMAT\_ZSK  
embroidery.h, [326](#)

emb\_identify\_format  
embroidery.h, [332](#)  
formats.c, [412](#)

EMB\_IMAGE  
embroidery.h, [326](#)

EMB\_INT16\_BIG  
embroidery\_internal.h, [368](#)

EMB\_INT16\_LITTLE  
embroidery\_internal.h, [368](#)

EMB\_INT32\_BIG  
embroidery\_internal.h, [368](#)

EMB\_INT32\_LITTLE  
embroidery\_internal.h, [368](#)

EMB\_LINE  
embroidery.h, [326](#)

EMB\_LITTLE\_ENDIAN  
embroidery\_internal.h, [368](#)

EMB\_MAX  
embroidery\_internal.h, [368](#)

EMB\_MAX\_LAYERS  
embroidery.h, [326](#)

EMB\_MIN  
embroidery\_internal.h, [368](#)

emb\_optOut  
embroidery\_internal.h, [377](#)  
main.c, [477](#)

EMB\_PATH  
embroidery.h, [326](#)

EMB\_POINT  
embroidery.h, [326](#)

EMB\_POLYGON  
embroidery.h, [326](#)

EMB\_POLYLINE  
embroidery.h, [326](#)

EMB\_PUBLIC  
embroidery.h, [327](#)

emb\_readline  
embroidery\_internal.h, [377](#)  
main.c, [478](#)

EMB\_RECT  
embroidery.h, [327](#)

emb\_round  
embroidery.h, [332](#)  
functions.c, [464](#)

EMB\_SPLINE  
embroidery.h, [327](#)

EMB\_STITCH  
embroidery.h, [327](#)

EMB\_TEXT\_MULTI  
embroidery.h, [327](#)

EMB\_TEXT\_SINGLE  
embroidery.h, [327](#)

EMB\_THREAD  
embroidery.h, [327](#)

EMB\_VECTOR  
     embroidery.h, 327  
 emb\_verbose  
     embroidery.h, 347  
     main.c, 482  
 EmbAlignedDim  
     embroidery.h, 329  
 EmbAlignedDim\_, 77  
     position, 77  
 EmbAngularDim  
     embroidery.h, 329  
 EmbAngularDim\_, 77  
     position, 77  
 EmbArc  
     embroidery.h, 329  
 EmbArc\_, 78  
     end, 78  
     mid, 78  
     start, 78  
 embArc\_arcLength  
     arc.c, 459  
 embArc\_area  
     arc.c, 459  
 embArc\_chord  
     arc.c, 459  
 embArc\_clockwise  
     arc.c, 459  
     embroidery.h, 332  
 embArc\_endAngle  
     arc.c, 459  
 embArc\_gripEdit  
     arc.c, 459  
 embArc\_includedAngle  
     arc.c, 459  
 embArc\_init  
     arc.c, 459  
     embroidery.h, 332  
 embArc\_mouseSnapPoint  
     arc.c, 460  
 embArc\_paint  
     arc.c, 460  
 embArc\_print  
     main.c, 478  
 embArc\_setCenter  
     arc.c, 460  
 embArc\_setEndAngle  
     arc.c, 460  
 embArc\_setRadius  
     arc.c, 460  
 embArc\_setStartAngle  
     arc.c, 460  
 embArc\_startAngle  
     arc.c, 460  
 embArc\_updatePath  
     arc.c, 460  
 embArc\_updateRubber  
     arc.c, 460  
 EmbArcLengthDim  
     embroidery.h, 330  
 EmbArcLengthDim\_, 78  
     position, 79  
 EmbArray  
     embroidery.h, 330  
 EmbArray\_, 79  
     count, 79  
     geometry, 79  
     length, 79  
     stitch, 79  
     thread, 79  
     type, 80  
 embArray\_addArc  
     array.c, 311  
     embroidery.h, 332  
 embArray\_addCircle  
     array.c, 311  
     embroidery.h, 332  
 embArray\_addEllipse  
     array.c, 311  
     embroidery.h, 332  
 embArray\_addFlag  
     array.c, 311  
     embroidery.h, 332  
 embArray\_addLine  
     array.c, 311  
     embroidery.h, 333  
 embArray\_addPath  
     array.c, 312  
     embroidery.h, 333  
 embArray\_addPoint  
     array.c, 312  
     embroidery.h, 333  
 embArray\_addPolygon  
     array.c, 312  
     embroidery.h, 333  
 embArray\_addPolyline  
     array.c, 312  
     embroidery.h, 333  
 embArray\_addRect  
     array.c, 312  
     embroidery.h, 333  
 embArray\_addStitch  
     array.c, 312  
     embroidery.h, 333  
 embArray\_addThread  
     embroidery.h, 333  
 embArray\_addVector  
     array.c, 312  
     embroidery.h, 333  
 embArray\_copy  
     array.c, 312  
     embroidery.h, 333  
 embArray\_create  
     array.c, 312  
     embroidery.h, 333  
 embArray\_free  
     array.c, 312



- embroidery.h, 334
- embArray\_resize
  - array.c, 312
  - embroidery.h, 334
- embBase\_setColorRGB
  - arc.c, 460
- EmbBezier
  - embroidery.h, 330
- EmbBezier\_, 80
  - control1, 80
  - control2, 80
  - end, 80
  - start, 80
- EmbBlock
  - embroidery.h, 330
- EmbBlock\_, 81
  - position, 81
- EmbCircle
  - embroidery.h, 330
- EmbCircle\_, 81
  - center, 81
  - radius, 81
- embCircle\_area
  - circle.c, 462
- embCircle\_circumference
  - circle.c, 462
- embCircle\_init
  - circle.c, 462
  - embroidery.h, 334
- embCircle\_prompt
  - arc.c, 460
- embCircle\_setArea
  - arc.c, 460
- embCircle\_setCircumference
  - arc.c, 461
- EmbColor
  - embroidery.h, 330
- EmbColor\_, 81
  - b, 82
  - g, 82
  - r, 82
- embColor\_create
  - embroidery.h, 334
- embColor\_distance
  - embroidery.h, 334
  - main.c, 478
- embColor\_fromHexStr
  - embroidery.h, 334
  - encoding.c, 403
- embColor\_make
  - embroidery.h, 334
- embColor\_read
  - embroidery\_internal.h, 377
  - main.c, 478
- embColor\_write
  - embroidery\_internal.h, 377
  - main.c, 478
- embConstantPi
  - embroidery.h, 347
  - main.c, 482
- EmbDetailsDialog, 82
  - ~EmbDetailsDialog, 83
  - boundingRect, 84
  - buttonBox, 84
  - colorChanges, 84
  - colorTotal, 84
  - createHistogram, 83
  - createMainWidget, 83
  - EmbDetailsDialog, 83
  - getInfo, 83
  - mainWidget, 84
  - stitchesJump, 84
  - stitchesReal, 84
  - stitchesTotal, 84
  - stitchesTrim, 84
- EmbDiameterDim
  - embroidery.h, 330
- EmbDiameterDim\_, 85
  - position, 85
- EmbEllipse
  - embroidery.h, 330
- EmbEllipse\_, 85
  - center, 85
  - radius, 85
  - rotation, 85
- embEllipse\_area
  - ellipse.c, 463
  - embroidery.h, 334
- embEllipse\_click
  - arc.c, 461
- embEllipse\_diameterX
  - ellipse.c, 463
  - embroidery.h, 334
- embEllipse\_diameterY
  - ellipse.c, 463
  - embroidery.h, 334
- embEllipse\_height
  - ellipse.c, 463
  - embroidery.h, 335
- embEllipse\_init
  - ellipse.c, 463
  - embroidery.h, 335
- embEllipse\_main
  - arc.c, 461
- embEllipse\_make
  - embroidery.h, 335
- embEllipse\_perimeter
  - ellipse.c, 463
  - embroidery.h, 335
- embEllipse\_setDiameterMajor
  - ellipse.c, 463
- embEllipse\_setDiameterMinor
  - ellipse.c, 464
- embEllipse\_setRadiusMajor
  - ellipse.c, 464
- embEllipse\_setRadiusMinor

- ellipse.c, 464
- embEllipse\_setSize
  - ellipse.c, 464
- embEllipse\_updatePath
  - ellipse.c, 464
- embEllipse\_width
  - ellipse.c, 464
- embroidery.h, 335
- EmbFlag
  - embroidery.h, 330
- embFormat\_getExtension
  - formats.c, 412
- EMBFORMAT\_MAXDESC
  - embroidery.h, 327
- EMBFORMAT\_MAXEXT
  - embroidery.h, 327
- EMBFORMAT\_OBJECTONLY
  - embroidery.h, 327
- EMBFORMAT\_STCHANDOBJ
  - embroidery.h, 327
- EMBFORMAT\_STITCHONLY
  - embroidery.h, 327
- EMBFORMAT\_UNSUPPORTED
  - embroidery.h, 327
- EmbFormatList
  - embroidery.h, 330
- EmbFormatList\_, 86
  - check\_for\_color\_file, 86
  - color\_only, 86
  - description, 86
  - extension, 86
  - reader\_state, 86
  - type, 86
  - write\_external\_color\_file, 86
  - writer\_state, 86
- EmbGeometry
  - embroidery.h, 330
- EmbGeometry\_, 87
  - arc, 87
  - circle, 87
  - color, 87
  - ellipse, 87
  - flag, 88
  - line, 88
  - lineType, 88
  - object, 88
  - path, 88
  - point, 88
  - polygon, 88
  - polyline, 88
  - rect, 88
  - spline, 88
  - stitch, 88
  - thread, 89
  - type, 89
  - vector, 89
- embGeometry\_boundingRect
  - embroidery.h, 335
- geometry.c, 457
- embGeometry\_free
  - embroidery.h, 335
- geometry.c, 457
- embGeometry\_init
  - embroidery.h, 335
- geometry.c, 457
- embGeometry\_move
  - embroidery.h, 335
- geometry.c, 457
- embGeometry\_vulcanize
  - embroidery.h, 335
- geometry.c, 457
- EmblImage
  - embroidery.h, 330
- EmblImage\_, 89
  - data, 89
  - dimensions, 89
  - height, 89
  - name, 90
  - path, 90
  - position, 90
  - width, 90
- emblImage\_create
  - embroidery.h, 336
- emblImage\_free
  - embroidery.h, 336
- emblImage\_read
  - embroidery.h, 336
- emblImage\_write
  - embroidery.h, 336
- EmblInfiniteLine
  - embroidery.h, 330
- EmblInfiniteLine\_, 90
  - position, 90
- emblInt\_read
  - embroidery\_internal.h, 378
- encoding.c, 404
- emblInt\_write
  - embroidery\_internal.h, 378
- encoding.c, 404
- Embird, 382, 420, 425
- EmbLayer
  - embroidery.h, 330
- EmbLayer\_, 90
  - geometry, 91
  - name, 91
- EmbLeaderDim
  - embroidery.h, 330
- EmbLeaderDim\_, 91
  - position, 91
- EmbLine
  - embroidery.h, 330
- EmbLine\_, 91
  - color, 92
  - end, 92
  - lineType, 92
  - start, 92

embLine\_intersectionPoint  
  embroidery.h, 336  
  line.c, 465

embLine\_make  
  embroidery.h, 336

embLine\_normalVector  
  embroidery.h, 336  
  line.c, 465

embLine\_toVector  
  line.c, 465

EmbLinearDim  
  embroidery.h, 330

EmbLinearDim\_, 92  
  position, 92

EmbOrdinateDim  
  embroidery.h, 330

EmbOrdinateDim\_, 93  
  position, 93

EmbPath  
  embroidery.h, 331

EmbPath\_, 93  
  color, 93  
  flagList, 93  
  lineType, 93  
  pointList, 93

EmbPattern  
  embroidery.h, 331

EmbPattern\_, 94  
  currentColorIndex, 94  
  dstJumpsPerTrim, 94  
  geometry, 94  
  home, 94  
  hoop\_height, 94  
  hoop\_width, 94  
  layer, 95  
  stitch\_list, 95  
  thread\_list, 95

embPattern\_addCircleAbs  
  embroidery.h, 336  
  pattern.c, 483

embPattern\_addEllipseAbs  
  embroidery.h, 336  
  pattern.c, 483

embPattern\_addLineAbs  
  embroidery.h, 336  
  pattern.c, 484

embPattern\_addPathAbs  
  embroidery.h, 337  
  pattern.c, 484

embPattern\_addPointAbs  
  embroidery.h, 337  
  pattern.c, 484

embPattern\_addPolygonAbs  
  embroidery.h, 337  
  pattern.c, 484

embPattern\_addPolylineAbs  
  embroidery.h, 337

embPattern\_addPolylineObjectAbs  
  pattern.c, 484

embPattern\_addRectAbs  
  embroidery.h, 337  
  pattern.c, 484

embPattern\_addStitchAbs  
  embroidery.h, 337  
  pattern.c, 484

embPattern\_addStitchRel  
  embroidery.h, 337  
  pattern.c, 484

embPattern\_addThread  
  embroidery.h, 337  
  pattern.c, 484

embPattern\_calcBoundingBox  
  embroidery.h, 337  
  pattern.c, 485

embPattern\_center  
  embroidery.h, 338  
  pattern.c, 485

embPattern\_changeColor  
  embroidery.h, 338  
  pattern.c, 485

embPattern\_color\_count  
  embroidery.h, 338  
  pattern.c, 485

embPattern\_combine  
  embroidery.h, 338  
  fill.c, 406

embPattern\_combineJumpStitches  
  embroidery.h, 338  
  pattern.c, 485

embPattern\_convertGeometry  
  embroidery.h, 338  
  fill.c, 406

embPattern\_copyPolylinesToStitch\_list  
  pattern.c, 485

embPattern\_copyPolylinesToStitchList  
  embroidery.h, 338

embPattern\_copystitch\_listToPolylines  
  pattern.c, 485

embPattern\_copyStitchListToPolylines  
  embroidery.h, 338

embPattern\_correctForMaxStitchLength  
  embroidery.h, 338  
  pattern.c, 485

embPattern\_create  
  embroidery.h, 338  
  pattern.c, 485

embPattern\_crossstitch  
  embroidery.h, 339  
  fill.c, 406

embPattern\_designDetails  
  embroidery.h, 339  
  pattern.c, 485

embPattern\_end  
  embroidery.h, 339  
  pattern.c, 486

embPattern\_fixColorCount

- embroidery.h, 339
- pattern.c, 486
- embPattern\_flip
  - embroidery.h, 339
  - pattern.c, 486
- embPattern\_flipHorizontal
  - embroidery.h, 339
  - pattern.c, 486
- embPattern\_flipVertical
  - embroidery.h, 339
  - pattern.c, 486
- embPattern\_free
  - embroidery.h, 339
  - pattern.c, 486
- embPattern\_hideStitchesOverLength
  - embroidery.h, 339
  - pattern.c, 486
- embPattern\_horizontal\_fill
  - embroidery.h, 339
  - fill.c, 406
- embPattern\_jumpStitches
  - embroidery.h, 340
  - pattern.c, 486
- embPattern\_lengthHistogram
  - embroidery.h, 340
  - pattern.c, 486
- embPattern\_loadExternalColorFile
  - embroidery.h, 340
  - pattern.c, 486
- embPattern\_maximumStitchLength
  - embroidery.h, 340
  - pattern.c, 486
- embPattern\_minimumStitchLength
  - embroidery.h, 340
  - pattern.c, 486
- embPattern\_movePolylinesToStitch\_list
  - pattern.c, 487
- embPattern\_movePolylinesToStitchList
  - embroidery.h, 340
- embPattern\_movestitch\_listToPolylines
  - pattern.c, 487
- embPattern\_moveStitchListToPolylines
  - embroidery.h, 340
- embPattern\_read
  - embroidery.h, 340
  - formats.c, 412
- embPattern\_readAuto
  - embroidery.h, 340
  - formats.c, 412
- embPattern\_realStitches
  - embroidery.h, 341
  - pattern.c, 487
- embPattern\_render
  - embroidery.h, 341
- embPattern\_scale
  - embroidery.h, 341
  - pattern.c, 487
- embPattern\_simulate
  - embroidery.h, 341
- embPattern\_stitchArc
  - fill.c, 406
- embPattern\_stitchCircle
  - fill.c, 406
- embPattern\_stitchEllipse
  - fill.c, 407
- embPattern\_stitchPath
  - fill.c, 407
- embPattern\_stitchPolygon
  - fill.c, 407
- embPattern\_stitchPolyline
  - fill.c, 407
- embPattern\_stitchRect
  - fill.c, 408
- embPattern\_stitchText
  - fill.c, 408
- embPattern\_totalStitchLength
  - embroidery.h, 341
  - pattern.c, 487
- embPattern\_trimStitches
  - embroidery.h, 341
  - pattern.c, 487
- embPattern\_write
  - embroidery.h, 341
  - formats.c, 412
- embPattern\_writeAuto
  - embroidery.h, 341
  - formats.c, 412
- EmbPoint
  - embroidery.h, 331
- EmbPoint\_, 95
  - color, 95
  - lineType, 95
  - position, 95
- EmbPolygon
  - embroidery.h, 331
- embPolygon\_reduceByDistance
  - fill.c, 408
- embPolygon\_reduceByNth
  - fill.c, 408
- EmbPolyline
  - embroidery.h, 331
- EmbRadiusDim
  - embroidery.h, 331
- EmbRadiusDim\_, 96
  - position, 96
- EmbRay
  - embroidery.h, 331
- EmbRay\_, 96
  - position, 96
- EmbReal
  - embroidery.h, 331
- EmbRect
  - embroidery.h, 331
- EmbRect\_, 96
  - bottom, 97
  - left, 97

- radius, [97](#)
- right, [97](#)
- rotation, [97](#)
- top, [97](#)
- embRect\_area
  - embroidery.h, [342](#)
  - rect.c, [466](#)
- embRect\_bottomLeft
  - arc.c, [461](#)
- embRect\_bottomRight
  - arc.c, [461](#)
- embRect\_init
  - embroidery.h, [342](#)
  - rect.c, [466](#)
- embroidermodder.cpp
  - \_appVer\_, [231](#)
  - exitApp, [231](#)
  - main, [230](#)
  - usage\_msg, [231](#)
- embroidermodder.h
  - \_mainWin, [246](#)
  - actionHash, [246](#)
  - activeScene, [237](#)
  - activeView, [237](#)
  - actuator, [237](#)
  - add\_polyline, [238](#)
  - add\_to\_path, [238](#)
  - BOOL\_TYPE, [235](#)
  - checkBoxes, [246](#)
  - comboBoxes, [246](#)
  - Command, [236](#)
  - config, [246](#)
  - config\_tables, [246](#)
  - construct\_command, [238](#)
  - contains, [238](#)
  - convert\_args\_to\_type, [239](#)
  - create\_menu, [239](#)
  - debug\_message, [239](#)
  - degrees\_, [239](#)
  - dialog, [246](#)
  - Dictionary, [236](#)
  - dockPropEdit, [246](#)
  - dockUndoEdit, [246](#)
  - doubleSpinBoxes, [246](#)
  - emb\_constant\_pi, [246](#)
  - fileExtension, [239](#)
  - FUNCTION\_TYPE, [235](#)
  - get\_bool, [240](#)
  - get\_int, [240](#)
  - get\_qstr, [240](#)
  - get\_real, [240](#)
  - get\_str, [240](#)
  - get\_str\_list, [240](#)
  - get\_uint, [240](#)
  - groupBoxes, [247](#)
  - INT\_TYPE, [235](#)
  - labels, [247](#)
  - lineEdits, [247](#)
  - make\_checkbox, [240](#)
  - make\_spinbox, [240](#)
  - make\_ui\_element, [240](#)
  - mdiArea, [247](#)
  - menuHash, [247](#)
  - Node, [236](#)
  - node\_bool, [241](#)
  - node\_int, [241](#)
  - node\_qstr, [241](#)
  - node\_real, [241](#)
  - node\_str, [241](#)
  - node\_str\_list, [242](#)
  - node\_uint, [242](#)
  - NodeList, [236](#)
  - OBJ\_COLOR, [236](#)
  - OBJ\_KEYS, [236](#)
  - OBJ\_LAYER, [236](#)
  - OBJ\_LTYPE, [236](#)
  - OBJ\_LWT, [236](#)
  - OBJ\_NAME, [236](#)
  - OBJ\_RUBBER, [236](#)
  - OBJ\_TYPE, [236](#)
  - OBJ\_TYPE\_ARC, [237](#)
  - OBJ\_TYPE\_BASE, [236](#)
  - OBJ\_TYPE\_BLOCK, [237](#)
  - OBJ\_TYPE\_CIRCLE, [237](#)
  - OBJ\_TYPE\_DIMALIGNED, [237](#)
  - OBJ\_TYPE\_DIMANGULAR, [237](#)
  - OBJ\_TYPE\_DIMARCLENGTH, [237](#)
  - OBJ\_TYPE\_DIMDIAMETER, [237](#)
  - OBJ\_TYPE\_DIMLEADER, [237](#)
  - OBJ\_TYPE\_DIMLINEAR, [237](#)
  - OBJ\_TYPE\_DIMORDINATE, [237](#)
  - OBJ\_TYPE\_DIMRADIUS, [237](#)
  - OBJ\_TYPE\_ELLIPSE, [237](#)
  - OBJ\_TYPE\_ELLIPSEARC, [237](#)
  - OBJ\_TYPE\_GRID, [237](#)
  - OBJ\_TYPE\_HATCH, [237](#)
  - OBJ\_TYPE\_IMAGE, [237](#)
  - OBJ\_TYPE\_INFINITELINE, [237](#)
  - OBJ\_TYPE\_LINE, [237](#)
  - OBJ\_TYPE\_NULL, [236](#)
  - OBJ\_TYPE\_PATH, [237](#)
  - OBJ\_TYPE\_POINT, [237](#)
  - OBJ\_TYPE\_POLYGON, [237](#)
  - OBJ\_TYPE\_POLYLINE, [237](#)
  - OBJ\_TYPE\_RAY, [237](#)
  - OBJ\_TYPE\_RECTANGLE, [237](#)
  - OBJ\_TYPE\_RUBBER, [237](#)
  - OBJ\_TYPE\_SLOT, [237](#)
  - OBJ\_TYPE\_SPLINE, [237](#)
  - OBJ\_TYPE\_TEXTMULTI, [237](#)
  - OBJ\_TYPE\_TEXTSINGLE, [237](#)
  - OBJ\_TYPE\_UNKNOWN, [237](#)
  - OBJ\_TYPE\_VALUES, [236](#)
  - operator\*, [242](#)
  - operator+, [242](#)
  - operator-, [242](#)

- prompt, [247](#)
- radians\_\_, [243](#)
- read\_configuration, [243](#)
- read\_settings, [243](#)
- read\_string\_setting, [243](#)
- REAL\_TYPE, [235](#)
- rotate\_vector, [243](#)
- run\_script, [243](#)
- run\_script\_file, [244](#)
- scripts, [247](#)
- set\_enabled, [244](#)
- set\_visibility, [244](#)
- settings, [247](#)
- spinBoxes, [247](#)
- statusbar, [247](#)
- String, [236](#)
- STRING\_LIST\_TYPE, [235](#)
- STRING\_TYPE, [235](#)
- StringList, [236](#)
- subMenuHash, [247](#)
- to\_EmbVector, [244](#)
- to\_qlist, [244](#)
- to\_QPointF, [245](#)
- to\_string\_vector, [245](#)
- to\_vector, [245](#)
- tokenize, [245](#)
- toolbarHash, [247](#)
- toolButtons, [247](#)
- translate\_str, [245](#)
- UNKNOWN\_TYPE, [235](#)
- validFileFormat, [245](#)
- VECTOR\_TYPE, [235](#)
- write\_settings, [246](#)
- embroidermodder2/cmdprompt.cpp, [230](#)
- embroidermodder2/em2\_dev\_script.py, [230](#)
- embroidermodder2/embdetails-dialog.cpp, [230](#)
- embroidermodder2/embroidermodder.cpp, [230](#)
- embroidermodder2/embroidermodder.h, [231](#), [247](#)
- embroidermodder2/imagewidget.cpp, [265](#)
- embroidermodder2/interface.cpp, [265](#)
- embroidermodder2/layer-manager.cpp, [272](#)
- embroidermodder2/mainwindow-menus.cpp, [273](#)
- embroidermodder2/mainwindow-toolbars.cpp, [273](#)
- embroidermodder2/mainwindow.cpp, [273](#)
- embroidermodder2/mdiarea.cpp, [304](#)
- embroidermodder2/mdiwindow.cpp, [304](#)
- embroidermodder2/objects.cpp, [305](#)
- embroidermodder2/preview-dialog.cpp, [306](#)
- embroidermodder2/property-editor.cpp, [306](#)
- embroidermodder2/README.md, [307](#)
- embroidermodder2/selectbox.cpp, [307](#)
- embroidermodder2/settings-dialog.cpp, [308](#)
- embroidermodder2/statusbar.cpp, [310](#)
- embroidermodder2/undo-commands.cpp, [310](#)
- embroidermodder2/undo-editor.cpp, [310](#)
- embroidermodder2/view.cpp, [310](#)
- embroidery.h
  - \_dxfColorTable, [347](#)
  - Arc\_Polyester, [322](#)
  - Arc\_Rayon, [322](#)
  - black\_thread, [347](#)
  - CHUNK\_SIZE, [322](#)
  - CoatsAndClark\_Rayon, [322](#)
  - convert, [332](#)
  - degrees, [332](#)
  - dxf\_color, [322](#)
  - EMB\_ARC, [322](#)
  - EMB\_ARRAY, [322](#)
  - EMB\_CIRCLE, [322](#)
  - EMB\_DIM\_DIAMETER, [323](#)
  - EMB\_DIM\_LEADER, [323](#)
  - EMB\_ELLIPSE, [323](#)
  - emb\_error, [347](#)
  - EMB\_FLAG, [323](#)
  - EMB\_FORMAT\_100, [323](#)
  - EMB\_FORMAT\_100O, [323](#)
  - EMB\_FORMAT\_ART, [323](#)
  - EMB\_FORMAT\_BMC, [323](#)
  - EMB\_FORMAT\_BRO, [323](#)
  - EMB\_FORMAT\_CND, [323](#)
  - EMB\_FORMAT\_COL, [323](#)
  - EMB\_FORMAT\_CSD, [323](#)
  - EMB\_FORMAT\_CSV, [323](#)
  - EMB\_FORMAT\_DAT, [323](#)
  - EMB\_FORMAT\_DEM, [323](#)
  - EMB\_FORMAT\_DSB, [323](#)
  - EMB\_FORMAT\_DST, [323](#)
  - EMB\_FORMAT\_DSZ, [323](#)
  - EMB\_FORMAT\_DXF, [324](#)
  - EMB\_FORMAT\_EDR, [324](#)
  - EMB\_FORMAT\_EMD, [324](#)
  - EMB\_FORMAT\_EXP, [324](#)
  - EMB\_FORMAT\_EXY, [324](#)
  - EMB\_FORMAT\_EYS, [324](#)
  - EMB\_FORMAT\_FXY, [324](#)
  - EMB\_FORMAT\_GC, [324](#)
  - EMB\_FORMAT\_GNC, [324](#)
  - EMB\_FORMAT\_GT, [324](#)
  - EMB\_FORMAT\_HUS, [324](#)
  - EMB\_FORMAT\_INB, [324](#)
  - EMB\_FORMAT\_INF, [324](#)
  - EMB\_FORMAT\_JEF, [324](#)
  - EMB\_FORMAT\_KSM, [324](#)
  - EMB\_FORMAT\_MAX, [324](#)
  - EMB\_FORMAT\_MIT, [324](#)
  - EMB\_FORMAT\_NEW, [324](#)
  - EMB\_FORMAT\_OFM, [325](#)
  - EMB\_FORMAT\_PCD, [325](#)
  - EMB\_FORMAT\_PCM, [325](#)
  - EMB\_FORMAT\_PCQ, [325](#)
  - EMB\_FORMAT\_PCS, [325](#)
  - EMB\_FORMAT\_PEC, [325](#)
  - EMB\_FORMAT\_PEL, [325](#)
  - EMB\_FORMAT\_PEM, [325](#)
  - EMB\_FORMAT\_PES, [325](#)
  - EMB\_FORMAT\_PHB, [325](#)

EMB\_FORMAT\_PHC, [325](#)  
EMB\_FORMAT\_PLT, [325](#)  
EMB\_FORMAT\_RGB, [325](#)  
EMB\_FORMAT\_SEW, [325](#)  
EMB\_FORMAT\_SHV, [325](#)  
EMB\_FORMAT\_SST, [325](#)  
EMB\_FORMAT\_STX, [325](#)  
EMB\_FORMAT\_SVG, [325](#)  
EMB\_FORMAT\_T01, [326](#)  
EMB\_FORMAT\_T09, [326](#)  
EMB\_FORMAT\_TAP, [326](#)  
EMB\_FORMAT\_THR, [326](#)  
EMB\_FORMAT\_TXT, [326](#)  
EMB\_FORMAT\_U00, [326](#)  
EMB\_FORMAT\_U01, [326](#)  
EMB\_FORMAT\_VIP, [326](#)  
EMB\_FORMAT\_VP3, [326](#)  
EMB\_FORMAT\_XXX, [326](#)  
EMB\_FORMAT\_ZSK, [326](#)  
emb\_identify\_format, [332](#)  
EMB\_IMAGE, [326](#)  
EMB\_LINE, [326](#)  
EMB\_MAX\_LAYERS, [326](#)  
EMB\_PATH, [326](#)  
EMB\_POINT, [326](#)  
EMB\_POLYGON, [326](#)  
EMB\_POLYLINE, [326](#)  
EMB\_PUBLIC, [327](#)  
EMB\_RECT, [327](#)  
emb\_round, [332](#)  
EMB\_SPLINE, [327](#)  
EMB\_STITCH, [327](#)  
EMB\_TEXT\_MULTI, [327](#)  
EMB\_TEXT\_SINGLE, [327](#)  
EMB\_THREAD, [327](#)  
EMB\_VECTOR, [327](#)  
emb\_verbose, [347](#)  
EmbAlignedDim, [329](#)  
EmbAngularDim, [329](#)  
EmbArc, [329](#)  
embArc\_clockwise, [332](#)  
embArc\_init, [332](#)  
EmbArcLengthDim, [330](#)  
EmbArray, [330](#)  
embArray\_addArc, [332](#)  
embArray\_addCircle, [332](#)  
embArray\_addEllipse, [332](#)  
embArray\_addFlag, [332](#)  
embArray\_addLine, [333](#)  
embArray\_addPath, [333](#)  
embArray\_addPoint, [333](#)  
embArray\_addPolygon, [333](#)  
embArray\_addPolyline, [333](#)  
embArray\_addRect, [333](#)  
embArray\_addStitch, [333](#)  
embArray\_addThread, [333](#)  
embArray\_addVector, [333](#)  
embArray\_copy, [333](#)  
embArray\_create, [333](#)  
embArray\_free, [334](#)  
embArray\_resize, [334](#)  
EmbBezier, [330](#)  
EmbBlock, [330](#)  
EmbCircle, [330](#)  
embCircle\_init, [334](#)  
EmbColor, [330](#)  
embColor\_create, [334](#)  
embColor\_distance, [334](#)  
embColor\_fromHexStr, [334](#)  
embColor\_make, [334](#)  
embConstantPi, [347](#)  
EmbDiameterDim, [330](#)  
EmbEllipse, [330](#)  
embEllipse\_area, [334](#)  
embEllipse\_diameterX, [334](#)  
embEllipse\_diameterY, [334](#)  
embEllipse\_height, [335](#)  
embEllipse\_init, [335](#)  
embEllipse\_make, [335](#)  
embEllipse\_perimeter, [335](#)  
embEllipse\_width, [335](#)  
EmbFlag, [330](#)  
EMBFORMAT\_MAXDESC, [327](#)  
EMBFORMAT\_MAXEXT, [327](#)  
EMBFORMAT\_OBJECTONLY, [327](#)  
EMBFORMAT\_STCHANDOBJ, [327](#)  
EMBFORMAT\_STITCHONLY, [327](#)  
EMBFORMAT\_UNSUPPORTED, [327](#)  
EmbFormatList, [330](#)  
EmbGeometry, [330](#)  
embGeometry\_boundingRect, [335](#)  
embGeometry\_free, [335](#)  
embGeometry\_init, [335](#)  
embGeometry\_move, [335](#)  
embGeometry\_vulcanize, [335](#)  
EmblImage, [330](#)  
emblImage\_create, [336](#)  
emblImage\_free, [336](#)  
emblImage\_read, [336](#)  
emblImage\_write, [336](#)  
EmblInfiniteLine, [330](#)  
EmbLayer, [330](#)  
EmbLeaderDim, [330](#)  
EmbLine, [330](#)  
embLine\_intersectionPoint, [336](#)  
embLine\_make, [336](#)  
embLine\_normalVector, [336](#)  
EmbLinearDim, [330](#)  
EmbOrdinateDim, [330](#)  
EmbPath, [331](#)  
EmbPattern, [331](#)  
embPattern\_addCircleAbs, [336](#)  
embPattern\_addEllipseAbs, [336](#)  
embPattern\_addLineAbs, [336](#)  
embPattern\_addPathAbs, [337](#)  
embPattern\_addPointAbs, [337](#)



- embPattern\_addPolygonAbs, 337
- embPattern\_addPolylineAbs, 337
- embPattern\_addRectAbs, 337
- embPattern\_addStitchAbs, 337
- embPattern\_addStitchRel, 337
- embPattern\_addThread, 337
- embPattern\_calcBoundingBox, 337
- embPattern\_center, 338
- embPattern\_changeColor, 338
- embPattern\_color\_count, 338
- embPattern\_combine, 338
- embPattern\_combineJumpStitches, 338
- embPattern\_convertGeometry, 338
- embPattern\_copyPolylinesToStitchList, 338
- embPattern\_copyStitchListToPolylines, 338
- embPattern\_correctForMaxStitchLength, 338
- embPattern\_create, 338
- embPattern\_crossstitch, 339
- embPattern\_designDetails, 339
- embPattern\_end, 339
- embPattern\_fixColorCount, 339
- embPattern\_flip, 339
- embPattern\_flipHorizontal, 339
- embPattern\_flipVertical, 339
- embPattern\_free, 339
- embPattern\_hideStitchesOverLength, 339
- embPattern\_horizontal\_fill, 339
- embPattern\_jumpStitches, 340
- embPattern\_lengthHistogram, 340
- embPattern\_loadExternalColorFile, 340
- embPattern\_maximumStitchLength, 340
- embPattern\_minimumStitchLength, 340
- embPattern\_movePolylinesToStitchList, 340
- embPattern\_moveStitchListToPolylines, 340
- embPattern\_read, 340
- embPattern\_readAuto, 340
- embPattern\_realStitches, 341
- embPattern\_render, 341
- embPattern\_scale, 341
- embPattern\_simulate, 341
- embPattern\_totalStitchLength, 341
- embPattern\_trimStitches, 341
- embPattern\_write, 341
- embPattern\_writeAuto, 341
- EmbPoint, 331
- EmbPolygon, 331
- EmbPolyline, 331
- EmbRadiusDim, 331
- EmbRay, 331
- EmbReal, 331
- EmbRect, 331
- embRect\_area, 342
- embRect\_init, 342
- EmbSatinOutline, 331
- embSatinOutline\_generateSatinOutline, 342
- embSatinOutline\_renderStitches, 342
- EmbSpline, 331
- EmbStitch, 331
- EmbTextMulti, 331
- EmbTextSingle, 331
- EmbThread, 331
- embThread\_findNearestColor, 342
- embThread\_findNearestThread, 342
- embThread\_getRandom, 343
- EmbTime, 331
- embTime\_initNow, 343
- embTime\_time, 343
- EmbVector, 331
- embVector\_add, 343
- embVector\_angle, 343
- embVector\_average, 343
- embVector\_cross, 343
- embVector\_distance, 344
- embVector\_dot, 344
- embVector\_length, 344
- embVector\_multiply, 344
- embVector\_normalize, 344
- embVector\_relativeX, 344
- embVector\_relativeY, 345
- embVector\_subtract, 345
- embVector\_transpose\_product, 345
- embVector\_unit, 345
- END, 327
- Exquisite\_Polyester, 327
- formatTable, 347
- Fufu\_Polyester, 327
- Fufu\_Rayon, 327
- full\_test\_matrix, 345
- getArcCenter, 345
- getArcDataFromBulge, 345
- getCircleCircleIntersections, 346
- getCircleTangentPoints, 346
- Hemingworth\_Polyester, 328
- hilbert\_curve, 346
- hus\_thread, 328
- husThreads, 347
- Isacord\_Polyester, 328
- Isafil\_Rayon, 328
- jef\_thread, 328
- jefThreads, 347
- JUMP, 328
- L\_system, 331
- LIBEMBROIDERY\_EMBEDDED\_VERSION, 328
- lindenmayer\_system, 346
- Madeira\_Polyester, 328
- Madeira\_Rayon, 328
- Marathon\_Polyester, 328
- Marathon\_Rayon, 328
- MAX\_STITCHES, 328
- MAX\_THREADS, 328
- Metro\_Polyester, 328
- NORMAL, 328
- numberOfFormats, 328
- Pantone, 328
- pcm\_thread, 328
- pcmThreads, 347



- pec\_thread, [329](#)
- pecThreadCount, [347](#)
- pecThreads, [347](#)
- radians, [346](#)
- report, [346](#)
- RobisonAnton\_Polyester, [329](#)
- RobisonAnton\_Rayon, [329](#)
- SEQUIN, [329](#)
- shv\_thread, [329](#)
- shvThreadCount, [347](#)
- shvThreads, [348](#)
- Sigma\_Polyester, [329](#)
- STOP, [329](#)
- Sulky\_Rayon, [329](#)
- SVG\_Colors, [329](#)
- testMain, [346](#)
- thread\_color, [332](#)
- ThreadArt\_Polyester, [329](#)
- ThreadArt\_Rayon, [329](#)
- threadColor, [346](#)
- threadColorName, [346](#)
- threadColorNum, [347](#)
- ThreaDelight\_Polyester, [329](#)
- TRIM, [329](#)
- vipDecodingTable, [348](#)
- Z102\_Isacord\_Polyester, [329](#)
- embroidery\_internal.h
  - bcf\_difat\_create, [372](#)
  - bcf\_directory, [371](#)
  - bcf\_directory\_entry, [371](#)
  - bcf\_directory\_free, [373](#)
  - bcf\_file, [371](#)
  - bcf\_file\_difat, [371](#)
  - bcf\_file\_difat\_free, [373](#)
  - bcf\_file\_fat, [371](#)
  - bcf\_file\_fat\_free, [373](#)
  - bcf\_file\_free, [373](#)
  - bcf\_file\_header, [371](#)
  - bcfFile\_read, [373](#)
  - bcfFileFat\_create, [373](#)
  - bcfFileHeader\_isValid, [373](#)
  - bcfFileHeader\_read, [373](#)
  - binaryReadString, [373](#)
  - binaryReadUnicodeString, [374](#)
  - binaryWriteInt, [374](#)
  - binaryWriteIntBE, [374](#)
  - binaryWriteShort, [374](#)
  - binaryWriteUInt, [374](#)
  - binaryWriteUIntBE, [374](#)
  - binaryWriteUShort, [374](#)
  - binaryWriteUShortBE, [375](#)
  - BULGETOCONTROL, [363](#)
  - BULGETOEND, [363](#)
  - check\_header\_present, [375](#)
  - CompoundFileDirectory, [375](#)
  - CompoundFileDirectoryEntry, [375](#)
  - CompoundFileSector\_DIFAT\_Sector, [363](#)
  - CompoundFileSector\_EndOfChain, [363](#)
  - CompoundFileSector\_FAT\_Sector, [364](#)
  - CompoundFileSector\_FreeSector, [364](#)
  - CompoundFileSector\_MaxRegSector, [364](#)
  - CompoundFileStreamId\_MaxRegularStreamId, [364](#)
  - CompoundFileStreamId\_NoStream, [364](#)
  - compress, [371](#)
  - compress\_get\_bits, [375](#)
  - compress\_get\_position, [375](#)
  - compress\_get\_token, [375](#)
  - compress\_load\_block, [376](#)
  - compress\_load\_character\_huffman, [376](#)
  - compress\_load\_character\_length\_huffman, [376](#)
  - compress\_load\_distance\_huffman, [376](#)
  - compress\_pop, [376](#)
  - compress\_read\_variable\_length, [376](#)
  - copy\_trim, [376](#)
  - create\_test\_file\_1, [376](#)
  - create\_test\_file\_2, [376](#)
  - create\_test\_file\_3, [376](#)
  - CSV\_EXPECT, [372](#)
  - CSV\_EXPECT\_COMMA, [372](#)
  - CSV\_EXPECT\_NULL, [372](#)
  - CSV\_EXPECT\_QUOTE1, [372](#)
  - CSV\_EXPECT\_QUOTE2, [372](#)
  - CSV\_MODE, [372](#)
  - CSV\_MODE\_COMMENT, [372](#)
  - CSV\_MODE\_NULL, [372](#)
  - CSV\_MODE\_STITCH, [372](#)
  - CSV\_MODE\_THREAD, [372](#)
  - CSV\_MODE\_VARIABLE, [372](#)
  - CUBICTOCONTROL1, [364](#)
  - CUBICTOCONTROL2, [364](#)
  - CUBICTOEND, [364](#)
  - decode\_t01\_record, [376](#)
  - decode\_tajima\_ternary, [377](#)
  - decodeNewStitch, [377](#)
  - DXF\_VERSION\_2000, [364](#)
  - DXF\_VERSION\_2002, [364](#)
  - DXF\_VERSION\_2004, [364](#)
  - DXF\_VERSION\_2006, [364](#)
  - DXF\_VERSION\_2007, [364](#)
  - DXF\_VERSION\_2009, [364](#)
  - DXF\_VERSION\_2010, [364](#)
  - DXF\_VERSION\_2013, [364](#)
  - DXF\_VERSION\_R10, [364](#)
  - DXF\_VERSION\_R11, [365](#)
  - DXF\_VERSION\_R12, [365](#)
  - DXF\_VERSION\_R13, [365](#)
  - DXF\_VERSION\_R14, [365](#)
  - DXF\_VERSION\_R15, [365](#)
  - DXF\_VERSION\_R18, [365](#)
  - DXF\_VERSION\_R21, [365](#)
  - DXF\_VERSION\_R24, [365](#)
  - DXF\_VERSION\_R27, [365](#)
  - ELEMENT\_A, [365](#)
  - ELEMENT\_ANIMATE, [365](#)
  - ELEMENT\_ANIMATECOLOR, [365](#)

ELEMENT\_ANIMATEMOTION, 365  
ELEMENT\_ANIMATETRANSFORM, 365  
ELEMENT\_ANIMATION, 365  
ELEMENT\_AUDIO, 365  
ELEMENT\_CIRCLE, 365  
ELEMENT\_DEFS, 365  
ELEMENT\_DESC, 366  
ELEMENT\_DISCARD, 366  
ELEMENT\_ELLIPSE, 366  
ELEMENT\_FONT, 366  
ELEMENT\_FONT\_FACE, 366  
ELEMENT\_FONT\_FACE\_SRC, 366  
ELEMENT\_FONT\_FACE\_URI, 366  
ELEMENT\_FOREIGN\_OBJECT, 366  
ELEMENT\_G, 366  
ELEMENT\_GLYPH, 366  
ELEMENT\_HANDLER, 366  
ELEMENT\_HKERN, 366  
ELEMENT\_IMAGE, 366  
ELEMENT\_LINE, 366  
ELEMENT\_LINEAR\_GRADIENT, 366  
ELEMENT\_LISTENER, 366  
ELEMENT\_METADATA, 366  
ELEMENT\_MISSING\_GLYPH, 366  
ELEMENT\_MPATH, 367  
ELEMENT\_PATH, 367  
ELEMENT\_POLYGON, 367  
ELEMENT\_POLYLINE, 367  
ELEMENT\_PREFETCH, 367  
ELEMENT\_RADIAL\_GRADIENT, 367  
ELEMENT\_RECT, 367  
ELEMENT\_SCRIPT, 367  
ELEMENT\_SET, 367  
ELEMENT\_SOLID\_COLOR, 367  
ELEMENT\_STOP, 367  
ELEMENT\_SVG, 367  
ELEMENT\_SWITCH, 367  
ELEMENT\_TBREAK, 367  
ELEMENT\_TEXT, 367  
ELEMENT\_TEXT\_AREA, 367  
ELEMENT\_TITLE, 367  
ELEMENT\_TSPAN, 367  
ELEMENT\_USE, 368  
ELEMENT\_VIDEO, 368  
ELEMENT\_XML, 368  
ELLIPSETOEND, 368  
ELLIPSETORAD, 368  
EMB\_BIG\_ENDIAN, 368  
EMB\_INT16\_BIG, 368  
EMB\_INT16\_LITTLE, 368  
EMB\_INT32\_BIG, 368  
EMB\_INT32\_LITTLE, 368  
EMB\_LITTLE\_ENDIAN, 368  
EMB\_MAX, 368  
EMB\_MIN, 368  
emb\_optOut, 377  
emb\_readline, 377  
embColor\_read, 377  
embColor\_write, 377  
emblnt\_read, 378  
emblnt\_write, 378  
encode\_t01\_record, 378  
encode\_tajima\_ternary, 378  
ENDIAN\_HOST, 368  
entriesInDifatSector, 378  
fpad, 378  
fread\_int16, 378  
fread\_int32\_be, 379  
fread\_uint16, 379  
GetFile, 379  
GREEN\_TERM\_COLOR, 368  
HOOP\_110X110, 368  
HOOP\_126X110, 368  
HOOP\_140X200, 369  
HOOP\_230X200, 369  
HOOP\_50X50, 369  
huffman, 371  
huffman\_build\_table, 379  
huffman\_table\_lookup, 379  
hus\_compress, 379  
hus\_decompress, 379  
imageWithFrame, 395  
LINETO, 369  
loadFatFromSector, 380  
mitDecodeStitch, 380  
mitEncodeStitch, 380  
MOVETO, 369  
N\_PES\_VERSIONS, 369  
numberOfEntriesInDifatSector, 380  
ObjectTypeRootEntry, 369  
ObjectTypeStorage, 369  
ObjectTypeStream, 369  
ObjectTypeUnknown, 369  
PES0001, 369  
PES0020, 369  
PES0022, 369  
PES0030, 369  
PES0040, 369  
PES0050, 369  
PES0055, 369  
PES0056, 370  
PES0060, 370  
PES0070, 370  
PES0080, 370  
PES0090, 370  
PES0100, 370  
pfaffDecode, 380  
pfaffEncode, 380  
printArcResults, 380  
QUADTOCONTROL, 370  
QUADTOEND, 370  
read100, 381  
read10o, 381  
readArt, 381  
readBmc, 381  
readBro, 381

readCnd, [381](#)  
readCol, [381](#)  
readCsd, [381](#)  
readCsv, [381](#)  
readDat, [381](#)  
readDem, [381](#)  
readDescriptions, [382](#)  
readDsb, [382](#)  
readDst, [382](#)  
readDsz, [382](#)  
readDxf, [382](#)  
readEdr, [382](#)  
readEmd, [382](#)  
readExp, [382](#)  
readExy, [382](#)  
readEys, [382](#)  
readFeatherPatterns, [383](#)  
readFullSector, [383](#)  
readFxy, [383](#)  
readGc, [383](#)  
readGnc, [383](#)  
readGt, [383](#)  
readHoopName, [383](#)  
readHus, [383](#)  
readImageString, [383](#)  
readInb, [384](#)  
readInf, [384](#)  
readJef, [384](#)  
readKsm, [384](#)  
readMax, [384](#)  
readMit, [384](#)  
readMotifPatterns, [384](#)  
readNew, [384](#)  
readNextSector, [384](#)  
readOfm, [384](#)  
readPcd, [385](#)  
readPcm, [385](#)  
readPcq, [385](#)  
readPcs, [385](#)  
readPec, [385](#)  
readPecStitches, [385](#)  
readPel, [385](#)  
readPem, [385](#)  
readPes, [386](#)  
readPESHeaderV10, [386](#)  
readPESHeaderV5, [386](#)  
readPESHeaderV6, [386](#)  
readPESHeaderV7, [386](#)  
readPESHeaderV8, [386](#)  
readPESHeaderV9, [386](#)  
readPhb, [386](#)  
readPhc, [386](#)  
readPlt, [386](#)  
readProgrammableFills, [387](#)  
readRgb, [387](#)  
readSew, [387](#)  
readShv, [387](#)  
readSst, [387](#)  
readStx, [387](#)  
readSvg, [387](#)  
readT01, [387](#)  
readT09, [387](#)  
readTap, [387](#)  
readThr, [388](#)  
readThreads, [388](#)  
readTxt, [388](#)  
readU00, [388](#)  
readU01, [388](#)  
readVip, [388](#)  
readVp3, [388](#)  
readXxx, [388](#)  
readZsk, [388](#)  
RED\_TERM\_COLOR, [370](#)  
RESET\_TERM\_COLOR, [370](#)  
safe\_free, [388](#)  
stringInArray, [389](#)  
StxThread, [371](#)  
SubDescriptor, [372](#)  
SVG\_ATTRIBUTE, [370](#)  
SVG\_CATCH\_ALL, [370](#)  
SVG\_CREATOR\_EMBROIDERMODDER, [370](#)  
SVG\_CREATOR\_ILLUSTRATOR, [370](#)  
SVG\_CREATOR\_INKSCAPE, [370](#)  
SVG\_CREATOR\_NULL, [370](#)  
SVG\_ELEMENT, [370](#)  
SVG\_EXPECT\_ATTRIBUTE, [370](#)  
SVG\_EXPECT\_ELEMENT, [371](#)  
SVG\_EXPECT\_NULL, [371](#)  
SVG\_EXPECT\_VALUE, [371](#)  
SVG\_MEDIA\_PROPERTY, [371](#)  
SVG\_NULL, [371](#)  
SVG\_PROPERTY, [371](#)  
SvgAttribute, [372](#)  
testEmbCircle, [389](#)  
testEmbCircle\_2, [389](#)  
testEmbFormat, [389](#)  
testGeomArc, [389](#)  
testTangentPoints, [389](#)  
testThreadColor, [389](#)  
ThredExtension, [372](#)  
ThredHeader, [372](#)  
VipHeader, [372](#)  
vp3Hoop, [372](#)  
write100, [389](#)  
write10o, [389](#)  
write\_24bit, [389](#)  
writeArt, [390](#)  
writeBmc, [390](#)  
writeBro, [390](#)  
writeCnd, [390](#)  
writeCol, [390](#)  
writeCsd, [390](#)  
writeCsv, [390](#)  
writeDat, [390](#)  
writeDem, [390](#)  
writeDsb, [390](#)

- writeDst, [390](#)
- writeDsz, [391](#)
- writeDxf, [391](#)
- writeEdr, [391](#)
- writeEmd, [391](#)
- writeExp, [391](#)
- writeExy, [391](#)
- writeEys, [391](#)
- writeFxy, [391](#)
- writeGc, [391](#)
- writeGnc, [391](#)
- writeGt, [391](#)
- writeHus, [392](#)
- writeInb, [392](#)
- writeInf, [392](#)
- writeJef, [392](#)
- writeKsm, [392](#)
- writeMax, [392](#)
- writeMit, [392](#)
- writeNew, [392](#)
- writeOfm, [392](#)
- writePcd, [392](#)
- writePcm, [392](#)
- writePcq, [393](#)
- writePcs, [393](#)
- writePec, [393](#)
- writePecStitches, [393](#)
- writePel, [393](#)
- writePem, [393](#)
- writePes, [393](#)
- writePhb, [393](#)
- writePhc, [393](#)
- writePlt, [393](#)
- writeRgb, [394](#)
- writeSew, [394](#)
- writeShv, [394](#)
- writeSst, [394](#)
- writeStx, [394](#)
- writeSvg, [394](#)
- writeT01, [394](#)
- writeT09, [394](#)
- writeTap, [394](#)
- writeThr, [394](#)
- writeTxt, [394](#)
- writeU00, [395](#)
- writeU01, [395](#)
- writeVip, [395](#)
- writeVp3, [395](#)
- writeXxx, [395](#)
- writeZsk, [395](#)
- YELLOW\_TERM\_COLOR, [371](#)
- EmbSatinOutline
  - embroidery.h, [331](#)
- EmbSatinOutline\_, [97](#)
  - length, [98](#)
  - side1, [98](#)
  - side2, [98](#)
- embSatinOutline\_generateSatinOutline
  - embroidery.h, [342](#)
  - main.c, [478](#)
- embSatinOutline\_renderStitches
  - embroidery.h, [342](#)
  - main.c, [478](#)
- EmbSpline
  - embroidery.h, [331](#)
- EmbSpline\_, [98](#)
  - beziers, [98](#)
- EmbStitch
  - embroidery.h, [331](#)
- EmbStitch\_, [99](#)
  - color, [99](#)
  - flags, [99](#)
  - x, [99](#)
  - y, [99](#)
- EmbTextMulti
  - embroidery.h, [331](#)
- EmbTextMulti\_, [99](#)
  - position, [100](#)
  - text, [100](#)
- EmbTextSingle
  - embroidery.h, [331](#)
- EmbTextSingle\_, [100](#)
  - position, [100](#)
  - text, [100](#)
- EmbThread
  - embroidery.h, [331](#)
- EmbThread\_, [100](#)
  - catalogNumber, [101](#)
  - color, [101](#)
  - description, [101](#)
- embThread\_findNearestColor
  - embroidery.h, [342](#)
  - main.c, [479](#)
- embThread\_findNearestThread
  - embroidery.h, [342](#)
  - main.c, [479](#)
- embThread\_getRandom
  - embroidery.h, [343](#)
  - main.c, [479](#)
- EmbTime
  - embroidery.h, [331](#)
- EmbTime\_, [101](#)
  - day, [101](#)
  - hour, [101](#)
  - minute, [102](#)
  - month, [102](#)
  - second, [102](#)
  - year, [102](#)
- embTime\_initNow
  - embroidery.h, [343](#)
  - main.c, [479](#)
- embTime\_time
  - embroidery.h, [343](#)
  - main.c, [479](#)
- EmbVector
  - embroidery.h, [331](#)

- EmbVector\_, 102
  - x, 102
  - y, 102
- embVector\_add
  - embroidery.h, 343
  - vector.c, 468
- embVector\_angle
  - embroidery.h, 343
  - vector.c, 468
- embVector\_average
  - embroidery.h, 343
  - vector.c, 468
- embVector\_cross
  - embroidery.h, 343
  - vector.c, 468
- embVector\_distance
  - embroidery.h, 344
  - vector.c, 469
- embVector\_dot
  - embroidery.h, 344
  - vector.c, 469
- embVector\_length
  - embroidery.h, 344
  - vector.c, 469
- embVector\_multiply
  - embroidery.h, 344
  - vector.c, 469
- embVector\_normalize
  - embroidery.h, 344
  - vector.c, 469
- embVector\_print
  - main.c, 479
- embVector\_relativeX
  - embroidery.h, 344
  - vector.c, 469
- embVector\_relativeY
  - embroidery.h, 345
  - vector.c, 470
- embVector\_subtract
  - embroidery.h, 345
  - vector.c, 470
- embVector\_transpose\_product
  - embroidery.h, 345
  - vector.c, 470
- embVector\_unit
  - embroidery.h, 345
  - vector.c, 470
- emd, 9, 426
- emdDecode
  - format\_emd.c, 426
- encode\_record
  - format\_dst.c, 424
- encode\_t01\_record
  - embroidery\_internal.h, 378
  - encoding.c, 404
- encode\_tajima\_ternary
  - embroidery\_internal.h, 378
  - encoding.c, 404
- encode\_tap\_record
  - format\_tap.c, 450
- encoding.c
  - decode\_t01\_record, 403
  - decode\_tajima\_ternary, 403
  - decodeNewStitch, 403
  - embColor\_fromHexStr, 403
  - embInt\_read, 404
  - embInt\_write, 404
  - encode\_t01\_record, 404
  - encode\_tajima\_ternary, 404
  - mitDecodeStitch, 404
  - mitEncodeStitch, 404
  - pfaffDecode, 404
  - pfaffEncode, 405
  - reverse\_byte\_order, 405
  - write\_24bit, 405
- END
  - embroidery.h, 327
- end
  - EmbArc\_, 78
  - EmbBezier\_, 80
  - EmbLine\_, 92
- end\_action
  - mainwindow.cpp, 290
- endCommand
  - CmdPromptInput, 70
- ENDIAN\_HOST
  - embroidery\_internal.h, 368
- enterEvent
  - View, 220
- entriesInDifatSector
  - embroidery\_internal.h, 378
  - main.c, 480
- error\_action
  - mainwindow.cpp, 290
- escapePressed
  - CmdPrompt, 54
  - CmdPromptInput, 70
  - MainWindow, 151
  - MdiWindow, 171
  - View, 220
- event
  - Application, 50
- eventFilter
  - CmdPromptInput, 70
  - PropertyEditor, 181
- exitApp
  - embroidermodder.cpp, 231
- exp, 9, 426
- expDecode
  - format\_exp.c, 426
- Exquisite\_Polyester
  - embroidery.h, 327
- extension
  - EmbFormatList\_, 86
- extensions
  - settings-dialog.cpp, 309

- extern/libembroidery/src/array.c, 311
- extern/libembroidery/src/compress.c, 313
- extern/libembroidery/src/embroidery.h, 315, 348
- extern/libembroidery/src/embroidery\_internal.h, 355, 395
- extern/libembroidery/src/encoding.c, 402
- extern/libembroidery/src/fill.c, 405
- extern/libembroidery/src/formats.c, 410
- extern/libembroidery/src/formats/format\_100.c, 414
- extern/libembroidery/src/formats/format\_10o.c, 414
- extern/libembroidery/src/formats/format\_art.c, 415
- extern/libembroidery/src/formats/format\_bmc.c, 415
- extern/libembroidery/src/formats/format\_bro.c, 416
- extern/libembroidery/src/formats/format\_cnd.c, 416
- extern/libembroidery/src/formats/format\_col.c, 417
- extern/libembroidery/src/formats/format\_csd.c, 418
- extern/libembroidery/src/formats/format\_csv.c, 419
- extern/libembroidery/src/formats/format\_dat.c, 420
- extern/libembroidery/src/formats/format\_dem.c, 421
- extern/libembroidery/src/formats/format\_dsb.c, 421
- extern/libembroidery/src/formats/format\_dst.c, 422
- extern/libembroidery/src/formats/format\_dsz.c, 424
- extern/libembroidery/src/formats/format\_dxf.c, 424
- extern/libembroidery/src/formats/format\_edr.c, 425
- extern/libembroidery/src/formats/format\_emd.c, 426
- extern/libembroidery/src/formats/format\_exp.c, 426
- extern/libembroidery/src/formats/format\_exy.c, 427
- extern/libembroidery/src/formats/format\_eyc.c, 427
- extern/libembroidery/src/formats/format\_fxy.c, 428
- extern/libembroidery/src/formats/format\_gc.c, 428
- extern/libembroidery/src/formats/format\_gnc.c, 429
- extern/libembroidery/src/formats/format\_gt.c, 429
- extern/libembroidery/src/formats/format\_hus.c, 430
- extern/libembroidery/src/formats/format\_inb.c, 431
- extern/libembroidery/src/formats/format\_inf.c, 431
- extern/libembroidery/src/formats/format\_jef.c, 432
- extern/libembroidery/src/formats/format\_ksm.c, 433
- extern/libembroidery/src/formats/format\_max.c, 433
- extern/libembroidery/src/formats/format\_mit.c, 434
- extern/libembroidery/src/formats/format\_new.c, 435
- extern/libembroidery/src/formats/format\_ofm.c, 435
- extern/libembroidery/src/formats/format\_pcd.c, 436
- extern/libembroidery/src/formats/format\_pcm.c, 437
- extern/libembroidery/src/formats/format\_pcq.c, 437
- extern/libembroidery/src/formats/format\_pcs.c, 438
- extern/libembroidery/src/formats/format\_pec.c, 438
- extern/libembroidery/src/formats/format\_pel.c, 440
- extern/libembroidery/src/formats/format\_pem.c, 440
- extern/libembroidery/src/formats/format\_pes.c, 441
- extern/libembroidery/src/formats/format\_phb.c, 443
- extern/libembroidery/src/formats/format\_phc.c, 444
- extern/libembroidery/src/formats/format\_plt.c, 444
- extern/libembroidery/src/formats/format\_rgb.c, 445
- extern/libembroidery/src/formats/format\_sew.c, 445
- extern/libembroidery/src/formats/format\_shv.c, 446
- extern/libembroidery/src/formats/format\_sst.c, 446
- extern/libembroidery/src/formats/format\_stx.c, 447
- extern/libembroidery/src/formats/format\_svg.c, 447
- extern/libembroidery/src/formats/format\_t01.c, 449
- extern/libembroidery/src/formats/format\_t09.c, 449
- extern/libembroidery/src/formats/format\_tap.c, 450
- extern/libembroidery/src/formats/format\_thr.c, 450
- extern/libembroidery/src/formats/format\_txt.c, 451
- extern/libembroidery/src/formats/format\_u00.c, 451
- extern/libembroidery/src/formats/format\_u01.c, 452
- extern/libembroidery/src/formats/format\_vip.c, 452
- extern/libembroidery/src/formats/format\_vp3.c, 454
- extern/libembroidery/src/formats/format\_xxx.c, 455
- extern/libembroidery/src/formats/format\_zsk.c, 456
- extern/libembroidery/src/geometry.c, 456
- extern/libembroidery/src/geometry/arc.c, 458
- extern/libembroidery/src/geometry/circle.c, 461
- extern/libembroidery/src/geometry/ellipse.c, 462
- extern/libembroidery/src/geometry/functions.c, 464
- extern/libembroidery/src/geometry/line.c, 465
- extern/libembroidery/src/geometry/path.c, 465
- extern/libembroidery/src/geometry/polygon.c, 465
- extern/libembroidery/src/geometry/polyline.c, 465
- extern/libembroidery/src/geometry/rect.c, 466
- extern/libembroidery/src/geometry/text.c, 466
- extern/libembroidery/src/geometry/vector.c, 468
- extern/libembroidery/src/image.c, 470
- extern/libembroidery/src/main.c, 471
- extern/libembroidery/src/pattern.c, 482
- extern/libembroidery/src/thread-color.c, 487
- exy, 9, 427
- eyc, 383, 428
- F10Pressed
  - CmdPrompt, 54
  - CmdPromptInput, 70
- F11Pressed
  - CmdPrompt, 54
  - CmdPromptInput, 71
- F12Pressed
  - CmdPrompt, 54
  - CmdPromptInput, 71
- F1Pressed
  - CmdPrompt, 54
  - CmdPromptInput, 71
- F2Pressed
  - CmdPrompt, 54
  - CmdPromptInput, 71
- F3Pressed
  - CmdPrompt, 55
  - CmdPromptInput, 71
- F4Pressed
  - CmdPrompt, 55
  - CmdPromptInput, 71
- F5Pressed
  - CmdPrompt, 55
  - CmdPromptInput, 71
- F6Pressed
  - CmdPrompt, 55
  - CmdPromptInput, 71
- F7Pressed
  - CmdPrompt, 55

- CmdPromptInput, 71
- F8Pressed
  - CmdPrompt, 55
  - CmdPromptInput, 71
- F9Pressed
  - CmdPrompt, 55
  - CmdPromptInput, 71
- factor
  - UndoableCommand, 213
- fat
  - \_bcf\_file, 43
- fatEntries
  - \_bcf\_file\_fat, 44
- fatEntryCount
  - \_bcf\_file\_fat, 44
- fatSectorCount
  - \_bcf\_file\_difat, 43
- fatSectorEntries
  - \_bcf\_file\_difat, 44
- fieldEdited
  - PropertyEditor, 181
- fieldNewText
  - property-editor.cpp, 306
- fieldNoText
  - property-editor.cpp, 307
- fieldOffText
  - property-editor.cpp, 307
- fieldOldText
  - property-editor.cpp, 307
- fieldOnText
  - property-editor.cpp, 307
- fieldVariesText
  - property-editor.cpp, 307
- fieldYesText
  - property-editor.cpp, 307
- fileExtension
  - embroidermodder.h, 239
  - mdiwindow.cpp, 304
- fileWasLoaded
  - MdiWindow, 176
- fill.c
  - dragon\_curve, 406
  - embPattern\_combine, 406
  - embPattern\_convertGeometry, 406
  - embPattern\_crossstitch, 406
  - embPattern\_horizontal\_fill, 406
  - embPattern\_stitchArc, 406
  - embPattern\_stitchCircle, 406
  - embPattern\_stitchEllipse, 407
  - embPattern\_stitchPath, 407
  - embPattern\_stitchPolygon, 407
  - embPattern\_stitchPolyline, 407
  - embPattern\_stitchRect, 408
  - embPattern\_stitchText, 408
  - embPolygon\_reduceByDistance, 408
  - embPolygon\_reduceByNth, 408
  - generate\_dragon\_curve, 408
  - greedy\_algorithm, 408
  - hilbert\_curve, 408
  - hilbert\_curve\_l\_system, 409
  - join\_short\_stitches, 409
  - lindenmayer\_system, 409
  - rules, 409
  - save\_points\_to\_pattern, 409
  - threshold\_method, 409
- filled
  - Geometry, 136
- findIndex
  - Geometry, 114
- findMdiWindow
  - MainWindow, 151
- firstDifatSectorLocation
  - \_bcf\_file\_header, 45
- firstDirectorySectorLocation
  - \_bcf\_file\_header, 45
- firstMiniFATSectorLocation
  - \_bcf\_file\_header, 45
- flag
  - EmbGeometry\_, 88
- FLAG\_CIRCLE
  - main.c, 473
- FLAG\_CIRCLE\_SHORT
  - main.c, 474
- FLAG\_COMBINE
  - main.c, 474
- FLAG\_CROSS\_STITCH
  - main.c, 474
- FLAG\_ELLIPSE
  - main.c, 474
- FLAG\_ELLIPSE\_SHORT
  - main.c, 474
- FLAG\_FILL
  - main.c, 474
- FLAG\_FILL\_SHORT
  - main.c, 474
- FLAG\_FORMATS
  - main.c, 474
- FLAG\_FORMATS\_SHORT
  - main.c, 474
- FLAG\_FULL\_TEST\_SUITE
  - main.c, 474
- FLAG\_HELP
  - main.c, 474
- FLAG\_HELP\_SHORT
  - main.c, 474
- FLAG\_HILBERT\_CURVE
  - main.c, 474
- FLAG\_LINE
  - main.c, 474
- FLAG\_LINE\_SHORT
  - main.c, 474
- FLAG\_POLYGON
  - main.c, 474
- FLAG\_POLYGON\_SHORT
  - main.c, 474
- FLAG\_POLYLINE



- main.c, [474](#)
- FLAG\_POLYLINE\_SHORT
  - main.c, [475](#)
- FLAG\_QUIET
  - main.c, [475](#)
- FLAG\_QUIET\_SHORT
  - main.c, [475](#)
- FLAG\_RENDER
  - main.c, [475](#)
- FLAG\_RENDER\_SHORT
  - main.c, [475](#)
- FLAG\_SATIN
  - main.c, [475](#)
- FLAG\_SATIN\_SHORT
  - main.c, [475](#)
- FLAG\_SIERPINSKI\_TRIANGLE
  - main.c, [475](#)
- FLAG\_SIMULATE
  - main.c, [475](#)
- FLAG\_STITCH
  - main.c, [475](#)
- FLAG\_STITCH\_SHORT
  - main.c, [475](#)
- FLAG\_TEST
  - main.c, [475](#)
- FLAG\_TO
  - main.c, [475](#)
- FLAG\_TO\_SHORT
  - main.c, [475](#)
- FLAG\_VERBOSE
  - main.c, [475](#)
- FLAG\_VERBOSE\_SHORT
  - main.c, [475](#)
- FLAG\_VERSION
  - main.c, [475](#)
- FLAG\_VERSION\_SHORT
  - main.c, [475](#)
- flagList
  - EmbPath\_, [93](#)
- flags
  - EmbStitch\_, [99](#)
- Flared
  - Geometry, [109](#)
- Fletching
  - Geometry, [109](#)
- floatingChanged
  - CmdPrompt, [55](#)
- floatingChangedToolBar
  - MainWindow, [151](#)
- focusWidget
  - PropertyEditor, [183](#)
  - UndoEditor, [215](#)
- forceRepaint
  - MdiArea, [164](#)
  - SelectBox, [196](#)
- format\_100.c
  - read100, [414](#)
  - write100, [414](#)
- format\_10o.c
  - read10o, [415](#)
  - write10o, [415](#)
- format\_art.c
  - readArt, [415](#)
  - writeArt, [415](#)
- format\_bmc.c
  - readBmc, [416](#)
  - writeBmc, [416](#)
- format\_bro.c
  - readBro, [416](#)
  - writeBro, [416](#)
- format\_cnd.c
  - readCnd, [417](#)
  - writeCnd, [417](#)
- format\_col.c
  - readCol, [418](#)
  - writeCol, [418](#)
- format\_csd.c
  - \_subMask, [419](#)
  - \_xorMask, [419](#)
  - BuildDecryptionTable, [418](#)
  - csd\_decryptArray, [419](#)
  - CsdSubMaskSize, [418](#)
  - CsdXorMaskSize, [418](#)
  - DecodeCsdByte, [419](#)
  - readCsd, [419](#)
  - writeCsd, [419](#)
- format\_csv.c
  - csvStitchFlagToStr, [420](#)
  - csvStrToStitchFlag, [420](#)
  - readCsv, [420](#)
  - writeCsv, [420](#)
- format\_dat.c
  - readDat, [420](#)
  - writeDat, [420](#)
- format\_dem.c
  - readDem, [421](#)
  - writeDem, [421](#)
- format\_dsb.c
  - readDsb, [422](#)
  - writeDsb, [422](#)
- format\_dst.c
  - cci, [423](#)
  - decode\_record\_flags, [423](#)
  - encode\_record, [424](#)
  - readDst, [424](#)
  - set\_dst\_variable, [424](#)
  - writeDst, [424](#)
- format\_dsz.c
  - readDsz, [424](#)
  - writeDsz, [424](#)
- format\_dxf.c
  - readDxf, [425](#)
  - readLine, [425](#)
  - writeDxf, [425](#)
- format\_edr.c
  - readEdr, [425](#)



- writeEdr, [425](#)
- format\_emd.c
  - emdDecode, [426](#)
  - readEmd, [426](#)
  - writeEmd, [426](#)
- format\_exp.c
  - expDecode, [426](#)
  - readExp, [426](#)
  - writeExp, [427](#)
- format\_exy.c
  - decode\_exy\_flags, [427](#)
  - readExy, [427](#)
  - writeExy, [427](#)
- format\_eys.c
  - readEys, [427](#)
  - writeEys, [428](#)
- format\_fxy.c
  - readFxy, [428](#)
  - writeFxy, [428](#)
- format\_gc.c
  - readGc, [428](#)
  - writeGc, [429](#)
- format\_gnc.c
  - readGnc, [429](#)
  - writeGnc, [429](#)
- format\_gt.c
  - readGt, [429](#)
  - writeGt, [430](#)
- format\_hus.c
  - husCompressData, [430](#)
  - husDecodeByte, [430](#)
  - husDecodeStitchType, [430](#)
  - husDecompressData, [430](#)
  - husEncodeByte, [430](#)
  - husEncodeStitchType, [430](#)
  - readHus, [431](#)
  - writeHus, [431](#)
- format\_inb.c
  - readInb, [431](#)
  - writeInb, [431](#)
- format\_inf.c
  - readInf, [431](#)
  - writeInf, [432](#)
- format\_jef.c
  - jefDecode, [432](#)
  - jefEncode, [432](#)
  - jefGetHoopSize, [432](#)
  - jefSetHoopFromId, [432](#)
  - read\_hoop, [432](#)
  - readJef, [433](#)
  - writeJef, [433](#)
- format\_ksm.c
  - ksmEncode, [433](#)
  - readKsm, [433](#)
  - writeKsm, [433](#)
- format\_max.c
  - max\_header, [434](#)
  - readMax, [434](#)
  - writeMax, [434](#)
- format\_mit.c
  - readMit, [434](#)
  - writeMit, [435](#)
- format\_new.c
  - readNew, [435](#)
  - writeNew, [435](#)
- format\_ofm.c
  - ofmDecode, [436](#)
  - ofmReadBlockHeader, [436](#)
  - ofmReadClass, [436](#)
  - ofmReadColorChange, [436](#)
  - ofmReadExpanded, [436](#)
  - ofmReadLibrary, [436](#)
  - ofmReadThreads, [436](#)
  - readOfm, [436](#)
  - writeOfm, [436](#)
- format\_pcd.c
  - readPcd, [437](#)
  - writePcd, [437](#)
- format\_pcm.c
  - readPcm, [437](#)
  - writePcm, [437](#)
- format\_pcq.c
  - readPcq, [438](#)
  - writePcq, [438](#)
- format\_pcs.c
  - readPcs, [438](#)
  - writePcs, [438](#)
- format\_pec.c
  - pecEncode, [439](#)
  - pecEncodeJump, [439](#)
  - pecEncodeStop, [439](#)
  - readPec, [439](#)
  - readPecStitches, [439](#)
  - writeImage, [439](#)
  - writePec, [439](#)
  - writePecStitches, [439](#)
- format\_pel.c
  - readPel, [440](#)
  - writePel, [440](#)
- format\_pem.c
  - readPem, [440](#)
  - writePem, [440](#)
- format\_pes.c
  - pes\_version, [443](#)
  - pes\_version\_strings, [443](#)
  - pesWriteEmbOneSection, [441](#)
  - pesWriteSewSegSection, [441](#)
  - readDescriptions, [441](#)
  - readFeatherPatterns, [441](#)
  - readHoopName, [441](#)
  - readImageString, [442](#)
  - readMotifPatterns, [442](#)
  - readPes, [442](#)
  - readPESHeaderV10, [442](#)
  - readPESHeaderV5, [442](#)
  - readPESHeaderV6, [442](#)

- readPESHeaderV7, 442
- readPESHeaderV8, 442
- readPESHeaderV9, 442
- readProgrammableFills, 442
- readThreads, 442
- writePes, 443
- format\_phb.c
  - readPhb, 443
  - writePhb, 443
- format\_phc.c
  - readPhc, 444
  - writePhc, 444
- format\_plt.c
  - readPlt, 444
  - writePlt, 444
- format\_rgb.c
  - readRgb, 445
  - writeRgb, 445
- format\_sew.c
  - readSew, 445
  - sewDecode, 445
  - writeSew, 445
- format\_shv.c
  - readShv, 446
  - shvDecode, 446
  - shvDecodeShort, 446
  - writeShv, 446
- format\_sst.c
  - readSst, 446
  - writeSst, 447
- format\_stx.c
  - readStx, 447
  - stxReadThread, 447
  - writeStx, 447
- format\_svg.c
  - attributeList, 448
  - current\_element\_id, 448
  - currentAttribute, 448
  - currentValue, 448
  - n\_attributes, 448
  - readSvg, 448
  - svgCreator, 448
  - svgExpect, 448
  - svgMultiValue, 448
  - writeSvg, 448
- format\_t01.c
  - readT01, 449
  - writeT01, 449
- format\_t09.c
  - readT09, 449
  - writeT09, 449
- format\_tap.c
  - decode\_tap\_record\_flags, 450
  - encode\_tap\_record, 450
  - readTap, 450
  - writeTap, 450
- format\_thr.c
  - readThr, 450
  - writeThr, 451
- format\_txt.c
  - readTxt, 451
  - writeTxt, 451
- format\_u00.c
  - readU00, 451
  - writeU00, 452
- format\_u01.c
  - readU01, 452
  - writeU01, 452
- format\_vip.c
  - readVip, 453
  - vipCompressData, 453
  - vipDecodeByte, 453
  - vipDecodeStitchType, 453
  - vipDecodingTable, 453
  - vipDecompressData, 453
  - vipEncodeByte, 453
  - vipEncodeStitchType, 453
  - writeVip, 453
- format\_vp3.c
  - readVp3, 454
  - vp3Decode, 454
  - vp3DecodeInt16, 454
  - vp3PatchByteCount, 454
  - vp3ReadHoopSection, 454
  - vp3ReadString, 455
  - vp3WriteString, 455
  - vp3WriteStringLen, 455
  - writeVp3, 455
- format\_xxx.c
  - readXxx, 455
  - writeXxx, 455
  - xxxDecodeByte, 455
  - xxxEncodeDesign, 456
  - xxxEncodeStitch, 456
  - xxxEncodeStop, 456
- format\_zsk.c
  - readZsk, 456
  - writeZsk, 456
- formatFilterOpen
  - MainWindow, 161
- formatFilterSave
  - MainWindow, 161
- formats.c
  - binaryWriteInt, 411
  - binaryWriteIntBE, 411
  - binaryWriteShort, 411
  - binaryWriteUInt, 411
  - binaryWriteUIntBE, 411
  - binaryWriteUShort, 411
  - binaryWriteUShortBE, 411
  - emb\_identify\_format, 412
  - embFormat\_getExtension, 412
  - embPattern\_read, 412
  - embPattern\_readAuto, 412
  - embPattern\_write, 412
  - embPattern\_writeAuto, 412

- formatTable, 413
- fpad, 413
- fread\_int16, 413
- fread\_int32\_be, 413
- fread\_uint16, 413
- imageWithFrame, 414
- safe\_free, 413
- formatTable
  - embroidery.h, 347
  - formats.c, 413
- formatType
  - SaveObject, 195
- Fortron, 383, 428
- fourier\_series
  - objects.cpp, 305
- fpad
  - embroidery\_internal.h, 378
  - formats.c, 413
- fread\_int16
  - embroidery\_internal.h, 378
  - formats.c, 413
- fread\_int32\_be
  - embroidery\_internal.h, 379
  - formats.c, 413
- fread\_uint16
  - embroidery\_internal.h, 379
  - formats.c, 413
- fromCenter
  - UndoableCommand, 213
- fromTransform
  - UndoableCommand, 213
- Fufu\_Polyester
  - embroidery.h, 327
- Fufu\_Rayon
  - embroidery.h, 327
- full\_test\_matrix
  - embroidery.h, 345
- FUNCTION\_TYPE
  - embroidermodder.h, 235
- functions.c
  - degrees, 464
  - emb\_round, 464
  - radians, 464
- fx, 9, 383, 428
- g
  - EmbColor\_, 82
- general\_props
  - settings-dialog.cpp, 309
- generate\_dragon\_curve
  - fill.c, 408
- Geometry, 103
  - ~Geometry, 112
  - allGripPoints, 113
  - arcEndPoint, 135
  - arcMidPoint, 135
  - arcStartPoint, 135
  - ArrowStyle, 108
  - arrowStyleAngle, 135
  - arrowStyleLength, 135
  - arrowStylePath, 135
  - boundingRect, 113
  - Box, 109
  - calculateArcData, 113
  - circle\_click, 113
  - Closed, 109
  - curved, 136
  - Dot, 109
  - drawRubberLine, 113
  - filled, 136
  - findIndex, 114
  - Flared, 109
  - Fletching, 109
  - Geometry, 109–112
  - gripEdit, 114
  - gripIndex, 136
  - init, 114
  - init\_arc, 114
  - init\_circle, 115
  - init\_ellipse, 115
  - init\_line, 115
  - init\_path, 116
  - init\_point, 116
  - init\_rect, 117
  - init\_text\_single, 117
  - lineStyle, 109
  - lineStyleAngle, 136
  - lineStyleLength, 136
  - lineStylePath, 136
  - lwtPen, 136
  - mouseSnapPoint, 117
  - NoArrow, 109
  - NoLine, 109
  - normalPath, 136
  - objectAngle, 118
  - objectArcLength, 118
  - objectArea, 118
  - objectBottomLeft, 118
  - objectBottomRight, 119
  - objectCenter, 119
  - objectChord, 119
  - objectCircumference, 119
  - objectClockwise, 119
  - objectCopyPath, 119
  - objectDelta, 120
  - objectDiameter, 120
  - objectDiameterMajor, 120
  - objectDiameterMinor, 120
  - objectEndAngle, 120
  - objectEndPoint, 120
  - objectEndPoint1, 120
  - objectEndPoint2, 121
  - objectHeight, 121
  - objectIncludedAngle, 121
  - objectLength, 121
  - objectLineType, 121
  - objectLineWeight, 121

- objectMidPoint, [122](#)
- objectPos, [122](#)
- objectQuadrant0, [122](#)
- objectQuadrant180, [122](#)
- objectQuadrant270, [122](#)
- objectQuadrant90, [122](#)
- objectRadius, [122](#)
- objectRadiusMajor, [123](#)
- objectRadiusMinor, [123](#)
- objectRubberPoint, [123](#)
- objectRubberText, [123](#)
- objectSavePath, [123](#)
- objectSavePathList, [123](#)
- objectStartAngle, [124](#)
- objectStartPoint, [124](#)
- objectTopLeft, [124](#)
- objectTopRight, [124](#)
- objectWidth, [124](#)
- objectX, [124](#)
- objectX1, [125](#)
- objectX2, [125](#)
- objectY, [125](#)
- objectY1, [125](#)
- objectY2, [125](#)
- objID, [136](#)
- objLine, [136](#)
- objPen, [136](#)
- objRubberMode, [137](#)
- objRubberPoints, [137](#)
- objRubberTexts, [137](#)
- objText, [137](#)
- objTextBackward, [137](#)
- objTextFont, [137](#)
- objTextJustify, [137](#)
- objTextPath, [137](#)
- objTextUpsideDown, [137](#)
- Open, [109](#)
- paint, [125](#)
- properties, [137](#)
- realRender, [125](#)
- rect, [126](#)
- script\_click, [126](#)
- script\_context, [126](#)
- script\_main, [126](#)
- script\_prompt, [126](#)
- setLine, [126](#), [127](#)
- setObjectArea, [127](#)
- setObjectCenter, [127](#)
- setObjectCenterX, [127](#)
- setObjectCenterY, [127](#)
- setObjectCircumference, [127](#)
- setObjectDiameter, [127](#)
- setObjectDiameterMajor, [128](#)
- setObjectDiameterMinor, [128](#)
- setObjectEndAngle, [128](#)
- setObjectEndPoint, [128](#)
- setObjectEndPoint1, [128](#)
- setObjectEndPoint2, [129](#)
- setObjectLineWeight, [129](#)
- setObjectMidPoint, [129](#)
- setObjectPos, [129](#)
- setObjectRadius, [130](#)
- setObjectRadiusMajor, [130](#)
- setObjectRadiusMinor, [130](#)
- setObjectRect, [130](#)
- setObjectSize, [130](#)
- setObjectStartAngle, [130](#)
- setObjectStartPoint, [131](#)
- setObjectText, [131](#)
- setObjectTextBackward, [131](#)
- setObjectTextBold, [131](#)
- setObjectTextFont, [131](#)
- setObjectTextItalic, [131](#)
- setObjectTextJustify, [132](#)
- setObjectTextOverline, [132](#)
- setObjectTextSize, [132](#)
- setObjectTextStrikeOut, [132](#)
- setObjectTextStyle, [132](#)
- setObjectTextUnderline, [133](#)
- setObjectTextUpsideDown, [133](#)
- setObjectX, [133](#)
- setObjectY, [133](#)
- setRect, [133](#)
- subPathList, [134](#)
- Tick, [109](#)
- Type, [137](#)
- type, [134](#)
- updateArcRect, [134](#)
- updateLeader, [134](#)
- updatePath, [134](#)
- updateRubber, [135](#)
- vulcanize, [135](#)
- x\_values, [138](#)
- y\_values, [138](#)
- geometry
  - EmbArray\_, [79](#)
  - EmbLayer\_, [91](#)
  - EmbPattern\_, [94](#)
- geometry.c
  - embGeometry\_boundingRect, [457](#)
  - embGeometry\_free, [457](#)
  - embGeometry\_init, [457](#)
  - embGeometry\_move, [457](#)
  - embGeometry\_vulcanize, [457](#)
- get\_bool
  - embroidermodder.h, [240](#)
  - interface.cpp, [267](#)
- get\_int
  - embroidermodder.h, [240](#)
  - interface.cpp, [267](#)
- get\_n\_reals
  - interface.cpp, [267](#)
- get\_qstr
  - embroidermodder.h, [240](#)
  - interface.cpp, [267](#)
- get\_real

- embroidermodder.h, 240
- interface.cpp, 268
- get\_str
  - embroidermodder.h, 240
  - interface.cpp, 268
- get\_str\_list
  - embroidermodder.h, 240
  - interface.cpp, 268
- get\_trim\_bounds
  - main.c, 480
- get\_uint
  - embroidermodder.h, 240
  - interface.cpp, 268
- getArcCenter
  - arc.c, 461
  - embroidery.h, 345
- getArcDataFromBulge
  - arc.c, 461
  - embroidery.h, 345
- getCircleCircleIntersections
  - circle.c, 462
  - embroidery.h, 346
- getCircleTangentPoints
  - circle.c, 462
  - embroidery.h, 346
- getCurrentColor
  - MainWindow, 152
- getCurrentLayer
  - MainWindow, 152
- getCurrentLineType
  - MainWindow, 152
- getCurrentLineWeight
  - MainWindow, 152
- GetFile
  - embroidery\_internal.h, 379
  - main.c, 480
- getFileSeparator
  - MainWindow, 152
- getInfo
  - EmbDetailsDialog, 83
- getShortCurrentFile
  - MdiWindow, 171
- getUndoStack
  - View, 220
- gnc, 9, 383, 429
- Gold Thread, 383, 430
- Great Notions, 383, 429
- greedy\_algorithm
  - fill.c, 408
- GREEN\_TERM\_COLOR
  - embroidery\_internal.h, 368
- gridColor
  - View, 225
- gridPath
  - View, 225
- gripBaseObj
  - View, 225
- gripColorCool
  - View, 225
- gripColorHot
  - View, 226
- gripEdit
  - Geometry, 114
- gripIndex
  - Geometry, 136
- grippingActive
  - View, 226
- gripSize
  - View, 226
- group\_box\_data
  - property-editor.cpp, 307
- group\_box\_types
  - property-editor.cpp, 307
- groupBoxes
  - embroidermodder.h, 247
  - mainwindow.cpp, 303
- gscene
  - MdiWindow, 176
  - SaveObject, 195
  - View, 226
- gt, 9, 383, 430
- gview
  - MdiWindow, 176
  - UndoableCommand, 213
- handleMoved
  - CmdPromptHandle, 61
- handlePressed
  - CmdPromptHandle, 61
- handleReleased
  - CmdPromptHandle, 61
- Happy, 450
- hashDeletedObjects
  - View, 226
- haveExtraDIFATSectors
  - main.c, 480
- header
  - \_bcf\_file, 43
  - em2\_dev\_script, 39
- height
  - \_vp3Hoop, 48
  - EmblImage\_, 89
- help\_action
  - mainwindow.cpp, 290
- Hemingworth\_Polyester
  - embroidery.h, 328
- hex\_code
  - thread\_color\_, 209
- hideAllGroups
  - PropertyEditor, 181
- hideUnimplemented
  - MainWindow, 152
- hilbert\_curve
  - embroidery.h, 346
  - fill.c, 408
- hilbert\_curve\_l\_system
  - fill.c, 409

- historyAppended
  - CmdPrompt, 55
  - CmdPromptHistory, 65
- home
  - EmbPattern\_, 94
- HOOP\_110X110
  - embroidery\_internal.h, 368
- HOOP\_126X110
  - embroidery\_internal.h, 368
- HOOP\_140X200
  - embroidery\_internal.h, 369
- HOOP\_230X200
  - embroidery\_internal.h, 369
- HOOP\_50X50
  - embroidery\_internal.h, 369
- hoop\_height
  - EmbPattern\_, 94
- hoop\_padding, 138
  - bottom, 138
  - left, 138
  - right, 138
  - top, 138
- hoop\_width
  - EmbPattern\_, 94
- hoopSize
  - ThredHeader\_, 210
- hoopX
  - ThredExtension\_, 209
- hoopY
  - ThredExtension\_, 209
- hour
  - EmbTime\_, 101
- Huffman, 139
  - default\_value, 139
  - lengths, 139
  - nlengths, 139
  - ntable, 139
  - table, 139
  - table\_width, 139
- huffman
  - embroidery\_internal.h, 371
- huffman\_build\_table
  - compress.c, 314
  - embroidery\_internal.h, 379
- huffman\_lookup
  - compress.c, 314
- huffman\_lookup\_data
  - compress.c, 315
- huffman\_table\_lookup
  - embroidery\_internal.h, 379
- hus, 9, 430
- hus\_compress
  - compress.c, 314
  - embroidery\_internal.h, 379
- hus\_decompress
  - compress.c, 314
  - embroidery\_internal.h, 379
- hus\_thread
  - embroidery.h, 328
- husCompressData
  - format\_hus.c, 430
- husDecodeByte
  - format\_hus.c, 430
- husDecodeStitchType
  - format\_hus.c, 430
- husDecompressData
  - format\_hus.c, 430
- husEncodeByte
  - format\_hus.c, 430
- husEncodeStitchType
  - format\_hus.c, 430
- Husqvarna Viking, 430, 446
- husThreads
  - embroidery.h, 347
  - thread-color.c, 488
- i
  - Node\_, 177
- icon\_action
  - mainwindow.cpp, 291
- iconDir
  - PropertyEditor, 183
  - UndoEditor, 215
- iconResize
  - MainWindow, 153
- iconSize
  - PropertyEditor, 184
  - UndoEditor, 215
- id
  - UndoableCommand, 212
- image.c
  - image\_diff, 470
  - writelnImage, 471
- image\_diff
  - image.c, 470
- ImageWidget, 140
  - ~ImageWidget, 140
  - ImageWidget, 140
  - img, 141
  - load, 141
  - paintEvent, 141
  - save, 141
- imageWithFrame
  - embroidery\_internal.h, 395
  - formats.c, 414
- img
  - ImageWidget, 141
- imgWidget
  - PreviewDialog, 178
- inb, 9, 384, 431
- Inbro, 384, 431
- include\_action
  - mainwindow.cpp, 291
- inf, 384, 432
- init
  - Geometry, 114
- init\_action

- mainwindow.cpp, 291
- init\_arc
  - Geometry, 114
- init\_circle
  - Geometry, 115
- init\_ellipse
  - Geometry, 115
- init\_line
  - Geometry, 115
- init\_path
  - Geometry, 116
- init\_point
  - Geometry, 116
- init\_rect
  - Geometry, 117
- init\_text\_single
  - Geometry, 117
- input\_data
  - Compress, 76
- input\_length
  - Compress, 76
- INT\_TYPE
  - embroidermodder.h, 235
- interface.cpp
  - add\_to\_path, 267
  - debug\_message, 267
  - degrees\_\_, 267
  - get\_bool, 267
  - get\_int, 267
  - get\_n\_reals, 267
  - get\_qstr, 267
  - get\_real, 268
  - get\_str, 268
  - get\_str\_list, 268
  - get\_uint, 268
  - make\_checkbox, 268
  - make\_spinbox, 268
  - make\_ui\_element, 268
  - node\_bool, 268
  - node\_int, 269
  - node\_qstr, 269
  - node\_real, 269
  - node\_str, 269
  - node\_str\_list, 269
  - node\_uint, 270
  - operator\*, 270
  - operator+, 270
  - operator-, 270
  - radians\_\_, 270
  - set\_enabled, 270
  - set\_visibility, 271
  - to\_EmbVector, 271
  - to\_qlist, 271
  - to\_QPointF, 271
  - to\_string\_vector, 271
  - to\_vector, 272
  - tokenize, 272
  - translate\_str, 272
- is\_int\_action
  - mainwindow.cpp, 291
- Isacord\_Polyester
  - embroidery.h, 328
- Isafil\_Rayon
  - embroidery.h, 328
- isBlinking
  - CmdPromptInput, 74
- isCommandActive
  - MainWindow, 153
- isLwtEnabled
  - View, 220
- isRealEnabled
  - View, 220
- isShiftPressed
  - MainWindow, 153
- Janome, 432, 445
- jef, 9, 432
- jef\_thread
  - embroidery.h, 328
- jefDecode
  - format\_jef.c, 432
- jefEncode
  - format\_jef.c, 432
- jefGetHoopSize
  - format\_jef.c, 432
- jefSetHoopFromId
  - format\_jef.c, 432
- jefThreads
  - embroidery.h, 347
  - thread-color.c, 488
- join\_short\_stitches
  - fill.c, 409
- JUMP
  - embroidery.h, 328
- ksm, 9, 433
- ksmEncode
  - format\_ksm.c, 433
- L\_system
  - embroidery.h, 331
- labels
  - embroidermodder.h, 247
  - mainwindow.cpp, 303
- labelTipOfTheDay
  - mainwindow.cpp, 303
- lastCmd
  - CmdPromptInput, 74
- layer
  - EmbPattern\_, 95
- layer\_manager\_action
  - mainwindow.cpp, 291
- layer\_previous\_action
  - mainwindow.cpp, 292
- LayerManager, 142
  - ~LayerManager, 142
  - addLayer, 142

- LayerManager, [142](#)
  - layerModel, [143](#)
  - layerModelSorted, [143](#)
  - treeView, [143](#)
- layerModel
  - LayerManager, [143](#)
- layerModelSorted
  - LayerManager, [143](#)
- layerSelector
  - MainWindow, [161](#)
- layerSelectorIndexChanged
  - MainWindow, [153](#)
- layoutState
  - MainWindow, [161](#)
- left
  - \_vp3Hoop, [48](#)
  - EmbRect\_, [97](#)
  - hoop\_padding, [138](#)
- left2
  - \_vp3Hoop, [48](#)
- leftBrush
  - SelectBox, [197](#)
- leftBrushColor
  - SelectBox, [197](#)
- leftPen
  - SelectBox, [197](#)
- leftPenColor
  - SelectBox, [197](#)
- leftSiblingId
  - \_bcf\_directory\_entry, [42](#)
- length
  - EmbArray\_, [79](#)
  - EmbSatinOutline\_, [98](#)
  - ThredHeader\_, [210](#)
- lengths
  - Huffman, [139](#)
- LIBEMBROIDERY\_EMBEDDED\_VERSION
  - embroidery.h, [328](#)
- lindenmayer\_system
  - embroidery.h, [346](#)
  - fill.c, [409](#)
- line
  - EmbGeometry\_, [88](#)
- line.c
  - embLine\_intersectionPoint, [465](#)
  - embLine\_normalVector, [465](#)
  - embLine\_toVector, [465](#)
- lineEdits
  - embroidermodder.h, [247](#)
  - mainwindow.cpp, [303](#)
- lineStyle
  - Geometry, [109](#)
- lineStyleAngle
  - Geometry, [136](#)
- lineStyleLength
  - Geometry, [136](#)
- lineStylePath
  - Geometry, [136](#)
- LINETO
  - embroidery\_internal.h, [369](#)
- lineType
  - EmbGeometry\_, [88](#)
  - EmbLine\_, [92](#)
  - EmbPath\_, [93](#)
  - EmbPoint\_, [95](#)
- linetypeSelector
  - MainWindow, [161](#)
- linetypeSelectorIndexChanged
  - MainWindow, [153](#)
- lineweightSelector
  - MainWindow, [161](#)
- lineweightSelectorIndexChanged
  - MainWindow, [153](#)
- listMdiWin
  - MainWindow, [161](#)
- load
  - ImageWidget, [141](#)
- load\_group\_box\_data\_from\_table
  - property-editor.cpp, [306](#)
- loadFatFromSector
  - embroidery\_internal.h, [380](#)
  - main.c, [480](#)
- loadFile
  - MdiWindow, [171](#)
- loadFormats
  - MainWindow, [153](#)
- loadRulerSettings
  - View, [220](#)
- logPromptInput
  - MainWindow, [154](#)
  - MdiWindow, [172](#)
- LSYSTEM, [143](#)
  - alphabet, [143](#)
  - axiom, [144](#)
  - constants, [144](#)
  - rules, [144](#)
- lwtPen
  - Geometry, [136](#)
- Madeira\_Polyester
  - embroidery.h, [328](#)
- Madeira\_Rayon
  - embroidery.h, [328](#)
- magicCode
  - VipHeader\_, [229](#)
- main
  - embroidermodder.cpp, [230](#)
- main.c
  - bcf\_difat\_create, [476](#)
  - bcf\_directory\_free, [476](#)
  - bcf\_file\_free, [476](#)
  - bcfFile\_read, [476](#)
  - bcfFileFat\_create, [476](#)
  - bcfFileHeader\_read, [476](#)
  - binaryReadString, [476](#)
  - binaryReadUnicodeString, [477](#)
  - black\_thread, [482](#)



- check\_header\_present, 477
- CompoundFileDirectory, 477
- CompoundFileDirectoryEntry, 477
- copy\_trim, 477
- difatEntriesInHeader, 482
- emb\_error, 482
- emb\_optOut, 477
- emb\_readline, 478
- emb\_verbose, 482
- embArc\_print, 478
- embColor\_distance, 478
- embColor\_read, 478
- embColor\_write, 478
- embConstantPi, 482
- embSatinOutline\_generateSatinOutline, 478
- embSatinOutline\_renderStitches, 478
- embThread\_findNearestColor, 479
- embThread\_findNearestThread, 479
- embThread\_getRandom, 479
- embTime\_initNow, 479
- embTime\_time, 479
- embVector\_print, 479
- entriesInDifatSector, 480
- FLAG\_CIRCLE, 473
- FLAG\_CIRCLE\_SHORT, 474
- FLAG\_COMBINE, 474
- FLAG\_CROSS\_STITCH, 474
- FLAG\_ELLIPSE, 474
- FLAG\_ELLIPSE\_SHORT, 474
- FLAG\_FILL, 474
- FLAG\_FILL\_SHORT, 474
- FLAG\_FORMATS, 474
- FLAG\_FORMATS\_SHORT, 474
- FLAG\_FULL\_TEST\_SUITE, 474
- FLAG\_HELP, 474
- FLAG\_HELP\_SHORT, 474
- FLAG\_HILBERT\_CURVE, 474
- FLAG\_LINE, 474
- FLAG\_LINE\_SHORT, 474
- FLAG\_POLYGON, 474
- FLAG\_POLYGON\_SHORT, 474
- FLAG\_POLYLINE, 474
- FLAG\_POLYLINE\_SHORT, 475
- FLAG\_QUIET, 475
- FLAG\_QUIET\_SHORT, 475
- FLAG\_RENDER, 475
- FLAG\_RENDER\_SHORT, 475
- FLAG\_SATIN, 475
- FLAG\_SATIN\_SHORT, 475
- FLAG\_SIERPINSKI\_TRIANGLE, 475
- FLAG\_SIMULATE, 475
- FLAG\_STITCH, 475
- FLAG\_STITCH\_SHORT, 475
- FLAG\_TEST, 475
- FLAG\_TO, 475
- FLAG\_TO\_SHORT, 475
- FLAG\_VERBOSE, 475
- FLAG\_VERBOSE\_SHORT, 475
- FLAG\_VERSION, 475
- FLAG\_VERSION\_SHORT, 475
- get\_trim\_bounds, 480
- GetFile, 480
- haveExtraDIFATSectors, 480
- loadFatFromSector, 480
- NUM\_FLAGS, 476
- parseDIFATSectors, 480
- parseDirectoryEntryName, 480
- parseTime, 480
- readFullSector, 481
- readNextSector, 481
- sectorSize, 481
- seekToSector, 481
- sizeofChainingEntryAtEndOfDifatSector, 482
- sizeofDifatEntry, 482
- sizeofDirectoryEntry, 482
- sizeofFatEntry, 482
- stringInArray, 481
- WHITESPACE, 482
- write\_24bit, 481
- mainWidget
  - EmbDetailsDialog, 84
- MainWindow, 144
  - ~MainWindow, 148
  - about, 148
  - activeCommand, 148
  - activeMdiWindow, 148
  - activeUndoStack, 148
  - buttonTipOfTheDayClicked, 149
  - checkForUpdates, 149
  - closeEvent, 149
  - closeToolBar, 149
  - colorSelector, 161
  - colorSelectorIndexChanged, 149
  - create\_icon, 150
  - create\_toolbar, 150
  - createAllActions, 150
  - createAllMenus, 150
  - createAllToolbars, 151
  - cutCopyObjectList, 161
  - deletePressed, 151
  - docIndex, 161
  - escapePressed, 151
  - findMdiWindow, 151
  - floatingChangedToolBar, 151
  - formatFilterOpen, 161
  - formatFilterSave, 161
  - getCurrentColor, 152
  - getCurrentLayer, 152
  - getCurrentLineType, 152
  - getCurrentLineWeight, 152
  - getFileSeparator, 152
  - hideUnimplemented, 152
  - iconResize, 153
  - isCommandActive, 153
  - isShiftPressed, 153
  - layerSelector, 161

layerSelectorIndexChanged, 153  
 layoutState, 161  
 linetypeSelector, 161  
 linetypeSelectorIndexChanged, 153  
 linewidthSelector, 161  
 linewidthSelectorIndexChanged, 153  
 listMdiWin, 161  
 loadFormats, 153  
 logPromptInput, 154  
 MainWindow, 148  
 myFileSeparator, 162  
 newFile, 154  
 numOfDocs, 162  
 onCloseMdiWin, 154  
 onCloseWindow, 154  
 onWindowActivated, 154  
 openFile, 154  
 openFilesSelected, 155  
 openrecentfile, 155  
 pickAddModeToggled, 155  
 platformString, 155  
 promptHistoryAppended, 155  
 promptInputNext, 156  
 promptInputPrevious, 156  
 quit, 156  
 recentMenuAboutToShow, 156  
 resizeEvent, 156  
 saveasfile, 156  
 savefile, 156  
 setShiftPressed, 156  
 setShiftReleased, 157  
 setTextFont, 157  
 setTextSize, 157  
 settingsPrompt, 157  
 setUndoCleanIcon, 157  
 shiftKeyPressedState, 162  
 stub\_testing, 157  
 textFontSelector, 162  
 textFontSelectorCurrentFontChanged, 158  
 textSizeSelector, 162  
 textSizeSelectorIndexChanged, 158  
 tipOfTheDay, 158  
 toggleGrid, 158  
 toggleLwt, 158  
 toggleRuler, 158  
 updateAllViewBackgroundColors, 158  
 updateAllViewCrossHairColors, 159  
 updateAllViewGridColors, 159  
 updateAllViewRulerColors, 159  
 updateAllViewScrollBars, 159  
 updateAllViewSelectBoxColors, 160  
 updateMenuToolbarStatusbar, 160  
 updatePickAddMode, 160  
 windowMenuAboutToShow, 160  
 windowMenuActivated, 160  
 mainwindow-menus.cpp  
   create\_menu, 273  
 mainwindow.cpp  
   \_mainWin, 302  
   about\_action, 280  
   actionHash, 302  
   activeScene, 280  
   activeView, 280  
   actuator, 280  
   add\_arc\_action, 281  
   add\_circle\_action, 281  
   add\_dim\_leader\_action, 281  
   add\_ellipse\_action, 281  
   add\_geometry\_action, 281  
   add\_horizontal\_dimension\_action, 282  
   add\_image\_action, 282  
   add\_infinite\_line\_action, 282  
   add\_line\_action, 282  
   add\_path\_action, 282  
   add\_point\_action, 282  
   add\_polygon\_action, 283  
   add\_polyline\_action, 283  
   add\_ray\_action, 283  
   add\_rectangle\_action, 283  
   add\_regular\_polygon\_action, 283  
   add\_rounded\_rectangle\_action, 283  
   add\_rubber\_action, 284  
   add\_slot\_action, 284  
   add\_text\_multi\_action, 284  
   add\_text\_single\_action, 284  
   add\_to\_selection\_action, 284  
   add\_triangle\_action, 286  
   add\_vertical\_dimension\_action, 286  
   alert\_action, 286  
   allow\_rubber\_action, 286  
   append\_history\_action, 286  
   append\_prompt\_history\_action, 287  
   blink\_prompt\_action, 287  
   calculate\_angle\_action, 287  
   calculate\_distance\_action, 287  
   changelog\_action, 287  
   checkboxes, 302  
   checkBoxTipOfTheDay, 302  
   clear\_rubber\_action, 288  
   clear\_selection\_action, 288  
   comboBoxes, 302  
   command\_map, 303  
   config, 303  
   config\_tables, 303  
   construct\_command, 288  
   convert\_args\_to\_type, 288  
   copy\_action, 288  
   copy\_selected\_action, 289  
   cut\_action, 289  
   cut\_selected\_action, 289  
   day\_vision\_action, 289  
   debug\_action, 289  
   delete\_selected\_action, 289  
   design\_details\_action, 289  
   dialog, 303  
   disable\_action, 289

do\_nothing\_action, 290  
dockPropEdit, 303  
dockUndoEdit, 303  
doubleSpinBoxes, 303  
end\_action, 290  
error\_action, 290  
groupBoxes, 303  
help\_action, 290  
icon\_action, 291  
include\_action, 291  
init\_action, 291  
is\_int\_action, 291  
labels, 303  
labelTipOfTheDay, 303  
layer\_manager\_action, 291  
layer\_previous\_action, 292  
lineEdits, 303  
make\_layer\_active\_action, 292  
mdiArea, 303  
menuHash, 303  
messagebox\_action, 292  
mirror\_selected\_action, 292  
mouse\_x\_action, 292  
mouse\_y\_action, 292  
move\_selected\_action, 293  
new\_action, 293  
night\_vision\_action, 293  
no\_argument\_debug, 293  
num\_selected\_action, 293  
OBJ\_LTYPE\_CENTER, 279  
OBJ\_LTYPE\_CONT, 279  
OBJ\_LTYPE\_DOT, 279  
OBJ\_LTYPE\_FISHBONE, 279  
OBJ\_LTYPE\_HIDDEN, 279  
OBJ\_LTYPE\_PHANTOM, 279  
OBJ\_LTYPE\_RUNNING, 279  
OBJ\_LTYPE\_SATIN, 279  
OBJ\_LTYPE\_VALUES, 278  
OBJ\_LTYPE\_ZIGZAG, 279  
OBJ\_LWT\_01, 279  
OBJ\_LWT\_02, 279  
OBJ\_LWT\_03, 279  
OBJ\_LWT\_04, 279  
OBJ\_LWT\_05, 279  
OBJ\_LWT\_06, 279  
OBJ\_LWT\_07, 279  
OBJ\_LWT\_08, 279  
OBJ\_LWT\_09, 279  
OBJ\_LWT\_10, 279  
OBJ\_LWT\_11, 279  
OBJ\_LWT\_12, 279  
OBJ\_LWT\_13, 279  
OBJ\_LWT\_14, 279  
OBJ\_LWT\_15, 279  
OBJ\_LWT\_16, 279  
OBJ\_LWT\_17, 279  
OBJ\_LWT\_18, 279  
OBJ\_LWT\_19, 279  
OBJ\_LWT\_20, 279  
OBJ\_LWT\_21, 279  
OBJ\_LWT\_22, 279  
OBJ\_LWT\_23, 279  
OBJ\_LWT\_24, 279  
OBJ\_LWT\_BYBLOCK, 279  
OBJ\_LWT\_BYLAYER, 279  
OBJ\_LWT\_DEFAULT, 279  
OBJ\_LWT\_VALUES, 279  
OBJ\_SNAP\_APPINTERSECTION, 280  
OBJ\_SNAP\_CENTER, 280  
OBJ\_SNAP\_ENDPOINT, 280  
OBJ\_SNAP\_EXTENSION, 280  
OBJ\_SNAP\_INSERTION, 280  
OBJ\_SNAP\_INTERSECTION, 280  
OBJ\_SNAP\_MIDPOINT, 280  
OBJ\_SNAP\_NEAREST, 280  
OBJ\_SNAP\_NODE, 280  
OBJ\_SNAP\_NULL, 279  
OBJ\_SNAP\_PARALLEL, 280  
OBJ\_SNAP\_PERPENDICULAR, 280  
OBJ\_SNAP\_QUADRANT, 280  
OBJ\_SNAP\_TANGENT, 280  
OBJ\_SNAP\_VALUES, 279  
open\_action, 293  
pan\_action, 294  
paste\_action, 294  
paste\_selected\_action, 294  
perpendicular\_distance\_action, 294  
platform\_action, 294  
platformString, 295  
preview\_off\_action, 295  
preview\_on\_action, 295  
print\_action, 295  
print\_area\_action, 296  
prompt, 303  
qsnap\_x\_action, 296  
qsnap\_y\_action, 296  
quit\_action, 296  
read\_configuration, 296  
read\_string\_list\_setting, 296  
read\_string\_setting, 297  
redo\_action, 297  
rotate\_selected\_action, 297  
rubber\_action, 297  
rubber\_modes, 303  
run\_script, 297  
run\_script\_file, 297  
scale\_selected\_action, 298  
scripts, 304  
select\_all\_action, 298  
set\_background\_color\_action, 298  
set\_crosshair\_color\_action, 298  
set\_cursor\_shape\_action, 298  
set\_grid\_color\_action, 299  
set\_prompt\_prefix\_action, 299  
set\_rubber\_filter\_action, 299  
set\_rubber\_mode\_action, 299

- set\_rubber\_point\_action, 299
- set\_rubber\_text\_action, 299
- SetRubberText, 300
- SetTextAngle\_action, 300
- settings, 304
- settings\_dialog\_action, 300
- spare\_rubber\_action, 300
- spinBoxes, 304
- statusbar, 304
- subMenuHash, 304
- tip\_of\_the\_day\_action, 300
- todo\_action, 300
- toolbarHash, 304
- toolButtons, 304
- undo\_action, 301
- validFileFormat, 301
- validRGB, 301
- version\_action, 301
- vulcanize\_action, 301
- whats\_this\_action, 301
- window\_action, 302
- wizardTipOfTheDay, 304
- zoom\_action, 302
- majorVersion
  - \_bcf\_file\_header, 45
- make\_checkbox
  - embroidermodder.h, 240
  - interface.cpp, 268
- make\_editing\_copy
  - settings-dialog.cpp, 308
- make\_layer\_active\_action
  - mainwindow.cpp, 292
- make\_spinbox
  - embroidermodder.h, 240
  - interface.cpp, 268
- make\_ui\_element
  - embroidermodder.h, 240
  - interface.cpp, 268
- manufacturer\_code
  - thread\_color\_, 209
- mapSignal
  - PropertyEditor, 182
- Marathon\_Polyester
  - embroidery.h, 328
- Marathon\_Rayon
  - embroidery.h, 328
- max, 434
- max\_header
  - format\_max.c, 434
- MAX\_STITCHES
  - embroidery.h, 328
- MAX\_THREADS
  - embroidery.h, 328
- maxNumberOfDirectoryEntries
  - \_bcf\_directory, 40
- MdiArea, 162
  - ~MdiArea, 164
  - bgColor, 166
  - bgLogo, 166
  - bgTexture, 166
  - cascade, 164
  - forceRepaint, 164
  - MdiArea, 163
  - mouseDoubleClickEvent, 164
  - paintEvent, 164
  - setBackgroundColor, 164
  - setBackgroundLogo, 165
  - setBackgroundTexture, 165
  - tile, 165
  - useBackgroundColor, 165
  - useBackgroundLogo, 165
  - useBackgroundTexture, 166
  - useColor, 166
  - useLogo, 166
  - useTexture, 166
  - zoomExtentsAllSubWindows, 166
- mdiArea
  - embroidermodder.h, 247
  - mainwindow.cpp, 303
  - MdiWindow, 176
- MdiWindow, 167
  - ~MdiWindow, 169
  - closeEvent, 169
  - curColor, 175
  - curFile, 175
  - curLayer, 175
  - curLineType, 176
  - curLineWeight, 176
  - currentColorChanged, 169
  - currentLayerChanged, 169
  - currentLinetypeChanged, 169
  - currentLineweightChanged, 171
  - deletePressed, 171
  - designDetails, 171
  - escapePressed, 171
  - fileWasLoaded, 176
  - getShortCurrentFile, 171
  - gscene, 176
  - gview, 176
  - loadFile, 171
  - logPromptInput, 172
  - mdiArea, 176
  - MdiWindow, 168
  - myIndex, 176
  - onWindowActivated, 172
  - print, 172
  - printer, 176
  - promptHistory, 176
  - promptHistoryAppended, 172
  - promptInputList, 176
  - promptInputNext, 172
  - promptInputNum, 176
  - promptInputPrevious, 172
  - promptInputPrevNext, 172
  - saveBMC, 173
  - saveFile, 173

- sendCloseMdiWin, [173](#)
- setCurrentFile, [173](#)
- setViewBackgroundColor, [174](#)
- setViewCrossHairColor, [174](#)
- setViewGridColor, [174](#)
- setViewRulerColor, [174](#)
- setViewSelectBoxColors, [174](#)
- showViewScrollBars, [175](#)
- sizeHint, [175](#)
- updateColorLinetypeLineweight, [175](#)
- mdiwindow.cpp
  - fileExtension, [304](#)
- Mega 2560 or another board with equal or, [16](#)
- Melco, [417](#), [421](#), [426](#), [436](#)
- menuHash
  - embroidermodder.h, [247](#)
  - mainwindow.cpp, [303](#)
- mergeWith
  - UndoableCommand, [212](#)
- messagebox\_action
  - mainwindow.cpp, [292](#)
- Metro\_Polyester
  - embroidery.h, [328](#)
- mid
  - EmbArc\_, [78](#)
- miniSectorShift
  - \_bcf\_file\_header, [46](#)
- miniStreamCutoffSize
  - \_bcf\_file\_header, [46](#)
- minorVersion
  - \_bcf\_file\_header, [46](#)
- minute
  - EmbTime\_, [102](#)
- mirror
  - UndoableCommand, [212](#)
- mirror\_selected\_action
  - mainwindow.cpp, [292](#)
- mirrorLine
  - UndoableCommand, [213](#)
- mirrorSelected
  - View, [220](#)
- mit, [384](#), [435](#)
- mitDecodeStitch
  - embroidery\_internal.h, [380](#)
  - encoding.c, [404](#)
- mitEncodeStitch
  - embroidery\_internal.h, [380](#)
  - encoding.c, [404](#)
- Mitsubishi, [384](#), [435](#)
- modifiedTime
  - \_bcf\_directory\_entry, [42](#)
- modifierName
  - ThredExtension\_, [209](#)
- month
  - EmbTime\_, [102](#)
- mouse\_x\_action
  - mainwindow.cpp, [292](#)
- mouse\_y\_action
  - mainwindow.cpp, [292](#)
- mouseDoubleClickEvent
  - MdiArea, [164](#)
  - View, [221](#)
- mouseMoveEvent
  - CmdPromptHandle, [61](#)
  - View, [221](#)
- mousePressEvent
  - CmdPromptHandle, [61](#)
  - View, [221](#)
- mouseReleaseEvent
  - CmdPromptHandle, [62](#)
  - View, [221](#)
- mouseSnapPoint
  - Geometry, [117](#)
- move\_selected\_action
  - mainwindow.cpp, [293](#)
- moveAction
  - View, [221](#)
- movePoint
  - View, [226](#)
- moveResizeHistory
  - CmdPromptSplitter, [75](#)
- moveSelected
  - View, [221](#)
- MOVETO
  - embroidery\_internal.h, [369](#)
- moveY
  - CmdPromptHandle, [62](#)
- movingActive
  - View, [226](#)
- myFileSeparator
  - MainWindow, [162](#)
- myIndex
  - MdiWindow, [176](#)
- n\_attributes
  - format\_svg.c, [448](#)
- N\_PES\_VERSIONS
  - embroidery\_internal.h, [369](#)
- name
  - EmblImage\_, [90](#)
  - Emblayer\_, [91](#)
  - SvgAttribute\_, [208](#)
  - thread\_color\_, [209](#)
- navType
  - UndoableCommand, [213](#)
- negativeXHoopSize
  - VipHeader\_, [229](#)
- negativeYHoopSize
  - VipHeader\_, [229](#)
- new, [384](#), [435](#)
- new\_action
  - mainwindow.cpp, [293](#)
- newFile
  - MainWindow, [154](#)
- next
  - \_bcf\_directory\_entry, [42](#)
- night\_vision\_action

- mainwindow.cpp, 293
- nlengths
  - Huffman, 139
- no\_argument\_debug
  - mainwindow.cpp, 293
- NoArrow
  - Geometry, 109
- Node
  - embroidermodder.h, 236
- Node\_, 177
  - b, 177
  - i, 177
  - r, 177
  - s, 177
  - sl, 177
  - type, 177
- node\_bool
  - embroidermodder.h, 241
  - interface.cpp, 268
- node\_int
  - embroidermodder.h, 241
  - interface.cpp, 269
- node\_qstr
  - embroidermodder.h, 241
  - interface.cpp, 269
- node\_real
  - embroidermodder.h, 241
  - interface.cpp, 269
- node\_str
  - embroidermodder.h, 241
  - interface.cpp, 269
- node\_str\_list
  - embroidermodder.h, 242
  - interface.cpp, 269
- node\_uint
  - embroidermodder.h, 242
  - interface.cpp, 270
- NodeList
  - embroidermodder.h, 236
- NoLine
  - Geometry, 109
- NORMAL
  - embroidery.h, 328
- normalPath
  - Geometry, 136
- ntable
  - Huffman, 139
- NUM\_FLAGS
  - main.c, 476
- num\_selected\_action
  - mainwindow.cpp, 293
- numberOfBytesRemaining
  - \_vp3Hoop, 48
- numberOfColors
  - \_vp3Hoop, 48
  - VipHeader\_, 229
- numberOfDifatSectors
  - \_bcf\_file\_header, 46
- numberOfDirectorySectors
  - \_bcf\_file\_header, 46
- numberOfEntriesInDifatSector
  - embroidery\_internal.h, 380
- numberOfEntriesInFatSector
  - \_bcf\_file\_fat, 44
- numberOfFATSectors
  - \_bcf\_file\_header, 46
- numberOfFormats
  - embroidery.h, 328
- numberOfMiniFatSectors
  - \_bcf\_file\_header, 46
- numberOfStitches
  - VipHeader\_, 229
- numOfDocs
  - MainWindow, 162
- numSelected
  - View, 221
- numStiches
  - ThredHeader\_, 210
- OBJ\_COLOR
  - embroidermodder.h, 236
- OBJ\_KEYS
  - embroidermodder.h, 236
- OBJ\_LAYER
  - embroidermodder.h, 236
- OBJ\_LTYPE
  - embroidermodder.h, 236
- OBJ\_LTYPE\_CENTER
  - mainwindow.cpp, 279
- OBJ\_LTYPE\_CONT
  - mainwindow.cpp, 279
- OBJ\_LTYPE\_DOT
  - mainwindow.cpp, 279
- OBJ\_LTYPE\_FISHBONE
  - mainwindow.cpp, 279
- OBJ\_LTYPE\_HIDDEN
  - mainwindow.cpp, 279
- OBJ\_LTYPE\_PHANTOM
  - mainwindow.cpp, 279
- OBJ\_LTYPE\_RUNNING
  - mainwindow.cpp, 279
- OBJ\_LTYPE\_SATIN
  - mainwindow.cpp, 279
- OBJ\_LTYPE\_VALUES
  - mainwindow.cpp, 278
- OBJ\_LTYPE\_ZIGZAG
  - mainwindow.cpp, 279
- OBJ\_LWT
  - embroidermodder.h, 236
- OBJ\_LWT\_01
  - mainwindow.cpp, 279
- OBJ\_LWT\_02
  - mainwindow.cpp, 279
- OBJ\_LWT\_03
  - mainwindow.cpp, 279
- OBJ\_LWT\_04
  - mainwindow.cpp, 279

- OBJ\_LWT\_05
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_06
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_07
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_08
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_09
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_10
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_11
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_12
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_13
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_14
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_15
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_16
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_17
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_18
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_19
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_20
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_21
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_22
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_23
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_24
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_BYBLOCK
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_BYLAYER
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_DEFAULT
  - mainwindow.cpp, [279](#)
- OBJ\_LWT\_VALUES
  - mainwindow.cpp, [279](#)
- OBJ\_NAME
  - embroidermodder.h, [236](#)
- OBJ\_RUBBER
  - embroidermodder.h, [236](#)
- OBJ\_SNAP\_APPINTERSECTION
  - mainwindow.cpp, [280](#)
- OBJ\_SNAP\_CENTER
  - mainwindow.cpp, [280](#)
- OBJ\_SNAP\_ENDPOINT
  - mainwindow.cpp, [280](#)
- OBJ\_SNAP\_EXTENSION
  - mainwindow.cpp, [280](#)
- OBJ\_SNAP\_INSERTION
  - mainwindow.cpp, [280](#)
- OBJ\_SNAP\_INTERSECTION
  - mainwindow.cpp, [280](#)
- OBJ\_SNAP\_MIDPOINT
  - mainwindow.cpp, [280](#)
- OBJ\_SNAP\_NEAREST
  - mainwindow.cpp, [280](#)
- OBJ\_SNAP\_NODE
  - mainwindow.cpp, [280](#)
- OBJ\_SNAP\_NULL
  - mainwindow.cpp, [279](#)
- OBJ\_SNAP\_PARALLEL
  - mainwindow.cpp, [280](#)
- OBJ\_SNAP\_PERPENDICULAR
  - mainwindow.cpp, [280](#)
- OBJ\_SNAP\_QUADRANT
  - mainwindow.cpp, [280](#)
- OBJ\_SNAP\_TANGENT
  - mainwindow.cpp, [280](#)
- OBJ\_SNAP\_VALUES
  - mainwindow.cpp, [279](#)
- OBJ\_TYPE
  - embroidermodder.h, [236](#)
- OBJ\_TYPE\_ARC
  - embroidermodder.h, [237](#)
- OBJ\_TYPE\_BASE
  - embroidermodder.h, [236](#)
- OBJ\_TYPE\_BLOCK
  - embroidermodder.h, [237](#)
- OBJ\_TYPE\_CIRCLE
  - embroidermodder.h, [237](#)
- OBJ\_TYPE\_DIMALIGNED
  - embroidermodder.h, [237](#)
- OBJ\_TYPE\_DIMANGULAR
  - embroidermodder.h, [237](#)
- OBJ\_TYPE\_DIMARCLENGTH
  - embroidermodder.h, [237](#)
- OBJ\_TYPE\_DIMDIAMETER
  - embroidermodder.h, [237](#)
- OBJ\_TYPE\_DIMLEADER
  - embroidermodder.h, [237](#)
- OBJ\_TYPE\_DIMLINEAR
  - embroidermodder.h, [237](#)
- OBJ\_TYPE\_DIMORDINATE
  - embroidermodder.h, [237](#)
- OBJ\_TYPE\_DIMRADIUS
  - embroidermodder.h, [237](#)
- OBJ\_TYPE\_ELLIPSE
  - embroidermodder.h, [237](#)
- OBJ\_TYPE\_ELLIPSEARC
  - embroidermodder.h, [237](#)
- OBJ\_TYPE\_GRID
  - embroidermodder.h, [237](#)
- OBJ\_TYPE\_HATCH
  - embroidermodder.h, [237](#)



- OBJ\_TYPE\_IMAGE
  - embroidermodder.h, 237
- OBJ\_TYPE\_INFINITELINE
  - embroidermodder.h, 237
- OBJ\_TYPE\_LINE
  - embroidermodder.h, 237
- OBJ\_TYPE\_NULL
  - embroidermodder.h, 236
- OBJ\_TYPE\_PATH
  - embroidermodder.h, 237
- OBJ\_TYPE\_POINT
  - embroidermodder.h, 237
- OBJ\_TYPE\_POLYGON
  - embroidermodder.h, 237
- OBJ\_TYPE\_POLYLINE
  - embroidermodder.h, 237
- OBJ\_TYPE\_RAY
  - embroidermodder.h, 237
- OBJ\_TYPE\_RECTANGLE
  - embroidermodder.h, 237
- OBJ\_TYPE\_RUBBER
  - embroidermodder.h, 237
- OBJ\_TYPE\_SLOT
  - embroidermodder.h, 237
- OBJ\_TYPE\_SPLINE
  - embroidermodder.h, 237
- OBJ\_TYPE\_TEXTMULTI
  - embroidermodder.h, 237
- OBJ\_TYPE\_TEXTSINGLE
  - embroidermodder.h, 237
- OBJ\_TYPE\_UNKNOWN
  - embroidermodder.h, 237
- OBJ\_TYPE\_VALUES
  - embroidermodder.h, 236
- object
  - EmbGeometry\_, 88
  - UndoableCommand, 213
- object\_names
  - property-editor.cpp, 307
- objectAngle
  - Geometry, 118
- objectArcLength
  - Geometry, 118
- objectArea
  - Geometry, 118
- objectBottomLeft
  - Geometry, 118
- objectBottomRight
  - Geometry, 119
- objectCenter
  - Geometry, 119
- objectChord
  - Geometry, 119
- objectCircumference
  - Geometry, 119
- objectClockwise
  - Geometry, 119
- objectCopyPath
  - Geometry, 119
- objectDelta
  - Geometry, 120
- objectDiameter
  - Geometry, 120
- objectDiameterMajor
  - Geometry, 120
- objectDiameterMinor
  - Geometry, 120
- objectEndAngle
  - Geometry, 120
- objectEndPoint
  - Geometry, 120
- objectEndPoint1
  - Geometry, 120
- objectEndPoint2
  - Geometry, 121
- objectHeight
  - Geometry, 121
- objectIncludedAngle
  - Geometry, 121
- objectLength
  - Geometry, 121
- objectLineType
  - Geometry, 121
- objectLineWeight
  - Geometry, 121
- objectMidPoint
  - Geometry, 122
- objectPos
  - Geometry, 122
- objectQuadrant0
  - Geometry, 122
- objectQuadrant180
  - Geometry, 122
- objectQuadrant270
  - Geometry, 122
- objectQuadrant90
  - Geometry, 122
- objectRadius
  - Geometry, 122
- objectRadiusMajor
  - Geometry, 123
- objectRadiusMinor
  - Geometry, 123
- objectRubberPoint
  - Geometry, 123
- objectRubberText
  - Geometry, 123
- objects.cpp
  - add\_polyline, 305
  - closest\_point, 305
  - fourier\_series, 305
  - rotate\_vector, 306
- objectSavePath
  - Geometry, 123
- objectSavePathList
  - Geometry, 123



- objectStartAngle
  - Geometry, [124](#)
- objectStartPoint
  - Geometry, [124](#)
- objectTopLeft
  - Geometry, [124](#)
- objectTopRight
  - Geometry, [124](#)
- objectType
  - \_bcf\_directory\_entry, [42](#)
- ObjectTypeRootEntry
  - embroidery\_internal.h, [369](#)
- ObjectTypeStorage
  - embroidery\_internal.h, [369](#)
- ObjectTypeStream
  - embroidery\_internal.h, [369](#)
- ObjectTypeUnknown
  - embroidery\_internal.h, [369](#)
- objectWidth
  - Geometry, [124](#)
- objectX
  - Geometry, [124](#)
- objectX1
  - Geometry, [125](#)
- objectX2
  - Geometry, [125](#)
- objectY
  - Geometry, [125](#)
- objectY1
  - Geometry, [125](#)
- objectY2
  - Geometry, [125](#)
- objID
  - Geometry, [136](#)
- objLine
  - Geometry, [136](#)
- objPen
  - Geometry, [136](#)
- objRubberMode
  - Geometry, [137](#)
- objRubberPoints
  - Geometry, [137](#)
- objRubberTexts
  - Geometry, [137](#)
- objText
  - Geometry, [137](#)
- objTextBackward
  - Geometry, [137](#)
- objTextFont
  - Geometry, [137](#)
- objTextJustify
  - Geometry, [137](#)
- objTextPath
  - Geometry, [137](#)
- objTextUpsideDown
  - Geometry, [137](#)
- ofm, [436](#)
- ofmDecode
  - format\_ofm.c, [436](#)
- ofmReadBlockHeader
  - format\_ofm.c, [436](#)
- ofmReadClass
  - format\_ofm.c, [436](#)
- ofmReadColorChange
  - format\_ofm.c, [436](#)
- ofmReadExpanded
  - format\_ofm.c, [436](#)
- ofmReadLibrary
  - format\_ofm.c, [436](#)
- ofmReadThreads
  - format\_ofm.c, [436](#)
- onCloseMdiWin
  - MainWindow, [154](#)
- onCloseWindow
  - MainWindow, [154](#)
- onWindowActivated
  - MainWindow, [154](#)
  - MdiWindow, [172](#)
- Open
  - Geometry, [109](#)
- open\_action
  - mainwindow.cpp, [293](#)
- openFile
  - MainWindow, [154](#)
- openFilesSelected
  - MainWindow, [155](#)
- openrecentfile
  - MainWindow, [155](#)
- opensave\_props
  - settings-dialog.cpp, [309](#)
- operator\*
  - embroidermodder.h, [242](#)
  - interface.cpp, [270](#)
- operator+
  - embroidermodder.h, [242](#)
  - interface.cpp, [270](#)
- operator-
  - embroidermodder.h, [242](#)
  - interface.cpp, [270](#)
- originPath
  - View, [226](#)
- paint
  - Geometry, [125](#)
- paintEvent
  - ImageWidget, [141](#)
  - MdiArea, [164](#)
  - SelectBox, [196](#)
- pan\_action
  - mainwindow.cpp, [294](#)
- panDistance
  - View, [226](#)
- panDown
  - View, [221](#)
- panLeft
  - View, [221](#)
- panningActive

- View, 226
- panningPointActive
  - View, 226
- panningRealTimeActive
  - View, 226
- panPoint
  - View, 221
- panRealTime
  - View, 221
- panRight
  - View, 221
- panStart
  - View, 221
- panStartX
  - View, 226
- panStartY
  - View, 226
- Pantone
  - embroidery.h, 328
- panUp
  - View, 221
- parseDIFATSectors
  - main.c, 480
- parseDirectoryEntryName
  - main.c, 480
- parseTime
  - main.c, 480
- paste
  - View, 222
- paste\_action
  - mainwindow.cpp, 294
- paste\_selected\_action
  - mainwindow.cpp, 294
- pasteClip
  - CmdPromptInput, 72
- pasteDelta
  - View, 226
- pasteObjectItemGroup
  - View, 226
- pastePressed
  - CmdPrompt, 56
  - CmdPromptInput, 72
- pastingActive
  - View, 226
- path
  - EmbGeometry\_, 88
  - EmbImage\_, 90
- pattern.c
  - convert, 483
  - embPattern\_addCircleAbs, 483
  - embPattern\_addEllipseAbs, 483
  - embPattern\_addLineAbs, 484
  - embPattern\_addPathAbs, 484
  - embPattern\_addPointAbs, 484
  - embPattern\_addPolygonAbs, 484
  - embPattern\_addPolylineObjectAbs, 484
  - embPattern\_addRectAbs, 484
  - embPattern\_addStitchAbs, 484
  - embPattern\_addStitchRel, 484
  - embPattern\_addThread, 484
  - embPattern\_calcBoundingBox, 485
  - embPattern\_center, 485
  - embPattern\_changeColor, 485
  - embPattern\_color\_count, 485
  - embPattern\_combineJumpStitches, 485
  - embPattern\_copyPolylinesToStitch\_list, 485
  - embPattern\_copystitch\_listToPolylines, 485
  - embPattern\_correctForMaxStitchLength, 485
  - embPattern\_create, 485
  - embPattern\_designDetails, 485
  - embPattern\_end, 486
  - embPattern\_fixColorCount, 486
  - embPattern\_flip, 486
  - embPattern\_flipHorizontal, 486
  - embPattern\_flipVertical, 486
  - embPattern\_free, 486
  - embPattern\_hideStitchesOverLength, 486
  - embPattern\_jumpStitches, 486
  - embPattern\_lengthHistogram, 486
  - embPattern\_loadExternalColorFile, 486
  - embPattern\_maximumStitchLength, 486
  - embPattern\_minimumStitchLength, 486
  - embPattern\_movePolylinesToStitch\_list, 487
  - embPattern\_movestitch\_listToPolylines, 487
  - embPattern\_realStitches, 487
  - embPattern\_scale, 487
  - embPattern\_totalStitchLength, 487
  - embPattern\_trimStitches, 487
- pcd, 9, 385, 437
- pcm, 9, 385, 437
- pcm\_thread
  - embroidery.h, 328
- pcmThreads
  - embroidery.h, 347
  - thread-color.c, 488
- pcq, 9, 385, 438
- pcs, 9, 385, 438
- pec, 9, 385, 439
- pec\_thread
  - embroidery.h, 329
- pecEncode
  - format\_pec.c, 439
- pecEncodeJump
  - format\_pec.c, 439
- pecEncodeStop
  - format\_pec.c, 439
- pecThreadCount
  - embroidery.h, 347
  - thread-color.c, 488
- pecThreads
  - embroidery.h, 347
  - thread-color.c, 489
- pel, 9, 385, 440
- pem, 9, 386, 440
- perpendicular\_distance\_action
  - mainwindow.cpp, 294

- pes, [9](#), [443](#)
- PES0001
  - embroidery\_internal.h, [369](#)
- PES0020
  - embroidery\_internal.h, [369](#)
- PES0022
  - embroidery\_internal.h, [369](#)
- PES0030
  - embroidery\_internal.h, [369](#)
- PES0040
  - embroidery\_internal.h, [369](#)
- PES0050
  - embroidery\_internal.h, [369](#)
- PES0055
  - embroidery\_internal.h, [369](#)
- PES0056
  - embroidery\_internal.h, [370](#)
- PES0060
  - embroidery\_internal.h, [370](#)
- PES0070
  - embroidery\_internal.h, [370](#)
- PES0080
  - embroidery\_internal.h, [370](#)
- PES0090
  - embroidery\_internal.h, [370](#)
- PES0100
  - embroidery\_internal.h, [370](#)
- pes\_version
  - format\_pes.c, [443](#)
- pes\_version\_strings
  - format\_pes.c, [443](#)
- pesWriteEmbOneSection
  - format\_pes.c, [441](#)
- pesWriteSewSegSection
  - format\_pes.c, [441](#)
- Pfaff, [348](#), [385](#), [387](#), [433](#), [434](#), [437](#), [438](#), [449](#), [454](#), [455](#)
- pfaffDecode
  - embroidery\_internal.h, [380](#)
  - encoding.c, [404](#)
- pfaffEncode
  - embroidery\_internal.h, [380](#)
  - encoding.c, [405](#)
- phb, [9](#), [386](#), [443](#)
- phc, [9](#), [386](#), [444](#)
- pickAdd
  - PropertyEditor, [184](#)
- pickAddModeToggled
  - MainWindow, [155](#)
  - PropertyEditor, [182](#)
- pickBoxSize
  - View, [226](#)
- pivot
  - UndoableCommand, [213](#)
- platform\_action
  - mainwindow.cpp, [294](#)
- platformString
  - MainWindow, [155](#)
  - mainwindow.cpp, [295](#)
- plt, [387](#), [444](#)
- point
  - EmbGeometry\_, [88](#)
- pointList
  - EmbPath\_, [93](#)
- polygon
  - EmbGeometry\_, [88](#)
- polyline
  - EmbGeometry\_, [88](#)
- position
  - EmbAlignedDim\_, [77](#)
  - EmbAngularDim\_, [77](#)
  - EmbArcLengthDim\_, [79](#)
  - EmbBlock\_, [81](#)
  - EmbDiameterDim\_, [85](#)
  - EmbImage\_, [90](#)
  - EmbInfiniteLine\_, [90](#)
  - EmbLeaderDim\_, [91](#)
  - EmbLinearDim\_, [92](#)
  - EmbOrdinateDim\_, [93](#)
  - EmbPoint\_, [95](#)
  - EmbRadiusDim\_, [96](#)
  - EmbRay\_, [96](#)
  - EmbTextMulti\_, [100](#)
  - EmbTextSingle\_, [100](#)
- positiveXHoopSize
  - VipHeader\_, [229](#)
- positiveYHoopSize
  - VipHeader\_, [229](#)
- precisionAngle
  - PropertyEditor, [184](#)
- precisionLength
  - PropertyEditor, [184](#)
- prefix
  - CmdPromptInput, [74](#)
- pressPoint
  - View, [227](#)
- pressResizeHistory
  - CmdPromptSplitter, [75](#)
- pressY
  - CmdPromptHandle, [62](#)
- preview
  - settings-dialog.cpp, [309](#)
- preview\_off\_action
  - mainwindow.cpp, [295](#)
- preview\_on\_action
  - mainwindow.cpp, [295](#)
- previewActive
  - View, [227](#)
- previewData
  - View, [227](#)
- PreviewDialog, [178](#)
  - ~PreviewDialog, [178](#)
  - imgWidget, [178](#)
  - PreviewDialog, [178](#)
- previewMode
  - View, [227](#)
- previewObjectItemGroup

- View, 227
- previewObjectList
  - View, 227
- previewOff
  - View, 222
- previewOn
  - View, 222
- previewPoint
  - View, 227
- print
  - MdiWindow, 172
- print\_action
  - mainwindow.cpp, 295
- print\_area\_action
  - mainwindow.cpp, 296
- printArcResults
  - embroidery\_internal.h, 380
- printer
  - MdiWindow, 176
- privacy\_policy.md, 489
- processInput
  - CmdPromptInput, 72
- prompt
  - embroidermodder.h, 247
  - mainwindow.cpp, 303
- prompt\_props
  - settings-dialog.cpp, 309
- promptDivider
  - CmdPrompt, 59
- promptHistory
  - CmdPrompt, 59
  - MdiWindow, 176
- promptHistoryAppended
  - MainWindow, 155
  - MdiWindow, 172
- promptInput
  - CmdPrompt, 59
- promptInputList
  - MdiWindow, 176
- promptInputNext
  - MainWindow, 156
  - MdiWindow, 172
- promptInputNum
  - MdiWindow, 176
- promptInputPrevious
  - MainWindow, 156
  - MdiWindow, 172
- promptInputPrevNext
  - MdiWindow, 172
- promptSplitter
  - CmdPrompt, 59
- promptVBoxLayout
  - CmdPrompt, 59
- properties
  - Geometry, 137
- property-editor.cpp
  - comboBoxTextSingleFont, 306
  - fieldNewText, 306

- fieldNoText, 307
- fieldOffText, 307
- fieldOldText, 307
- fieldOnText, 307
- fieldVariesText, 307
- fieldYesText, 307
- group\_box\_data, 307
- group\_box\_types, 307
- load\_group\_box\_data\_from\_table, 306
- object\_names, 307
- PropertyEditor, 179
  - ~PropertyEditor, 180
  - clearAllFields, 180
  - comboBoxSelected, 183
  - createComboBoxSelected, 180
  - createGroupBox, 181
  - createLineEdit, 181
  - createToolButton, 181
  - createToolButtonPickAdd, 181
  - createToolButtonQSelect, 181
  - eventFilter, 181
  - fieldEdited, 181
  - focusWidget, 183
  - hideAllGroups, 181
  - iconDir, 183
  - iconSize, 184
  - mapSignal, 182
  - pickAdd, 184
  - pickAddModeToggled, 182
  - precisionAngle, 184
  - precisionLength, 184
  - PropertyEditor, 180
  - propertyEditorButtonStyle, 184
  - selectedItemList, 184
  - setSelectedItems, 182
  - showGroups, 182
  - showOneType, 182
  - signalMapper, 184
  - togglePickAddMode, 182
  - toolButtonPickAdd, 184
  - toolButtonQSelect, 184
  - updateComboBoxBoolIfVaries, 182
  - updateComboBoxStrIfVaries, 183
  - updateFontComboBoxStrIfVaries, 183
  - updateLineEditNumIfVaries, 183
  - updateLineEditStrIfVaries, 183
  - updatePickAddModeButton, 183
- propertyEditorButtonStyle
  - PropertyEditor, 184
- qsnap\_x\_action
  - mainwindow.cpp, 296
- qsnap\_y\_action
  - mainwindow.cpp, 296
- qSnapActive
  - View, 227
- qsnapApertureSize
  - View, 227
- qsnapLocatorColor

- View, 227
- qsnapLocatorSize
  - View, 227
- qSnapToggle
  - View, 227
- QUADTOCONTROL
  - embroidery\_internal.h, 370
- QUADTOEND
  - embroidery\_internal.h, 370
- quick\_snap\_props
  - settings-dialog.cpp, 309
- quit
  - MainWindow, 156
- quit\_action
  - mainwindow.cpp, 296
- r
  - EmbColor\_, 82
  - Node\_, 177
- radians
  - embroidery.h, 346
  - functions.c, 464
- radians\_\_
  - embroidermodder.h, 243
  - interface.cpp, 270
- radius
  - EmbCircle\_, 81
  - EmbEllipse\_, 85
  - EmbRect\_, 97
- rapidFireEnabled
  - CmdPromptInput, 74
- rapidMoveActive
  - View, 227
- read100
  - embroidery\_internal.h, 381
  - format\_100.c, 414
- read10o
  - embroidery\_internal.h, 381
  - format\_10o.c, 415
- read\_configuration
  - embroidermodder.h, 243
  - mainwindow.cpp, 296
- read\_hoop
  - format\_jef.c, 432
- read\_settings
  - embroidermodder.h, 243
  - settings-dialog.cpp, 308
- read\_string\_list\_setting
  - mainwindow.cpp, 296
- read\_string\_setting
  - embroidermodder.h, 243
  - mainwindow.cpp, 297
- readArt
  - embroidery\_internal.h, 381
  - format\_art.c, 415
- readBmc
  - embroidery\_internal.h, 381
  - format\_bmc.c, 416
- readBro
  - embroidery\_internal.h, 381
  - format\_bro.c, 416
- readCnd
  - embroidery\_internal.h, 381
  - format\_cnd.c, 417
- readCol
  - embroidery\_internal.h, 381
  - format\_col.c, 418
- readCsd
  - embroidery\_internal.h, 381
  - format\_csd.c, 419
- readCsv
  - embroidery\_internal.h, 381
  - format\_csv.c, 420
- readDat
  - embroidery\_internal.h, 381
  - format\_dat.c, 420
- readDem
  - embroidery\_internal.h, 381
  - format\_dem.c, 421
- readDescriptions
  - embroidery\_internal.h, 382
  - format\_pes.c, 441
- readDsb
  - embroidery\_internal.h, 382
  - format\_dsb.c, 422
- readDst
  - embroidery\_internal.h, 382
  - format\_dst.c, 424
- readDsz
  - embroidery\_internal.h, 382
  - format\_dsz.c, 424
- readDxf
  - embroidery\_internal.h, 382
  - format\_dxf.c, 425
- readEdr
  - embroidery\_internal.h, 382
  - format\_edr.c, 425
- readEmd
  - embroidery\_internal.h, 382
  - format\_emd.c, 426
- reader\_state
  - EmbFormatList\_, 86
- readExp
  - embroidery\_internal.h, 382
  - format\_exp.c, 426
- readExy
  - embroidery\_internal.h, 382
  - format\_exy.c, 427
- readEys
  - embroidery\_internal.h, 382
  - format\_eyes.c, 427
- readFeatherPatterns
  - embroidery\_internal.h, 383
  - format\_pes.c, 441
- readFullSector
  - embroidery\_internal.h, 383
  - main.c, 481

readFxy  
    embroidery\_internal.h, 383  
    format\_fxy.c, 428

readGc  
    embroidery\_internal.h, 383  
    format\_gc.c, 428

readGnc  
    embroidery\_internal.h, 383  
    format\_gnc.c, 429

readGt  
    embroidery\_internal.h, 383  
    format\_gt.c, 429

readHoopName  
    embroidery\_internal.h, 383  
    format\_pes.c, 441

readHus  
    embroidery\_internal.h, 383  
    format\_hus.c, 431

readImageString  
    embroidery\_internal.h, 383  
    format\_pes.c, 442

readInb  
    embroidery\_internal.h, 384  
    format\_inb.c, 431

readInf  
    embroidery\_internal.h, 384  
    format\_inf.c, 431

readJef  
    embroidery\_internal.h, 384  
    format\_jef.c, 433

readKsm  
    embroidery\_internal.h, 384  
    format\_ksm.c, 433

readLine  
    format\_dxf.c, 425

readMax  
    embroidery\_internal.h, 384  
    format\_max.c, 434

readMit  
    embroidery\_internal.h, 384  
    format\_mit.c, 434

readMotifPatterns  
    embroidery\_internal.h, 384  
    format\_pes.c, 442

readNew  
    embroidery\_internal.h, 384  
    format\_new.c, 435

readNextSector  
    embroidery\_internal.h, 384  
    main.c, 481

readOfm  
    embroidery\_internal.h, 384  
    format\_ofm.c, 436

readPcd  
    embroidery\_internal.h, 385  
    format\_pcd.c, 437

readPcm  
    embroidery\_internal.h, 385  
    format\_pcm.c, 437

readPcq  
    embroidery\_internal.h, 385  
    format\_pcq.c, 438

readPcs  
    embroidery\_internal.h, 385  
    format\_pcs.c, 438

readPec  
    embroidery\_internal.h, 385  
    format\_pec.c, 439

readPecStitches  
    embroidery\_internal.h, 385  
    format\_pec.c, 439

readPel  
    embroidery\_internal.h, 385  
    format\_pel.c, 440

readPem  
    embroidery\_internal.h, 385  
    format\_pem.c, 440

readPes  
    embroidery\_internal.h, 386  
    format\_pes.c, 442

readPESHeaderV10  
    embroidery\_internal.h, 386  
    format\_pes.c, 442

readPESHeaderV5  
    embroidery\_internal.h, 386  
    format\_pes.c, 442

readPESHeaderV6  
    embroidery\_internal.h, 386  
    format\_pes.c, 442

readPESHeaderV7  
    embroidery\_internal.h, 386  
    format\_pes.c, 442

readPESHeaderV8  
    embroidery\_internal.h, 386  
    format\_pes.c, 442

readPESHeaderV9  
    embroidery\_internal.h, 386  
    format\_pes.c, 442

readPhb  
    embroidery\_internal.h, 386  
    format\_phb.c, 443

readPhc  
    embroidery\_internal.h, 386  
    format\_phc.c, 444

readPlt  
    embroidery\_internal.h, 386  
    format\_plt.c, 444

readProgrammableFills  
    embroidery\_internal.h, 387  
    format\_pes.c, 442

readRgb  
    embroidery\_internal.h, 387  
    format\_rgb.c, 445

readSew  
    embroidery\_internal.h, 387  
    format\_sew.c, 445

- readShv
  - embroidery\_internal.h, [387](#)
  - format\_shv.c, [446](#)
- readSst
  - embroidery\_internal.h, [387](#)
  - format\_sst.c, [446](#)
- readStx
  - embroidery\_internal.h, [387](#)
  - format\_stx.c, [447](#)
- readSvg
  - embroidery\_internal.h, [387](#)
  - format\_svg.c, [448](#)
- readT01
  - embroidery\_internal.h, [387](#)
  - format\_t01.c, [449](#)
- readT09
  - embroidery\_internal.h, [387](#)
  - format\_t09.c, [449](#)
- readTap
  - embroidery\_internal.h, [387](#)
  - format\_tap.c, [450](#)
- readThr
  - embroidery\_internal.h, [388](#)
  - format\_thr.c, [450](#)
- readThreads
  - embroidery\_internal.h, [388](#)
  - format\_pes.c, [442](#)
- readTxt
  - embroidery\_internal.h, [388](#)
  - format\_txt.c, [451](#)
- readU00
  - embroidery\_internal.h, [388](#)
  - format\_u00.c, [451](#)
- readU01
  - embroidery\_internal.h, [388](#)
  - format\_u01.c, [452](#)
- readVip
  - embroidery\_internal.h, [388](#)
  - format\_vip.c, [453](#)
- readVp3
  - embroidery\_internal.h, [388](#)
  - format\_vp3.c, [454](#)
- readXxx
  - embroidery\_internal.h, [388](#)
  - format\_xxx.c, [455](#)
- readZsk
  - embroidery\_internal.h, [388](#)
  - format\_zsk.c, [456](#)
- REAL\_TYPE
  - embroidermodder.h, [235](#)
- realRender
  - Geometry, [125](#)
- recalculateLimits
  - View, [222](#)
- recentMenuAboutToShow
  - MainWindow, [156](#)
- rect
  - EmbGeometry\_, [88](#)
  - Geometry, [126](#)
- rect.c
  - embRect\_area, [466](#)
  - embRect\_init, [466](#)
- RED\_TERM\_COLOR
  - embroidery\_internal.h, [370](#)
- redo
  - UndoableCommand, [212](#)
  - UndoEditor, [214](#)
- redo\_action
  - mainwindow.cpp, [297](#)
- redoPressed
  - CmdPrompt, [56](#)
  - CmdPromptInput, [72](#)
- redoText
  - UndoEditor, [214](#)
- rejectChanges
  - Settings\_Dialog, [205](#)
- releasePoint
  - View, [227](#)
- releaseResizeHistory
  - CmdPromptSplitter, [75](#)
- releaseY
  - CmdPromptHandle, [62](#)
- repeatAction
  - View, [222](#)
- report
  - embroidery.h, [346](#)
- reserved
  - ThredExtension\_, [209](#)
  - ThredHeader\_, [210](#)
- reserved1
  - \_bcf\_file\_header, [46](#)
- reserved2
  - \_bcf\_file\_header, [46](#)
- RESET\_TERM\_COLOR
  - embroidery\_internal.h, [370](#)
- resizeEvent
  - MainWindow, [156](#)
- resizeHistory
  - CmdPromptHistory, [65](#)
- reverse\_byte\_order
  - encoding.c, [405](#)
- rgb, [9](#), [387](#), [445](#)
- right
  - \_vp3Hoop, [48](#)
  - EmbRect\_, [97](#)
  - hoop\_padding, [138](#)
- right2
  - \_vp3Hoop, [48](#)
- rightBrush
  - SelectBox, [197](#)
- rightBrushColor
  - SelectBox, [197](#)
- rightPen
  - SelectBox, [197](#)
- rightPenColor
  - SelectBox, [197](#)

- rightSiblingId
  - \_bcf\_directory\_entry, [42](#)
- RobisonAnton\_Polyester
  - embroidery.h, [329](#)
- RobisonAnton\_Rayon
  - embroidery.h, [329](#)
- rotate
  - UndoableCommand, [212](#)
- rotate\_selected\_action
  - mainwindow.cpp, [297](#)
- rotate\_vector
  - embroidermodder.h, [243](#)
  - objects.cpp, [306](#)
- rotateAction
  - View, [222](#)
- rotateSelected
  - View, [222](#)
- rotation
  - EmbEllipse\_, [85](#)
  - EmbRect\_, [97](#)
- roundToMultiple
  - View, [222](#)
- rubber\_action
  - mainwindow.cpp, [297](#)
- rubber\_modes
  - mainwindow.cpp, [303](#)
- rubberRoomList
  - View, [227](#)
- rulerColor
  - View, [227](#)
- rulerMetric
  - View, [227](#)
- rulerPixelSize
  - View, [227](#)
- rules
  - fill.c, [409](#)
  - LSYSTEM, [144](#)
- run\_script
  - embroidermodder.h, [243](#)
  - mainwindow.cpp, [297](#)
- run\_script\_file
  - embroidermodder.h, [244](#)
  - mainwindow.cpp, [297](#)
- runCommand
  - CmdPrompt, [56](#)
  - CmdPromptInput, [72](#)
- s
  - em2\_dev\_script, [40](#)
  - Node\_, [177](#)
- safe\_free
  - embroidery\_internal.h, [388](#)
  - formats.c, [413](#)
- save
  - ImageWidget, [141](#)
  - SaveObject, [194](#)
- save\_points\_to\_pattern
  - fill.c, [409](#)
- saveasfile
  - MainWindow, [156](#)
- saveBMC
  - MdiWindow, [173](#)
- saveFile
  - MdiWindow, [173](#)
- savefile
  - MainWindow, [156](#)
- saveHistory
  - CmdPrompt, [56](#)
- SaveObject, [185](#)
  - ~SaveObject, [186](#)
  - addArc, [187](#)
  - addBlock, [187](#)
  - addCircle, [187](#)
  - addDimAligned, [187](#)
  - addDimAngular, [188](#)
  - addDimArcLength, [188](#)
  - addDimDiameter, [188](#)
  - addDimLeader, [188](#)
  - addDimLinear, [189](#)
  - addDimOrdinate, [189](#)
  - addDimRadius, [189](#)
  - addEllipse, [190](#)
  - addEllipseArc, [190](#)
  - addGrid, [190](#)
  - addHatch, [190](#)
  - addImage, [191](#)
  - addInfiniteLine, [191](#)
  - addLine, [191](#)
  - addPath, [191](#)
  - addPoint, [192](#)
  - addPolygon, [192](#)
  - addPolyline, [192](#)
  - addRay, [193](#)
  - addRectangle, [193](#)
  - addSlot, [193](#)
  - addSpline, [193](#)
  - addTextMulti, [194](#)
  - addTextSingle, [194](#)
  - formatType, [195](#)
  - gscene, [195](#)
  - save, [194](#)
  - SaveObject, [186](#)
  - toPolyline, [194](#)
- scale\_selected\_action
  - mainwindow.cpp, [298](#)
- scaleAction
  - View, [222](#)
- scaleSelected
  - View, [222](#)
- sceneGripPoint
  - View, [228](#)
- sceneMousePoint
  - View, [228](#)
- sceneMovePoint
  - View, [228](#)
- scenePressPoint
  - View, [228](#)



- sceneReleasePoint
  - View, 228
- script\_click
  - Geometry, 126
- script\_context
  - Geometry, 126
- script\_main
  - Geometry, 126
- script\_prompt
  - Geometry, 126
- scripts
  - embroidermodder.h, 247
  - mainwindow.cpp, 304
- second
  - EmbTime\_, 102
- sectionName
  - StxThread\_, 207
- sectorShift
  - \_bcf\_file\_header, 46
- sectorSize
  - \_bcf\_file\_difat, 44
  - main.c, 481
- seekToSector
  - main.c, 481
- select\_all\_action
  - mainwindow.cpp, 298
- selectAll
  - View, 222
- selectAllPressed
  - CmdPrompt, 56
  - CmdPromptInput, 72
- SelectBox, 195
  - alpha, 197
  - boxDir, 197
  - dirBrush, 197
  - dirPen, 197
  - forceRepaint, 196
  - leftBrush, 197
  - leftBrushColor, 197
  - leftPen, 197
  - leftPenColor, 197
  - paintEvent, 196
  - rightBrush, 197
  - rightBrushColor, 197
  - rightPen, 197
  - rightPenColor, 197
  - SelectBox, 196
  - setColors, 197
  - setDirection, 197
- selectBox
  - View, 228
- selected\_items
  - View, 222
- selectedItemList
  - PropertyEditor, 184
- selectingActive
  - View, 228
- selectionChanged
  - View, 222
- sendCloseMdiWin
  - MdiWindow, 173
- SEQUIN
  - embroidery.h, 329
- set\_background\_color\_action
  - mainwindow.cpp, 298
- set\_crosshair\_color\_action
  - mainwindow.cpp, 298
- set\_cursor\_shape\_action
  - mainwindow.cpp, 298
- set\_dst\_variable
  - format\_dst.c, 424
- set\_enabled
  - embroidermodder.h, 244
  - interface.cpp, 270
- set\_grid\_color\_action
  - mainwindow.cpp, 299
- set\_object\_color
  - arc.c, 461
- set\_prompt\_prefix\_action
  - mainwindow.cpp, 299
- set\_rubber\_filter\_action
  - mainwindow.cpp, 299
- set\_rubber\_mode\_action
  - mainwindow.cpp, 299
- set\_rubber\_point\_action
  - mainwindow.cpp, 299
- set\_rubber\_text\_action
  - mainwindow.cpp, 299
- set\_visibility
  - embroidermodder.h, 244
  - interface.cpp, 271
- setBackgroundColor
  - MdiArea, 164
  - View, 223
- setBackgroundLogo
  - MdiArea, 165
- setBackgroundTexture
  - MdiArea, 165
- setColors
  - SelectBox, 197
- setCornerButton
  - View, 223
- setCrossHairColor
  - View, 223
- setCrossHairSize
  - View, 223
- setCurrentFile
  - MdiWindow, 173
- setCurrentText
  - CmdPrompt, 56
- setDirection
  - SelectBox, 197
- setGridColor
  - View, 223
- setHistory
  - CmdPrompt, 56

- setLine
  - Geometry, [126](#), [127](#)
- setMainWin
  - Application, [51](#)
- setMouseCoord
  - StatusBar, [206](#)
- setObjectArea
  - Geometry, [127](#)
- setObjectCenter
  - Geometry, [127](#)
- setObjectCenterX
  - Geometry, [127](#)
- setObjectCenterY
  - Geometry, [127](#)
- setObjectCircumference
  - Geometry, [127](#)
- setObjectDiameter
  - Geometry, [127](#)
- setObjectDiameterMajor
  - Geometry, [128](#)
- setObjectDiameterMinor
  - Geometry, [128](#)
- setObjectEndAngle
  - Geometry, [128](#)
- setObjectEndPoint
  - Geometry, [128](#)
- setObjectEndPoint1
  - Geometry, [128](#)
- setObjectEndPoint2
  - Geometry, [129](#)
- setObjectLineWeight
  - Geometry, [129](#)
- setObjectMidPoint
  - Geometry, [129](#)
- setObjectPos
  - Geometry, [129](#)
- setObjectRadius
  - Geometry, [130](#)
- setObjectRadiusMajor
  - Geometry, [130](#)
- setObjectRadiusMinor
  - Geometry, [130](#)
- setObjectRect
  - Geometry, [130](#)
- setObjectSize
  - Geometry, [130](#)
- setObjectStartAngle
  - Geometry, [130](#)
- setObjectStartPoint
  - Geometry, [131](#)
- setObjectText
  - Geometry, [131](#)
- setObjectTextBackward
  - Geometry, [131](#)
- setObjectTextBold
  - Geometry, [131](#)
- setObjectTextFont
  - Geometry, [131](#)
- setObjectTextItalic
  - Geometry, [131](#)
- setObjectTextJustify
  - Geometry, [132](#)
- setObjectTextOverline
  - Geometry, [132](#)
- setObjectTextSize
  - Geometry, [132](#)
- setObjectTextStrikeOut
  - Geometry, [132](#)
- setObjectTextStyle
  - Geometry, [132](#)
- setObjectTextUnderline
  - Geometry, [133](#)
- setObjectTextUpsideDown
  - Geometry, [133](#)
- setObjectX
  - Geometry, [133](#)
- setObjectY
  - Geometry, [133](#)
- setPrefix
  - CmdPrompt, [56](#)
- setPromptBackgroundColor
  - CmdPrompt, [57](#)
- setPromptFontFamily
  - CmdPrompt, [57](#)
- setPromptFontSize
  - CmdPrompt, [57](#)
- setPromptFontStyle
  - CmdPrompt, [57](#)
- setPromptTextColor
  - CmdPrompt, [58](#)
- setRect
  - Geometry, [133](#)
- setRubberMode
  - View, [223](#)
- setRubberPoint
  - View, [223](#)
- SetRubberText
  - mainwindow.cpp, [300](#)
- setRubberText
  - View, [223](#)
- setRulerColor
  - View, [223](#)
- setSelectBoxColors
  - View, [223](#)
- setSelectedItems
  - PropertyEditor, [182](#)
- setShiftPressed
  - MainWindow, [156](#)
- setShiftReleased
  - MainWindow, [157](#)
- SetTextAngle\_action
  - mainwindow.cpp, [300](#)
- setTextFont
  - MainWindow, [157](#)
- setTextSize
  - MainWindow, [157](#)

- settings
  - embroidermodder.h, [247](#)
  - mainwindow.cpp, [304](#)
- settings-dialog.cpp
  - accept\_, [308](#)
  - display\_props, [308](#)
  - extensions, [309](#)
  - general\_props, [309](#)
  - make\_editing\_copy, [308](#)
  - opensave\_props, [309](#)
  - preview, [309](#)
  - prompt\_props, [309](#)
  - quick\_snap\_props, [309](#)
  - read\_settings, [308](#)
  - write\_settings, [308](#)
- Settings\_Dialog, [198](#)
  - ~Settings\_Dialog, [199](#)
  - acceptChanges, [200](#)
  - addColorsToComboBox, [200](#)
  - buttonBox, [205](#)
  - buttonCustomFilterClearAll, [200](#)
  - buttonCustomFilterClearAllClicked, [200](#)
  - buttonCustomFilterSelectAll, [200](#)
  - buttonCustomFilterSelectAllClicked, [200](#)
  - buttonQSnapClearAll, [200](#)
  - buttonQSnapClearAllClicked, [200](#)
  - buttonQSnapSelectAll, [200](#)
  - buttonQSnapSelectAllClicked, [200](#)
  - checkBoxCustomFilterStateChanged, [200](#)
  - checkBoxGeneralMdiBGUseColorStateChanged, [200](#)
  - checkBoxGeneralMdiBGUseLogoStateChanged, [200](#)
  - checkBoxGeneralMdiBGUseTextureStateChanged, [200](#)
  - checkBoxGridCenterOnOriginStateChanged, [201](#)
  - checkBoxGridColorMatchCrossHairStateChanged, [201](#)
  - checkBoxGridLoadFromFileStateChanged, [201](#)
  - checkBoxLwtRealRenderStateChanged, [201](#)
  - checkBoxLwtShowLwtStateChanged, [201](#)
  - checkBoxPromptSaveHistoryAsHtmlStateChanged, [201](#)
  - checkBoxRulerShowOnLoadStateChanged, [201](#)
  - checkBoxShowScrollBarsStateChanged, [201](#)
  - chooseDisplayBackgroundColor, [201](#)
  - chooseDisplayCrossHairColor, [201](#)
  - chooseDisplaySelectBoxLeftColor, [201](#)
  - chooseDisplaySelectBoxLeftFill, [201](#)
  - chooseDisplaySelectBoxRightColor, [202](#)
  - chooseDisplaySelectBoxRightFill, [202](#)
  - chooseGeneralMdiBackgroundColor, [202](#)
  - chooseGeneralMdiBackgroundLogo, [202](#)
  - chooseGeneralMdiBackgroundTexture, [202](#)
  - chooseGridColor, [202](#)
  - choosePromptBackgroundColor, [202](#)
  - choosePromptTextColor, [202](#)
  - chooseRulerColor, [202](#)
  - comboBoxGridTypeCurrentIndexChanged, [202](#)
  - comboBoxIconSizeCurrentIndexChanged, [202](#)
  - comboBoxPromptFontFamilyCurrentIndexChanged, [203](#)
  - comboBoxPromptFontStyleCurrentIndexChanged, [203](#)
  - comboBoxQSnapLocatorColorCurrentIndexChanged, [203](#)
  - comboBoxRulerMetricCurrentIndexChanged, [203](#)
  - comboBoxScrollBarWidgetCurrentIndexChanged, [203](#)
  - comboBoxSelectionCoolGripColorCurrentIndexChanged, [203](#)
  - comboBoxSelectionHotGripColorCurrentIndexChanged, [203](#)
  - create\_checkbox, [203](#)
  - create\_float\_spinbox, [203](#)
  - createTabDisplay, [203](#)
  - createTabFilePaths, [203](#)
  - createTabGeneral, [204](#)
  - createTabGridRuler, [204](#)
  - createTabLineWeight, [204](#)
  - createTabOpenSave, [204](#)
  - createTabOrthoPolar, [204](#)
  - createTabPrinting, [204](#)
  - createTabPrompt, [204](#)
  - createTabQuickSnap, [204](#)
  - createTabQuickTrack, [204](#)
  - createTabSelection, [204](#)
  - createTabSnap, [204](#)
  - currentDisplayBackgroundColorChanged, [204](#)
  - currentDisplayCrossHairColorChanged, [204](#)
  - currentDisplaySelectBoxLeftColorChanged, [204](#)
  - currentDisplaySelectBoxLeftFillChanged, [204](#)
  - currentDisplaySelectBoxRightColorChanged, [204](#)
  - currentDisplaySelectBoxRightFillChanged, [205](#)
  - currentGeneralMdiBackgroundColorChanged, [205](#)
  - currentGridColorChanged, [205](#)
  - currentPromptBackgroundColorChanged, [205](#)
  - currentPromptTextColorChanged, [205](#)
  - currentRulerColorChanged, [205](#)
  - rejectChanges, [205](#)
  - Settings\_Dialog, [199](#)
  - spinBoxDisplaySelectBoxAlphaValueChanged, [205](#)
  - spinBoxPromptFontSizeValueChanged, [205](#)
  - spinBoxRulerPixelSizeValueChanged, [205](#)
  - tabWidget, [205](#)
- settings\_dialog\_action
  - mainwindow.cpp, [300](#)
- settingsPrompt
  - MainWindow, [157](#)
- setUndoCleanIcon
  - MainWindow, [157](#)
- setViewBackgroundColor
  - MdiWindow, [174](#)
- setViewCrossHairColor
  - MdiWindow, [174](#)

- setViewGridColor
  - MdiWindow, 174
- setViewRulerColor
  - MdiWindow, 174
- setViewSelectBoxColors
  - MdiWindow, 174
- sew, 9, 445
- sewDecode
  - format\_sew.c, 445
- shiftKeyPressedState
  - MainWindow, 162
- shiftPressed
  - CmdPrompt, 58
  - CmdPromptInput, 72
- shiftReleased
  - CmdPrompt, 58
  - CmdPromptInput, 72
- showGroups
  - PropertyEditor, 182
- showOneType
  - PropertyEditor, 182
- showScrollBars
  - View, 223
- showSettings
  - CmdPrompt, 58
  - CmdPromptInput, 72
- showViewScrollBars
  - MdiWindow, 175
- shv, 9, 446
- shv\_thread
  - embroidery.h, 329
- shvDecode
  - format\_shv.c, 446
- shvDecodeShort
  - format\_shv.c, 446
- shvThreadCount
  - embroidery.h, 347
  - thread-color.c, 489
- shvThreads
  - embroidery.h, 348
  - thread-color.c, 489
- side1
  - EmbSatinOutline\_, 98
- side2
  - EmbSatinOutline\_, 98
- Sierra Expanded, 383, 428
- Sigma\_Polyester
  - embroidery.h, 329
- signalMapper
  - PropertyEditor, 184
- signature
  - \_bcf\_file\_header, 46
- sigVersion
  - ThredHeader\_, 210
- Singer, 418, 456
- sizeHint
  - MdiWindow, 175
- sizeOfChainingEntryAtEndOfDifatSector
  - main.c, 482
- sizeOfDifatEntry
  - main.c, 482
- sizeOfDirectoryEntry
  - main.c, 482
- sizeOfFatEntry
  - main.c, 482
- sl
  - Node\_, 177
- someInt
  - SubDescriptor\_, 208
- someNum
  - SubDescriptor\_, 208
- someOtherInt
  - SubDescriptor\_, 208
- spare\_rubber\_action
  - mainwindow.cpp, 300
- spareRubber
  - View, 223
- spareRubberList
  - View, 228
- spinBoxDisplaySelectBoxAlphaValueChanged
  - Settings\_Dialog, 205
- spinBoxes
  - embroidermodder.h, 247
  - mainwindow.cpp, 304
- spinBoxPromptFontSizeValueChanged
  - Settings\_Dialog, 205
- spinBoxRulerPixelSizeValueChanged
  - Settings\_Dialog, 205
- spline
  - EmbGeometry\_, 88
- sst, 9, 387, 447
- start
  - EmbArc\_, 78
  - EmbBezier\_, 80
  - EmbLine\_, 92
- startBlinking
  - CmdPrompt, 58
- startCommand
  - CmdPrompt, 58
  - CmdPromptInput, 72
- startGripping
  - View, 223
- startingSectorLocation
  - \_bcf\_directory\_entry, 42
- startResizeHistory
  - CmdPromptHistory, 65
- state
  - View, 228
- stateBits
  - \_bcf\_directory\_entry, 42
- StatusBar, 206
  - buttons, 207
  - context\_menu\_action, 206
  - context\_menu\_event, 206
  - setMouseCoord, 206
  - StatusBar, 206

- statusBarMouseCoord, 207
- toggle, 206
- statusbar
  - embroidermodder.h, 247
  - mainwindow.cpp, 304
- statusBarMouseCoord
  - StatusBar, 207
- stitch
  - EmbArray\_, 79
  - EmbGeometry\_, 88
- stitch\_list
  - EmbPattern\_, 95
- stitchesJump
  - EmbDetailsDialog, 84
- stitchesReal
  - EmbDetailsDialog, 84
- stitchesTotal
  - EmbDetailsDialog, 84
- stitchesTrim
  - EmbDetailsDialog, 84
- stitchGranularity
  - ThredExtension\_, 209
- STOP
  - embroidery.h, 329
- stopBlinking
  - CmdPrompt, 58
  - CmdPromptInput, 73
- stopGripping
  - View, 224
- stopResizeHistory
  - CmdPromptHistory, 65
- streamSize
  - \_bcf\_directory\_entry, 42
- streamSizeHigh
  - \_bcf\_directory\_entry, 42
- String
  - embroidermodder.h, 236
- STRING\_LIST\_TYPE
  - embroidermodder.h, 235
- STRING\_TYPE
  - embroidermodder.h, 235
- stringInArray
  - embroidery\_internal.h, 389
  - main.c, 481
- StringList
  - embroidermodder.h, 236
- stringVal
  - VipHeader\_, 229
- stub\_testing
  - MainWindow, 157
- stx, 447
- stxColor
  - StxThread\_, 207
- stxReadThread
  - format\_stx.c, 447
- StxThread
  - embroidery\_internal.h, 371
- StxThread\_, 207
- colorCode, 207
- colorName, 207
- sectionName, 207
- stxColor, 207
- subDescriptors, 207
- styleHash
  - CmdPrompt, 59
- SubDescriptor
  - embroidery\_internal.h, 372
- SubDescriptor\_, 207
  - colorCode, 208
  - colorName, 208
  - someInt, 208
  - someNum, 208
  - someOtherInt, 208
- subDescriptors
  - StxThread\_, 207
- subMenuHash
  - embroidermodder.h, 247
  - mainwindow.cpp, 304
- subPathList
  - Geometry, 134
- Sulky\_Rayon
  - embroidery.h, 329
- Sunstar, 387, 447
- svg, 9, 448
- SVG\_ATTRIBUTE
  - embroidery\_internal.h, 370
- SVG\_CATCH\_ALL
  - embroidery\_internal.h, 370
- SVG\_Colors
  - embroidery.h, 329
- SVG\_CREATOR\_EMBROIDERMODDER
  - embroidery\_internal.h, 370
- SVG\_CREATOR\_ILLUSTRATOR
  - embroidery\_internal.h, 370
- SVG\_CREATOR\_INKSCAPE
  - embroidery\_internal.h, 370
- SVG\_CREATOR\_NULL
  - embroidery\_internal.h, 370
- SVG\_ELEMENT
  - embroidery\_internal.h, 370
- SVG\_EXPECT\_ATTRIBUTE
  - embroidery\_internal.h, 370
- SVG\_EXPECT\_ELEMENT
  - embroidery\_internal.h, 371
- SVG\_EXPECT\_NULL
  - embroidery\_internal.h, 371
- SVG\_EXPECT\_VALUE
  - embroidery\_internal.h, 371
- SVG\_MEDIA\_PROPERTY
  - embroidery\_internal.h, 371
- SVG\_NULL
  - embroidery\_internal.h, 371
- SVG\_PROPERTY
  - embroidery\_internal.h, 371
- SvgAttribute
  - embroidery\_internal.h, 372

- SvgAttribute\_, 208
  - name, 208
  - value, 208
- svgCreator
  - format\_svg.c, 448
- svgExpect
  - format\_svg.c, 448
- svgMultiValue
  - format\_svg.c, 448
- t01, 387, 449
- t09, 387, 449
- table
  - Huffman, 139
- table\_width
  - Huffman, 139
- tabPressed
  - CmdPrompt, 58
  - CmdPromptInput, 73
- tabWidget
  - Settings\_Dialog, 205
- Tajima, 422
- tap, 9, 450
- tempBaseObj
  - View, 228
- testEmbCircle
  - embroidery\_internal.h, 389
- testEmbCircle\_2
  - embroidery\_internal.h, 389
- testEmbFormat
  - embroidery\_internal.h, 389
- testGeomArc
  - embroidery\_internal.h, 389
- testMain
  - embroidery.h, 346
- testTangentPoints
  - embroidery\_internal.h, 389
- testThreadColor
  - embroidery\_internal.h, 389
- text
  - EmbTextMulti\_, 100
  - EmbTextSingle\_, 100
- text.c
  - textSingle\_gripEdit, 466
  - textSingle\_mouseSnapPoint, 466
  - textSingle\_paint, 467
  - textSingle\_setJustify, 467
  - textSingle\_setTextBackward, 467
  - textSingle\_setTextBold, 467
  - textSingle\_setTextFont, 467
  - textSingle\_setTextItalic, 467
  - textSingle\_setTextOverline, 467
  - textSingle\_setTextSize, 467
  - textSingle\_setTextStrikeOut, 467
  - textSingle\_setTextStyle, 467
  - textSingle\_setTextUnderline, 467
  - textSingle\_setTextUpsideDown, 467
  - textSingle\_updateRubber, 467
- textFontSelector
  - MainWindow, 162
- textFontSelectorCurrentFontChanged
  - MainWindow, 158
- textSingle\_gripEdit
  - text.c, 466
- textSingle\_mouseSnapPoint
  - text.c, 466
- textSingle\_paint
  - text.c, 467
- textSingle\_setJustify
  - text.c, 467
- textSingle\_setTextBackward
  - text.c, 467
- textSingle\_setTextBold
  - text.c, 467
- textSingle\_setTextFont
  - text.c, 467
- textSingle\_setTextItalic
  - text.c, 467
- textSingle\_setTextOverline
  - text.c, 467
- textSingle\_setTextSize
  - text.c, 467
- textSingle\_setTextStrikeOut
  - text.c, 467
- textSingle\_setTextStyle
  - text.c, 467
- textSingle\_setTextUnderline
  - text.c, 467
- textSingle\_setTextUpsideDown
  - text.c, 467
- textSingle\_updateRubber
  - text.c, 467
- textSizeSelector
  - MainWindow, 162
- textSizeSelectorIndexChanged
  - MainWindow, 158
- thr, 388, 451
- thread
  - EmbArray\_, 79
  - EmbGeometry\_, 89
- thread-color.c
  - \_dxfColorTable, 488
  - brand\_codes, 488
  - brand\_codes\_files, 488
  - husThreads, 488
  - jefThreads, 488
  - pcmThreads, 488
  - pecThreadCount, 488
  - pecThreads, 489
  - shvThreadCount, 489
  - shvThreads, 489
  - threadColor, 488
  - threadColorName, 488
  - threadColorNum, 488
- thread\_color
  - embroidery.h, 332
- thread\_color\_, 208

- hex\_code, [209](#)
- manufacturer\_code, [209](#)
- name, [209](#)
- thread\_list
  - EmbPattern\_, [95](#)
- ThreadArt\_Polyester
  - embroidery.h, [329](#)
- ThreadArt\_Rayon
  - embroidery.h, [329](#)
- threadColor
  - embroidery.h, [346](#)
  - thread-color.c, [488](#)
- threadColorName
  - embroidery.h, [346](#)
  - thread-color.c, [488](#)
- threadColorNum
  - embroidery.h, [347](#)
  - thread-color.c, [488](#)
- ThreaDelight\_Polyester
  - embroidery.h, [329](#)
- threadLength
  - \_vp3Hoop, [48](#)
- ThreadWorks, [388](#), [451](#)
- ThredExtension
  - embroidery\_internal.h, [372](#)
- ThredExtension\_, [209](#)
  - auxFormat, [209](#)
  - creatorName, [209](#)
  - hoopX, [209](#)
  - hoopY, [209](#)
  - modifierName, [209](#)
  - reserved, [209](#)
  - stitchGranularity, [209](#)
- ThredHeader
  - embroidery\_internal.h, [372](#)
- ThredHeader\_, [210](#)
  - hoopSize, [210](#)
  - length, [210](#)
  - numStiches, [210](#)
  - reserved, [210](#)
  - sigVersion, [210](#)
- threshold\_method
  - fill.c, [409](#)
- Tick
  - Geometry, [109](#)
- tile
  - MdiArea, [165](#)
- tip\_of\_the\_day\_action
  - mainwindow.cpp, [300](#)
- tipOfTheDay
  - MainWindow, [158](#)
- tmpHeight
  - CmdPromptHistory, [65](#)
- to\_EmbVector
  - embroidermodder.h, [244](#)
  - interface.cpp, [271](#)
- to\_qlist
  - embroidermodder.h, [244](#)
- interface.cpp, [271](#)
- to\_QPointF
  - embroidermodder.h, [245](#)
  - interface.cpp, [271](#)
- to\_string\_vector
  - embroidermodder.h, [245](#)
  - interface.cpp, [271](#)
- to\_vector
  - embroidermodder.h, [245](#)
  - interface.cpp, [272](#)
- toCenter
  - UndoableCommand, [213](#)
- todo\_action
  - mainwindow.cpp, [300](#)
- toggle
  - StatusBar, [206](#)
- toggleGrid
  - MainWindow, [158](#)
  - View, [224](#)
- toggleLwt
  - MainWindow, [158](#)
  - View, [224](#)
- toggleOrtho
  - View, [224](#)
- togglePickAddMode
  - PropertyEditor, [182](#)
- togglePolar
  - View, [224](#)
- toggleQSnap
  - View, [224](#)
- toggleQTrack
  - View, [224](#)
- toggleReal
  - View, [224](#)
- toggleRuler
  - MainWindow, [158](#)
  - View, [224](#)
- toggleSnap
  - View, [224](#)
- tokenize
  - embroidermodder.h, [245](#)
  - interface.cpp, [272](#)
- toolbarHash
  - embroidermodder.h, [247](#)
  - mainwindow.cpp, [304](#)
- toolButtonPickAdd
  - PropertyEditor, [184](#)
- toolButtonQSelect
  - PropertyEditor, [184](#)
- toolButtons
  - embroidermodder.h, [247](#)
  - mainwindow.cpp, [304](#)
- top
  - \_vp3Hoop, [48](#)
  - EmbRect\_, [97](#)
  - hoop\_padding, [138](#)
- top2
  - \_vp3Hoop, [49](#)

- toPolyline
  - SaveObject, 194
- toTransform
  - UndoableCommand, 213
- Toyota, 414
- transactionSignatureNumber
  - \_bcf\_file\_header, 47
- translate\_str
  - embroidermodder.h, 245
  - interface.cpp, 272
- treeView
  - LayerManager, 143
- TRIM
  - embroidery.h, 329
- txt, 388, 451
- Type
  - Geometry, 137
- type
  - EmbArray\_, 80
  - EmbFormatList\_, 86
  - EmbGeometry\_, 89
  - Geometry, 134
  - Node\_, 177
- u00, 388, 452
- u01, 9, 388, 452
- undo
  - UndoableCommand, 212
  - UndoEditor, 214
- undo\_action
  - mainwindow.cpp, 301
- UndoableCommand, 210
  - after, 212
  - angle, 212
  - before, 212
  - command, 212
  - delta, 213
  - done, 213
  - factor, 213
  - fromCenter, 213
  - fromTransform, 213
  - gview, 213
  - id, 212
  - mergeWith, 212
  - mirror, 212
  - mirrorLine, 213
  - navType, 213
  - object, 213
  - pivot, 213
  - redo, 212
  - rotate, 212
  - toCenter, 213
  - toTransform, 213
  - undo, 212
  - UndoableCommand, 211, 212
- UndoEditor, 213
  - ~UndoEditor, 214
  - addStack, 214
  - canRedo, 214
  - canUndo, 214
  - focusWidget, 215
  - iconDir, 215
  - iconSize, 215
  - redo, 214
  - redoText, 214
  - undo, 214
  - UndoEditor, 214
  - undoGroup, 215
  - undoText, 214
  - undoView, 215
  - updateCleanIcon, 214
- undoGroup
  - UndoEditor, 215
- undoPressed
  - CmdPrompt, 58
  - CmdPromptInput, 73
- undoStack
  - View, 228
- undoText
  - UndoEditor, 214
- undoView
  - UndoEditor, 215
- unknown
  - VipHeader\_, 229
- unknown2
  - \_vp3Hoop, 49
- unknown3
  - \_vp3Hoop, 49
- unknown4
  - \_vp3Hoop, 49
- UNKNOWN\_TYPE
  - embroidermodder.h, 235
- updateAllViewBackgroundColors
  - MainWindow, 158
- updateAllViewCrossHairColors
  - MainWindow, 159
- updateAllViewGridColors
  - MainWindow, 159
- updateAllViewRulerColors
  - MainWindow, 159
- updateAllViewScrollBars
  - MainWindow, 159
- updateAllViewSelectBoxColors
  - MainWindow, 160
- updateArcRect
  - Geometry, 134
- updateCleanIcon
  - UndoEditor, 214
- updateColorLinetypeLineweight
  - MdiWindow, 175
- updateComboBoxBoolIfVaries
  - PropertyEditor, 182
- updateComboBoxStrIfVaries
  - PropertyEditor, 183
- updateCurrentText
  - CmdPromptInput, 73
- updateFontComboBoxStrIfVaries



- PropertyEditor, 183
- updateLeader
  - Geometry, 134
- updateLineEditNumIfVaries
  - PropertyEditor, 183
- updateLineEditStrIfVaries
  - PropertyEditor, 183
- updateMenuToolBarStatusBar
  - MainWindow, 160
- updateMouseCoords
  - View, 224
- updatePath
  - Geometry, 134
- updatePickAddMode
  - MainWindow, 160
- updatePickAddModeButton
  - PropertyEditor, 183
- updateRubber
  - Geometry, 135
- updateStyle
  - CmdPrompt, 59
- upPressed
  - CmdPrompt, 59
  - CmdPromptInput, 73
- usage\_msg
  - embroidermodder.cpp, 231
- useBackgroundColor
  - MdiArea, 165
- useBackgroundLogo
  - MdiArea, 165
- useBackgroundTexture
  - MdiArea, 166
- useColor
  - MdiArea, 166
- useLogo
  - MdiArea, 166
- useTexture
  - MdiArea, 166
- validFileFormat
  - embroidermodder.h, 245
  - mainwindow.cpp, 301
- validRGB
  - mainwindow.cpp, 301
- value
  - SvgAttribute\_, 208
- vector
  - EmbGeometry\_, 89
- vector.c
  - embVector\_add, 468
  - embVector\_angle, 468
  - embVector\_average, 468
  - embVector\_cross, 468
  - embVector\_distance, 469
  - embVector\_dot, 469
  - embVector\_length, 469
  - embVector\_multiply, 469
  - embVector\_normalize, 469
  - embVector\_relativeX, 469
  - embVector\_relativeY, 470
  - embVector\_subtract, 470
  - embVector\_transpose\_product, 470
  - embVector\_unit, 470
- VECTOR\_TYPE
  - embroidermodder.h, 235
- version\_action
  - mainwindow.cpp, 301
- View, 215
  - ~View, 218
  - addObject, 218
  - addToRubberRoom, 218
  - alignScenePointWithViewPoint, 218
  - allowRubber, 219
  - allowZoomIn, 219
  - allowZoomOut, 219
  - center, 219
  - centerAt, 219
  - clearRubberRoom, 219
  - clearSelection, 219
  - contextMenuEvent, 219
  - copy, 219
  - copySelected, 219
  - cornerButtonClicked, 219
  - createGrid, 219
  - createGridIso, 219
  - createGridPolar, 219
  - createGridRect, 219
  - createObjectList, 219
  - createOrigin, 220
  - createRulerTextPath, 220
  - crosshairColor, 225
  - crosshairSize, 225
  - cut, 220
  - cutCopyMousePoint, 225
  - deleteObject, 220
  - deletePressed, 220
  - deleteSelected, 220
  - drawBackground, 220
  - drawForeground, 220
  - enterEvent, 220
  - escapePressed, 220
  - getUndoStack, 220
  - gridColor, 225
  - gridPath, 225
  - gripBaseObj, 225
  - gripColorCool, 225
  - gripColorHot, 226
  - grippingActive, 226
  - gripSize, 226
  - gscene, 226
  - hashDeletedObjects, 226
  - isLwtEnabled, 220
  - isRealEnabled, 220
  - loadRulerSettings, 220
  - mirrorSelected, 220
  - mouseDoubleClickEvent, 221
  - mouseMoveEvent, 221

[mousePressEvent](#), 221  
[mouseReleaseEvent](#), 221  
[moveAction](#), 221  
[movePoint](#), 226  
[moveSelected](#), 221  
[movingActive](#), 226  
[numSelected](#), 221  
[originPath](#), 226  
[panDistance](#), 226  
[panDown](#), 221  
[panLeft](#), 221  
[panningActive](#), 226  
[panningPointActive](#), 226  
[panningRealTimeActive](#), 226  
[panPoint](#), 221  
[panRealTime](#), 221  
[panRight](#), 221  
[panStart](#), 221  
[panStartX](#), 226  
[panStartY](#), 226  
[panUp](#), 221  
[paste](#), 222  
[pasteDelta](#), 226  
[pasteObjectItemGroup](#), 226  
[pastingActive](#), 226  
[pickBoxSize](#), 226  
[pressPoint](#), 227  
[previewActive](#), 227  
[previewData](#), 227  
[previewMode](#), 227  
[previewObjectItemGroup](#), 227  
[previewObjectList](#), 227  
[previewOff](#), 222  
[previewOn](#), 222  
[previewPoint](#), 227  
[qSnapActive](#), 227  
[qsnapApertureSize](#), 227  
[qsnapLocatorColor](#), 227  
[qsnapLocatorSize](#), 227  
[qSnapToggle](#), 227  
[rapidMoveActive](#), 227  
[recalculateLimits](#), 222  
[releasePoint](#), 227  
[repeatAction](#), 222  
[rotateAction](#), 222  
[rotateSelected](#), 222  
[roundToMultiple](#), 222  
[rubberRoomList](#), 227  
[rulerColor](#), 227  
[rulerMetric](#), 227  
[rulerPixelSize](#), 227  
[scaleAction](#), 222  
[scaleSelected](#), 222  
[sceneGripPoint](#), 228  
[sceneMousePoint](#), 228  
[sceneMovePoint](#), 228  
[scenePressPoint](#), 228  
[sceneReleasePoint](#), 228  
[selectAll](#), 222  
[selectBox](#), 228  
[selected\\_items](#), 222  
[selectingActive](#), 228  
[selectionChanged](#), 222  
[setBackgroundColor](#), 223  
[setCornerButton](#), 223  
[setCrossHairColor](#), 223  
[setCrossHairSize](#), 223  
[setGridColor](#), 223  
[setRubberMode](#), 223  
[setRubberPoint](#), 223  
[setRubberText](#), 223  
[setRulerColor](#), 223  
[setSelectBoxColors](#), 223  
[showScrollBars](#), 223  
[spareRubber](#), 223  
[spareRubberList](#), 228  
[startGripping](#), 223  
[state](#), 228  
[stopGripping](#), 224  
[tempBaseObj](#), 228  
[toggleGrid](#), 224  
[toggleLwt](#), 224  
[toggleOrtho](#), 224  
[togglePolar](#), 224  
[toggleQSnap](#), 224  
[toggleQTrack](#), 224  
[toggleReal](#), 224  
[toggleRuler](#), 224  
[toggleSnap](#), 224  
[undoStack](#), 228  
[updateMouseCoords](#), 224  
[View](#), 218  
[viewMousePoint](#), 228  
[vulcanizeObject](#), 224  
[vulcanizeRubberRoom](#), 224  
[wheelEvent](#), 224  
[willOverflowInt32](#), 225  
[willUnderflowInt32](#), 225  
[zoomExtents](#), 225  
[zoomIn](#), 225  
[zoomOut](#), 225  
[zoomSelected](#), 225  
[zoomToPoint](#), 225  
[zoomWindow](#), 225  
[zoomWindowActive](#), 228  
[view.cpp](#)  
     [contains](#), 310  
[viewMousePoint](#)  
     [View](#), 228  
[vip](#), 9, 348, 454  
[vipCompressData](#)  
     [format\\_vip.c](#), 453  
[vipDecodeByte](#)  
     [format\\_vip.c](#), 453  
[vipDecodeStitchType](#)  
     [format\\_vip.c](#), 453

- vipDecodingTable
  - embroidery.h, 348
  - format\_vip.c, 453
- vipDecompressData
  - format\_vip.c, 453
- vipEncodeByte
  - format\_vip.c, 453
- vipEncodeStitchType
  - format\_vip.c, 453
- VipHeader
  - embroidery\_internal.h, 372
- VipHeader\_, 228
  - attributeOffset, 229
  - colorLength, 229
  - magicCode, 229
  - negativeXHoopSize, 229
  - negativeYHoopSize, 229
  - numberOfColors, 229
  - numberOfStitches, 229
  - postitiveXHoopSize, 229
  - postitiveYHoopSize, 229
  - stringVal, 229
  - unknown, 229
  - xOffset, 229
  - yOffset, 229
- vp3, 9, 455
- vp3Decode
  - format\_vp3.c, 454
- vp3DecodeInt16
  - format\_vp3.c, 454
- vp3Hoop
  - embroidery\_internal.h, 372
- vp3PatchByteCount
  - format\_vp3.c, 454
- vp3ReadHoopSection
  - format\_vp3.c, 454
- vp3ReadString
  - format\_vp3.c, 455
- vp3WriteString
  - format\_vp3.c, 455
- vp3WriteStringLen
  - format\_vp3.c, 455
- vulcanize
  - Geometry, 135
- vulcanize\_action
  - mainwindow.cpp, 301
- vulcanizeObject
  - View, 224
- vulcanizeRubberRoom
  - View, 224
- whats\_this\_action
  - mainwindow.cpp, 301
- wheelEvent
  - View, 224
- WHITESPACE
  - main.c, 482
- width
  - \_vp3Hoop, 49
- EmblImage\_, 90
- willOverflowInt32
  - View, 225
- willUnderflowInt32
  - View, 225
- window\_action
  - mainwindow.cpp, 302
- windowMenuAboutToShow
  - MainWindow, 160
- windowMenuActivated
  - MainWindow, 160
- wizardTipOfTheDay
  - mainwindow.cpp, 304
- write100
  - embroidery\_internal.h, 389
  - format\_100.c, 414
- write10o
  - embroidery\_internal.h, 389
  - format\_10o.c, 415
- write\_24bit
  - embroidery\_internal.h, 389
  - encoding.c, 405
  - main.c, 481
- write\_external\_color\_file
  - EmbFormatList\_, 86
- write\_settings
  - embroidermodder.h, 246
  - settings-dialog.cpp, 308
- writeArt
  - embroidery\_internal.h, 390
  - format\_art.c, 415
- writeBmc
  - embroidery\_internal.h, 390
  - format\_bmc.c, 416
- writeBro
  - embroidery\_internal.h, 390
  - format\_bro.c, 416
- writeCnd
  - embroidery\_internal.h, 390
  - format\_cnd.c, 417
- writeCol
  - embroidery\_internal.h, 390
  - format\_col.c, 418
- writeCsd
  - embroidery\_internal.h, 390
  - format\_csd.c, 419
- writeCsv
  - embroidery\_internal.h, 390
  - format\_csv.c, 420
- writeDat
  - embroidery\_internal.h, 390
  - format\_dat.c, 420
- writeDem
  - embroidery\_internal.h, 390
  - format\_dem.c, 421
- writeDsb
  - embroidery\_internal.h, 390
  - format\_dsb.c, 422

writeDst  
    embroidery\_internal.h, 390  
    format\_dst.c, 424

writeDsz  
    embroidery\_internal.h, 391  
    format\_dsz.c, 424

writeDxf  
    embroidery\_internal.h, 391  
    format\_dxf.c, 425

writeEdr  
    embroidery\_internal.h, 391  
    format\_edr.c, 425

writeEmd  
    embroidery\_internal.h, 391  
    format\_emd.c, 426

writeExp  
    embroidery\_internal.h, 391  
    format\_exp.c, 427

writeExy  
    embroidery\_internal.h, 391  
    format\_exy.c, 427

writeEys  
    embroidery\_internal.h, 391  
    format\_ey.c, 428

writeFxy  
    embroidery\_internal.h, 391  
    format\_fxy.c, 428

writeGc  
    embroidery\_internal.h, 391  
    format\_gc.c, 429

writeGnc  
    embroidery\_internal.h, 391  
    format\_gnc.c, 429

writeGt  
    embroidery\_internal.h, 391  
    format\_gt.c, 430

writeHus  
    embroidery\_internal.h, 392  
    format\_hus.c, 431

writelnImage  
    format\_pec.c, 439  
    image.c, 471

writelnb  
    embroidery\_internal.h, 392  
    format\_inb.c, 431

writelnf  
    embroidery\_internal.h, 392  
    format\_inf.c, 432

writeJef  
    embroidery\_internal.h, 392  
    format\_jef.c, 433

writeKsm  
    embroidery\_internal.h, 392  
    format\_ksm.c, 433

writeMax  
    embroidery\_internal.h, 392  
    format\_max.c, 434

writeMit  
    embroidery\_internal.h, 392  
    format\_mit.c, 435

writeNew  
    embroidery\_internal.h, 392  
    format\_new.c, 435

writeOfm  
    embroidery\_internal.h, 392  
    format\_ofm.c, 436

writePcd  
    embroidery\_internal.h, 392  
    format\_pcd.c, 437

writePcm  
    embroidery\_internal.h, 392  
    format\_pcm.c, 437

writePcq  
    embroidery\_internal.h, 393  
    format\_pcq.c, 438

writePcs  
    embroidery\_internal.h, 393  
    format\_pcs.c, 438

writePec  
    embroidery\_internal.h, 393  
    format\_pec.c, 439

writePecStitches  
    embroidery\_internal.h, 393  
    format\_pec.c, 439

writePel  
    embroidery\_internal.h, 393  
    format\_pel.c, 440

writePem  
    embroidery\_internal.h, 393  
    format\_pem.c, 440

writePes  
    embroidery\_internal.h, 393  
    format\_pes.c, 443

writePhb  
    embroidery\_internal.h, 393  
    format\_phb.c, 443

writePhc  
    embroidery\_internal.h, 393  
    format\_phc.c, 444

writePlt  
    embroidery\_internal.h, 393  
    format\_plt.c, 444

writer\_state  
    EmbFormatList\_, 86

writeRgb  
    embroidery\_internal.h, 394  
    format\_rgb.c, 445

writeSew  
    embroidery\_internal.h, 394  
    format\_sew.c, 445

writeShv  
    embroidery\_internal.h, 394  
    format\_shv.c, 446

writeSst  
    embroidery\_internal.h, 394  
    format\_sst.c, 447

- writeStx
  - embroidery\_internal.h, [394](#)
  - format\_stx.c, [447](#)
- writeSvg
  - embroidery\_internal.h, [394](#)
  - format\_svg.c, [448](#)
- writeT01
  - embroidery\_internal.h, [394](#)
  - format\_t01.c, [449](#)
- writeT09
  - embroidery\_internal.h, [394](#)
  - format\_t09.c, [449](#)
- writeTap
  - embroidery\_internal.h, [394](#)
  - format\_tap.c, [450](#)
- writeThr
  - embroidery\_internal.h, [394](#)
  - format\_thr.c, [451](#)
- writeTxt
  - embroidery\_internal.h, [394](#)
  - format\_txt.c, [451](#)
- writeU00
  - embroidery\_internal.h, [395](#)
  - format\_u00.c, [452](#)
- writeU01
  - embroidery\_internal.h, [395](#)
  - format\_u01.c, [452](#)
- writeVip
  - embroidery\_internal.h, [395](#)
  - format\_vip.c, [453](#)
- writeVp3
  - embroidery\_internal.h, [395](#)
  - format\_vp3.c, [455](#)
- writeXxx
  - embroidery\_internal.h, [395](#)
  - format\_xxx.c, [455](#)
- writeZsk
  - embroidery\_internal.h, [395](#)
  - format\_zsk.c, [456](#)
- x
  - EmbStitch\_, [99](#)
  - EmbVector\_, [102](#)
- x\_values
  - Geometry, [138](#)
- xOffset
  - \_vp3Hoop, [49](#)
  - VipHeader\_, [229](#)
- xxx, [9](#), [456](#)
- xxxDecodeByte
  - format\_xxx.c, [455](#)
- xxxEncodeDesign
  - format\_xxx.c, [456](#)
- xxxEncodeStitch
  - format\_xxx.c, [456](#)
- xxxEncodeStop
  - format\_xxx.c, [456](#)
- y
  - EmbStitch\_, [99](#)
  - EmbVector\_, [102](#)
- y\_values
  - Geometry, [138](#)
- year
  - EmbTime\_, [102](#)
- YELLOW\_TERM\_COLOR
  - embroidery\_internal.h, [371](#)
- yOffset
  - \_vp3Hoop, [49](#)
  - VipHeader\_, [229](#)
- Z102\_Isacord\_Polyester
  - embroidery.h, [329](#)
- zoom\_action
  - mainwindow.cpp, [302](#)
- zoomExtents
  - View, [225](#)
- zoomExtentsAllSubWindows
  - MdiArea, [166](#)
- zoomIn
  - View, [225](#)
- zoomOut
  - View, [225](#)
- zoomSelected
  - View, [225](#)
- zoomToPoint
  - View, [225](#)
- zoomWindow
  - View, [225](#)
- zoomWindowActive
  - View, [228](#)
- zsk, [9](#), [456](#)
- ZSK USA, [382](#), [424](#), [456](#)