Embroidermodder

Generated by Doxygen 1.9.4

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.

2 About

2.1 The Embroidermodder Project and Team

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

The tools and associated documents are:

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

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

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

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

2.1.1 "Core Development Team"

Embroidermodder 2:

- `Jonathan Greig`
- `Josh Varga`
- `Robin Swift`

Embroidermodder 1:

- `Josh Varga`
- `Mark Pontius`

2.2 for Embroidermodder 2, libembroidery and all other related code

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

2.2.1 Embroidermodder 1

The Embroidermodder Team is also inspired by the original Embroidermodder that was built by Mark Pontius and the same Josh Varga on SourceForge which unfortunately appears to have died from linkrot. We may create a distribution on here to be the official "legacy" Embroidermodder code but likely in a seperate 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

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
`rw-none`	Either the format produces no output, reporting an error. Or it produces a Tajima dst file as an alternative.
`rw-poor`	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.
`rw-basic`	Simple files in this format run well on machines that use this format.
`rw-standard`	Files with non-standard features work on machines and we have good documentation on the format.
`rw-reliable`	All known features don't cause crashes. Almost all work as expected.
`rw-complete`	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
`test-none`	No tests have been written to test the specifics of the format.
`test-basic`	Stitch Lists and/or colors have read/write tests.
`test-thorough`	All features of that format has at least one test.
`test-fuzz`	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.
`test-complete`	Both thorough and fuzz testing is covered.

6.1.1.1 Test Support Levels

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

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

6.1.1.2 Documentation Support Levels

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

6.1.1.3 Overall Support

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

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

6.1.2 Table of Format Support Levels

Overview of documentation support by format.

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

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

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

6.1.3 Format Support

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

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

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

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

7 Geometry and Algorithms

7.1 To Do

Todo (Arduino) Fix emb-outline files

Todo (Arduino)Fix thread-color files

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

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

Todo (Arduino) Smoothieboard experiments

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

7.1.1 Development

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

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

7.1.2 Testing

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

```
$ embroidermodder --test
```

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

7.1.3 Contributing

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

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

- Low level C developers are be needed for the base library `libembroidery`.
- Low level assembly programmers are needed for translating some of `libembroidery` to `EmbroiderBot`.
- 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 dolphi. 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 \texttt{_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 dependancies 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 `libembrodiery\_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 beore 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 `stitchList\_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

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. Secondarily, 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, La↩
TeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

9 Contributor Covenant Code of Conduct

9.1 Our Pledge

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

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

9.2 Our Standards

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

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

Examples of unacceptable behavior include:

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

9.3 Enforcement Responsibilities

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

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

9.4 Scope

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

9.5 Enforcement

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

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

9.6 Enforcement Guidelines

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

9.6.1 1. Correction

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

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

9.6.2 2. Warning

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

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

9.6.3 3. Temporary Ban

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

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

9.6.4 4. Permanent Ban

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

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

9.7 Attribution

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

Community Impact Guidelines were inspired by Mozilla's code of conduct enforcement ladder.

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

10 Privacy Policy for Embroidery Viewer

Last updated December 15, 2021

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

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

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

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

10.0.1 CONTACT US

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

Embroidermodder@gmail.com

11 Todo List

Member about_action (String args)

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

Member bcf_directory

possibly add a directory tree in the future.

Member bcf_file_header

CLSID should be a separate type.

Member binaryWriteInt (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 lineweight

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 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.

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 &obj←Path, 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:

13 Hierarchical Index

13.1 Class Hierarchy

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

14 Class Index

14.1 Class List

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

15 File Index

15.1 File List

Here is a list of all files with brief descriptions:

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:

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

"""
```

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

argc	
argv	

17.8.3 Member Function Documentation

17.8.3.1 event()

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

Application::event.

Parameters

event	

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

parent	

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

txt	

17.9.3.2 appendHistory

```
void appendHistory (
    QString txt ) [slot]
```

CmdPrompt::appendHistory.

Parameters

txt	

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

isFloating	

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

fileName	
html	

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

txt	

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

CmdPrompt::setPromptBackgroundColor.

Parameters

color	

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

CmdPrompt::setPromptFontFamily.

Parameters

family	

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

CmdPrompt::setPromptFontSize.

Parameters

size	

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

CmdPrompt::setPromptFontStyle.

Parameters

style	

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

CmdPrompt::setPromptTextColor.

Parameters

color	

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

orientation	
parent	

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

e	The mouse event.

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

CmdPromptHandle::mousePressEvent.

Parameters

e	

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

CmdPromptHandle::mouseReleaseEvent.

Parameters

e	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

parent	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

txt	
prefixLength	

17.11.3.2 applyFormatting() `QString applyFormatting (`

```
QString txt,
int prefixLength )
```

CmdPromptHistory::applyFormatting.

Parameters

txt	
prefixLength	

Returns

17.11.3.3 contextMenuEvent() `void contextMenuEvent (`

```
QContextMenuEvent * event )   [protected]
```

CmdPromptHistory::contextMenuEvent.

Parameters

event	

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

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

CmdPromptHistory::resizeHistory.

Parameters

y	

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

parent	

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

formats	

17.12.2.4 checkChangedText void checkChangedText (

```
QString txt ) [slot]
```

CmdPromptInput::checkChangedText.

Parameters

txt	

17.12.2.5 checkCursorPosition

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

CmdPromptInput::checkCursorPosition.

Parameters

oldpos	
newpos	

17.12.2.6 checkEditedText

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

CmdPromptInput::checkEditedText.

Parameters

txt	

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

event	

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

obj	
event	

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

txt	

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

parent	

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

theScene	
parent	

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 EmbImage_ Struct Reference

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

Public Attributes

- EmbVector position
- EmbVector dimensions
- unsigned char ∗ data
- int width
- int height
- char path [200]
- char name [200]

17.29.1 Member Data Documentation

17.29.1.1 data `unsigned char∗ data`

17.29.1.2 dimensions `EmbVector dimensions`

17.29.1.3 height `int height`

17.29.1.4 name `char name[200]`

17.29.1.5 path `char path[200]`

17.29.1.6 position `EmbVector position`

17.29.1.7 width `int width`

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

- extern/libembroidery/src/embroidery.h

17.30 EmbInfiniteLine_ Struct Reference

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

Public Attributes

- EmbVector position

17.30.1 Member Data Documentation

17.30.1.1 position `EmbVector position`

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

- extern/libembroidery/src/embroidery.h

17.31 EmbLayer_ Struct Reference

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

Public Attributes

- char name [100]
- EmbArray ∗ geometry

17.31.1 Member Data Documentation

17.31.1.1 geometry `EmbArray* geometry`

17.31.1.2 name `char name[100]`

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

- extern/libembroidery/src/embroidery.h

17.32 EmbLeaderDim_ Struct Reference

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

Public Attributes

- EmbVector position

17.32.1 Member Data Documentation

17.32.1.1 position `EmbVector position`

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

- extern/libembroidery/src/embroidery.h

17.33 EmbLine_ Struct Reference

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

Public Attributes

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

17.33.1 Member Data Documentation

17.33.1.1 color `EmbColor color`

17.33.1.2 end `EmbVector end`

17.33.1.3 lineType `int lineType`

17.33.1.4 start `EmbVector start`

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

- extern/libembroidery/src/embroidery.h

17.34 EmbLinearDim_ Struct Reference

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

Public Attributes

- EmbVector position

17.34.1 Member Data Documentation

17.34.1.1 position `EmbVector position`

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

- extern/libembroidery/src/embroidery.h

17.35 EmbOrdinateDim_ Struct Reference

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

Public Attributes

- EmbVector position

17.35.1 Member Data Documentation

17.35.1.1 position `EmbVector position`

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

- extern/libembroidery/src/embroidery.h

17.36 EmbPath_ Struct Reference

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

Public Attributes

- EmbArray ∗ pointList
- EmbArray ∗ flagList
- int lineType
- EmbColor color

17.36.1 Member Data Documentation

17.36.1.1 color `EmbColor color`

17.36.1.2 flagList `EmbArray* flagList`

17.36.1.3 lineType `int lineType`

17.36.1.4 pointList `EmbArray* pointList`

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

- extern/libembroidery/src/embroidery.h

17.37 EmbPattern_ Struct Reference

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

Public Attributes

- unsigned int dstJumpsPerTrim
- EmbVector home
- EmbReal hoop_width
- EmbReal hoop_height
- EmbArray ∗ thread_list
- EmbArray ∗ stitch_list
- EmbArray ∗ geometry
- EmbLayer layer [EMB_MAX_LAYERS]
- int currentColorIndex

17.37.1 Member Data Documentation

17.37.1.1 currentColorIndex `int currentColorIndex`

17.37.1.2 dstJumpsPerTrim `unsigned int dstJumpsPerTrim`

17.37.1.3 geometry `EmbArray* geometry`

17.37.1.4 home `EmbVector home`

17.37.1.5 hoop_height `EmbReal hoop_height`

17.37.1.6 hoop_width `EmbReal hoop_width`

17.37.1.7 layer `EmbLayer layer[EMB_MAX_LAYERS]`

17.37.1.8 stitch_list `EmbArray* stitch_list`

17.37.1.9 thread_list `EmbArray* thread_list`

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

- extern/libembroidery/src/embroidery.h

17.38 EmbPoint_ Struct Reference

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

Public Attributes

- EmbVector position
- int lineType
- EmbColor color

17.38.1 Member Data Documentation

17.38.1.1 color `EmbColor color`

17.38.1.2 lineType `int lineType`

17.38.1.3 position `EmbVector position`

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

- extern/libembroidery/src/embroidery.h

17.39 EmbRadiusDim_ Struct Reference

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

Public Attributes

- EmbVector position

17.39.1 Member Data Documentation

17.39.1.1 position `EmbVector position`

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

- extern/libembroidery/src/embroidery.h

17.40 EmbRay_ Struct Reference

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

Public Attributes

- EmbVector position

17.40.1 Member Data Documentation

17.40.1.1 position `EmbVector position`

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

- extern/libembroidery/src/embroidery.h

17.41 EmbRect_ Struct Reference

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

Public Attributes

- EmbReal top
- EmbReal left
- EmbReal bottom
- EmbReal right
- EmbReal rotation
- EmbReal radius

17.41.1 Member Data Documentation

17.41.1.1 bottom `EmbReal bottom`

17.41.1.2 left `EmbReal left`

17.41.1.3 radius `EmbReal radius`

17.41.1.4 right `EmbReal right`

17.41.1.5 rotation `EmbReal rotation`

17.41.1.6 top `EmbReal top`

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

- extern/libembroidery/src/embroidery.h

17.42 EmbSatinOutline_ Struct Reference

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

Public Attributes

- int length
- EmbArray ∗ side1
- EmbArray ∗ side2

17.42.1 Member Data Documentation

17.42.1.1 length `int length`

17.42.1.2 side1 `EmbArray* side1`

17.42.1.3 side2 `EmbArray* side2`

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

- extern/libembroidery/src/embroidery.h

17.43 EmbSpline_ Struct Reference

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

Public Attributes

- EmbArray ∗ beziers

17.43.1 Member Data Documentation

17.43.1.1 beziers `EmbArray* beziers`

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

- extern/libembroidery/src/embroidery.h

17.44 EmbStitch_ Struct Reference

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

Public Attributes

- int flags
- EmbReal x
- EmbReal y
- int color

17.44.1 Member Data Documentation

17.44.1.1 color `int color`

positive is up, units are in mm

17.44.1.2 flags `int flags`

17.44.1.3 x `EmbReal x`

uses codes defined above

17.44.1.4 y `EmbReal y`

absolute position (not relative)

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

- extern/libembroidery/src/embroidery.h

17.45 EmbTextMulti_ Struct Reference

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

Public Attributes

- EmbVector position
- char text [200]

17.45.1 Member Data Documentation

17.45.1.1 position `EmbVector position`

17.45.1.2 text `char text[200]`

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

- extern/libembroidery/src/embroidery.h

17.46 EmbTextSingle_ Struct Reference

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

Public Attributes

- EmbVector position
- char text [200]

17.46.1 Member Data Documentation

17.46.1.1 position `EmbVector position`

17.46.1.2 text `char text[200]`

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

- extern/libembroidery/src/embroidery.h

17.47 EmbThread_ Struct Reference

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

Public Attributes

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

17.47.1 Member Data Documentation

17.47.1.1 catalogNumber `char catalogNumber[30]`

17.47.1.2 color `EmbColor color`

17.47.1.3 description `char description[50]`

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

- extern/libembroidery/src/embroidery.h

17.48 EmbTime_ Struct Reference

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

Public Attributes

- unsigned int year
- unsigned int month
- unsigned int day
- unsigned int hour
- unsigned int minute
- unsigned int second

17.48.1 Member Data Documentation

17.48.1.1 day `unsigned int day`

17.48.1.2 hour `unsigned int hour`

17.48.1.3 minute `unsigned int minute`

17.48.1.4 month `unsigned int month`

17.48.1.5 second `unsigned int second`

17.48.1.6 year `unsigned int year`

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

- extern/libembroidery/src/embroidery.h

17.49 EmbVector_ Struct Reference

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

Public Attributes

- EmbReal x
- EmbReal y

17.49.1 Detailed Description

The basic type to represent points absolutely or represent directions.

Positive y is up, units are in mm.

17.49.2 Member Data Documentation

17.49.2.1 x `EmbReal x`

17.49.2.2 y `EmbReal y`

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

- extern/libembroidery/src/embroidery.h

17.50 Geometry Class Reference

The Geometry class.

`#include <embroidermodder.h>`

Public Types

- enum ArrowStyle {
  NoArrow , Open , Closed , Dot ,
  Box , Tick }
- enum lineStyle { NoLine , Flared , Fletching }

Public Member Functions

- virtual int type ()
- Geometry (int object_type=OBJ_TYPE_BASE, QGraphicsItem ∗parent=0)
- Geometry (Geometry ∗obj, QGraphicsItem ∗parent=0)

  Geometry::Geometry.
- Geometry (EmbArc arc, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem ∗parent=0)

  Construct a new Geometry:: Geometry object.
- Geometry (EmbCircle circle, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem ∗parent=0)

  Construct a new Geometry:: Geometry object.
- Geometry (EmbLine line, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem ∗parent=0)
- Geometry (EmbEllipse ellipse, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem ∗parent=0)

  Construct a new Geometry:: Geometry object.
- Geometry (EmbRect rect, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem ∗parent=0)

  Construct a new Geometry object.
- Geometry (QString str, EmbVector position, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem ∗parent=0)

  Construct a new Geometry object.
- Geometry (EmbLine line, int Type_, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem ∗parent)

  Construct a new Geometry:: Geometry object.
- Geometry (QPainterPath p, int type_, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem ∗parent=0)

  Construct a new Geometry:: Geometry object.
- Geometry (EmbVector pos, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem ∗parent=0)

  Construct a new Geometry:: Geometry object.
- void init_arc (EmbArc arc, QRgb rgb, Qt::PenStyle lineType)

  Geometry::init.
- void init_circle (EmbCircle circle, QRgb rgb, Qt::PenStyle lineType)

  Geometry::init_circle.
- void init_line (EmbLine line, QRgb rgb, Qt::PenStyle lineType)

  Geometry::init_line.
- void init_ellipse (EmbEllipse ellipse, QRgb rgb, Qt::PenStyle lineType)

Geometry::init_ellipse.

- void init_rect (EmbRect rect, QRgb rgb, Qt::PenStyle lineType)

  Geometry::init_line.
- void init_text_single (QString str, EmbVector position, QRgb rgb, Qt::PenStyle lineType)

  Geometry::init_line.
- void init_path (QPainterPath p, QRgb rgb, Qt::PenStyle lineType)

  Geometry::init_line.
- void init_point (EmbVector pos, QRgb rgb, Qt::PenStyle lineType)

  Geometry::init_line.
- void init (void)
- ∼Geometry ()

  Geometry::∼Geometry.
- Qt::PenStyle objectLineType ()
- EmbReal objectLineWeight ()
- QPointF objectRubberPoint (QString key)

  Geometry::objectRubberPoint.
- QString objectRubberText (QString key)

  Geometry::objectRubberText.
- QPointF objectCenter ()
- QPointF objectPos ()
- EmbReal objectX ()
- EmbReal objectY ()
- QPointF objectTopLeft ()
- QPointF objectTopRight ()
- QPointF objectBottomLeft ()
- QPointF objectBottomRight ()
- EmbReal objectArea ()

  Geometry::objectArea.
- QPointF objectStartPoint ()

  Geometry::objectStartPoint.
- QPointF objectMidPoint ()

  Geometry::objectMidPoint.
- QPointF objectEndPoint ()

  Geometry::objectEndPoint.
- QRectF rect ()
- void circle_click (Dictionary global, EmbVector v)
- EmbReal objectWidth ()
- EmbReal objectHeight ()
- EmbReal objectRadiusMajor ()
- EmbReal objectRadiusMinor ()
- EmbReal objectDiameterMajor ()
- EmbReal objectDiameterMinor ()
- QPointF objectEndPoint1 ()

  DimLeaderObject::objectEndPoint1.
- QPointF objectEndPoint2 ()

  Geometry::objectEndPoint2.
- EmbReal objectStartAngle ()

  Geometry::objectStartAngle.
- EmbReal objectEndAngle ()

  Geometry::objectEndAngle.
- EmbReal objectArcLength ()

  Geometry::objectArcLength.

- EmbReal objectChord ()

  Geometry::objectChord.
- EmbReal objectIncludedAngle ()

  Geometry::objectIncludedAngle.
- bool objectClockwise ()

  Geometry::objectClockwise.
- EmbReal objectX1 ()
- EmbReal objectY1 ()
- EmbReal objectX2 ()
- EmbReal objectY2 ()
- EmbReal objectAngle ()

  DimLeaderObject::objectAngle.
- QPointF objectDelta ()
- EmbReal objectLength ()
- EmbReal objectRadius ()
- EmbReal objectDiameter ()
- EmbReal objectCircumference ()
- QPointF objectQuadrant0 ()
- QPointF objectQuadrant90 ()
- QPointF objectQuadrant180 ()
- QPointF objectQuadrant270 ()
- QPainterPath objectCopyPath ()

  PathObject::objectCopyPath.
- QPainterPath objectSavePath ()

  Geometry::objectSavePath.
- std::vector< QPainterPath > objectSavePathList ()
- std::vector< QPainterPath > subPathList ()
- int findIndex (const QPointF &point)

  Geometry::findIndex.
- void setObjectEndPoint1 (EmbVector endPt1)

  DimLeaderObject::setObjectEndPoint1.
- void setObjectEndPoint2 (EmbVector endPt2)

  DimLeaderObject::setObjectEndPoint2.
- void updatePath ()

  Geometry::updatePath.
- void updatePath (const QPainterPath &p)

  Geometry::updatePath.
- void updateLeader (void)

  DimLeaderObject::updateLeader.
- virtual QRectF boundingRect ()
- void drawRubberLine (const QLineF &rubLine, QPainter ∗painter=0, const char ∗colorFromScene=0)

  Geometry::drawRubberLine.
- void updateRubber (QPainter ∗painter=0)

  DimLeaderObject::updateRubber.
- void vulcanize (void)

  DimLeaderObject::vulcanize.
- QPointF mouseSnapPoint (const QPointF &mousePoint)

  Geometry::mouseSnapPoint.
- std::vector< QPointF > allGripPoints ()

  Geometry::allGripPoints.
- void gripEdit (const QPointF &before, const QPointF &after)

  DimLeaderObject::gripEdit.

- void realRender (QPainter ∗painter, const QPainterPath &renderPath)

  Geometry::realRender.
- void paint (QPainter ∗, const QStyleOptionGraphicsItem ∗, QWidget ∗)

  Geometry::paint.
- void calculateArcData (EmbArc arc)

  Geometry::calculateArcData.
- void updateArcRect (EmbReal radius)

  Geometry::updateArcRect.
- void setObjectPos (const QPointF &point)
- void setObjectX (EmbReal x)
- void setObjectY (EmbReal y)
- void setObjectCenter (EmbVector center)
- void setObjectCenterX (EmbReal centerX)
- void setObjectCenterY (EmbReal centerY)
- void setObjectSize (EmbReal width, EmbReal height)
- void setObjectRect (EmbReal x, EmbReal y, EmbReal w, EmbReal h)
- void setRect (const QRectF &r)
- void setRect (EmbReal x, EmbReal y, EmbReal w, EmbReal h)
- void setLine (const QLineF &li)
- void setLine (EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2)
- void setObjectLineWeight (String lineWeight)

  Geometry::setObjectLineWeight.
- void setObjectRadius (EmbReal radius)

  Geometry::setObjectRadius.
- void setObjectStartAngle (EmbReal angle)

  Geometry::setObjectStartAngle.
- void setObjectEndAngle (EmbReal angle)

  Geometry::setObjectEndAngle.
- void setObjectStartPoint (EmbVector point)

  Geometry::setObjectStartPoint.
- void setObjectMidPoint (EmbVector point)

  Geometry::setObjectMidPoint.
- void setObjectEndPoint (EmbVector point)

  Geometry::setObjectEndPoint.
- void setObjectDiameter (EmbReal diameter)

  Geometry::setObjectDiameter.
- void setObjectArea (EmbReal area)

  Geometry::setObjectArea.
- void setObjectCircumference (EmbReal circumference)

  Geometry::setObjectCircumference.
- void setObjectPos (EmbReal x, EmbReal y)
- void setObjectText (QString str)
- void setObjectTextFont (QString font)
- void setObjectTextJustify (QString justify)

  Verify the string is a valid option, otherwise default to "Left".
- void setObjectTextSize (EmbReal size)
- void setObjectTextStyle (bool bold, bool italic, bool under, bool strike, bool over)
- void setObjectTextBold (bool val)
- void setObjectTextItalic (bool val)
- void setObjectTextUnderline (bool val)
- void setObjectTextStrikeOut (bool val)
- void setObjectTextOverline (bool val)

- void setObjectTextBackward (bool val)
- void setObjectTextUpsideDown (bool val)
- void setObjectRadiusMajor (EmbReal radius)
- void setObjectRadiusMinor (EmbReal radius)
- void setObjectDiameterMajor (EmbReal diameter)
- void setObjectDiameterMinor (EmbReal diameter)
- void script_main (void)
- void script_click (EmbVector v)

    circle_click
- void script_context (String str)
- void script_prompt (String str)

Public Attributes

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

17.50.1 Detailed Description

The Geometry class.

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

17.50.2 Member Enumeration Documentation

17.50.2.1 ArrowStyle `enum ArrowStyle`

Enumerator

NoArrow	
Open	
Closed	
Dot	
Box	
Tick	

17.50.2.2 lineStyle enum lineStyle

Enumerator

NoLine	
Flared	
Fletching	

17.50.3 Constructor & Destructor Documentation

17.50.3.1 Geometry() [1/11] Geometry (

```
int object_type = OBJ_TYPE_BASE,
QGraphicsItem * parent = 0 )
```

17.50.3.2 Geometry() [2/11] Geometry (

```
Geometry * obj,
QGraphicsItem * parent = 0 )
```

Geometry::Geometry.

Parameters

*obj	
*parent	

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

arc	
rgb	
lineType	
parent	

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

circle	
rgb	
lineType	
parent	

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

ellipse	
rgb	
lineType	
parent	

17.50.3.7 Geometry() [7/11]

```
Geometry (
        EmbRect rect,
        QRgb rgb,
        Qt::PenStyle lineType,
        QGraphicsItem * parent = 0 )
```

Construct a new Geometry object.

Parameters

rect	
rgb	
lineType	
parent	

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

str	
v	
rgb	
lineType	
parent	

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

line	
*Type_*	

Parameters

rgb	
lineType	
parent	

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

p	
*Type_*	
rgb	
lineType	
parent	

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

vector	
rgb	
lineType	
parent	

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

arc	

Todo convert this to update and make it Type sensitive.

17.50.4.4 circle_click()

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

17.50.4.5 drawRubberLine()

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

Geometry::drawRubberLine.

Parameters

rubLine	
painter	
colorFromScene	

17.50.4.6 findIndex()

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

Geometry::findIndex.

Parameters

point	

Returns

17.50.4.7 gripEdit()

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

DimLeaderObject::gripEdit.

Parameters

before	
after	

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

arc	
rgb	
lineType	

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

17.50.4.10 init_circle()

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

Geometry::init_circle.

Parameters

circle	
rgb	
lineType	

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

17.50.4.11 init_ellipse()

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

Geometry::init_ellipse.

Parameters

ellipse	
rgb	
lineType	

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.

17.50.4.12 init_line()

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

Geometry::init_line.

Parameters

ellipse	
rgb	
lineType	

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.

17.50.4.13 init_path()

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

Geometry::init_line.

Parameters

ellipse	
rgb	
lineType	

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.

17.50.4.14 init_point()

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

Geometry::init_line.

Parameters

ellipse	
rgb	
lineType	

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

ellipse	
rgb	
lineType	

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

ellipse	
rgb	
lineType	

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 lineweight

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

Geometry::mouseSnapPoint.

Parameters

mousePoint	

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

key	

Returns

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

Geometry::objectRubberText.

Parameters

key	

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

painter	
option	

17.50.4.66 realRender()

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

Geometry::realRender.

Parameters

painter	
renderPath	

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

str	

17.50.4.70 script_main() `void script_main (`

`void )`

17.50.4.71 script_prompt() `void script_prompt (`

`String str )`

Parameters

str	

17.50.4.72 setLine() `[1/2]` `void setLine (`

```
const QLineF & li )
```

17.50.4.73 setLine() [2/2]

```
void setLine (
    EmbReal x1,
    EmbReal y1,
    EmbReal x2,
    EmbReal y2 )
```

17.50.4.74 setObjectArea()

```
void setObjectArea (
    EmbReal area )
```

Geometry::setObjectArea.

Parameters

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

diameter	

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

angle	

17.50.4.83 setObjectEndPoint()

```
void setObjectEndPoint (
    EmbVector point )
```

Geometry::setObjectEndPoint.

Parameters

point	

17.50.4.84 setObjectEndPoint1()

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

DimLeaderObject::setObjectEndPoint1.

Parameters

x1	
y1	

17.50.4.85 setObjectEndPoint2()

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

DimLeaderObject::setObjectEndPoint2.

Parameters

x2	
y2	

17.50.4.86 setObjectLineWeight()

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

Geometry::setObjectLineWeight.

Parameters

lineWeight	

17.50.4.87 setObjectMidPoint()

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

Geometry::setObjectMidPoint.

Parameters

point	

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

radius	

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

angle	

17.50.4.96 setObjectStartPoint()

```
void setObjectStartPoint (
    EmbVector point )
```

Geometry::setObjectStartPoint.

Parameters

point	

17.50.4.97 setObjectText()

```
void setObjectText (
    QString str )
```

17.50.4.98 setObjectTextBackward()

```
void setObjectTextBackward (
    bool val )
```

Parameters

val	

17.50.4.99 setObjectTextBold()

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

Parameters

val	

17.50.4.100 setObjectTextFont()

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

Parameters

font	

17.50.4.101 setObjectTextItalic()

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

Parameters

val	

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

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

Parameters

justify	

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

Parameters

val	

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

Parameters

size	

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

Parameters

val	

17.50.4.106 setObjectTextStyle() `void setObjectTextStyle ( bool bold,`

```
bool italic,
bool under,
bool strike,
bool over )
```

Parameters

bold	
italic	
under	
strike	
over	

17.50.4.107 setObjectTextUnderline()

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

Parameters

val	

17.50.4.108 setObjectTextUpsideDown()

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

Parameters

val	

17.50.4.109 setObjectX()

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

17.50.4.110 setObjectY()

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

17.50.4.111 setRect() [1/2]

```
void setRect (
    const QRectF & r )
```

17.50.4.112 setRect() [2/2]

```
void setRect (
    EmbReal x,
    EmbReal y,
    EmbReal w,
    EmbReal h )
```

17.50.4.113 subPathList()

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

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

radius	

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

p	

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

DimLeaderObject::updateRubber.

Parameters

painter	

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

filename	
parent	

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

fileName	

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

fileName	

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 QRgb 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 QRgb color,
QString lineType,
QString lineWeight,
const bool print )
```

LayerManager::addLayer.

Parameters

name	
visible	
frozen	
zValue	
color	
lineType	
lineWeight	
print	

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.
- QRgb getCurrentColor ()

    MainWindow::getCurrentColor.
- QString getCurrentLineType ()

    MainWindow::getCurrentLineType.
- QString getCurrentLineWeight ()

    MainWindow::getCurrentLineWeight.
- bool isShiftPressed ()
- void setShiftPressed ()
- void setShiftReleased ()
- void deletePressed ()

    MainWindow::deletePressed.
- void escapePressed ()

    MainWindow::escapePressed.

Public Member Functions

- MainWindow ()

    MainWindow::MainWindow.
- ∼MainWindow ()

    MainWindow::∼MainWindow.
- MdiWindow $*$ activeMdiWindow ()

    MainWindow::activeMdiWindow.
- QUndoStack $*$ activeUndoStack ()

    MainWindow::activeUndoStack.
- void setUndoCleanIcon (bool opened)

    MainWindow::setUndoCleanIcon.
- virtual void updateMenuToolbarStatusbar ()

    MainWindow::updateMenuToolbarStatusbar.
- bool isCommandActive ()
- QString activeCommand ()
- QIcon create_icon (QString stub)

    MainWindow::create_icon.
- void create_toolbar (String toolbar, String label, StringList entries)

    MainWindow::create_toolbar.
- QString platformString ()

Public Attributes

- std::vector< QGraphicsItem ∗ > cutCopyObjectList
- QString formatFilterOpen
- QString formatFilterSave

Protected Member Functions

- virtual void resizeEvent (QResizeEvent ∗)

  MainWindow::resizeEvent.
- void closeEvent (QCloseEvent ∗event)

  MainWindow::closeEvent.
- QAction ∗ getFileSeparator ()

  MainWindow::getFileSeparator.
- void loadFormats ()

  MainWindow::loadFormats.
- QMdiSubWindow ∗ findMdiWindow (String fileName)

  MainWindow::findMdiWindow.
- void createAllActions ()

  MainWindow::createAllActions.
- void createAllMenus ()

  MainWindow::createAllMenus.
- void createAllToolbars ()

  MainWindow::createAllToolbars.

Protected Attributes

- bool shiftKeyPressedState
- QByteArray layoutState
- int numOfDocs
- int docIndex
- std::vector< MdiWindow ∗ > listMdiWin
- QAction ∗ myFileSeparator
- QComboBox ∗ layerSelector
- QComboBox ∗ colorSelector
- QComboBox ∗ linetypeSelector
- QComboBox ∗ lineweightSelector
- QFontComboBox ∗ textFontSelector
- QComboBox ∗ textSizeSelector

Private Slots

- void hideUnimplemented ()

  MainWindow::hideUnimplemented.

17.56.1 Detailed Description

The MainWindow class.

17.56.2 Constructor & Destructor Documentation

17.56.2.1 MainWindow() `MainWindow ( )`

MainWindow::MainWindow.

17.56.2.2 ∼MainWindow() `∼MainWindow ( )`

MainWindow::∼MainWindow.

17.56.3 Member Function Documentation

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

about_action

Parameters

args	

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

event	

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

MainWindow::closeToolBar.

Parameters

action	

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

MainWindow::colorSelectorIndexChanged.

Parameters

index	

17.56.3.10 create_icon() `QIcon create_icon ( QString stub )`

MainWindow::create_icon.

Parameters

stub	

Returns

17.56.3.11 create_toolbar() `void create_toolbar ( String toolbar, String label, StringList entries )`

MainWindow::create_toolbar.

Parameters

toolbar	
label	
entries	

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

fileName	

Returns

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

MainWindow::floatingChangedToolBar.

Parameters

isFloating	

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

index	

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

17.56.3.30 lineweightSelectorIndexChanged `void lineweightSelectorIndexChanged ( int index ) [slot]`

MainWindow::lineweightSelectorIndexChanged.

Parameters

index	

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

MainWindow::loadFormats.

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

MainWindow::logPromptInput.

Parameters

txt	

17.56.3.33 newFile void newFile ( ) [slot]

MainWindow::newFile.

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

MainWindow::onCloseMdiWin.

Parameters

theMdiWin	

17.56.3.35 onCloseWindow void onCloseWindow ( ) [slot]

MainWindow::onCloseWindow.

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

MainWindow::onWindowActivated.

Parameters

w	

17.56.3.37 openFile

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

MainWindow::openFile.

Parameters

recent	
recentFile	

17.56.3.38 openFilesSelected

```
void openFilesSelected (
    StringList filesToOpen ) [slot]
```

MainWindow::openFilesSelected.

Parameters

filesToOpen	

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

txt	

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

e	

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::setTextFont.

Parameters

str	

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

MainWindow::setTextSize.

Parameters

num	

17.56.3.54 settingsPrompt `void settingsPrompt ( ) [slot]`

MainWindow::settingsPrompt.

17.56.3.55 setUndoCleanIcon() `void setUndoCleanIcon ( bool opened )`

MainWindow::setUndoCleanIcon.

Parameters

opened	

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

font	

17.56.3.58 textSizeSelectorIndexChanged

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

MainWindow::textSizeSelectorIndexChanged.

Parameters

index	

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

color	

17.56.3.64 updateAllViewCrossHairColors

```
void updateAllViewCrossHairColors (
        QRgb color ) [slot]
```

MainWindow::updateAllViewCrossHairColors.

Parameters

color	

17.56.3.65 updateAllViewGridColors

```
void updateAllViewGridColors (
        QRgb color ) [slot]
```

MainWindow::updateAllViewGridColors.

Parameters

color	

17.56.3.66 updateAllViewRulerColors

```
void updateAllViewRulerColors (
        QRgb color ) [slot]
```

MainWindow::updateAllViewRulerColors.

Parameters

color	

17.56.3.67 updateAllViewScrollBars

```
void updateAllViewScrollBars (
        bool val ) [slot]
```

MainWindow::updateAllViewScrollBars.

Parameters

val	

17.56.3.68 updateAllViewSelectBoxColors

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

MainWindow::updateAllViewSelectBoxColors.

Parameters

colorL	
fillL	
colorR	
fillR	
alpha	

17.56.3.69 updateMenuToolbarStatusbar()

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

MainWindow::updateMenuToolbarStatusbar.

17.56.3.70 updatePickAddMode

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

MainWindow::updatePickAddMode.

Parameters

val	

17.56.3.71 windowMenuAboutToShow

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

MainWindow::windowMenuAboutToShow.

17.56.3.72 windowMenuActivated

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

MainWindow::windowMenuActivated.

Parameters

checked	

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

mw	
parent	

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

color	

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

MdiArea::setBackgroundLogo.

Parameters

fileName	

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

MdiArea::setBackgroundTexture.

Parameters

fileName	

17.57.2.8 tile `void tile ( ) [slot]`

MdiArea::tile.

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

Parameters

use	

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

MdiArea::useBackgroundLogo.

Parameters

use	

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

MdiArea::useBackgroundTexture.

Parameters

use	

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 QRgb &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 (QRgb color)
- void setViewBackgroundColor (QRgb color)
- void setViewSelectBoxColors (QRgb colorL, QRgb fillL, QRgb colorR, QRgb fillR, int alpha)
- void setViewGridColor (QRgb color)
- void setViewRulerColor (QRgb color)
- void print ()

  MdiWindow::print.
- void saveBMC ()

  MdiWindow::saveBMC.
- void promptHistoryAppended (QString txt)
- void logPromptInput (QString txt)
- void promptInputPrevious ()
- void promptInputNext ()

  MdiWindow::promptInputNext.

Signals

- void sendCloseMdiWin (MdiWindow ∗)

Public Member Functions

- MdiWindow (const int theIndex, QMdiArea ∗parent, Qt::WindowFlags wflags)

  Construct a new MdiWindow object.
- ∼MdiWindow ()

  MdiWindow::∼MdiWindow.
- void setCurrentFile (QString fileName)

  MdiWindow::setCurrentFile.
- void promptInputPrevNext (bool prev)

  MdiWindow::promptInputPrevNext.
- virtual QSize sizeHint ()

  MdiWindow::sizeHint.
- QString getShortCurrentFile ()

  MdiWindow::getShortCurrentFile.
- void designDetails ()
- bool loadFile (String fileName)

  MdiWindow::loadFile.
- bool saveFile (String fileName)

  MdiWindow::saveFile.

Public Attributes

- QMdiArea ∗ mdiArea
- QGraphicsScene ∗ gscene
- View ∗ gview
- bool fileWasLoaded
- QString promptHistory
- std::vector< QString > promptInputList
- int promptInputNum
- QPrinter printer
- QString curFile
- int myIndex
- QString curLayer
- QRgb curColor
- QString curLineType
- QString curLineWeight

17.58.1 Constructor & Destructor Documentation

17.58.1.1 MdiWindow() MdiWindow (

```
const int theIndex,
QMdiArea * parent,
Qt::WindowFlags wflags )
```

Construct a new MdiWindow object.

Parameters

theIndex	
parent	
wflags	

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

color	

17.58.2.3 currentLayerChanged `void currentLayerChanged ( QString layer ) [slot]`

MdiWindow::currentLayerChanged.

Parameters

layer	

17.58.2.4 currentLinetypeChanged `void currentLinetypeChanged ( QString type ) [slot]`

MdiWindow::currentLinetypeChanged.

Parameters

type	

17.58.2.5 currentLineweightChanged `void currentLineweightChanged ( QString weight ) [slot]`

MdiWindow::currentLineweightChanged.

Parameters

weight	

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

fileName	

Returns

17.58.2.11 logPromptInput `void logPromptInput ( QString txt ) [slot]`

Parameters

txt	

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

txt	

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

prev	

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

fileName	

Returns

17.58.2.20 sendCloseMdiWin `void sendCloseMdiWin ( MdiWindow * ) [signal]`

17.58.2.21 setCurrentFile() `void setCurrentFile ( QString fileName )`

MdiWindow::setCurrentFile.

Parameters

fileName	

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

Parameters

color	

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

Parameters

color	

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

Parameters

color	

17.58.2.25 setViewRulerColor `void setViewRulerColor ( QRgb color ) [slot]`

Parameters

color	

17.58.2.26 setViewSelectBoxColors `void setViewSelectBoxColors (`

```
QRgb colorL,
QRgb fillL,
QRgb colorR,
QRgb fillR,
int alpha )  [slot]
```

Parameters

colorL	
fillL	
colorR	
fillR	
alpha	

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

Parameters

val	

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 ()
- QToolButton ∗ createToolButton (QString iconName, QString txt)
- QLineEdit ∗ createLineEdit (QString validatorType=QString(), bool readOnly=false)
- void updateLineEditStrIfVaries (QLineEdit ∗lineEdit, QString str)
- void updateLineEditNumIfVaries (QLineEdit ∗lineEdit, EmbReal num, bool useAnglePrecision)
- void updateFontComboBoxStrIfVaries (QFontComboBox ∗fontComboBox, QString str)
- void updateComboBoxStrIfVaries (QComboBox ∗comboBox, QString str, StringList strList)
- void updateComboBoxBoolIfVaries (QComboBox ∗comboBox, bool val, bool yesOrNoText)
- void mapSignal (QObject ∗fieldObj, QString name, QVariant value)

  PropertyEditor::mapSignal.
- QComboBox ∗ createComboBoxSelected ()
- QToolButton ∗ createToolButtonQSelect ()
- QToolButton ∗ createToolButtonPickAdd ()
- void createGroupBox (String group_box_key, const char ∗title)

Public Attributes

- QWidget ∗ focusWidget
- QString iconDir
- int iconSize
- Qt::ToolButtonStyle propertyEditorButtonStyle
- bool pickAdd
- std::vector< QGraphicsItem ∗ > selectedItemList
- int precisionAngle
- int precisionLength
- QSignalMapper ∗ signalMapper
- QComboBox ∗ comboBoxSelected
- QToolButton ∗ toolButtonQSelect
- QToolButton ∗ toolButtonPickAdd

Protected Member Functions

- bool eventFilter (QObject ∗obj, QEvent ∗event)

Private Slots

- void fieldEdited (QObject ∗fieldObj)
- void showGroups (int objType)
- void showOneType (int index)
- void hideAllGroups ()
- void clearAllFields ()
- void togglePickAddMode ()

17.61.1 Constructor & Destructor Documentation

17.61.1.1 PropertyEditor()

```
PropertyEditor (
        QString iconDirectory = QString(),
        bool pickAddMode = true,
        QWidget ∗ widgetToFocus = 0,
        QWidget ∗ parent = 0 )
```

17.61.1.2 ∼PropertyEditor()

```
∼PropertyEditor ( )
```

Todo document this

17.61.2 Member Function Documentation

17.61.2.1 clearAllFields

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

Todo DimAligned

DimAngular

DimArcLength

DimDiameter

DimLeader

DimLinear

DimOrdinate

DimRadius

17.61.2.2 createComboBoxSelected() `QComboBox * createComboBoxSelected ( )`

Todo document this

17.61.2.3 createGroupBox()

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

17.61.2.4 createLineEdit()

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

17.61.2.5 createToolButton()

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

17.61.2.6 createToolButtonPickAdd() `QToolButton * createToolButtonPickAdd ( )`

17.61.2.7 createToolButtonQSelect() `QToolButton * createToolButtonQSelect ( )`

Todo document this

17.61.2.8 eventFilter()

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

Todo document this

17.61.2.9 fieldEdited

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

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

Note

General group will never be hidden.

17.61.2.11 mapSignal()

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

PropertyEditor::mapSignal.

Parameters

fieldObj	
name	
value	

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

theScene	
parent	

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

pattern	
item	

17.62.2.2 addBlock()

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

SaveObject::addBlock.

Parameters

pattern	
item	

17.62.2.3 addCircle()

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

SaveObject::addCircle.

Parameters

pattern	
item	

17.62.2.4 addDimAligned()

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

SaveObject::addDimAligned.

Parameters

pattern	
item	

17.62.2.5 addDimAngular()

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

SaveObject::addDimAngular.

Parameters

pattern	
item	

17.62.2.6 addDimArcLength()

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

SaveObject::addDimArcLength.

Parameters

pattern	
item	

17.62.2.7 addDimDiameter()

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

SaveObject::addDimDiameter.

Parameters

pattern	
item	

17.62.2.8 addDimLeader()

```
void addDimLeader (
            EmbPattern * pattern,
            QGraphicsItem * item )
```

SaveObject::addDimLeader.

Parameters

pattern	
item	

17.62.2.9 addDimLinear()

```
void addDimLinear (
            EmbPattern * pattern,
            QGraphicsItem * item )
```

SaveObject::addDimLinear.

Parameters

pattern	
item	

17.62.2.10 addDimOrdinate()

```
void addDimOrdinate (
            EmbPattern * pattern,
            QGraphicsItem * item )
```

SaveObject::addDimOrdinate.

Parameters

pattern	
item	

17.62.2.11 addDimRadius()

```
void addDimRadius (
            EmbPattern * pattern,
            QGraphicsItem * item )
```

SaveObject::addDimRadius.

Parameters

pattern	
item	

17.62.2.12 addEllipse()

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

SaveObject::addEllipse.

Parameters

pattern	
item	

17.62.2.13 addEllipseArc()

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

SaveObject::addEllipseArc.

Parameters

pattern	
item	

17.62.2.14 addGrid()

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

SaveObject::addGrid.

Parameters

pattern	
item	

17.62.2.15 addHatch()

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

SaveObject::addHatch.

Parameters

pattern	
item	

17.62.2.16 addImage()

```
void addImage (
        EmbPattern * pattern,
        QGraphicsItem * item )
```

SaveObject::addImage.

Parameters

pattern	
item	

17.62.2.17 addInfiniteLine()

```
void addInfiniteLine (
        EmbPattern * pattern,
        QGraphicsItem * item )
```

SaveObject::addInfiniteLine.

Parameters

pattern	
item	

17.62.2.18 addLine()

```
void addLine (
        EmbPattern * pattern,
        QGraphicsItem * item )
```

SaveObject::addLine.

Parameters

pattern	
item	

17.62.2.19 addPath()

```
void addPath (
EmbPattern * pattern,
QGraphicsItem * item )
```

SaveObject::addPath.

Parameters

pattern	
item	

Todo Reimplement addPolyline() using the libembroidery C API

17.62.2.20 addPoint()

```
void addPoint (
    EmbPattern * pattern,
    QGraphicsItem * item )
```

SaveObject::addPoint.

Parameters

pattern	
item	

17.62.2.21 addPolygon()

```
void addPolygon (
    EmbPattern * pattern,
    QGraphicsItem * item )
```

SaveObject::addPolygon.

Parameters

pattern	
item	

17.62.2.22 addPolyline()

```
void addPolyline (
    EmbPattern * pattern,
    QGraphicsItem * item )
```

SaveObject::addPolyline.

Parameters

pattern	
item	

17.62.2.23 addRay()

```
void addRay (
        EmbPattern * pattern,
        QGraphicsItem * item )
```

SaveObject::addRay.

Parameters

pattern	
item	

17.62.2.24 addRectangle()

```
void addRectangle (
        EmbPattern * pattern,
        QGraphicsItem * item )
```

SaveObject::addRectangle.

Parameters

pattern	
item	

17.62.2.25 addSlot()

```
void addSlot (
        EmbPattern * pattern,
        QGraphicsItem * item )
```

SaveObject::addSlot.

Parameters

pattern	
item	

17.62.2.26 addSpline()

```
void addSpline (
        EmbPattern * pattern,
        QGraphicsItem * item )
```

SaveObject::addSpline.

Parameters

pattern	
item	

17.62.2.27 addTextMulti()

```
void addTextMulti (
    EmbPattern * pattern,
    QGraphicsItem * item )
```

SaveObject::addTextMulti.

Parameters

pattern	
item	

17.62.2.28 addTextSingle()

```
void addTextSingle (
    EmbPattern * pattern,
    QGraphicsItem * item )
```

SaveObject::addTextSingle.

Parameters

pattern	
item	

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

pattern	
objPos	
objPath	
layer	
color	
lineType	
lineWeight	

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.

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

checked	

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

type	

17.64.2.37 comboBoxIconSizeCurrentIndexChanged `void comboBoxIconSizeCurrentIndexChanged ( int index ) [private], [slot]`

Settings_Dialog::comboBoxIconSizeCurrentIndexChanged.

Parameters

index	

17.64.2.38 comboBoxPromptFontFamilyCurrentIndexChanged

```
void comboBoxPromptFontFamily↩
CurrentIndexChanged (
            QString family ) [private], [slot]
```

17.64.2.39 comboBoxPromptFontStyleCurrentIndexChanged

```
void comboBoxPromptFontStyleCurrent↩
IndexChanged (
            QString style ) [private], [slot]
```

17.64.2.40 comboBoxQSnapLocatorColorCurrentIndexChanged

```
void comboBoxQSnapLocatorColor↩
CurrentIndexChanged (
            int index ) [private], [slot]
```

17.64.2.41 comboBoxRulerMetricCurrentIndexChanged

```
void comboBoxRulerMetricCurrentIndex↩
Changed (
            int index ) [private], [slot]
```

17.64.2.42 comboBoxScrollBarWidgetCurrentIndexChanged

```
void comboBoxScrollBarWidgetCurrent↩
IndexChanged (
            int index ) [private], [slot]
```

17.64.2.43 comboBoxSelectionCoolGripColorCurrentIndexChanged

```
void comboBoxSelectionCoolGrip↩
ColorCurrentIndexChanged (
            int index ) [private], [slot]
```

17.64.2.44 comboBoxSelectionHotGripColorCurrentIndexChanged

```
void comboBoxSelectionHotGrip↩
ColorCurrentIndexChanged (
            int index ) [private], [slot]
```

17.64.2.45 create_checkbox()

```
QCheckBox * create_checkbox (
            QGroupBox * groupbox,
            String label )
```

17.64.2.46 create_float_spinbox()

```
void create_float_spinbox (
            QGroupBox * gb,
            QGridLayout * gridLayout,
            const char * label_in,
            EmbReal single_step,
            EmbReal lower,
            EmbReal upper,
            String key,
            int row )
```

17.64.2.47 createTabDisplay()

```
QWidget * createTabDisplay ( )
```

17.64.2.48 createTabFilesPaths() `QWidget * createTabFilesPaths ( )`

17.64.2.49 createTabGeneral() `QWidget * createTabGeneral ( )`

17.64.2.50 createTabGridRuler() `QWidget * createTabGridRuler ( )`

17.64.2.51 createTabLineWeight() `QWidget * createTabLineWeight ( )`

17.64.2.52 createTabOpenSave() `QWidget * createTabOpenSave ( )`

17.64.2.53 createTabOrthoPolar() `QWidget * createTabOrthoPolar ( )`

17.64.2.54 createTabPrinting() `QWidget * createTabPrinting ( )`

17.64.2.55 createTabPrompt() `QWidget * createTabPrompt ( )`

17.64.2.56 createTabQuickSnap() `QWidget * createTabQuickSnap ( )`

17.64.2.57 createTabQuickTrack() `QWidget * createTabQuickTrack ( )`

17.64.2.58 createTabSelection() `QWidget * createTabSelection ( )`

17.64.2.59 createTabSnap() `QWidget * createTabSnap ( )`

17.64.2.60 currentDisplayBackgroundColorChanged

```
void currentDisplayBackgroundColorChanged (
        const QColor & color ) [private], [slot]
```

17.64.2.61 currentDisplayCrossHairColorChanged

```
void currentDisplayCrossHairColorChanged (
        const QColor & color ) [private], [slot]
```

17.64.2.62 currentDisplaySelectBoxLeftColorChanged

```
void currentDisplaySelectBoxLeftColor↩
Changed (
        const QColor & color ) [private], [slot]
```

17.64.2.63 currentDisplaySelectBoxLeftFillChanged

```
void currentDisplaySelectBoxLeftFillChanged (
        const QColor & color ) [private], [slot]
```

17.64.2.64 currentDisplaySelectBoxRightColorChanged `void currentDisplaySelectBoxRightColor↵ Changed (`

`const QColor & color ) [private], [slot]`

17.64.2.65 currentDisplaySelectBoxRightFillChanged `void currentDisplaySelectBoxRightFillChanged (`

`const QColor & color ) [private], [slot]`

17.64.2.66 currentGeneralMdiBackgroundColorChanged `void currentGeneralMdiBackgroundColor↵ Changed (`

`const QColor & color ) [private], [slot]`

17.64.2.67 currentGridColorChanged `void currentGridColorChanged (`

`const QColor & color ) [private], [slot]`

17.64.2.68 currentPromptBackgroundColorChanged `void currentPromptBackgroundColorChanged (`

`const QColor & color ) [private], [slot]`

17.64.2.69 currentPromptTextColorChanged `void currentPromptTextColorChanged (`

`const QColor & color ) [private], [slot]`

17.64.2.70 currentRulerColorChanged `void currentRulerColorChanged (`

`const QColor & color ) [private], [slot]`

17.64.2.71 rejectChanges `void rejectChanges ( ) [private], [slot]`

Settings_Dialog::rejectChanges.

17.64.2.72 spinBoxDisplaySelectBoxAlphaValueChanged `void spinBoxDisplaySelectBoxAlphaValue↵ Changed (`

`int value ) [private], [slot]`

17.64.2.73 spinBoxPromptFontSizeValueChanged `void spinBoxPromptFontSizeValueChanged (`

`int value ) [private], [slot]`

17.64.2.74 spinBoxRulerPixelSizeValueChanged `void spinBoxRulerPixelSizeValueChanged (`

`double value ) [private], [slot]`

17.64.3 Member Data Documentation

17.64.3.1 buttonBox `QDialogButtonBox* buttonBox`

17.64.3.2 tabWidget `QTabWidget∗ tabWidget`

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

- embroidermodder2/embroidermodder.h
- embroidermodder2/settings-dialog.cpp

17.65 StatusBar Class Reference

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

Public Member Functions

- StatusBar (QWidget ∗parent=0)
- void setMouseCoord (EmbReal x, EmbReal y)
- void context_menu_action (QToolButton ∗button, const char ∗icon, const char ∗label, QMenu ∗menu, String setting_page)
- void toggle (String key, bool on)
- void context_menu_event (QContextMenuEvent ∗event, QToolButton ∗button)

Public Attributes

- std::unordered_map< String, QToolButton ∗ > buttons
- QLabel ∗ statusBarMouseCoord

17.65.1 Detailed Description

17.65.2 Constructor & Destructor Documentation

17.65.2.1 StatusBar()

```
StatusBar (
    QWidget ∗ parent = 0 )
```

17.65.3 Member Function Documentation

17.65.3.1 context_menu_action()

```
void context_menu_action (
    QToolButton ∗ button,
    const char ∗ icon,
    const char ∗ label,
    QMenu ∗ menu,
    String setting_page )
```

17.65.3.2 context_menu_event()

```
void context_menu_event (
    QContextMenuEvent ∗ event,
    QToolButton ∗ button )
```

17.65.3.3 setMouseCoord()

```
void setMouseCoord (
    EmbReal x,
    EmbReal y )
```

17.65.3.4 toggle()

```
void toggle (
    String key,
    bool on )
```

17.65.4 Member Data Documentation

17.65.4.1 buttons `std::unordered_map<String, QToolButton*> buttons`

17.65.4.2 statusBarMouseCoord `QLabel* statusBarMouseCoord`

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

- embroidermodder2/embroidermodder.h
- embroidermodder2/statusbar.cpp

17.66 StxThread_ Struct Reference

`#include <embroidery_internal.h>`

Public Attributes

- char ∗ colorCode
- char ∗ colorName
- char ∗ sectionName
- SubDescriptor ∗ subDescriptors
- EmbColor stxColor

17.66.1 Member Data Documentation

17.66.1.1 colorCode `char* colorCode`

17.66.1.2 colorName `char* colorName`

17.66.1.3 sectionName `char* sectionName`

17.66.1.4 stxColor `EmbColor stxColor`

17.66.1.5 subDescriptors `SubDescriptor* subDescriptors`

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

- extern/libembroidery/src/embroidery_internal.h

17.67 SubDescriptor_ Struct Reference

`#include <embroidery_internal.h>`

Public Attributes

- int someNum
- int someInt
- int someOtherInt
- char ∗ colorCode
- char ∗ colorName

17.67.1 Member Data Documentation

17.67.1.1 colorCode `char* colorCode`

Todo better variable naming

17.67.1.2 colorName `char* colorName`

17.67.1.3 someInt `int someInt`

Todo better variable naming

17.67.1.4 someNum `int someNum`

17.67.1.5 someOtherInt `int someOtherInt`

Todo better variable naming

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

- extern/libembroidery/src/embroidery_internal.h

17.68 SvgAttribute_ Struct Reference

`#include <embroidery_internal.h>`

Public Attributes

- char * name
- char * value

17.68.1 Member Data Documentation

17.68.1.1 name `char* name`

17.68.1.2 value `char* value`

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

- extern/libembroidery/src/embroidery_internal.h

17.69 thread_color_ Struct Reference

`#include <embroidery.h>`

Public Attributes

- char name [22]
- unsigned int hex_code
- int manufacturer_code

17.69.1 Member Data Documentation

17.69.1.1 hex_code `unsigned int hex_code`

17.69.1.2 manufacturer_code `int manufacturer_code`

17.69.1.3 name `char name[22]`

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

- extern/libembroidery/src/embroidery.h

17.70 ThredExtension_ Struct Reference

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

Public Attributes

- float hoopX
- float hoopY
- float stitchGranularity
- char creatorName [50]
- char modifierName [50]
- char auxFormat
- char reserved [31]

17.70.1 Member Data Documentation

17.70.1.1 auxFormat `char auxFormat`

17.70.1.2 creatorName `char creatorName[50]`

17.70.1.3 hoopX `float hoopX`

17.70.1.4 hoopY `float hoopY`

17.70.1.5 modifierName `char modifierName[50]`

17.70.1.6 reserved `char reserved[31]`

17.70.1.7 stitchGranularity `float stitchGranularity`

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

- extern/libembroidery/src/embroidery_internal.h

17.71 ThredHeader_ Struct Reference

`#include <embroidery_internal.h>`

Public Attributes

- unsigned int sigVersion
- unsigned int length
- unsigned short numStiches
- unsigned short hoopSize
- unsigned short reserved [7]

17.71.1 Member Data Documentation

17.71.1.1 hoopSize `unsigned short hoopSize`

17.71.1.2 length `unsigned int length`

17.71.1.3 numStiches `unsigned short numStiches`

17.71.1.4 reserved `unsigned short reserved[7]`

17.71.1.5 sigVersion `unsigned int sigVersion`

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

- extern/libembroidery/src/embroidery_internal.h

17.72 UndoableCommand Class Reference

`#include <embroidermodder.h>`

Public Member Functions

- UndoableCommand (String command, QString text, Geometry ∗obj, View ∗v, QUndoCommand ∗parent=0)
- UndoableCommand (EmbVector d, QString text, Geometry ∗obj, View ∗v, QUndoCommand ∗parent=0)
- UndoableCommand (String command, EmbVector pivot, EmbReal angle, QString text, Geometry ∗obj, View ∗v, QUndoCommand ∗parent=0)
- UndoableCommand (QString type, View ∗v, QUndoCommand ∗parent=0)
- UndoableCommand (const QPointF beforePoint, const QPointF afterPoint, QString text, Geometry ∗obj, View ∗v, QUndoCommand ∗parent=0)
- UndoableCommand (EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, QString text, Geometry ∗obj, View ∗v, QUndoCommand ∗parent=0)
- int id ()
- bool mergeWith (const QUndoCommand ∗command)
- void undo ()
- void redo ()
- void mirror ()
- void rotate (EmbVector pivot, EmbReal rot)

Public Attributes

- Geometry ∗ object
- View ∗ gview
- String command
- EmbVector delta
- EmbVector pivot
- QPointF before
- QPointF after
- EmbReal angle
- EmbReal factor
- QString navType
- QTransform fromTransform
- QTransform toTransform
- QPointF fromCenter
- QPointF toCenter
- QLineF mirrorLine
- bool done

17.72.1 Constructor & Destructor Documentation

17.72.1.1 UndoableCommand() [1/6]

```
UndoableCommand (
        String command,
        QString text,
        Geometry * obj,
        View * v,
        QUndoCommand * parent = 0 )
```

17.72.1.2 UndoableCommand() [2/6]

```
UndoableCommand (
        EmbVector d,
        QString text,
        Geometry * obj,
        View * v,
        QUndoCommand * parent = 0 )
```

17.72.1.3 UndoableCommand() [3/6]

```
UndoableCommand (
        String command,
        EmbVector pivot,
        EmbReal angle,
        QString text,
        Geometry * obj,
        View * v,
        QUndoCommand * parent = 0 )
```

17.72.1.4 UndoableCommand() [4/6]

```
UndoableCommand (
        QString type,
        View * v,
        QUndoCommand * parent = 0 )
```

17.72.1.5 UndoableCommand() [5/6]

```
UndoableCommand (
    const QPointF beforePoint,
    const QPointF afterPoint,
    QString text,
    Geometry * obj,
    View * v,
    QUndoCommand * parent = 0 )
```

17.72.1.6 UndoableCommand() [6/6]

```
UndoableCommand (
    EmbReal x1,
    EmbReal y1,
    EmbReal x2,
    EmbReal y2,
    QString text,
    Geometry * obj,
    View * v,
    QUndoCommand * parent = 0 )
```

17.72.2 Member Function Documentation

17.72.2.1 id()

```
int id ( )  [inline]
```

17.72.2.2 mergeWith()

```
bool mergeWith (
    const QUndoCommand * command )
```

17.72.2.3 mirror()

```
void mirror ( )
```

17.72.2.4 redo()

```
void redo ( )
```

17.72.2.5 rotate()

```
void rotate (
    EmbVector pivot,
    EmbReal rot )
```

17.72.2.6 undo()

```
void undo ( )
```

17.72.3 Member Data Documentation

17.72.3.1 after

```
QPointF after
```

17.72.3.2 angle

```
EmbReal angle
```

17.72.3.3 before

```
QPointF before
```

17.72.3.4 command `String command`

17.72.3.5 delta `EmbVector delta`

17.72.3.6 done `bool done`

17.72.3.7 factor `EmbReal factor`

17.72.3.8 fromCenter `QPointF fromCenter`

17.72.3.9 fromTransform `QTransform fromTransform`

17.72.3.10 gview `View* gview`

17.72.3.11 mirrorLine `QLineF mirrorLine`

17.72.3.12 navType `QString navType`

17.72.3.13 object `Geometry* object`

17.72.3.14 pivot `EmbVector pivot`

17.72.3.15 toCenter `QPointF toCenter`

17.72.3.16 toTransform `QTransform toTransform`

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

- embroidermodder2/embroidermodder.h
- embroidermodder2/undo-commands.cpp

17.73 UndoEditor Class Reference

`#include <embroidermodder.h>`

Public Slots

- void undo ()
- void redo ()
- void updateCleanIcon (bool opened)

Public Member Functions

- UndoEditor (QString iconDirectory=QString(), QWidget ∗widgetToFocus=0, QWidget ∗parent=0)
- ∼UndoEditor ()
- void addStack (QUndoStack ∗stack)
- bool canUndo ()
- bool canRedo ()
- QString undoText ()
- QString redoText ()

Public Attributes

- QWidget ∗ focusWidget
- QString iconDir
- int iconSize
- QUndoGroup ∗ undoGroup
- QUndoView ∗ undoView

17.73.1 Constructor & Destructor Documentation

17.73.1.1 UndoEditor()

```
UndoEditor (
        QString iconDirectory = QString(),
        QWidget * widgetToFocus = 0,
        QWidget * parent = 0 )
```

17.73.1.2 ∼UndoEditor()

```
~UndoEditor ( )
```

17.73.2 Member Function Documentation

17.73.2.1 addStack()

```
void addStack (
        QUndoStack * stack )
```

17.73.2.2 canRedo()

```
bool canRedo ( )
```

17.73.2.3 canUndo()

```
bool canUndo ( )
```

17.73.2.4 redo

```
void redo ( ) [slot]
```

17.73.2.5 redoText()

```
QString redoText ( )
```

17.73.2.6 undo

```
void undo ( ) [slot]
```

17.73.2.7 undoText()

```
QString undoText ( )
```

17.73.2.8 updateCleanIcon `void updateCleanIcon ( bool opened ) [slot]`

17.73.3 Member Data Documentation

17.73.3.1 focusWidget `QWidget* focusWidget`

17.73.3.2 iconDir `QString iconDir`

17.73.3.3 iconSize `int iconSize`

17.73.3.4 undoGroup `QUndoGroup* undoGroup`

17.73.3.5 undoView `QUndoView* undoView`

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

- embroidermodder2/embroidermodder.h
- embroidermodder2/undo-editor.cpp

17.74 View Class Reference

`#include <embroidermodder.h>`

Public Slots

- void zoomIn ()
- void zoomOut ()
- void zoomWindow ()
- void zoomSelected ()
- void zoomExtents ()
- void panRealTime ()
- void panPoint ()
- void panLeft ()
- void panRight ()
- void panUp ()
- void panDown ()
- void selectAll ()
- void selectionChanged ()
- void clearSelection ()
- void deleteSelected ()
- void moveSelected (EmbReal dx, EmbReal dy)
- void cut ()
- void copy ()
- void paste ()
- void repeatAction ()
- void moveAction ()
- void scaleAction ()
- void scaleSelected (EmbReal x, EmbReal y, EmbReal factor)
- void rotateAction ()
- void rotateSelected (EmbReal x, EmbReal y, EmbReal rot)

- void mirrorSelected (EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2)
- int numSelected ()
- void deletePressed ()
- void escapePressed ()
- void cornerButtonClicked ()
- void showScrollBars (bool val)
- void setCornerButton ()
- void setCrossHairColor (QRgb color)
- void setCrossHairSize (uint8_t percent)
- void 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 multple 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

18.5.1.1 main()

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

qMain

Parameters

argc	
argv	

Returns

18.5.2 Variable Documentation

18.5.2.1 _appVer_

```
const char* _appVer_ = "v2.0.0-alpha3" [static]
```

18.5.2.2 exitApp

```
bool exitApp = false [static]
```

18.5.2.3 usage_msg

```
const char* usage_msg
```

Initial value:

```
= ""
    " ___ _____ ___  ___  __  _ ___  ___ ___   _____  __  ___  ___  ___ ___   ___ \n"
    "| __|      | _ \\| _ \\ /  \\| |   \\| __| _ \\ |      |/  \\|   \\|   \\| __| _ \\  |__ \\\n"
    "| __| | | | _ <|   /| () | | |) | __|   / | | | | () | |) | |) | __|   /  / __/\n"
    "|___|_|_|_|___/|_|\\_\\\\__/|_|___/|___|_|\\_\\|_|_|_|\\__/|___/|___/|___|_|\\_\\ |___|\n"
    " _____________________________________________________________________________ \n"
    "|                                                                             |\n"
    "|                     https://www.libembroidery.org                           |\n"
    "|_____________________________________________________________________________|\n"
    "\n"
    "Usage:  embroidermodder [options] files ...\n"
    "\n"
    "Options:\n"
    "  -d, --debug      Print lots of debugging information.\n"
    "  -h, --help       Print this message and exit.\n"
    "  -v, --version    Print the version number of embroidermodder and exit.\n"
    "\n"
```

18.6 embroidermodder2/embroidermodder.h File Reference

```
#include <cstdio>
#include <cmath>
#include <ctime>
#include <cinttypes>
#include <cstdarg>
#include <vector>
#include <unordered_map>
#include <string>
#include <filesystem>
#include "embroidery.h"
#include "toml.h"
#include <QAction>
#include <QApplication>
#include <QtPrintSupport>
```

Classes

- struct Node_
- class Geometry

    The Geometry class.
- class SaveObject
- class Application
- class CmdPromptInput
- class CmdPromptHistory

    The Command Prompt History class.
- class CmdPromptSplitter
- class CmdPromptHandle
- class CmdPrompt
- class EmbDetailsDialog
- class ImageWidget
- class LayerManager
- class MainWindow

    The MainWindow class.
- class MdiWindow
- class MdiArea
- class PreviewDialog
- class PropertyEditor
- class SelectBox
- class Settings_Dialog
- class StatusBar
- class UndoEditor
- class UndoableCommand
- class View

Macros

- #define STRING_TYPE 0
- #define STRING_LIST_TYPE 1
- #define REAL_TYPE 2
- #define INT_TYPE 3
- #define BOOL_TYPE 4
- #define FUNCTION_TYPE 5
- #define VECTOR_TYPE 6
- #define UNKNOWN_TYPE 7

Typedefs

- typedef std::string String
- typedef std::vector< String > StringList
- typedef struct Node_ Node
- typedef String(∗ Command) (String)
- typedef std::vector< Node > NodeList
- typedef std::unordered_map< String, Node > Dictionary

Enumerations

- enum OBJ_TYPE_VALUES {
  OBJ_TYPE_NULL = 0 , OBJ_TYPE_BASE = 100000 , OBJ_TYPE_ARC = 100001 , OBJ_TYPE_BLOCK = 100002 ,
  OBJ_TYPE_CIRCLE = 100003 , OBJ_TYPE_DIMALIGNED = 100004 , OBJ_TYPE_DIMANGULAR = 100005 , OBJ_TYPE_DIMARCLENGTH = 100006 ,
  OBJ_TYPE_DIMDIAMETER = 100007 , OBJ_TYPE_DIMLEADER = 100008 , OBJ_TYPE_DIMLINEAR = 100009 , OBJ_TYPE_DIMORDINATE = 100010 ,
  OBJ_TYPE_DIMRADIUS = 100011 , OBJ_TYPE_ELLIPSE = 100012 , OBJ_TYPE_ELLIPSEARC = 100013 , OBJ_TYPE_RUBBER = 100014 ,
  OBJ_TYPE_GRID = 100015 , OBJ_TYPE_HATCH = 100016 , OBJ_TYPE_IMAGE = 100017 , OBJ_TYPE_INFINITELINE = 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 **Todo** Use color chart in formats/format-dxf.h for this
OBJ_LWT	
OBJ_RUBBER	

18.6.4.2 OBJ_TYPE_VALUES `enum OBJ_TYPE_VALUES`

Enumerator

OBJ_TYPE_NULL	
OBJ_TYPE_BASE	

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

command	

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.toUpper() + "')", fileName); }

18.6.5.7 add_polyline()

```
void add_polyline (
        QPainterPath p,
        String rubberMode )
```

add_polyline

Parameters

p	
rubberMode	

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

command	
fmt	

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

menu	
def	
topLevel	

18.6.5.13 debug_message()

```
void debug_message (
            std::string msg )
```

debug_message

Parameters

msg	

18.6.5.14 degrees__()

```
EmbReal degrees__ (
            EmbReal radian )
```

degrees__

Parameters

radian	

Returns

18.6.5.15 fileExtension()

```
QString fileExtension (
            String fileName )
```

MdiWindow::fileExtension.

Parameters

fileName	

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

node	
value	

18.6.5.31 node_str_list()

```
Node node_str_list (
        StringList value )
```

set_node

Parameters

node	
value	

18.6.5.32 node_uint()

```
Node node_uint (
        uint32_t value )
```

create_node

Parameters

mode	

Returns

18.6.5.33 operator*()

```
EmbVector operator* (
        EmbVector v,
        EmbReal s )
```

operator *

Parameters

v	
s	

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

degrees	

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

fname	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

parent	
key	
enabled	

Todo error reporting.

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

set_visibility

Parameters

parent	
key	
visibility	

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

list	

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

list	

Returns

18.6.5.49 to_vector()

```
std::vector< QGraphicsItem * > to_vector (
        QList< QGraphicsItem * > list )
```

to_vector

Parameters

list	

Returns

18.6.5.50 tokenize()

```
StringList tokenize (
        String str,
        const char delim )
```

tokenize

Parameters

str	
delim	

Returns

18.6.5.51 translate_str()

```
QString translate_str (
        const char * str )
```

18.6.5.52 validFileFormat() `bool validFileFormat ( String fileName )`

MainWindow::validFileFormat.

Parameters

fileName	

Returns

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

18.6.5.53 write_settings() `void write_settings ( void )`

MainWindow::writeSettings.

This file needs to be read from the users home directory to ensure it is writable

18.6.6 Variable Documentation

18.6.6.1 _mainWin `MainWindow* _mainWin [extern]`

18.6.6.2 actionHash `std::unordered_map<String, QAction*> actionHash [extern]`

18.6.6.3 checkBoxes `std::unordered_map<String, QCheckBox *> checkBoxes [extern]`

18.6.6.4 comboBoxes `std::unordered_map<String, QComboBox *> comboBoxes [extern]`

18.6.6.5 config `Dictionary config`

18.6.6.6 config_tables `std::unordered_map<String, Dictionary> config_tables [extern]`

18.6.6.7 dialog `Dictionary dialog`

18.6.6.8 dockPropEdit `PropertyEditor* dockPropEdit [extern]`

18.6.6.9 dockUndoEdit `UndoEditor* dockUndoEdit [extern]`

18.6.6.10 doubleSpinBoxes `std::unordered_map<String, QDoubleSpinBox *> doubleSpinBoxes [extern]`

18.6.6.11 emb_constant_pi

`const EmbReal emb_constant_pi = 3.14159265358979323846 [static]`

18.6.6.12 groupBoxes

`std::unordered_map<String, QGroupBox *> groupBoxes [extern]`

18.6.6.13 labels

`std::unordered_map<String, QLabel *> labels [extern]`

18.6.6.14 lineEdits

`std::unordered_map<String, QLineEdit *> lineEdits [extern]`

18.6.6.15 mdiArea

`MdiArea* mdiArea [extern]`

18.6.6.16 menuHash

`std::unordered_map<String, QMenu*> menuHash [extern]`

18.6.6.17 prompt

`CmdPrompt* prompt [extern]`

18.6.6.18 scripts

`std::unordered_map<String, 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.

```
/*
*  Embroidermodder 2.
*
*  ------------------------------------------------------------
*
*  Copyright 2013-2023 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/
*/

#ifndef __EMBROIDERMODDER_UTILITY_H__
#define __EMBROIDERMODDER_UTILITY_H__

/*
* C/C++ Standard Libraries.
*/
#include <cstdio>
#include <cmath>
#include <ctime>
#include <cinttypes>
#include <cstdarg>
#include <vector>
#include <unordered_map>
#include <string>
#include <filesystem>

/*
* Libraries included in "extern/".
*/
#include "embroidery.h"
#include "toml.h"

/*
* Qt 6.0+ libraries.
*/
#include <QAction>
#include <QApplication>

#include <QtPrintSupport>

#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

class ImageWidget;
class MdiArea;
class MdiWindow;
class View;
class StatusBar;
class CmdPrompt;
class PropertyEditor;
class UndoEditor;
class MainWindow;
class Geometry;

typedef std::string String;
typedef std::vector<String> StringList;

typedef struct Node_ {
    String s;
    EmbReal r;
    int32_t i;
    bool b;
    StringList sl;
    int type;
} Node;

typedef String (*Command)(String);
typedef std::vector<Node> NodeList;
typedef std::unordered_map<String, Node> Dictionary;

//Values
enum OBJ_TYPE_VALUES {
    OBJ_TYPE_NULL =      0,
    /*< NOTE: Allow this enum to evaluate false */
    OBJ_TYPE_BASE = 100000,
    /*< NOTE: Values >= 65536 ensure compatibility with qgraphicsitem_cast() */
    OBJ_TYPE_ARC = 100001,
    OBJ_TYPE_BLOCK = 100002,
    /*< For the block type, that has to exist for SVG. */
    OBJ_TYPE_CIRCLE = 100003,
    OBJ_TYPE_DIMALIGNED = 100004,
    /*< For the Aligned Dimension, that has to exist for DXF drawings.  */
    OBJ_TYPE_DIMANGULAR = 100005,
    /*< For the Angular Dimension, that has to exist for DXF drawings.  */
    OBJ_TYPE_DIMARCLENGTH = 100006,
    /*< For the Arc Length Dimension, that has to exist for DXF drawings.  */
    OBJ_TYPE_DIMDIAMETER = 100007,
    OBJ_TYPE_DIMLEADER = 100008,
    OBJ_TYPE_DIMLINEAR = 100009,
    /*< For the Linear Dimension, that has to exist for DXF drawings.  */
    OBJ_TYPE_DIMORDINATE = 100010,
    /*< For the Ordinate Dimension, that has to exist for DXF drawings.  */
    OBJ_TYPE_DIMRADIUS = 100011,
    /*< For the Radial Dimension, that has to exist for DXF drawings.  */
    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_INFINITELINE = 100018,
    /*< For the Infinite Line object.  Which should be removed from output as it exists
for drafting reasons.  */
    OBJ_TYPE_LINE = 100019,
    OBJ_TYPE_PATH = 100020,
    OBJ_TYPE_POINT = 100021,
    OBJ_TYPE_POLYGON = 100022,
    OBJ_TYPE_POLYLINE = 100023,
    OBJ_TYPE_RAY = 100024,
    /*< For the Ray object.  */
    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,
    /*< value type - int:  See OBJ_TYPE_VALUES */
    OBJ_NAME = 1,
    /*< value type - str:  See OBJ_NAME_VALUES */
    OBJ_LAYER = 2,
    /*< value type - str:  "USER", "DEFINED", "STRINGS", etc...  */
    OBJ_COLOR = 3,
    OBJ_LTYPE = 4,
    /*< value type - int:  See OBJ_LTYPE_VALUES */
    OBJ_LWT = 5, //value type - int:  0-27
    OBJ_RUBBER = 6  //value type - int:  See OBJ_RUBBER_VALUES
};

static const EmbReal emb_constant_pi = 3.14159265358979323846;

/* Global variables
* ----------------
*/
extern MdiArea* mdiArea;

extern Dictionary settings, dialog, config;
extern std::unordered_map<String, StringList> scripts;
extern std::unordered_map<String, QGroupBox *> groupBoxes;
extern std::unordered_map<String, QCheckBox *> checkBoxes;
extern std::unordered_map<String, QSpinBox *> spinBoxes;
extern std::unordered_map<String, QDoubleSpinBox *> doubleSpinBoxes;
extern std::unordered_map<String, QLabel *> labels;
extern std::unordered_map<String, QComboBox *> comboBoxes;
extern std::unordered_map<String, QLineEdit *> lineEdits;
extern std::unordered_map<String, QToolButton *> toolButtons;
extern std::unordered_map<String, Dictionary> config_tables;
extern std::unordered_map<String, QAction*> actionHash;
extern std::unordered_map<String, QToolBar*> toolbarHash;
extern std::unordered_map<String, QMenu*> menuHash;
extern std::unordered_map<String, QMenu*> subMenuHash;

extern MainWindow* _mainWin;
extern CmdPrompt* prompt;
extern PropertyEditor* dockPropEdit;
extern UndoEditor* dockUndoEdit;
extern StatusBar* statusbar;

/* Functions in the global namespace
* ---------------------------------
*/
int read_configuration(const char *file);
void read_settings(void);
void write_settings(void);
EmbVector rotate_vector(EmbVector v, EmbReal alpha);

QString translate_str(const char *str);
bool contains(StringList, String);
bool validFileFormat(String fileName);
QString fileExtension(String fileName);

void add_polyline(QPainterPath p, String rubberMode);
```

```

String read_string_setting(toml_table_t *table, const char *key);
StringList tokenize(String str, const char delim);
String convert_args_to_type(String label, StringList args,
    const char *args_template, NodeList a);

View *activeView(void);
QGraphicsScene* activeScene();

void debug_message(String msg);
void set_enabled(QObject *parent, const char *key, bool enabled);
void set_visibility(QObject *parent, const char *name, bool visibility);
QPainterPath add_to_path(QPainterPath path, EmbVector scale, String s);

String actuator(String line);
String run_script_file(String fname);
String run_script(StringList script);
String construct_command(String command, const char *fmt, ...);

void create_menu(String menu, StringList def, bool topLevel);

QPointF to_QPointF(EmbVector a);
EmbVector to_EmbVector(QPointF a);
EmbVector operator+(EmbVector a, EmbVector b);
EmbVector operator-(EmbVector a, EmbVector b);
EmbVector operator*(EmbVector v, EmbReal s);
EmbReal radians__(EmbReal degrees);
EmbReal degrees__(EmbReal radian);

std::vector<QGraphicsItem*> to_vector(QList<QGraphicsItem*> list);
QList<QGraphicsItem*> to_qlist(std::vector<QGraphicsItem*> list);

StringList to_string_vector(QStringList list);

/* Interface creation functions.
*/
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);

/* Dictionary management functions.
*/
Node node_bool(bool value);
Node node_int(int32_t value);
Node node_uint(uint32_t value);
Node node_real(EmbReal value);
Node node_str(String value);
Node node_qstr(QString value);
Node node_str_list(StringList value);

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);

class Geometry :  public QGraphicsPathItem
{
public:
    enum ArrowStyle {
        NoArrow, //NOTE: Allow this enum to evaluate false
        Open,
        Closed,
        Dot,
        Box,
        Tick
    };

    enum lineStyle {
        NoLine, //NOTE: Allow this enum to evaluate false
        Flared,
        Fletching
    };

    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;
    virtual int type(){ return Type; }

    Geometry(int object_type = OBJ_TYPE_BASE, QGraphicsItem* parent = 0);
    Geometry(Geometry *obj, QGraphicsItem* parent = 0);
    Geometry(EmbArc arc, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
    Geometry(EmbCircle circle, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
    Geometry(EmbLine line, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
    Geometry(EmbEllipse ellipse, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
    Geometry(EmbRect rect, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
    Geometry(QString str, EmbVector position, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent =
  0);
    Geometry(EmbLine line, int Type_, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent);
    Geometry(QPainterPath p, int type_, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);
    Geometry(EmbVector pos, QRgb rgb, Qt::PenStyle lineType, QGraphicsItem* parent = 0);

    void init_arc(EmbArc arc, QRgb rgb, Qt::PenStyle lineType);
    void init_circle(EmbCircle circle, QRgb rgb, Qt::PenStyle lineType);
    void init_line(EmbLine line, QRgb rgb, Qt::PenStyle lineType);
    void init_ellipse(EmbEllipse ellipse, QRgb rgb, Qt::PenStyle lineType);
    void init_rect(EmbRect rect, QRgb rgb, Qt::PenStyle lineType);
    void init_text_single(QString str, EmbVector position, QRgb rgb, Qt::PenStyle lineType);
    void init_path(QPainterPath p, QRgb rgb, Qt::PenStyle lineType);
    void init_point(EmbVector pos, QRgb rgb, Qt::PenStyle lineType);

    void init(void);

    ~Geometry();

    /* Getters */
    Qt::PenStyle objectLineType() { return objPen.style(); }
    EmbReal objectLineWeight() { return lwtPen.widthF(); }
    QPointF objectRubberPoint(QString key);
    QString objectRubberText(QString key);

    QPointF objectCenter() { return scenePos(); }
    QPointF objectPos() { return scenePos(); }
    EmbReal objectX(){ return scenePos().x(); }
    EmbReal objectY(){ return scenePos().y(); }

    QPointF objectTopLeft();
    QPointF objectTopRight();
    QPointF objectBottomLeft();
    QPointF objectBottomRight();
    EmbReal objectArea();
    QPointF objectStartPoint();
    QPointF objectMidPoint();
    QPointF objectEndPoint();

    QRectF rect();
    void circle_click(Dictionary global, EmbVector v);
    EmbReal objectWidth();
    EmbReal objectHeight();
    EmbReal objectRadiusMajor();
    EmbReal objectRadiusMinor();
    EmbReal objectDiameterMajor();
    EmbReal objectDiameterMinor();
    QPointF objectEndPoint1();
    QPointF objectEndPoint2();
    EmbReal objectStartAngle();
    EmbReal objectEndAngle();
    EmbReal objectArcLength();
    EmbReal objectChord();
    EmbReal objectIncludedAngle();
    bool objectClockwise();
    EmbReal objectX1() { return objectEndPoint1().x(); }
    EmbReal objectY1() { return objectEndPoint1().y(); }
    EmbReal objectX2() { return objectEndPoint2().x(); }
    EmbReal objectY2() { return objectEndPoint2().y(); }
    EmbReal objectAngle();
    QPointF objectDelta() { return objectEndPoint2() - objectEndPoint1(); }
    EmbReal objectLength() { return objLine.length()*scale(); }
    EmbReal objectRadius();
    EmbReal objectDiameter();
    EmbReal objectCircumference();
    QPointF objectQuadrant0();
    QPointF objectQuadrant90();
    QPointF objectQuadrant180();
    QPointF objectQuadrant270();
    QPainterPath objectCopyPath();
    QPainterPath objectSavePath();

    std::vector<QPainterPath> objectSavePathList() { return subPathList(); }
    std::vector<QPainterPath> subPathList();

    int findIndex(const QPointF& point);

    void setObjectEndPoint1(EmbVector endPt1);
    void setObjectEndPoint2(EmbVector endPt2);

    void updatePath();
    void updatePath(const QPainterPath& p);
    void updateLeader(void);

    virtual QRectF boundingRect();

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

    void updateRubber(QPainter* painter = 0);
    void vulcanize(void);
    QPointF mouseSnapPoint(const QPointF& mousePoint);
    std::vector<QPointF> allGripPoints();
    void gripEdit(const QPointF& before, const QPointF& after);

    void realRender(QPainter* painter, const QPainterPath& renderPath);
    void paint(QPainter*, const QStyleOptionGraphicsItem*, QWidget*);

    /* Updaters, todo:  combine */
    void calculateArcData(EmbArc arc);
    void updateArcRect(EmbReal radius);

    /* Setters */
    void setObjectPos(const QPointF& point) { setPos(point.x(), point.y()); }
    void setObjectX(EmbReal x) { setPos(x, objectY()); }
    void setObjectY(EmbReal y) { setPos(objectX(), 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);
    void setObjectRadius(EmbReal radius);
    void setObjectStartAngle(EmbReal angle);
    void setObjectEndAngle(EmbReal angle);
    void setObjectStartPoint(EmbVector point);
    void setObjectMidPoint(EmbVector point);
    void setObjectEndPoint(EmbVector point);
    void setObjectDiameter(EmbReal diameter);
    void setObjectArea(EmbReal area);
    void setObjectCircumference(EmbReal circumference);
    void setObjectPos(EmbReal x, EmbReal y) { setPos(x, y); }
    void setObjectText(QString str);
    void setObjectTextFont(QString font);
    void setObjectTextJustify(QString justify);
    void setObjectTextSize(EmbReal size);
    void setObjectTextStyle(bool bold, bool italic, bool under, bool strike, bool over);
    void setObjectTextBold(bool val);
    void setObjectTextItalic(bool val);
    void setObjectTextUnderline(bool val);
    void setObjectTextStrikeOut(bool val);
    void setObjectTextOverline(bool val);
    void setObjectTextBackward(bool val);
    void setObjectTextUpsideDown(bool val);
    void setObjectRadiusMajor(EmbReal radius);
    void setObjectRadiusMinor(EmbReal radius);
    void setObjectDiameterMajor(EmbReal diameter);
    void setObjectDiameterMinor(EmbReal diameter);

    /* Scripted commands, uses the script string in */
    void script_main(void);
    void script_click(EmbVector v);
    void script_context(String str);
    void script_prompt(String str);
};

class SaveObject :  public QObject
{
    Q_OBJECT

public:
    SaveObject(QGraphicsScene* theScene, QObject* parent = 0);
    ~SaveObject();

    bool save(QString fileName);

    void addArc(EmbPattern* pattern, QGraphicsItem* item);
    void addBlock(EmbPattern* pattern, QGraphicsItem* item);
    void addCircle(EmbPattern* pattern, QGraphicsItem* item);
    void addDimAligned(EmbPattern* pattern, QGraphicsItem* item);
    void addDimAngular(EmbPattern* pattern, QGraphicsItem* item);
    void addDimArcLength(EmbPattern* pattern, QGraphicsItem* item);
    void addDimDiameter(EmbPattern* pattern, QGraphicsItem* item);
    void addDimLeader(EmbPattern* pattern, QGraphicsItem* item);
    void addDimLinear(EmbPattern* pattern, QGraphicsItem* item);
    void addDimOrdinate(EmbPattern* pattern, QGraphicsItem* item);
    void addDimRadius(EmbPattern* pattern, QGraphicsItem* item);
    void addEllipse(EmbPattern* pattern, QGraphicsItem* item);
    void addEllipseArc(EmbPattern* pattern, QGraphicsItem* item);
    void addGrid(EmbPattern* pattern, QGraphicsItem* item);
    void addHatch(EmbPattern* pattern, QGraphicsItem* item);
    void addImage(EmbPattern* pattern, QGraphicsItem* item);
    void addInfiniteLine(EmbPattern* pattern, QGraphicsItem* item);
    void addLine(EmbPattern* pattern, QGraphicsItem* item);
    void addPath(EmbPattern* pattern, QGraphicsItem* item);
    void addPoint(EmbPattern* pattern, QGraphicsItem* item);
    void addPolygon(EmbPattern* pattern, QGraphicsItem* item);
    void addPolyline(EmbPattern* pattern, QGraphicsItem* item);
    void addRay(EmbPattern* pattern, QGraphicsItem* item);
    void addRectangle(EmbPattern* pattern, QGraphicsItem* item);
    void addSlot(EmbPattern* pattern, QGraphicsItem* item);
    void addSpline(EmbPattern* pattern, QGraphicsItem* item);
    void addTextMulti(EmbPattern* pattern, QGraphicsItem* item);
    void addTextSingle(EmbPattern* pattern, QGraphicsItem* item);

    QGraphicsScene* gscene;
    int formatType;

    void toPolyline(EmbPattern* pattern, const QPointF& objPos, const QPainterPath& objPath, QString
  layer, const QColor& color, QString  lineType, QString  lineWeight);
};

class Application :  public QApplication
{
    Q_OBJECT
public:
    Application(int argc, char **argv);
    void setMainWin(MainWindow* mainWin) { __mainWin = _mainWin; }
    MainWindow* __mainWin;
protected:
    virtual bool event(QEvent *e);
};


class CmdPromptInput :  public QLineEdit
{
    Q_OBJECT

public:
    CmdPromptInput(QWidget* parent = 0);
    ~CmdPromptInput() {}

    QString curText;
    QString defaultPrefix;
    QString prefix;

    QString lastCmd;
    QString curCmd;
    bool cmdActive;

    bool rapidFireEnabled;
    bool isBlinking;

    void changeFormatting(std::vector<QTextLayout::FormatRange> formats);
    void clearFormatting();
    void applyFormatting();

protected:
    void contextMenuEvent(QContextMenuEvent *event);
    bool eventFilter(QObject *obj, QEvent *event);

signals:
    void appendHistory(QString  txt, int prefixLength);

    //These connect to the CmdPrompt signals
    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 slots:
    void endCommand();
    void processInput(void);
    void checkSelection();
    void updateCurrentText(QString  txt);
    void checkEditedText(QString  txt);
    void checkChangedText(QString  txt);
    void checkCursorPosition(int oldpos, int newpos);
private slots:
    void copyClip();
    void pasteClip();
};

class CmdPromptHistory :  public QTextBrowser
{
    Q_OBJECT

public:
    CmdPromptHistory(QWidget* parent = 0);
    ~CmdPromptHistory();

    int tmpHeight;
    QString applyFormatting(QString  txt, int prefixLength);

protected:
    void contextMenuEvent(QContextMenuEvent* event);

public slots:
    void appendHistory(QString  txt, int prefixLength);
    void startResizeHistory(int y);
    void stopResizeHistory(int y);
    void resizeHistory(int y);

signals:
    void historyAppended(QString  txt);
};

class CmdPromptSplitter :  public QSplitter
{
    Q_OBJECT

public:
    CmdPromptSplitter(QWidget* parent = 0);
    ~CmdPromptSplitter();

protected:
    QSplitterHandle* createHandle();

signals:
    void pressResizeHistory(int y);
    void releaseResizeHistory(int y);
    void moveResizeHistory(int y);
};

class CmdPromptHandle :  public QSplitterHandle
{
    Q_OBJECT

public:
    CmdPromptHandle(Qt::Orientation orientation, QSplitter* parent);
    ~CmdPromptHandle();

    int pressY;
    int releaseY;
    int moveY;

protected:
    void mousePressEvent(QMouseEvent* e);
    void mouseReleaseEvent(QMouseEvent* e);
    void mouseMoveEvent(QMouseEvent* e);

signals:
    void handlePressed(int y);
    void handleReleased(int y);
    void handleMoved(int y);
};

class CmdPrompt :  public QWidget
{
    Q_OBJECT

public:
    CmdPrompt(QWidget* parent = 0);
    ~CmdPrompt();

    CmdPromptInput* promptInput;
    CmdPromptHistory* promptHistory;
    QVBoxLayout* promptVBoxLayout;
    QFrame* promptDivider;

    CmdPromptSplitter* promptSplitter;

    QHash<QString, QString>*  styleHash;
    void updateStyle();
    QTimer* blinkTimer;
    bool blinkState;

public slots:
    void setCurrentText(QString  txt) {
        promptInput->curText = promptInput->prefix + txt;
        promptInput->setText(promptInput->curText);
    }
    void setHistory(QString txt) {
        promptHistory->setHtml(txt);
        promptHistory->moveCursor(QTextCursor::End, QTextCursor::MoveAnchor);
    }
    void setPrefix(QString  txt);
    void appendHistory(QString  txt);

    void alert(QString  txt);

    void startBlinking();
    void stopBlinking();
    void blink();

    void setPromptTextColor(const QColor&);
    void setPromptBackgroundColor(const QColor&);
    void setPromptFontFamily(QString );
    void setPromptFontStyle(QString );
    void setPromptFontSize(int);

    void floatingChanged(bool);

    void saveHistory(QString  fileName, bool html);

signals:
    void appendTheHistory(QString  txt, int prefixLength);

    //For connecting outside of command prompt
    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);
};

class EmbDetailsDialog :  public QDialog
{
    Q_OBJECT

public:
    EmbDetailsDialog(QGraphicsScene* theScene, QWidget *parent = 0);
    ~EmbDetailsDialog();

    QWidget* mainWidget;

    void getInfo();
    QWidget* createMainWidget();
    QWidget* createHistogram();

    QDialogButtonBox* buttonBox;

    uint32_t stitchesTotal;
    uint32_t stitchesReal;
    uint32_t stitchesJump;
    uint32_t stitchesTrim;
    uint32_t colorTotal;
    uint32_t colorChanges;

    QRectF boundingRect;
};

class ImageWidget :  public QWidget
{
    Q_OBJECT

public:
    QImage img;
    ImageWidget(QString filename, QWidget* parent = 0);
    ~ImageWidget();

    bool load(QString fileName);
    bool save(QString fileName);

protected:
    void paintEvent(QPaintEvent* event);
};

class LayerManager :  public QDialog
{
    Q_OBJECT

public:
    QStandardItemModel* layerModel;
    QSortFilterProxyModel* layerModelSorted;
    QTreeView* treeView;

    LayerManager(QWidget *parent = 0);
    ~LayerManager();

    void addLayer(QString name, const bool visible, const bool frozen,
        const EmbReal zValue, const QRgb color, QString lineType,
        QString lineWeight, const bool print);
};

class MainWindow:  public QMainWindow
{
    Q_OBJECT

public:
    MainWindow();
    ~MainWindow();

    MdiWindow* activeMdiWindow();
    QUndoStack* activeUndoStack();

    void setUndoCleanIcon(bool opened);

    virtual void updateMenuToolbarStatusbar();

    std::vector<QGraphicsItem*> cutCopyObjectList;

    QString formatFilterOpen;
    QString formatFilterSave;

    bool isCommandActive() { return prompt->promptInput->cmdActive; }
    QString activeCommand() { return prompt->promptInput->curCmd; }
    QIcon create_icon(QString stub);
    void create_toolbar(String toolbar, String label, StringList entries);

    QString platformString();

public slots:

    void onCloseWindow();
    virtual void onCloseMdiWin(MdiWindow*);

    void recentMenuAboutToShow();

    void onWindowActivated(QMdiSubWindow* w);
    void windowMenuAboutToShow();
    void windowMenuActivated( bool checked/*int id*/ );

    void updateAllViewScrollBars(bool val);
    void updateAllViewCrossHairColors(QRgb color);
    void updateAllViewBackgroundColors(QRgb color);
    void updateAllViewSelectBoxColors(QRgb colorL, QRgb fillL, QRgb colorR, QRgb fillR, int alpha);
    void updateAllViewGridColors(QRgb color);
    void updateAllViewRulerColors(QRgb color);

    void updatePickAddMode(bool val);
    void pickAddModeToggled();

    void settingsPrompt();

protected:
    virtual void resizeEvent(QResizeEvent*);
    void closeEvent(QCloseEvent *event);
    QAction* getFileSeparator();
    void loadFormats();

    bool shiftKeyPressedState;

    QByteArray layoutState;

    int numOfDocs;
    int docIndex;

    std::vector<MdiWindow*> listMdiWin;
    QMdiSubWindow* findMdiWindow(String fileName);

    QAction* myFileSeparator;

    void createAllActions();
    void createAllMenus();
    void createAllToolbars();

    // Selectors
    QComboBox* layerSelector;
    QComboBox* colorSelector;
    QComboBox* linetypeSelector;
    QComboBox* lineweightSelector;
    QFontComboBox* textFontSelector;
    QComboBox* textSizeSelector;

private slots:
    void hideUnimplemented();

public slots:
    void stub_testing();

    void promptHistoryAppended(QString  txt);
    void logPromptInput(QString  txt);
    void promptInputPrevious();
    void promptInputNext();

    void about(void);
    void tipOfTheDay(void);

    void newFile();
    void openFile(bool recent = false, String recentFile = "");
    void openFilesSelected(StringList files);
    void openrecentfile();
    void savefile();
    void saveasfile();
    void quit();
    void checkForUpdates();
    // Help Menu
    void buttonTipOfTheDayClicked(int);

    void closeToolBar(QAction*);
    void floatingChangedToolBar(bool);

    void toggleGrid();
    void toggleRuler();
    void toggleLwt();

    // Icons
    void iconResize(int iconSize);

    //Selectors
    void layerSelectorIndexChanged(int index);
    void colorSelectorIndexChanged(int index);
    void linetypeSelectorIndexChanged(int index);
    void lineweightSelectorIndexChanged(int index);
    void textFontSelectorCurrentFontChanged(const QFont& font);
    void textSizeSelectorIndexChanged(int index);

    void setTextFont(QString  str);
    void setTextSize(EmbReal num);

    QString getCurrentLayer();
    QRgb getCurrentColor();
    QString getCurrentLineType();
    QString getCurrentLineWeight();

    bool isShiftPressed();
    void setShiftPressed();
    void setShiftReleased();

    void deletePressed();
    void escapePressed();
};

class MdiWindow:  public QMdiSubWindow
{
    Q_OBJECT

public:
    MdiWindow(const int theIndex, QMdiArea* parent, Qt::WindowFlags wflags);
    ~MdiWindow();

    QMdiArea* mdiArea;
    QGraphicsScene* gscene;
    View* gview;

    bool fileWasLoaded;

    QString promptHistory;
    std::vector<QString> promptInputList;
    int promptInputNum;

    QPrinter printer;

    QString curFile;
    void setCurrentFile(QString  fileName);

    int myIndex;

    QString curLayer;
    QRgb curColor;
    QString curLineType;
    QString curLineWeight;

    void promptInputPrevNext(bool prev);

    virtual QSize sizeHint();
    QString getShortCurrentFile();
    void designDetails();
    bool loadFile(String fileName);
    bool saveFile(String fileName);
signals:
    void sendCloseMdiWin(MdiWindow*);

public slots:
    void closeEvent(QCloseEvent* e);
    void onWindowActivated();

    void currentLayerChanged(QString layer);
    void currentColorChanged(const QRgb& color);
    void currentLinetypeChanged(QString  type);
    void currentLineweightChanged(QString  weight);

    void updateColorLinetypeLineweight();
    void deletePressed();
    void escapePressed();

    void showViewScrollBars(bool val);
    void setViewCrossHairColor(QRgb color);
    void setViewBackgroundColor(QRgb color);
    void setViewSelectBoxColors(QRgb colorL, QRgb fillL, QRgb colorR, QRgb fillR, int alpha);
    void setViewGridColor(QRgb color);
    void setViewRulerColor(QRgb color);

    void print();
    void saveBMC();

    void promptHistoryAppended(QString  txt);
    void logPromptInput(QString  txt);
    void promptInputPrevious();
    void promptInputNext();
};

class MdiArea :  public QMdiArea
{
    Q_OBJECT

public:
    bool useLogo;
    bool useTexture;
    bool useColor;

    QPixmap bgLogo;
    QPixmap bgTexture;
    QColor  bgColor;

    void zoomExtentsAllSubWindows();
    void forceRepaint();

    MdiArea(QWidget* parent = 0);
    ~MdiArea();

    void useBackgroundLogo(bool use);
    void useBackgroundTexture(bool use);
    void useBackgroundColor(bool use);

    void setBackgroundLogo(QString  fileName);
    void setBackgroundTexture(QString  fileName);
    void setBackgroundColor(const QColor& color);

public slots:
    void cascade();
    void tile();
protected:
    virtual void mouseDoubleClickEvent(QMouseEvent* e);
    virtual void paintEvent(QPaintEvent* e);
};

class PreviewDialog :  public QFileDialog
{
    Q_OBJECT

public:
    PreviewDialog(QWidget* parent = 0,
       QString  caption = QString(),
       QString  directory = QString(),
       QString  filter = QString());
    ~PreviewDialog();

    ImageWidget* imgWidget;
};


class PropertyEditor :  public QDockWidget
{
    Q_OBJECT

public:
    PropertyEditor(QString  iconDirectory = QString(), bool pickAddMode = true, QWidget* widgetToFocus = 0, QWidget* parent = 0); //, Qt::WindowFlags flags = 0);
    ~PropertyEditor();

    QWidget* focusWidget;

    QString iconDir;
    int iconSize;
    Qt::ToolButtonStyle propertyEditorButtonStyle;

    bool pickAdd;

    std::vector<QGraphicsItem*> selectedItemList;

    QToolButton* createToolButton(QString  iconName, QString  txt);
    QLineEdit* createLineEdit(QString  validatorType = QString(), bool readOnly = false);

    int precisionAngle;
    int precisionLength;

    void updateLineEditStrIfVaries(QLineEdit* lineEdit, QString  str);
    void updateLineEditNumIfVaries(QLineEdit* lineEdit, EmbReal num, bool useAnglePrecision);
    void updateFontComboBoxStrIfVaries(QFontComboBox* fontComboBox, QString  str);
    void updateComboBoxStrIfVaries(QComboBox* comboBox, QString str, StringList strList);
    void updateComboBoxBoolIfVaries(QComboBox* comboBox, bool val, bool yesOrNoText);

    QSignalMapper* signalMapper;
    void mapSignal(QObject* fieldObj, QString  name, QVariant value);

    // Selection
    // ====================
    QComboBox*   createComboBoxSelected();
    QToolButton* createToolButtonQSelect();
    QToolButton* createToolButtonPickAdd();

    QComboBox*   comboBoxSelected;
    QToolButton* toolButtonQSelect;
    QToolButton* toolButtonPickAdd;

    //TODO: Alphabetic/Categorized TabWidget

    void createGroupBox(String group_box_key, const char *title);

protected:
    bool eventFilter(QObject *obj, QEvent *event);

signals:
    void pickAddModeToggled();

public slots:
    void setSelectedItems(std::vector<QGraphicsItem*> itemList);
    void updatePickAddModeButton(bool pickAddMode);

private slots:
    void fieldEdited(QObject* fieldObj);
    void showGroups(int objType);
    void showOneType(int index);
    void hideAllGroups();
    void clearAllFields();
    void togglePickAddMode();
};


class SelectBox :  public QRubberBand
{
    Q_OBJECT

public:
    SelectBox(Shape s, QWidget* parent = 0);

    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;

    void forceRepaint();

public slots:
    void setDirection(int dir);
    void setColors(const QColor& colorL, const QColor& fillL, const QColor& colorR, const QColor& fillR, int newAlpha);

protected:
    void paintEvent(QPaintEvent*);
};

class Settings_Dialog :  public QDialog
{
    Q_OBJECT

public:
    Settings_Dialog(QString  showTab = QString(), QWidget *parent = 0);
    ~Settings_Dialog();

    QTabWidget* tabWidget;

    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();

    QDialogButtonBox* buttonBox;

    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);

private slots:
    void comboBoxIconSizeCurrentIndexChanged(int);
    void checkBoxGeneralMdiBGUseLogoStateChanged(int);
    void chooseGeneralMdiBackgroundLogo();
    void checkBoxGeneralMdiBGUseTextureStateChanged(int);
    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 );
    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();

signals:
    void buttonCustomFilterSelectAll(bool);
    void buttonCustomFilterClearAll(bool);
    void buttonQSnapSelectAll(bool);
    void buttonQSnapClearAll(bool);
};

class StatusBar :  public QStatusBar
{
    Q_OBJECT

public:
    StatusBar(QWidget* parent = 0);
    std::unordered_map<String, QToolButton*> buttons;
    QLabel* statusBarMouseCoord;
    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);
};

class UndoEditor :  public QDockWidget
{
    Q_OBJECT

public:
    UndoEditor(QString  iconDirectory = QString(), QWidget* widgetToFocus = 0, QWidget* parent = 0); //, Qt::WindowFlags flags = 0);
    ~UndoEditor();

    void addStack(QUndoStack* stack);

    bool canUndo();
    bool canRedo();

    QWidget* focusWidget;

    QString iconDir;
    int iconSize;

    QUndoGroup* undoGroup;
    QUndoView*  undoView;

    QString undoText();
    QString redoText();
protected:

public slots:
    void undo();
    void redo();

    void updateCleanIcon(bool opened);
};

class UndoableCommand :  public QUndoCommand
{
```

```
public:
    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(){ return 1234; }
    bool mergeWith(const QUndoCommand* command);
    void undo();
    void redo();
    void mirror();
    void rotate(EmbVector pivot, EmbReal rot);

    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;
};

class View :  public QGraphicsView
{
    Q_OBJECT

public:
    View(QGraphicsScene* theScene, QWidget* parent);
    ~View();

    Dictionary state;

    std::vector<QGraphicsItem*> selected_items();

    bool allowZoomIn();
    bool allowZoomOut();

    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;

    void updateMouseCoords(int x, int y);
    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;

    void recalculateLimits();
    void zoomToPoint(const QPoint& mousePoint, int zoomDir);
    void centerAt(const QPointF& centerPoint);
    QPointF center() { return mapToScene(rect().center()); }

    QUndoStack* getUndoStack() { return undoStack; }
    void addObject(Geometry* obj);
    void deleteObject(Geometry* obj);
    void vulcanizeObject(Geometry* obj);

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);

protected:
    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:
    QHash<int64_t, QGraphicsItem*> hashDeletedObjects;

    StringList spareRubberList;

    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);

    QList<QGraphicsItem*> previewObjectList;
    QGraphicsItemGroup* previewObjectItemGroup;
    QPointF previewPoint;
    EmbReal previewData;
    String previewMode;

    std::vector<QGraphicsItem*> createObjectList(std::vector<QGraphicsItem*> list);
    QPointF cutCopyMousePoint;
    QGraphicsItemGroup* pasteObjectItemGroup;
    QPointF pasteDelta;

    std::vector<QGraphicsItem*> rubberRoomList;

    void copySelected();

    void startGripping(Geometry* obj);
    void stopGripping(bool accept = false);

    void panStart(const QPoint& point);
    int panDistance;
    int panStartX;
    int panStartY;

    void alignScenePointWithViewPoint(const QPointF& scenePoint, const QPoint& viewPoint);
};

#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

msg	

18.9.2.3 degrees__()

```
EmbReal degrees__ (
        EmbReal radian )
```

degrees__

Parameters

radian	

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

node	
value	

18.9.2.16 node_int()

```
Node node_int (
          int32_t value )
```

create_node

Parameters

mode	

Returns

18.9.2.17 node_qstr()

```
Node node_qstr (
          QString value )
```

set_node

Parameters

node	
value	

18.9.2.18 node_real()

```
Node node_real (
          EmbReal value )
```

set_node

Parameters

node	
value	

18.9.2.19 node_str()

```
Node node_str (
          String value )
```

set_node

Parameters

node	
value	

18.9.2.20 node_str_list()

```
Node node_str_list (
          StringList value )
```

set_node

Parameters

node	
value	

18.9.2.21 node_uint()

```
Node node_uint (
    uint32_t value )
```

create_node

Parameters

mode	

Returns

18.9.2.22 operator*()

```
EmbVector operator* (
    EmbVector v,
    EmbReal s )
```

operator *

Parameters

v	
s	

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

degrees	

Returns

18.9.2.26 set_enabled()

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

set_enabled

Parameters

parent	
key	
enabled	

Todo error reporting.

18.9.2.27 set_visibility()

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

set_visibility

Parameters

parent	
key	
visibility	

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

list	

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

list	

Returns

18.9.2.32 to_vector()

```
std::vector< QGraphicsItem * > to_vector (
        QList< QGraphicsItem * > list )
```

to_vector

Parameters

list	

Returns

18.9.2.33 tokenize()

```
StringList tokenize (
        String str,
        const char delim )
```

tokenize

Parameters

str	
delim	

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

menu	
def	
topLevel	

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

command	

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

args	

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

args	

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

args	

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

args	

Returns

18.13.2.19 add_point_action() [2/2]

```
static String add_point_action (
        String args ) [static]
```

add_point_action

Parameters

args	

Returns

18.13.2.20 add_polygon_action()

```
static String add_polygon_action (
        String args ) [static]
```

add_polygon_action

Parameters

args	

Returns

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

18.13.2.21 add_polyline_action()

```
static String add_polyline_action (
        String args ) [static]
```

18.13.2.22 add_ray_action()

```
static String add_ray_action (
        String args ) [static]
```

EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, EmbReal rot

18.13.2.23 add_rectangle_action()

```
static String add_rectangle_action (
        String args ) [static]
```

add_rectangle_action

Parameters

args	

Returns

18.13.2.24 add_regular_polygon_action()

```
static String add_regular_polygon_action (
        String args ) [static]
```

AddRegularPolygon.

Returns

EmbReal centerX, EmbReal centerY, quint16 sides, uint8_t mode, EmbReal rad, EmbReal rot, bool fill

18.13.2.25 add_rounded_rectangle_action()

```
static String add_rounded_rectangle_action (
        String args ) [static]
```

add_rounded_rectangle_action

Parameters

args	

Returns

EmbReal x, EmbReal y, EmbReal w, EmbReal h, EmbReal rad, EmbReal rot, bool fill

18.13.2.26 add_rubber_action()

```
String add_rubber_action (
        String args ) [static]
```

add_rubber_action

Parameters

args	

Returns

18.13.2.27 add_slot_action()

```
static String add_slot_action (
        String args ) [static]
```

add_slot_action

Parameters

args	

Returns

EmbReal centerX, EmbReal centerY, EmbReal diameter, EmbReal length, EmbReal rot, bool fill, String rubberMode

18.13.2.28 add_text_multi_action()

```
static String add_text_multi_action (
        String args ) [static]
```

add_text_multi_action

Returns

QString str, EmbReal x, EmbReal y, EmbReal rot, bool fill, String rubberMode

18.13.2.29 add_text_single_action()

```
static String add_text_single_action (
        String args ) [static]
```

add_text_single_action

Returns

QString str, EmbReal x, EmbReal y, EmbReal rot, bool fill, String rubberMode

18.13.2.30 add_to_selection_action()

```
static String add_to_selection_action (
            String args ) [static]
```

add_to_selection_action

Parameters

args	

Returns

18.13.2.31 add_triangle_action() `static String add_triangle_action ( String args ) [static]`

add_triangle_action

Parameters

args	

Returns

18.13.2.32 add_vertical_dimension_action() `static String add_vertical_dimension_action ( String args ) [static]`

EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2, EmbReal legHeight

18.13.2.33 alert_action() `static String alert_action ( String args ) [static]`

alert_action

Parameters

args	

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

args	

Returns

18.13.2.36 append_prompt_history_action()

```
String append_prompt_history_action (
        String args )  [static]
```

AppendPromptHistory.

Parameters

a	

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

args	

Returns

18.13.2.39 calculate_distance_action()

```
static String calculate_distance_action (
        String args )  [static]
```

calculate_distance

Parameters

args	

Returns

18.13.2.40 changelog_action()

```
static String changelog_action (
        String args )  [static]
```

changelog_action

Parameters

args	

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

command	
fmt	

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

label	The caller's name.
args	The list of strings passed from the user.
args_template	The string of characters describing the types of the output.
result	The fixed length array of results.

Returns

An error message if an error occured or an empty string if it passes.

18.13.2.45 copy_action()

```
static String copy_action (
    String args ) [static]
```

copy_action

Parameters

args	

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

args	

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

variable	

Returns

18.13.2.54 do_nothing_action()

```
String do_nothing_action (
        String args )  [static]
```

do_nothing_action This action intensionally does nothing.

Parameters

args	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

args	

Returns

18.13.2.56 error_action()

```
String error_action (
        String args )  [static]
```

Error.

Parameters

a	

Returns

18.13.2.57 help_action()

```
static String help_action (
        String args )  [static]
```

help_action

Parameters

args	

Returns

18.13.2.58 icon_action()

```
static String icon_action (
        String command ) [static]
```

icon_action

Parameters

command	

Returns

18.13.2.59 include_action()

```
String include_action (
        NodeList a )
```

Include.

Parameters

a	

Returns

18.13.2.60 init_action()

```
static String init_action (
        String args ) [static]
```

init_action

Parameters

args	

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

args	

Returns

18.13.2.63 layer_previous_action()

```
String layer_previous_action (
        String args )
```

layer_previous_action

Parameters

args	

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

args	

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

function_name	
args	

18.13.2.73 num_selected_action()

```
static String num_selected_action (
    String args ) [static]
```

NumSelected.

Parameters

args	

Returns

18.13.2.74 open_action() `static String open_action ( String args ) [static]`

open_action

Parameters

args	

Returns

18.13.2.75 pan_action() `String pan_action ( String mode ) [static]`

pan_action

Parameters

mode	

Returns

18.13.2.76 paste_action() `static String paste_action ( String args ) [static]`

paste_action

Parameters

args	

Returns

18.13.2.77 paste_selected_action() `static String paste_selected_action ( String args ) [static]`

PasteSelected *x y*.

18.13.2.78 perpendicular_distance_action() `static String perpendicular_distance_action ( String args ) [static]`

18.13.2.79 platform_action() `static String platform_action ( String args ) [static]`

platform_action

Parameters

args	

Returns

18.13.2.80 platformString()

```
String platformString (
        void )
```

platformString

Returns

18.13.2.81 preview_off_action()

```
static String preview_off_action (
        String args ) [static]
```

PreviewOff.

18.13.2.82 preview_on_action() [1/2]

```
String preview_on_action (
        String args ) [static]
```

preview_on_action

Parameters

args	

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

clone	
mode	
x	
y	
data	

18.13.2.84 print_action()

```
String print_action (
        String args ) [static]
```

print_action

Parameters

args	

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

args	

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 sttring 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

fname	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

args	

Returns

18.13.2.99 set_background_color_action()

```
static String set_background_color_action (
        String args ) [static]
```

set_background_color_action

Parameters

r	
g	
b	

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

r	
g	
b	

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

r	
g	
b	

18.13.2.105 set_prompt_prefix_action()

```
static String set_prompt_prefix_action (
        String args ) [static]
```

set_prompt_prefix_action

Parameters

args	

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

args	

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

showTab	

18.13.2.113 spare_rubber_action()

```
static String spare_rubber_action (
    String args ) [static]
```

SpareRubber.

Parameters

qint64	id

18.13.2.114 tip_of_the_day_action()

```
String tip_of_the_day_action (
    String args ) [static]
```

tip_of_the_day_action

Parameters

args	

Returns

18.13.2.115 todo_action()

```
String todo_action (
    String args ) [static]
```

Todo.

Parameters

a	

Returns

18.13.2.116 undo_action()

```
static String undo_action (
            String args ) [static]
```

undo_action

Parameters

args	

Returns

18.13.2.117 validFileFormat()

```
bool validFileFormat (
            String fileName )
```

MainWindow::validFileFormat.

Parameters

fileName	

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

args	

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

args	

Returns

18.13.2.122 window_action()

```
static String window_action (
        String args ) [static]
```

window_action

Parameters

args	

Returns

18.13.2.123 zoom_action()

```
String zoom_action (
        String mode ) [static]
```

zoom_action

Parameters

mode	

Returns

18.13.3 Variable Documentation

18.13.3.1 _mainWin

```
MainWindow* _mainWin = 0
```

18.13.3.2 actionHash

```
std::unordered_map<String, QAction*> actionHash
```

18.13.3.3 checkBoxes

```
std::unordered_map<String, QCheckBox *> checkBoxes
```

18.13.3.4 checkBoxTipOfTheDay

```
QCheckBox* checkBoxTipOfTheDay
```

18.13.3.5 comboBoxes `std::unordered_map<String, QComboBox *> comboBoxes`

18.13.3.6 command_map `std::unordered_map<String, Command> command_map`

18.13.3.7 config `Dictionary config`

18.13.3.8 config_tables `std::unordered_map<String, Dictionary> config_tables`

18.13.3.9 dialog `Dictionary dialog`

18.13.3.10 dockPropEdit `PropertyEditor* dockPropEdit = 0`

18.13.3.11 dockUndoEdit `UndoEditor* dockUndoEdit = 0`

18.13.3.12 doubleSpinBoxes `std::unordered_map<String, QDoubleSpinBox *> doubleSpinBoxes`

18.13.3.13 groupBoxes `std::unordered_map<String, QGroupBox *> groupBoxes`

18.13.3.14 labels `std::unordered_map<String, QLabel *> labels`

18.13.3.15 labelTipOfTheDay `QLabel* labelTipOfTheDay`

18.13.3.16 lineEdits `std::unordered_map<String, QLineEdit *> lineEdits`

18.13.3.17 mdiArea `MdiArea* mdiArea = 0`

18.13.3.18 menuHash `std::unordered_map<String, QMenu*> menuHash`

18.13.3.19 prompt `CmdPrompt* prompt = 0`

18.13.3.20 rubber_modes `StringList rubber_modes`

Initial value:

```
= {
    "CIRCLE_1P_RAD",
    "CIRCLE_1P_DIA",
    "CIRCLE_2P",
    "CIRCLE_3P",
    "CIRCLE_TTR",
    "CIRCLE_TTT",
    "DIMLEADER_LINE",
    "ELLIPSE_LINE",
    "ELLIPSE_MAJORDIAMETER_MINORRADIUS",
    "ELLIPSE_MAJORRADIUS_MINORRADIUS",
    "ELLIPSE_ROTATION",
    "LINE",
    "POLYGON",
    "POLYGON_INSCRIBE",
    "POLYGON_CIRCUMSCRIBE",
    "POLYLINE",
    "RECTANGLE",
    "TEXTSINGLE"
}
```

18.13.3.21 scripts `std::unordered_map<String, StringList> scripts`

18.13.3.22 settings `Dictionary settings`

Settings System.

Rather than pollute the global namespace, we collect together all the global settings into a structure that stores them. This also allows us to create a complete copy of the settings for the purpose of restoring them if the user cancels out of the Settings Dialog.

18.13.3.23 spinBoxes `std::unordered_map<String, QSpinBox *> spinBoxes`

18.13.3.24 statusbar `StatusBar* statusbar = 0`

18.13.3.25 subMenuHash `std::unordered_map<String, QMenu*> subMenuHash`

18.13.3.26 toolbarHash `std::unordered_map<String, QToolBar*> toolbarHash`

18.13.3.27 toolButtons `std::unordered_map<String, QToolButton *> toolButtons`

18.13.3.28 wizardTipOfTheDay `QWizard* wizardTipOfTheDay`

18.14 embroidermodder2/mdiarea.cpp File Reference

```
#include "embroidermodder.h"
```

18.15 embroidermodder2/mdiwindow.cpp File Reference

```
#include "embroidermodder.h"
```

Functions

- QString fileExtension (String fileName)

  MdiWindow::fileExtension.

18.15.1 Function Documentation

18.15.1.1 fileExtension() `QString fileExtension ( String fileName )`

MdiWindow::fileExtension.

Parameters

fileName	

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

p	
rubberMode	

18.16.1.2 closest_point()

```
QPointF closest_point (
        QPointF position,
        std::vector< QPointF > points )
```

mouse_snap_point

Parameters

points	

Returns

18.16.1.3 fourier_series()

```
EmbReal fourier_series (
        EmbReal arg,
        std::vector< EmbReal > terms )
```

fourier_series

Parameters

arg	
terms	

18.16.1.4 rotate_vector() `EmbVector rotate_vector ( EmbVector v, EmbReal alpha )`

Returns

18.17 embroidermodder2/preview-dialog.cpp File Reference

```
#include "embroidermodder.h"
```

18.18 embroidermodder2/property-editor.cpp File Reference

```
#include "embroidermodder.h"
```

Functions

- std::vector< Dictionary > load_group_box_data_from_table (String key)

Variables

- QString fieldOldText
- QString fieldNewText
- QString fieldVariesText
- QString fieldYesText
- QString fieldNoText
- QString fieldOnText
- QString fieldOffText
- StringList object_names
- std::vector< std::pair< String, int > > group_box_types
- QFontComboBox ∗ comboBoxTextSingleFont
- std::unordered_map< String, Dictionary > group_box_data

18.18.1 Function Documentation

18.18.1.1 load_group_box_data_from_table() `std::vector< Dictionary > load_group_box_data_from_table ( String key )`

18.18.2 Variable Documentation

18.18.2.1 comboBoxTextSingleFont `QFontComboBox* comboBoxTextSingleFont`

18.18.2.2 fieldNewText `QString fieldNewText`

18.18.2.3 fieldNoText `QString fieldNoText`

18.18.2.4 fieldOffText `QString fieldOffText`

18.18.2.5 fieldOldText `QString fieldOldText`

18.18.2.6 fieldOnText `QString fieldOnText`

18.18.2.7 fieldVariesText `QString fieldVariesText`

18.18.2.8 fieldYesText `QString fieldYesText`

18.18.2.9 group_box_data `std::unordered_map<String, Dictionary> group_box_data`

18.18.2.10 group_box_types `std::vector<std::pair<String, int> > group_box_types`

18.18.2.11 object_names `StringList object_names`

Initial value:

```
= {
    "Base",
    "Arc",
    "Block",
    "Circle",
    "Aligned Dimension",
    "Angular Dimension",
    "Arclength Dimension",
    "Diameter Dimension",
    "Leader Dimension",
    "Linear Dimension",
    "Ordinate Dimension",
    "Radius Dimension",
    "Ellipse",
    "Image",
    "Infinite Line",
    "Line",
    "Path",
    "Point",
    "Polygon",
    "Polyline",
    "Ray",
    "Rectangle",
    "Multiline Text",
    "Text",
    "Unknown"
}
```

18.19 embroidermodder2/README.md File Reference

18.20 embroidermodder2/selectbox.cpp File Reference

```
#include "embroidermodder.h"
```

18.21 embroidermodder2/settings-dialog.cpp File Reference

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

Functions

- void make_editing_copy (StringList props)
- void read_settings (void)

  read_settings
- void write_settings (void)

  MainWindow::writeSettings.

Variables

- Dictionary preview
- Dictionary accept_
- StringList extensions
- StringList general_props
- StringList display_props
- StringList prompt_props
- StringList quick_snap_props
- StringList opensave_props

18.21.1 Function Documentation

18.21.1.1 make_editing_copy()

```
void make_editing_copy (
        StringList props )
```

18.21.1.2 read_settings()

```
void read_settings (
        void )
```

read_settings

This file needs to be read from the users home directory to ensure it is writable.

18.21.1.3 write_settings()

```
void write_settings (
        void )
```

MainWindow::writeSettings.

This file needs to be read from the users home directory to ensure it is writable

18.21.2 Variable Documentation

18.21.2.1 accept_

```
Dictionary accept_
```

18.21.2.2 display_props

```
StringList display_props
```

Initial value:

```
= {
    "display_use_opengl",
    "display_renderhint_aa",
    "display_renderhint_text_aa",
    "display_renderhint_smooth_pix",
    "display_renderhint_high_aa",
    "display_renderhint_noncosmetic",
    "display_show_scrollbars",
    "display_scrollbar_widget_num",
    "display_crosshair_color",
    "display_bg_color",
    "display_selectbox_left_color",
    "display_selectbox_left_fill",
    "display_selectbox_right_color",
    "display_selectbox_right_fill",
    "display_selectbox_alpha",
    "display_zoomscale_in",
    "display_zoomscale_out"
}
```

18.21.2.3 extensions StringList extensions

Initial value:

```
= {
    "100", "10o", "ART", "BMC", "BRO",
    "CND", "COL", "CSD", "CSV", "DAT",
    "DEM", "DSB", "DST", "DSZ", "DXF",
    "EDR", "EMD", "EXP", "EXY", "EYS",
    "FXY", "GNC", "GT", "HUS", "INB",
    "JEF", "KSM", "PCD", "PCM", "PCQ",
    "PCS", "PEC", "PEL", "PEM", "PES",
    "PHB", "PHC", "RGB", "SEW", "SHV",
    "SST", "STX", "SVG", "T09", "TAP",
    "THR", "TXT", "U00", "U01", "VIP",
    "VP3", "XXX", "ZSK"
}
```

18.21.2.4 general_props StringList general_props

Initial value:

```
= {
    "general_icon_theme",
    "general_icon_size",
    "general_mdi_bg_use_logo",
    "general_mdi_bg_logo",
    "general_mdi_bg_use_texture",
    "general_mdi_bg_texture",
    "general_mdi_bg_use_color",
    "general_mdi_bg_color",
    "general_tip_of_the_day"
}
```

18.21.2.5 opensave_props StringList opensave_props

Initial value:

```
= {
    "opensave_custom_filter"
}
```

18.21.2.6 preview Dictionary preview

Embroidermodder 2.

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/EmbroidePy/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_10O 1
- #define EMB_FORMAT_ART 2
- #define EMB_FORMAT_BMC 3
- #define EMB_FORMAT_BRO 4
- #define EMB_FORMAT_CND 5
- #define EMB_FORMAT_COL 6
- #define EMB_FORMAT_CSD 7
- #define EMB_FORMAT_CSV 8
- #define EMB_FORMAT_DAT 9
- #define EMB_FORMAT_DEM 10
- #define EMB_FORMAT_DSB 11
- #define EMB_FORMAT_DST 12
- #define EMB_FORMAT_DSZ 13
- #define EMB_FORMAT_DXF 14
- #define EMB_FORMAT_EDR 15
- #define EMB_FORMAT_EMD 16
- #define EMB_FORMAT_EXP 17
- #define EMB_FORMAT_EXY 18
- #define EMB_FORMAT_EYS 19
- #define EMB_FORMAT_FXY 20
- #define EMB_FORMAT_GC 21
- #define EMB_FORMAT_GNC 22
- #define EMB_FORMAT_GT 23
- #define EMB_FORMAT_HUS 24
- #define EMB_FORMAT_INB 25
- #define EMB_FORMAT_INF 26
- #define EMB_FORMAT_JEF 27
- #define EMB_FORMAT_KSM 28
- #define EMB_FORMAT_MAX 29
- #define EMB_FORMAT_MIT 30
- #define EMB_FORMAT_NEW 31
- #define EMB_FORMAT_OFM 32
- #define EMB_FORMAT_PCD 33
- #define EMB_FORMAT_PCM 34
- #define EMB_FORMAT_PCQ 35
- #define EMB_FORMAT_PCS 36
- #define EMB_FORMAT_PEC 37
- #define EMB_FORMAT_PEL 38
- #define EMB_FORMAT_PEM 39
- #define EMB_FORMAT_PES 40
- #define EMB_FORMAT_PHB 41
- #define EMB_FORMAT_PHC 42
- #define EMB_FORMAT_PLT 43
- #define EMB_FORMAT_RGB 44
- #define EMB_FORMAT_SEW 45

- #define EMB_FORMAT_SHV 46
- #define EMB_FORMAT_SST 47
- #define EMB_FORMAT_STX 48
- #define EMB_FORMAT_SVG 49
- #define EMB_FORMAT_T01 50
- #define EMB_FORMAT_T09 51
- #define EMB_FORMAT_TAP 52
- #define EMB_FORMAT_THR 53
- #define EMB_FORMAT_TXT 54
- #define EMB_FORMAT_U00 55
- #define EMB_FORMAT_U01 56
- #define EMB_FORMAT_VIP 57
- #define EMB_FORMAT_VP3 58
- #define EMB_FORMAT_XXX 59
- #define EMB_FORMAT_ZSK 60
- #define Arc_Polyester 0
- #define Arc_Rayon 1
- #define CoatsAndClark_Rayon 2
- #define Exquisite_Polyester 3
- #define Fufu_Polyester 4
- #define Fufu_Rayon 5
- #define Hemingworth_Polyester 6
- #define Isacord_Polyester 7
- #define Isafil_Rayon 8
- #define Marathon_Polyester 9
- #define Marathon_Rayon 10
- #define Madeira_Polyester 11
- #define Madeira_Rayon 12
- #define Metro_Polyester 13
- #define Pantone 14
- #define RobisonAnton_Polyester 15
- #define RobisonAnton_Rayon 16
- #define Sigma_Polyester 17
- #define Sulky_Rayon 18
- #define ThreadArt_Rayon 19
- #define ThreadArt_Polyester 20
- #define ThreaDelight_Polyester 21
- #define Z102_Isacord_Polyester 22
- #define SVG_Colors 23
- #define hus_thread 24
- #define jef_thread 25
- #define pcm_thread 26
- #define pec_thread 27
- #define shv_thread 28
- #define dxf_color 29
- #define EMB_ARRAY 0
- #define EMB_ARC 1
- #define EMB_CIRCLE 2
- #define EMB_DIM_DIAMETER 3
- #define EMB_DIM_LEADER 4
- #define EMB_ELLIPSE 5
- #define EMB_FLAG 6
- #define EMB_LINE 7
- #define EMB_IMAGE 8
- #define EMB_PATH 9

- #define EMB_POINT 10
- #define EMB_POLYGON 11
- #define EMB_POLYLINE 12
- #define EMB_RECT 13
- #define EMB_SPLINE 14
- #define EMB_STITCH 15
- #define EMB_TEXT_SINGLE 16
- #define EMB_TEXT_MULTI 17
- #define EMB_VECTOR 18
- #define EMB_THREAD 19
- #define EMBFORMAT_UNSUPPORTED 0
- #define EMBFORMAT_STITCHONLY 1
- #define EMBFORMAT_OBJECTONLY 2
- #define EMBFORMAT_STCHANDOBJ 3 /∗ binary operation: 1+2=3 ∗/
- #define numberOfFormats 61
- #define CHUNK_SIZE 128
- #define EMB_MAX_LAYERS 10
- #define MAX_THREADS 256
- #define EMBFORMAT_MAXEXT 3
- #define EMBFORMAT_MAXDESC 50
- #define MAX_STITCHES 1000000
- #define EMB_PUBLIC

Typedefs

- typedef float EmbReal
- typedef struct EmbColor_ EmbColor
- typedef struct EmbVector_ EmbVector
- typedef struct EmbArray_ EmbArray
- typedef struct EmbImage_ EmbImage
- typedef struct EmbBlock_ EmbBlock
- typedef struct EmbAlignedDim_ EmbAlignedDim
- typedef struct EmbAngularDim_ EmbAngularDim
- typedef struct EmbArcLengthDim_ EmbArcLengthDim
- typedef struct EmbDiameterDim_ EmbDiameterDim
- typedef struct EmbLeaderDim_ EmbLeaderDim
- typedef struct EmbLinearDim_ EmbLinearDim
- typedef struct EmbOrdinateDim_ EmbOrdinateDim
- typedef struct EmbRadiusDim_ EmbRadiusDim
- typedef struct EmbInfiniteLine_ EmbInfiniteLine
- typedef struct EmbRay_ EmbRay
- typedef struct EmbTextMulti_ EmbTextMulti
- typedef struct EmbTextSingle_ EmbTextSingle
- typedef struct EmbTime_ EmbTime
- typedef struct EmbPoint_ EmbPoint
- typedef struct EmbLine_ EmbLine
- typedef struct EmbPath_ EmbPath
- typedef struct EmbStitch_ EmbStitch
- typedef struct EmbThread_ EmbThread
- typedef struct thread_color_ thread_color
- typedef struct EmbArc_ EmbArc

  absolute position (not relative)
- typedef struct EmbRect_ EmbRect
- typedef struct EmbCircle_ EmbCircle
- typedef EmbPath EmbPolygon

- typedef EmbPath EmbPolyline
- typedef int EmbFlag
- typedef struct EmbSatinOutline_ EmbSatinOutline
- typedef struct EmbEllipse_ EmbEllipse
- typedef struct EmbBezier_ EmbBezier
- typedef struct EmbSpline_ EmbSpline
- typedef struct LSYSTEM L_system
- typedef struct EmbGeometry_ EmbGeometry
- typedef struct EmbLayer_ EmbLayer
- typedef struct EmbPattern_ EmbPattern
- typedef struct EmbFormatList_ EmbFormatList

Functions

- EMB_PUBLIC int lindenmayer_system (L_system L, char ∗state, int iteration, int complete)
- EMB_PUBLIC int hilbert_curve (EmbPattern ∗pattern, int iterations)
- EMB_PUBLIC int emb_identify_format (const char ∗ending)

  fileName
- EMB_PUBLIC void testMain (int level)
- EMB_PUBLIC int convert (const char ∗inf, const char ∗outf)
- EMB_PUBLIC EmbColor embColor_make (unsigned char r, unsigned char g, unsigned char b)
- EMB_PUBLIC EmbColor ∗ embColor_create (unsigned char r, unsigned char g, unsigned char b)
- EMB_PUBLIC EmbColor embColor_fromHexStr (char ∗val)

  Converts a 6 digit hex string (I.E. "00FF00") into an EmbColor and returns it.
- EMB_PUBLIC int embColor_distance (EmbColor a, EmbColor b)

  a b
- EMB_PUBLIC EmbArray ∗ embArray_create (int type)
- EMB_PUBLIC int embArray_resize (EmbArray ∗g)
- EMB_PUBLIC void embArray_copy (EmbArray ∗dst, EmbArray ∗src)
- EMB_PUBLIC int embArray_addArc (EmbArray ∗g, EmbArc arc)
- EMB_PUBLIC int embArray_addCircle (EmbArray ∗g, EmbCircle circle)
- EMB_PUBLIC int embArray_addEllipse (EmbArray ∗g, EmbEllipse ellipse)
- EMB_PUBLIC int embArray_addFlag (EmbArray ∗g, int flag)
- EMB_PUBLIC int embArray_addLine (EmbArray ∗g, EmbLine line)
- EMB_PUBLIC int embArray_addRect (EmbArray ∗g, EmbRect rect)
- EMB_PUBLIC int embArray_addPath (EmbArray ∗g, EmbPath p)
- EMB_PUBLIC int embArray_addPoint (EmbArray ∗g, EmbPoint p)
- EMB_PUBLIC int embArray_addPolygon (EmbArray ∗g, EmbPolygon p)
- EMB_PUBLIC int embArray_addPolyline (EmbArray ∗g, EmbPolyline p)
- EMB_PUBLIC int embArray_addStitch (EmbArray ∗g, EmbStitch st)
- EMB_PUBLIC int embArray_addThread (EmbArray ∗g, EmbThread p)
- EMB_PUBLIC int embArray_addVector (EmbArray ∗g, EmbVector)
- EMB_PUBLIC void embArray_free (EmbArray ∗p)
- EMB_PUBLIC EmbLine embLine_make (EmbReal x1, EmbReal y1, EmbReal x2, EmbReal y2)
- EMB_PUBLIC void embLine_normalVector (EmbLine line, EmbVector ∗result, int clockwise)
- EMB_PUBLIC EmbVector embLine_intersectionPoint (EmbLine line1, EmbLine line2)
- EMB_PUBLIC int embThread_findNearestColor (EmbColor color, EmbColor ∗colors, int n_colors)
- EMB_PUBLIC int embThread_findNearestThread (EmbColor color, EmbThread ∗threads, int n_threads)

  color thread_list n_threads
- EMB_PUBLIC EmbThread embThread_getRandom (void)
- EMB_PUBLIC void embVector_normalize (EmbVector vector, EmbVector ∗result)
- EMB_PUBLIC void embVector_multiply (EmbVector vector, EmbReal magnitude, EmbVector ∗result)
- EMB_PUBLIC EmbVector embVector_add (EmbVector v1, EmbVector v2)
- EMB_PUBLIC EmbVector embVector_average (EmbVector v1, EmbVector v2)

- EMB_PUBLIC EmbVector embVector_subtract (EmbVector v1, EmbVector v2)
- EMB_PUBLIC EmbReal embVector_dot (EmbVector v1, EmbVector v2)
- EMB_PUBLIC EmbReal embVector_cross (EmbVector v1, EmbVector v2)

  The "cross product" as vectors a and b returned as a real value.
- EMB_PUBLIC void embVector_transpose_product (EmbVector v1, EmbVector v2, EmbVector ∗result)
- EMB_PUBLIC EmbReal embVector_length (EmbVector vector)
- EMB_PUBLIC EmbReal embVector_relativeX (EmbVector a1, EmbVector a2, EmbVector a3)
- EMB_PUBLIC EmbReal embVector_relativeY (EmbVector a1, EmbVector a2, EmbVector a3)
- EMB_PUBLIC EmbReal embVector_angle (EmbVector v)
- EMB_PUBLIC EmbReal embVector_distance (EmbVector a, EmbVector b)
- EMB_PUBLIC EmbVector embVector_unit (EmbReal angle)
- EMB_PUBLIC EmbArc embArc_init (void)
- EMB_PUBLIC char embArc_clockwise (EmbArc arc)
- EMB_PUBLIC void getArcCenter (EmbArc arc, EmbVector ∗arcCenter)
- EMB_PUBLIC char getArcDataFromBulge (EmbReal bulge, EmbArc ∗arc, EmbReal ∗arcCenterX, EmbReal ∗arcCenterY, EmbReal ∗radius, EmbReal ∗diameter, EmbReal ∗chord, EmbReal ∗chordMidX, EmbReal ∗chordMidY, EmbReal ∗sagitta, EmbReal ∗apothem, EmbReal ∗incAngleInDegrees, char ∗clockwise)
- EMB_PUBLIC EmbCircle embCircle_init (void)
- EMB_PUBLIC int getCircleCircleIntersections (EmbCircle c0, EmbCircle c1, EmbVector ∗v0, EmbVector ∗v1)
- EMB_PUBLIC int getCircleTangentPoints (EmbCircle c, EmbVector p, EmbVector ∗v0, EmbVector ∗v1)
- EMB_PUBLIC EmbEllipse embEllipse_init (void)
- EMB_PUBLIC EmbEllipse embEllipse_make (EmbReal cx, EmbReal cy, EmbReal rx, EmbReal ry)
- EMB_PUBLIC EmbReal embEllipse_diameterX (EmbEllipse ellipse)
- EMB_PUBLIC EmbReal embEllipse_diameterY (EmbEllipse ellipse)
- EMB_PUBLIC EmbReal embEllipse_width (EmbEllipse ellipse)
- EMB_PUBLIC EmbReal embEllipse_height (EmbEllipse ellipse)
- EMB_PUBLIC EmbReal embEllipse_area (EmbEllipse ellipse)
- EMB_PUBLIC EmbReal embEllipse_perimeter (EmbEllipse ellipse)
- EMB_PUBLIC EmbImage embImage_create (int, int)
- EMB_PUBLIC void embImage_read (EmbImage ∗image, char ∗fname)
- EMB_PUBLIC int embImage_write (EmbImage ∗image, char ∗fname)
- EMB_PUBLIC void embImage_free (EmbImage ∗image)
- EMB_PUBLIC EmbRect embRect_init (void)
- EMB_PUBLIC EmbReal embRect_area (EmbRect)
- EMB_PUBLIC int threadColor (const char ∗, int brand)
- EMB_PUBLIC int threadColorNum (unsigned int color, int brand)
- EMB_PUBLIC const char ∗ threadColorName (unsigned int color, int brand)
- EMB_PUBLIC void embTime_initNow (EmbTime ∗t)

  t
- EMB_PUBLIC EmbTime embTime_time (EmbTime ∗t)

  t
- EMB_PUBLIC void embSatinOutline_generateSatinOutline (EmbArray ∗lines, EmbReal thickness, EmbSatinOutline ∗result)

  lines thickness result
- EMB_PUBLIC EmbArray ∗ embSatinOutline_renderStitches (EmbSatinOutline ∗result, EmbReal density)

  result density
- EMB_PUBLIC EmbGeometry ∗ embGeometry_init (int type_in)

  Our generic object interface backends to each individual type.
- EMB_PUBLIC void embGeometry_free (EmbGeometry ∗obj)

  Free the memory occupied by a non-stitch geometry object.
- EMB_PUBLIC void embGeometry_move (EmbGeometry ∗obj, EmbVector delta)

  Translate obj by the vector delta.
- EMB_PUBLIC EmbRect embGeometry_boundingRect (EmbGeometry ∗obj)

Calculate the bounding box of geometry obj based on what kind of geometric object it is.

- EMB_PUBLIC void embGeometry_vulcanize (EmbGeometry ∗obj)

  Toggle the rubber mode of the object.
- EMB_PUBLIC EmbPattern ∗ embPattern_create (void)
- EMB_PUBLIC void embPattern_hideStitchesOverLength (EmbPattern ∗p, int length)
- EMB_PUBLIC void embPattern_fixColorCount (EmbPattern ∗p)
- EMB_PUBLIC int embPattern_addThread (EmbPattern ∗p, EmbThread thread)
- EMB_PUBLIC void embPattern_addStitchAbs (EmbPattern ∗p, EmbReal x, EmbReal y, int flags, int isAutoColorIndex)
- EMB_PUBLIC void embPattern_addStitchRel (EmbPattern ∗p, EmbReal dx, EmbReal dy, int flags, int isAutoColorIndex)
- 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 threshhold)
- EMB_PUBLIC void embPattern_horizontal_fill (EmbPattern ∗pattern, EmbImage ∗, int threshhold)
- 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_10O `#define EMB_FORMAT_10O 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_copyPolylinesToStitchList ( EmbPattern * pattern )`

18.28.3.68 embPattern_copyStitchListToPolylines() `EMB_PUBLIC void embPattern_copyStitchListToPolylines ( EmbPattern * pattern )`

18.28.3.69 embPattern_correctForMaxStitchLength() `EMB_PUBLIC void embPattern_correctForMaxStitchLength (`

```
EmbPattern * p,
EmbReal maxStitchLength,
EmbReal maxJumpLength )
```

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

18.28.3.70 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 *threshhold*)

pattern image threshhold

Uses a threshhold method to determine where to put crosses in the fill.

To improve this, we can remove the vertical stitches when two crosses neighbour. Currently the simple way to do this is to chain crosses that are neighbours exactly one ahead.

18.28.3.72 embPattern_designDetails() EMB_PUBLIC void embPattern_designDetails (EmbPattern ∗ *p*)

18.28.3.73 embPattern_end() EMB_PUBLIC void embPattern_end (EmbPattern ∗ *p*)

18.28.3.74 embPattern_fixColorCount() EMB_PUBLIC void embPattern_fixColorCount (EmbPattern ∗ *p*)

p

18.28.3.75 embPattern_flip() EMB_PUBLIC void embPattern_flip (EmbPattern ∗ *p*, int *horz*, int *vert*)

Flips the entire pattern (*p*) horizontally about the x-axis if (*horz*) is true. Flips the entire pattern (*p*) vertically about the y-axis if (*vert*) is true.

18.28.3.76 embPattern_flipHorizontal() EMB_PUBLIC void embPattern_flipHorizontal (EmbPattern ∗ *p*)

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

18.28.3.77 embPattern_flipVertical() EMB_PUBLIC void embPattern_flipVertical (EmbPattern ∗ *p*)

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

18.28.3.78 embPattern_free() EMB_PUBLIC void embPattern_free (EmbPattern ∗ *p*)

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

18.28.3.79 embPattern_hideStitchesOverLength() EMB_PUBLIC void embPattern_hideStitchesOver←Length (EmbPattern ∗ *p*, int *length*)

p length

18.28.3.80 embPattern_horizontal_fill() EMB_PUBLIC void embPattern_horizontal_fill (

```
EmbPattern * pattern,
EmbImage * image,
int threshhold )
```

pattern image threshhold

Uses a threshhold 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) x (-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 v1^{T} I_{2} v2 for our vectors *v1* and \v2 so a "component-wise product". The result is stored at the pointer *result*.
That is (1 0) (a) = (xa) (x y)(0 1) (b) (yb)

18.28.3.120 embVector_unit()

```
EMB_PUBLIC EmbVector embVector_unit (
            EmbReal alpha )
```

The unit vector in the direction *angle*.

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

18.28.3.121 full_test_matrix()

```
EMB_PUBLIC int full_test_matrix (
            char * fname )
```

18.28.3.122 getArcCenter()

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

18.28.3.123 getArcDataFromBulge()

```
EMB_PUBLIC char getArcDataFromBulge (
            EmbReal bulge,
            EmbArc * arc,
            EmbReal * arcCenterX,
            EmbReal * arcCenterY,
            EmbReal * radius,
            EmbReal * diameter,
            EmbReal * chord,
            EmbReal * chordMidX,
            EmbReal * chordMidY,
            EmbReal * sagitta,
            EmbReal * apothem,
            EmbReal * incAngleInDegrees,
            char * clockwise )
```

18.28.3.124 getCircleCircleIntersections() `EMB_PUBLIC int getCircleCircleIntersections (`

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

18.28.3.125 getCircleTangentPoints() `EMB_PUBLIC int getCircleTangentPoints (`

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

18.28.3.126 hilbert_curve() `EMB_PUBLIC int hilbert_curve (`

```
EmbPattern * pattern,
int iterations )
```

pattern iterations

`https://en.wikipedia.org/wiki/Hilbert_curve`

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

18.28.3.127 lindenmayer_system() `EMB_PUBLIC int lindenmayer_system (`

```
L_system L,
char * state,
int iterations,
int complete )
```

L state iterations complete

Returns

int

This is a slow generation algorithm.

18.28.3.128 radians() `EMB_PUBLIC EmbReal radians (`

```
EmbReal degree )
```

18.28.3.129 report() `EMB_PUBLIC void report (`

```
int result,
char * label )
```

18.28.3.130 testMain() `EMB_PUBLIC void testMain (`

```
int level )
```

18.28.3.131 threadColor() `EMB_PUBLIC int threadColor (`

```
const char * name,
int brand )
```

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

```
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.

```
#ifndef LIBEMBROIDERY_HEADER__
#define LIBEMBROIDERY_HEADER__

#ifdef __cplusplus
extern "C" {
#endif

#ifndef LIBEMBROIDERY_EMBEDDED_VERSION
#define LIBEMBROIDERY_EMBEDDED_VERSION 0
#endif

/* MACROS
****************************************************************************/

#define NORMAL                  0
#define JUMP                    1
#define TRIM                    2
#define STOP                    4
#define SEQUIN                  8
#define END                     16
#define EMB_FORMAT_100          0
#define EMB_FORMAT_10O          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

/* Thread color */
#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
/* maximum length of extension without dot */
#define EMBFORMAT_MAXDESC            50
/* the longest possible description string length */
#define MAX_STITCHES             1000000



```

```
#if defined(_WIN32) && !defined(WIN32)
#define WIN32
#endif

/* When building a shared library,
* use the proper export keyword depending on the compiler */
#define EMB_PUBLIC
#if defined(LIBEMBROIDERY_SHARED)
#undef EMB_PUBLIC
#if defined(__WIN32__) || defined(WIN32)
#define EMB_PUBLIC __declspec(dllexport)
#else
#define EMB_PUBLIC __attribute__ ((visibility("default")))
#endif
#endif

/* TYPEDEFS AND STRUCTS
*******************************************************************/

typedef float EmbReal;

typedef struct EmbColor_
{
    unsigned char r;
    unsigned char g;
    unsigned char b;
} EmbColor;

typedef struct EmbVector_
{
    EmbReal x;
    EmbReal y;
} EmbVector;

typedef struct EmbArray_ EmbArray;

typedef struct EmbImage_ {
    EmbVector position;
    EmbVector dimensions;
    unsigned char* data;
    int width;
    int height;
    char path[200];
    char name[200];
} EmbImage;

typedef struct EmbBlock_ {
    EmbVector position;
} EmbBlock;

typedef struct EmbAlignedDim_ {
    EmbVector position;
} EmbAlignedDim;

typedef struct EmbAngularDim_ {
    EmbVector position;
} EmbAngularDim;

typedef struct EmbArcLengthDim_ {
    EmbVector position;
} EmbArcLengthDim;

typedef struct EmbDiameterDim_ {
    EmbVector position;
} EmbDiameterDim;

typedef struct EmbLeaderDim_ {
    EmbVector position;
} EmbLeaderDim;

typedef struct EmbLinearDim_ {
    EmbVector position;
} EmbLinearDim;

typedef struct EmbOrdinateDim_ {
    EmbVector position;
} EmbOrdinateDim;

typedef struct EmbRadiusDim_ {
    EmbVector position;
} EmbRadiusDim;

typedef struct EmbInfiniteLine_ {
    EmbVector position;
} EmbInfiniteLine;

typedef struct EmbRay_ {
    EmbVector position;
} EmbRay;

typedef struct EmbTextMulti_ {
    EmbVector position;
    char text[200];
} EmbTextMulti;

typedef struct EmbTextSingle_ {
    EmbVector position;
    char text[200];
} EmbTextSingle;

typedef struct EmbTime_
{
    unsigned int year;
    unsigned int month;
    unsigned int day;
    unsigned int hour;
    unsigned int minute;
    unsigned int second;
} EmbTime;

typedef struct EmbPoint_
{
    EmbVector position;
    int lineType;
    EmbColor color;
} EmbPoint;

typedef struct EmbLine_
{
    EmbVector start;
    EmbVector end;
    int lineType;
    EmbColor color;
} EmbLine;

typedef struct EmbPath_
{
    EmbArray* pointList;
    EmbArray* flagList;
    int lineType;
    EmbColor color;
} EmbPath;

typedef struct EmbStitch_
{
    int flags;
    EmbReal x;
    EmbReal y;
    int color;
} EmbStitch;

typedef struct EmbThread_
{
    EmbColor color;
    char description[50];
    char catalogNumber[30];
} EmbThread;

typedef struct thread_color_ {
    char name[22];
    unsigned int hex_code;
    int manufacturer_code;
} thread_color;

typedef struct EmbArc_
{
    EmbVector start;
    EmbVector mid;
    EmbVector end;
} EmbArc;

typedef struct EmbRect_
{
    EmbReal top;
    EmbReal left;
    EmbReal bottom;
    EmbReal right;
    EmbReal rotation;
    EmbReal radius;
} EmbRect;

typedef struct EmbCircle_
{
    EmbVector center;
    EmbReal radius;
} EmbCircle;

typedef EmbPath EmbPolygon;

typedef EmbPath EmbPolyline;

typedef int EmbFlag;

typedef struct EmbSatinOutline_
{
    int length;
    EmbArray* side1;
    EmbArray* side2;
} EmbSatinOutline;

typedef struct EmbEllipse_
{
    EmbVector center;
    EmbVector radius;
    EmbReal rotation;
} EmbEllipse;

typedef struct EmbBezier_ {
    EmbVector start;
    EmbVector control1;
    EmbVector control2;
    EmbVector end;
} EmbBezier;

typedef struct EmbSpline_ {
    EmbArray *beziers;
} EmbSpline;

typedef struct LSYSTEM {
    char axiom;
    char *alphabet;
    char *constants;
    char **rules;
} L_system;

typedef struct EmbGeometry_ {
    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;
    EmbColor color;
} EmbGeometry;

struct EmbArray_ {
    EmbGeometry *geometry;
    EmbStitch *stitch;
    EmbThread *thread;
    int count;
    int length;
    int type;
};

typedef struct EmbLayer_
{
    char name[100];
    EmbArray *geometry;
} EmbLayer;

typedef struct EmbPattern_
{
    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;
} EmbPattern;

typedef struct EmbFormatList_
{
    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;
} EmbFormatList;

/* Function Declarations
*****************************************************************************/
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);
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);
EMB_PUBLIC int embColor_distance(EmbColor a, EmbColor 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_addSpline(EmbArray* g, EmbSpline 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);
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);
EMB_PUBLIC void embVector_transpose_product(EmbVector v1, EmbVector v2, EmbVector* result);
EMB_PUBLIC EmbReal embVector_length(EmbVector vector);
EMB_PUBLIC EmbReal embVector_relativeX(EmbVector a1, EmbVector a2, EmbVector a3);
EMB_PUBLIC EmbReal embVector_relativeY(EmbVector a1, EmbVector a2, EmbVector a3);
EMB_PUBLIC EmbReal embVector_angle(EmbVector v);
EMB_PUBLIC EmbReal embVector_distance(EmbVector a, EmbVector b);
EMB_PUBLIC EmbVector embVector_unit(EmbReal angle);

EMB_PUBLIC EmbArc embArc_init(void);
EMB_PUBLIC char embArc_clockwise(EmbArc arc);

EMB_PUBLIC void getArcCenter(EmbArc arc, EmbVector *arcCenter);
EMB_PUBLIC char getArcDataFromBulge(EmbReal bulge,
                         EmbArc *arc,
                         EmbReal* arcCenterX,        EmbReal* arcCenterY,
                         EmbReal* radius,            EmbReal* diameter,
                         EmbReal* chord,
                         EmbReal* chordMidX,         EmbReal* chordMidY,
                         EmbReal* sagitta,           EmbReal* apothem,
                         EmbReal* incAngleInDegrees, char*   clockwise);

```

```
EMB_PUBLIC EmbCircle embCircle_init(void);
EMB_PUBLIC int getCircleCircleIntersections(
     EmbCircle c0, EmbCircle c1, EmbVector *v0, EmbVector *v1);
EMB_PUBLIC int getCircleTangentPoints(
     EmbCircle c, EmbVector p, EmbVector *v0, EmbVector *v1);

EMB_PUBLIC EmbEllipse embEllipse_init(void);
EMB_PUBLIC EmbEllipse embEllipse_make(EmbReal cx, EmbReal cy, EmbReal rx, EmbReal ry);
EMB_PUBLIC EmbReal embEllipse_diameterX(EmbEllipse ellipse);
EMB_PUBLIC EmbReal embEllipse_diameterY(EmbEllipse ellipse);
EMB_PUBLIC EmbReal embEllipse_width(EmbEllipse ellipse);
EMB_PUBLIC EmbReal embEllipse_height(EmbEllipse ellipse);
EMB_PUBLIC EmbReal embEllipse_area(EmbEllipse ellipse);
EMB_PUBLIC EmbReal embEllipse_perimeter(EmbEllipse ellipse);

EMB_PUBLIC EmbImage embImage_create(int, int);
EMB_PUBLIC void embImage_read(EmbImage *image, char *fname);
EMB_PUBLIC int embImage_write(EmbImage *image, char *fname);
EMB_PUBLIC void embImage_free(EmbImage *image);

EMB_PUBLIC EmbRect embRect_init(void);
EMB_PUBLIC EmbReal embRect_area(EmbRect);

EMB_PUBLIC int threadColor(const char*, int brand);
EMB_PUBLIC int threadColorNum(unsigned int color, int brand);
EMB_PUBLIC const char* threadColorName(unsigned int color, int brand);

EMB_PUBLIC void embTime_initNow(EmbTime* t);
EMB_PUBLIC EmbTime embTime_time(EmbTime* t);

EMB_PUBLIC void embSatinOutline_generateSatinOutline(EmbArray* lines, EmbReal thickness,
  EmbSatinOutline* result);
EMB_PUBLIC EmbArray* embSatinOutline_renderStitches(EmbSatinOutline* result, EmbReal density);

EMB_PUBLIC EmbGeometry *embGeometry_init(int type_in);
EMB_PUBLIC void embGeometry_free(EmbGeometry *obj);
EMB_PUBLIC void embGeometry_move(EmbGeometry *obj, EmbVector delta);
EMB_PUBLIC EmbRect embGeometry_boundingRect(EmbGeometry *obj);
EMB_PUBLIC void embGeometry_vulcanize(EmbGeometry *obj);

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
  isAutoColorIndex);
EMB_PUBLIC void embPattern_addStitchRel(EmbPattern* p, EmbReal dx, EmbReal dy, int flags, int
  isAutoColorIndex);
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 threshhold);
EMB_PUBLIC void embPattern_horizontal_fill(EmbPattern *pattern, EmbImage *, int threshhold);
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);
EMB_PUBLIC char embPattern_write(EmbPattern *pattern, const char* fileName, int format);

EMB_PUBLIC char embPattern_readAuto(EmbPattern *pattern, const char* fileName);
EMB_PUBLIC char embPattern_writeAuto(EmbPattern *pattern, const char* 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);

/* NON-MACRO CONSTANTS
******************************************************************************/

extern EmbFormatList formatTable[numberOfFormats];
extern const int pecThreadCount;
extern const int shvThreadCount;
extern const EmbReal embConstantPi;
extern const EmbThread husThreads[];
extern const EmbThread jefThreads[];
extern const EmbThread shvThreads[];
extern const EmbThread pcmThreads[];
extern const EmbThread pecThreads[];
extern const unsigned char _dxfColorTable[][3];
extern EmbThread black_thread;
extern const unsigned char vipDecodingTable[];

/* VARIABLES
******************************************************************************/

extern int emb_error;

extern int emb_verbose;

#ifdef __cplusplus
}
#endif /* __cplusplus */

#endif /* LIBEMBROIDERY_HEADER__ */

```

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 0xFFFFFFFA
- #define CompoundFileSector_DIFAT_Sector 0xFFFFFFFC
- #define CompoundFileSector_FAT_Sector 0xFFFFFFFD
- #define CompoundFileSector_EndOfChain 0xFFFFFFFE
- #define CompoundFileSector_FreeSector 0xFFFFFFFF
- #define ObjectTypeUnknown 0x00
- #define ObjectTypeStorage 0x01
- #define ObjectTypeStream 0x02
- #define ObjectTypeRootEntry 0x05
- #define CompoundFileStreamId_MaxRegularStreamId 0xFFFFFFFA
- #define CompoundFileStreamId_NoStream 0xFFFFFFFF
- #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 embInt_read (FILE ∗f, char ∗label, void ∗b, int mode)
- void embInt_write (FILE ∗f, char ∗label, void ∗b, int mode)
- int emb_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 0xFFFFFFFC`

18.30.1.4 CompoundFileSector_EndOfChain `#define CompoundFileSector_EndOfChain 0xFFFFFFFE`

18.30.1.5 CompoundFileSector_FAT_Sector `#define CompoundFileSector_FAT_Sector 0xFFFFFFFD`

18.30.1.6 CompoundFileSector_FreeSector `#define CompoundFileSector_FreeSector 0xFFFFFFFF`

18.30.1.7 CompoundFileSector_MaxRegSector `#define CompoundFileSector_MaxRegSector 0xFFFFFFFA`

Type of sector

18.30.1.8 CompoundFileStreamId_MaxRegularStreamId `#define CompoundFileStreamId_MaxRegularStreamId 0xFFFFFFFA`

Special values for Stream Identifiers All real stream Ids are less than this

18.30.1.9 CompoundFileStreamId_NoStream `#define CompoundFileStreamId_NoStream 0xFFFFFFFF`

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 embInt_read

18.30.4.13 binaryWriteIntBE()

```
void binaryWriteIntBE (
        FILE * f,
        int data )
```

f data

Todo replace with embInt_read

18.30.4.14 binaryWriteShort()

```
void binaryWriteShort (
        FILE * f,
        short data )
```

f data

Todo replace with embInt_read

18.30.4.15 binaryWriteUInt()

```
void binaryWriteUInt (
        FILE * f,
        unsigned int data )
```

f data

Todo replace with embInt_read

18.30.4.16 binaryWriteUIntBE()

```
void binaryWriteUIntBE (
        FILE * f,
        unsigned int data )
```

f data

Todo replace with embInt_read

18.30.4.17 binaryWriteUShort()

```
void binaryWriteUShort (
        FILE * f,
        unsigned short data )
```

f data

Todo replace with embInt_read

18.30.4.18 binaryWriteUShortBE()

```
void binaryWriteUShortBE (
        FILE * f,
        unsigned short data )
```

f data

Todo replace with embInt_read

18.30.4.19 check_header_present()

```
int check_header_present (
        FILE * file,
        int minimum_header_length )
```

file minimum_header_length

Returns

int

Checks that there are enough bytes to interpret the header, stops possible segfaults when reading in the header bytes.
Returns 0 if there aren't enough, or the length of the file if there are.

18.30.4.20 CompoundFileDirectory()

```
bcf_directory * CompoundFileDirectory (
        const unsigned int maxNumberOfDirectoryEntries )
```

maxNumberOfDirectoryEntries

Returns

bcf_directory*

18.30.4.21 CompoundFileDirectoryEntry()

```
bcf_directory_entry * CompoundFileDirectoryEntry (
        FILE * file )
```

file

Returns

bcf_directory_entry*

18.30.4.22 compress_get_bits()

```
int compress_get_bits (
        compress * c,
        int length )
```

c length Returns .

18.30.4.23 compress_get_position()

```
int compress_get_position (
        compress * c )
```

c . Returns the position as an int.

18.30.4.24 compress_get_token()

```
int compress_get_token (
        compress * c )
```

c . Returns the token as an int.

18.30.4.25 compress_load_block()

```
void compress_load_block (
        compress * c )
```

c . Returns nothing.

18.30.4.26 compress_load_character_huffman()

```
void compress_load_character_huffman (
        compress * c )
```

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

18.30.4.27 compress_load_character_length_huffman()

```
void compress_load_character_length_↩
huffman (
        compress * c )
```

c . Returns.

18.30.4.28 compress_load_distance_huffman()

```
void compress_load_distance_huffman (
        compress * c )
```

c . Returns nothing.

18.30.4.29 compress_pop()

```
int compress_pop (
        compress * c,
        int bit_count )
```

c bit_count . Returns.

18.30.4.30 compress_read_variable_length()

```
int compress_read_variable_length (
        compress * c )
```

c. Returns.

18.30.4.31 copy_trim()

```
char * copy_trim (
        char const * s )
```

s

Returns

char∗

Todo decription

18.30.4.32 create_test_file_1()

```
int create_test_file_1 (
        const char * outf )
```

18.30.4.33 create_test_file_2()

```
int create_test_file_2 (
        const char * outf )
```

18.30.4.34 create_test_file_3()

```
int create_test_file_3 (
        const char * outf )
```

18.30.4.35 decode_t01_record()

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

b x y flags .

Todo remove the unused return argument.

18.30.4.36 decode_tajima_ternary()

```
void decode_tajima_ternary (
        unsigned char b[3],
        int * x,
        int * y )
```

Decode the signed ternary of the tajima format from *b* to the position values *x* and *y*.
There is no return argument.

18.30.4.37 decodeNewStitch()

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

value

Returns

int

18.30.4.38 emb_optOut()

```
char * emb_optOut (
        EmbReal num,
        char * str )
```

Optimizes the number (*num*) for output to a text file and returns it as a string (*str*).
num str

Returns

char*

18.30.4.39 emb_readline()

```
int emb_readline (
        FILE * file,
        char * line,
        int maxLength )
```

file line maxLength

Returns

int

18.30.4.40 embColor_read()

```
void embColor_read (
        FILE * f,
        EmbColor * c,
        int toRead )
```

f c toRead

18.30.4.41 embColor_write()

```
void embColor_write (
    FILE * f,
    EmbColor c,
    int toWrite )
```

f c toWrite

18.30.4.42 embInt_read()

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

f label b mode

Read and write system for multiple byte types.

The caller passes the function to read/write from, the memory location as a void pointer and a mode identifier that describes the type. This way we can abstract out the endianness of the system running the library and don't have to maintain many functions, just two.

18.30.4.43 embInt_write()

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

f label b mode

18.30.4.44 encode_t01_record()

```
void encode_t01_record (
    unsigned char b[3],
    int x,
    int y,
    int flags )
```

Encode into bytes *b* the values of the x-position *x*, y-position *y* and the *flags*.

18.30.4.45 encode_tajima_ternary()

```
int encode_tajima_ternary (
    unsigned char b[3],
    int x,
    int y )
```

Encode the signed ternary of the tajima format into *b* the position values *x* and *y*.

If the values of *x* or *y* fall outside of the valid range of -121 and +121 then it returns 0 and 1.

18.30.4.46 entriesInDifatSector()

```
unsigned int entriesInDifatSector (
    bcf_file_difat * fat )
```

fat

Returns

unsigned int

18.30.4.47 fpad()

```
void fpad (
    FILE * file,
    char c,
    int n )
```

f

Returns

int

18.30.4.48 fread_int16()

```
short fread_int16 (
    FILE * f )
```

f

Returns

short

18.30.4.49 fread_int32_be()

```
int fread_int32_be (
    FILE * f )
```

f

Returns

int

Todo replace with embInt_read

18.30.4.50 fread_uint16()

```
unsigned short fread_uint16 (
    FILE * f )
```

f

Returns

unsigned short

Todo replace with embInt_read

18.30.4.51 GetFile()

```
FILE * GetFile (
    bcf_file * bcfFile,
    FILE * file,
    char * fileToFind )
```

Get the File object.
bcfFile file fileToFind

Returns

FILE∗

18.30.4.52 huffman_build_table()

```
void huffman_build_table (
    huffman * h )
```

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

18.30.4.53 huffman_table_lookup()

```
int * huffman_table_lookup (
    huffman * h,
    int byte_lookup,
    int * lengths )
```

18.30.4.54 hus_compress()

```
int hus_compress (
    char * data,
    int length,
    char * output,
    int * output_length )
```

data length output output_length . Returns whether it was successful as an int.
This avoids the now unnecessary compression by placing a minimal header of 6 bytes and using only literals in the huffman compressed part (see the sources above).

18.30.4.55 hus_decompress()

```
int hus_decompress (
        char * data,
        int length,
        char * output,
        int * output_length )
```

data length output output_length . Returns whether the decompression was successful.

18.30.4.56 loadFatFromSector()

```
void loadFatFromSector (
        bcf_file_fat * fat,
        FILE * file )
```

fat file

18.30.4.57 mitDecodeStitch()

```
int mitDecodeStitch (
        unsigned char value )
```

value

Returns

int

18.30.4.58 mitEncodeStitch()

```
unsigned char mitEncodeStitch (
        EmbReal value )
```

value

Returns

unsigned char

18.30.4.59 numberOfEntriesInDifatSector()

```
unsigned int numberOfEntriesInDifatSector (
        bcf_file_difat * fat )
```

18.30.4.60 pfaffDecode()

```
EmbReal pfaffDecode (
        unsigned char a1,
        unsigned char a2,
        unsigned char a3 )
```

Decode the bytes *a1*, *a2* and *a3* . Returns the EmbReal floating-point value.

18.30.4.61 pfaffEncode()

```
void pfaffEncode (
        FILE * file,
        int dx,
        int dy,
        int flags )
```

file dx dy flags

18.30.4.62 printArcResults()

```
void printArcResults (
        EmbReal bulge,
        EmbArc arc,
        EmbReal centerX,
        EmbReal centerY,
        EmbReal radius,
        EmbReal diameter,
        EmbReal chord,
        EmbReal chordMidX,
        EmbReal chordMidY,
        EmbReal sagitta,
        EmbReal apothem,
        EmbReal incAngle,
        char clockwise )
```

18.30.4.63 read100()

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

18.30.4.64 read10o()

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

18.30.4.65 readArt()

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

18.30.4.66 readBmc()

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

18.30.4.67 readBro()

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

18.30.4.68 readCnd()

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

18.30.4.69 readCol()

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

18.30.4.70 readCsd()

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

18.30.4.71 readCsv()

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

18.30.4.72 readDat()

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

18.30.4.73 readDem()

```
char readDem (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.74 readDescriptions()

```
void readDescriptions (
    FILE * file,
    EmbPattern * pattern )
```

18.30.4.75 readDsb()

```
char readDsb (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.76 readDst()

```
char readDst (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.77 readDsz()

```
char readDsz (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.78 ZSK USA Embroidery Format (.dsz)

The ZSK USA dsz format is stitch-only.

18.30.4.79 readDxf()

```
char readDxf (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.80 readEdr()

```
char readEdr (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.81 Embird Embroidery Format (.edr)

Stitch Only Format

18.30.4.82 readEmd()

```
char readEmd (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.83 readExp()

```
char readExp (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.84 readExy()

```
char readExy (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.85 readEys()

```
char readEys (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.86 Sierra Expanded Embroidery Format (.eys) Stitch Only Format.

Smoothie G-Code Embroidery Format (.fxy)?

18.30.4.87 readFeatherPatterns()

```
void readFeatherPatterns (
    FILE * file,
    EmbPattern * pattern )
```

18.30.4.88 readFullSector()

```
unsigned int readFullSector (
    FILE * file,
    bcf_file_difat * bcfFile,
    unsigned int * difatEntriesToRead )
```

file bcfFile difatEntriesToRead

Returns

unsigned int

18.30.4.89 readFxy()

```
char readFxy (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.90 Embroidery Format (.fxy) Stitch Only Format.

18.30.4.91 readGc()

```
char readGc (
    EmbPattern * pattern,
    FILE * file )
```

Smoothie G-Code
Main Reference: Machinery's Handbook Guide A Guide to Tables, Formulas, & More in the 31st Edition by John Milton Amiss, Franklin D. Jones and Henry Ryffel

18.30.4.92 readGnc()

```
char readGnc (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.93 Great Notions Embroidery Format (.gnc) Stitch Only Format.

18.30.4.94 readGt()

```
char readGt (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.95 Gold Thread Embroidery Format (.gt) Stitch Only Format.

18.30.4.96 readHoopName()

```
void readHoopName (
    FILE * file,
    EmbPattern * pattern )
```

18.30.4.97 readHus()

```
char readHus (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.98 readImageString()

```
void readImageString (
    FILE * file,
    EmbPattern * pattern )
```

18.30.4.99 readInb()

```
char readInb (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.100 Inbro Embroidery Format (.inb)

Stitch Only Format.

18.30.4.101 readInf()

```
char readInf (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.102 Embroidery Color Format (.inf)

Stitch Only Format.

18.30.4.103 readJef()

```
char readJef (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.104 readKsm()

```
char readKsm (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.105 readMax()

```
char readMax (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.106 readMit()

```
char readMit (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.107 Mitsubishi Embroidery Format (.mit)

Stitch Only Format.

18.30.4.108 readMotifPatterns()

```
void readMotifPatterns (
    FILE * file,
    EmbPattern * pattern )
```

18.30.4.109 readNew()

```
char readNew (
    EmbPattern * pattern,
    FILE * file )
```

18.30.4.110 Ameco Embroidery Format (.new)

Stitch Only Format.

18.30.4.111 readNextSector()

```
void readNextSector (
    FILE * file,
    bcf_directory * dir )
```

file dir

18.30.4.112 readOfm()

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

18.30.4.113 readPcd()

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

18.30.4.114 Pfaff PCD File Format (.pcd)

Stitch Only Format.
The format uses a signed 3 byte-length number type.
See the description here ([5](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 (.pcq)

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 (.pec)

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.

```
#ifndef LIBEMBROIDERY_INTERNAL_HEADER__
#define LIBEMBROIDERY_INTERNAL_HEADER__

#include "embroidery.h"

/* For FILE * */
#include <stdio.h>

#define CompoundFileSector_MaxRegSector 0xFFFFFFFA
#define CompoundFileSector_DIFAT_Sector 0xFFFFFFFC
#define CompoundFileSector_FAT_Sector   0xFFFFFFFD
#define CompoundFileSector_EndOfChain   0xFFFFFFFE
#define CompoundFileSector_FreeSector   0xFFFFFFFF

#define ObjectTypeUnknown   0x00
#define ObjectTypeStorage   0x01
#define ObjectTypeStream    0x02
#define ObjectTypeRootEntry 0x05
#define CompoundFileStreamId_MaxRegularStreamId 0xFFFFFFFA
#define CompoundFileStreamId_NoStream           0xFFFFFFFF
#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

/* INTERNAL DEFINES */
#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))

/* Libembroidery’s handling of integer types.
*/
#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

/* DXF Version Identifiers */
#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

/*  SVG_TYPES
*  ---------
*/
#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

/* path flag codes */
#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

/* STRUCTS
********************************************************************/

/* double-indirection file allocation table references */

typedef struct _bcf_file_difat
{
    unsigned int fatSectorCount;
    unsigned int fatSectorEntries[109];
    unsigned int sectorSize;
} bcf_file_difat;

typedef struct _bcf_file_fat
{
    int          fatEntryCount;
    unsigned int fatEntries[255]; /* maybe make this dynamic */
    unsigned int numberOfEntriesInFatSector;
} bcf_file_fat;

typedef struct _bcf_directory_entry
{
    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; /* should be long long but in our case we shouldn’t need it, and hard to support on c89 cross platform */
    unsigned int                 streamSizeHigh; /* store the high int of streamsize */
    struct _bcf_directory_entry* next;
} bcf_directory_entry;

typedef struct _bcf_directory
{
    bcf_directory_entry* dirEntries;
    unsigned int         maxNumberOfDirectoryEntries;
} bcf_directory;

typedef struct _bcf_file_header
{
    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;
} bcf_file_header;

typedef struct _bcf_file
{
    bcf_file_header header;
    bcf_file_difat* difat;
    bcf_file_fat* fat;
    bcf_directory* directory;
} bcf_file;

typedef struct _vp3Hoop
{
    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;

    /* Centered hoop dimensions */
    int right2;
    int left2;
    int bottom2;
    int top2;

    int width;
    int height;
} vp3Hoop;

typedef struct ThredHeader_     /* thred file header */
{
    unsigned int sigVersion;    /* signature and version */
    unsigned int length;        /* length of ThredHeader + length of stitch data */
    unsigned short numStiches;  /* number of stitches */
    unsigned short hoopSize;    /* size of hoop */
    unsigned short reserved[7]; /* reserved for expansion */
} ThredHeader;

typedef struct ThredExtension_  /* thred v1.0 file header extension */
{
    float hoopX;                /* hoop size x dimension in 1/6 mm units */
    float hoopY;                /* hoop size y dimension in 1/6 mm units */
    float stitchGranularity;    /* stitches per millimeter--not implemented */
    char creatorName[50];       /* name of the file creator */
    char modifierName[50];      /* name of last file modifier */
    char auxFormat;             /* auxiliary file format, 0=PCS,1=DST,2=PES */
    char reserved[31];          /* reserved for expansion */
} ThredExtension;

typedef struct SubDescriptor_
{
    int someNum;
    int someInt;
    int someOtherInt;
    char* colorCode;
    char* colorName;
} SubDescriptor;

typedef struct StxThread_
{
    char* colorCode;
    char* colorName;
    char* sectionName;
    SubDescriptor* subDescriptors;
    EmbColor stxColor;
} StxThread;

typedef struct VipHeader_ {
    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;
} VipHeader;

typedef enum
{
    CSV_EXPECT_NULL,
    CSV_EXPECT_QUOTE1,
    CSV_EXPECT_QUOTE2,
    CSV_EXPECT_COMMA
} CSV_EXPECT;

typedef enum
{
    CSV_MODE_NULL,
    CSV_MODE_COMMENT,
    CSV_MODE_VARIABLE,
    CSV_MODE_THREAD,
    CSV_MODE_STITCH
} CSV_MODE;

typedef struct SvgAttribute_
{
    char* name;
    char* value;
} SvgAttribute;

typedef struct Huffman {
    int default_value;
    int lengths[1000];
    int nlengths;
    int table[1000];
    int table_width;
    int ntable;
} huffman;

typedef struct Compress {
    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;
} compress;

/* Function Declarations
****************************************************************************/
```

```
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);

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);
int mitDecodeStitch(unsigned char 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 embInt_read(FILE* f, char *label, void *b, int mode);
void embInt_write(FILE* f, char *label, void *b, int mode);
int emb_readline(FILE* file, char *line, int maxLength);

int bcfFile_read(FILE* file, bcf_file* bcfFile);
FILE* GetFile(bcf_file* bcfFile, FILE* file, char* fileToFind);
void bcf_file_free(bcf_file* bcfFile);

void binaryReadString(FILE* file, char *buffer, int maxLength);
void binaryReadUnicodeString(FILE* file, char *buffer, const int stringLength);

int stringInArray(const char *s, const char **array);
void fpad(FILE *f, char c, int n);
char *copy_trim(char const *s);
char* emb_optOut(EmbReal num, char* str);

void write_24bit(FILE* file, int);
int check_header_present(FILE* file, int minimum_header_length);

unsigned short fread_uint16(FILE *file);
short fread_int16(FILE* f);
int fread_int32_be(FILE* f);
void safe_free(void *data);
void embInt_read(FILE* f, char *label, void *b, int mode);

void binaryWriteUIntBE(FILE* f, unsigned int data);
void binaryWriteUInt(FILE* f, unsigned int data);
void binaryWriteIntBE(FILE* f, int data);
void binaryWriteInt(FILE* f, int data);
void binaryWriteUShort(FILE* f, unsigned short data);
void binaryWriteUShortBE(FILE* f, unsigned short data);
void binaryWriteShort(FILE* f, short data);

bcf_file_difat* bcf_difat_create(FILE* file, unsigned int fatSectors, const unsigned int sectorSize);
unsigned int readFullSector(FILE* file, bcf_file_difat* bcfFile, unsigned int*
  numberOfDifatEntriesStillToRead);
unsigned int numberOfEntriesInDifatSector(bcf_file_difat* fat);
void bcf_file_difat_free(bcf_file_difat* difat);

unsigned int entriesInDifatSector(bcf_file_difat* fat);
bcf_file_fat* bcfFileFat_create(const unsigned int sectorSize);
void loadFatFromSector(bcf_file_fat* fat, FILE* file);
void bcf_file_fat_free(bcf_file_fat** fat);
```

```c

bcf_directory_entry* CompoundFileDirectoryEntry(FILE* file);
bcf_directory* CompoundFileDirectory(const unsigned int maxNumberOfDirectoryEntries);
void readNextSector(FILE* file, bcf_directory* dir);
void bcf_directory_free(bcf_directory** dir);

bcf_file_header bcfFileHeader_read(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);

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 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);
void embColor_write(FILE *f, EmbColor c, int 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);

extern const char imageWithFrame[38][48];

#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 embInt_read (FILE ∗f, char ∗label, void ∗b, int mode)
- void embInt_write (FILE ∗f, char ∗label, void ∗b, int mode)

18.32.1 Detailed Description

The functions in this file are grouped together to aid the developer's understanding of the similarities between the file formats. This also helps reduce errors between reimplementation of the same idea.
For example: the Tajima ternary encoding of positions is used by at least 4 formats and the only part that changes is the flag encoding.

18.32.2 Function Documentation

18.32.2.1 decode_t01_record()

```
int decode_t01_record (
        unsigned char b[3],
        int ∗ x,
        int ∗ y,
        int ∗ flags )
```

b x y flags .

Todo remove the unused return argument.

18.32.2.2 decode_tajima_ternary()

```
void decode_tajima_ternary (
        unsigned char b[3],
        int ∗ x,
        int ∗ y )
```

Decode the signed ternary of the tajima format from *b* to the position values *x* and *y*.
There is no return argument.

18.32.2.3 decodeNewStitch()

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

value

Returns

int

18.32.2.4 embColor_fromHexStr() `EmbColor embColor_fromHexStr ( char * val )`

Converts a 6 digit hex string (I.E. "00FF00") into an EmbColor and returns it.
val 6 byte code describing the color as a hex string, doesn't require null termination.

Returns

EmbColor the same color as our internal type.

18.32.2.5 embInt_read()

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

f label b mode
Read and write system for multiple byte types.
The caller passes the function to read/write from, the memory location as a void pointer and a mode identifier that describes the type. This way we can abstract out the endianness of the system running the library and don't have to maintain many functions, just two.

18.32.2.6 embInt_write()

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

f label b mode

18.32.2.7 encode_t01_record()

```
void encode_t01_record (
        unsigned char b[3],
        int x,
        int y,
        int flags )
```

Encode into bytes *b* the values of the x-position *x*, y-position *y* and the *flags*.

18.32.2.8 encode_tajima_ternary()

```
int encode_tajima_ternary (
        unsigned char b[3],
        int x,
        int y )
```

Encode the signed ternary of the tajima format into *b* the position values *x* and *y*.
If the values of *x* or *y* fall outside of the valid range of -121 and +121 then it returns 0 and 1.

18.32.2.9 mitDecodeStitch()

```
int mitDecodeStitch (
        unsigned char value )
```

value

Returns

int

18.32.2.10 mitEncodeStitch()

```
unsigned char mitEncodeStitch (
        EmbReal value )
```

value

Returns

unsigned char

18.32.2.11 pfaffDecode()

```
EmbReal pfaffDecode (
        unsigned char a1,
        unsigned char a2,
        unsigned char a3 )
```

Decode the bytes *a1*, *a2* and *a3* . Returns the EmbReal floating-point value.

18.32.2.12 pfaffEncode()

```
void pfaffEncode (
        FILE * file,
        int dx,
        int dy,
        int flags )
```

file dx dy flags

18.32.2.13 reverse_byte_order()

```
void reverse_byte_order (
        void * b,
        int bytes )
```

Reverses the byte order of *bytes* number of bytes at memory location *b*. Only works for 2 or 4 byte arrays.

18.32.2.14 write_24bit()

```
void write_24bit (
        FILE * file,
        int x )
```

file x

18.33 extern/libembroidery/src/fill.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "embroidery_internal.h"
```

Functions

- int lindenmayer_system (L_system L, char ∗state, int iterations, int complete)
- static void join_short_stitches (int ∗points, int ∗n_points, int width, int tolerence)
- 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 threshhold)
- void embPattern_crossstitch (EmbPattern ∗pattern, EmbImage ∗image, int threshhold)
- 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 threshhold )
```

pattern image threshhold
Uses a threshhold 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 threshhold )
```

pattern image threshhold
Uses a threshhold 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) x (-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 seperation s can be found by taking the point on the line to be c+sn where the n is the unit normal vector to d and the vector to be d again. The intersection points are therefore a right angled triangle, with one side r, another s and the third the length to be solved, by Pythagoras we have:

$(c + sn - \sqrt{r^2-s^2}(d/|d|), c + sn + \sqrt{r^2-s^2}(d/|d|))$

repeating this process gives us all the end points and the fill only alters these lines by splitting the ones longer than some tolerence.

18.33.1.8 embPattern_stitchEllipse()

```
void embPattern_stitchEllipse (
            EmbPattern * p,
            EmbEllipse ellipse,
            int thread_index,
            int style )
```

p ellipse thread_index style

Todo finish stitchEllipse

18.33.1.9 embPattern_stitchPath()

```
void embPattern_stitchPath (
            EmbPattern * p,
            EmbPath path,
            int thread_index,
            int style )
```

p rect thread_index style

Todo finish stitch path

18.33.1.10 embPattern_stitchPolygon()

```
void embPattern_stitchPolygon (
            EmbPattern * p,
            EmbPolygon polygon,
            int thread_index,
            int style )
```

p rect thread_index style

Todo finish stitch polygon

18.33.1.11 embPattern_stitchPolyline()

```
void embPattern_stitchPolyline (
            EmbPattern * p,
            EmbPolyline polyline,
            int thread_index,
            int style )
```

p rect thread_index style

Todo finish stitch polyline

18.33.1.12 embPattern_stitchRect()

```
void embPattern_stitchRect (
            EmbPattern * p,
            EmbRect rect,
            int thread_index,
            int style )
```

p rect thread_index style

Here we just stitch the rectangle in the direction of it's longer side.

18.33.1.13 embPattern_stitchText()

```
void embPattern_stitchText (
            EmbPattern * p,
            EmbRect rect,
            int thread_index,
            int style )
```

p rect thread_index style

18.33.1.14 embPolygon_reduceByDistance()

```
void embPolygon_reduceByDistance (
            EmbArray * vertices,
            EmbArray * simplified,
            float distance )
```

vertices simplified distance

Reduces the polygon by distance.

This is a non-destructive function, so the caller is responsible for freeing "vertices" if they choose to keep "simplified".

18.33.1.15 embPolygon_reduceByNth()

```
void embPolygon_reduceByNth (
            EmbArray * vertices,
            EmbArray * out,
            int nth )
```

vertices out nth

Reduces the polygon by removing the Nth vertex in the vertices list. This is a non-destructive function, so the caller is responsible for freeing vertices if they choose to keep out.

18.33.1.16 generate_dragon_curve()

```
void generate_dragon_curve (
            char * state,
            int iterations )
```

state iterations

using the "paper folding" method

Todo find citation for paper folding method

18.33.1.17 greedy_algorithm()

```
static void greedy_algorithm (
            int * points,
            int n_points,
            int width,
            EmbReal bias )  [static]
```

points n_points width bias

18.33.1.18 Greedy Algorithm

For each point in the list find the shortest distance to any possible neighbour, then perform a swap to make that neighbour the next item in the list.

To make the stitches lie more on one axis than the other bias the distance operator to prefer horizontal direction.

18.33.1.19 hilbert_curve() `int hilbert_curve (`

```
EmbPattern * pattern,
int iterations )
```

pattern iterations

https://en.wikipedia.org/wiki/Hilbert_curve

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

18.33.1.20 join_short_stitches() `static void join_short_stitches (`

```
int * points,
int * n_points,
int width,
int tolerence )  [static]
```

points n_points width tolerence

Remove points that lie in the middle of two short stitches that could be one longer stitch. Repeat until none are found.

18.33.1.21 lindenmayer_system() `int lindenmayer_system (`

```
L_system L,
char * state,
int iterations,
int complete )
```

L state iterations complete

Returns

int

This is a slow generation algorithm.

18.33.1.22 save_points_to_pattern() `static void save_points_to_pattern (`

```
EmbPattern * pattern,
int * points,
int n_points,
EmbReal scale,
int width,
int height )  [static]
```

pattern points n_points scale width height

18.33.1.23 threshold_method() `static int * threshold_method (`

```
EmbImage * image,
int * n_points,
int subsample_width,
int subsample_height,
int threshold )  [static]
```

image n_points subsample_width subsample_height threshold

Returns

int*

Identify darker pixels to put stitches in.

18.33.2 Variable Documentation

18.33.2.1 hilbert_curve_l_system `L_system hilbert_curve_l_system`

Initial value:

```
= {
    'A', "AB", "F+-", (char**)rules
}
```

18.33.2.2 rules `const char* rules[] = {"+BF-AFA-FB+", "-AF+BFB+FA-"}`

18.34 extern/libembroidery/src/formats.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <ctype.h>
#include "embroidery_internal.h"
```

Functions

- void safe_free (void ∗data)

  data
- int embFormat_getExtension (const char ∗fileName, char ∗ending)

  fileName ending
- int emb_identify_format (const char ∗fileName)

  fileName
- short fread_int16 (FILE ∗f)

  f
- unsigned short fread_uint16 (FILE ∗f)

  f
- int fread_int32_be (FILE ∗f)

  f
- void fpad (FILE ∗file, char c, int n)

  f
- void binaryWriteShort (FILE ∗f, short data)

  f data
- void binaryWriteUShort (FILE ∗f, unsigned short data)

  f data
- void binaryWriteUShortBE (FILE ∗f, unsigned short data)

  f data
- void binaryWriteInt (FILE ∗f, int data)

  f data
- void binaryWriteIntBE (FILE ∗f, int data)

  f data
- void binaryWriteUInt (FILE ∗f, unsigned int data)

  f data
- void binaryWriteUIntBE (FILE ∗f, unsigned int data)

  f data
- char embPattern_read (EmbPattern ∗pattern, const char ∗fileName, int format)

  pattern fileName format
- char embPattern_write (EmbPattern ∗pattern, const char ∗fileName, int format)

  pattern fileName format
- char embPattern_readAuto (EmbPattern ∗pattern, const char ∗fileName)

  pattern fileName
- char embPattern_writeAuto (EmbPattern ∗pattern, const char ∗fileName)

  pattern fileName

Variables

- EmbFormatList formatTable [numberOfFormats]
- const char imageWithFrame [38][48]

18.34.1 Function Documentation

18.34.1.1 binaryWriteInt()

```
void binaryWriteInt (
        FILE * f,
        int data )
```

f data

Todo replace with embInt_read

18.34.1.2 binaryWriteIntBE()

```
void binaryWriteIntBE (
        FILE * f,
        int data )
```

f data

Todo replace with embInt_read

18.34.1.3 binaryWriteShort()

```
void binaryWriteShort (
        FILE * f,
        short data )
```

f data

Todo replace with embInt_read

18.34.1.4 binaryWriteUInt()

```
void binaryWriteUInt (
        FILE * f,
        unsigned int data )
```

f data

Todo replace with embInt_read

18.34.1.5 binaryWriteUIntBE()

```
void binaryWriteUIntBE (
        FILE * f,
        unsigned int data )
```

f data

Todo replace with embInt_read

18.34.1.6 binaryWriteUShort()

```
void binaryWriteUShort (
        FILE * f,
        unsigned short data )
```

f data

Todo replace with embInt_read

18.34.1.7 binaryWriteUShortBE()

```
void binaryWriteUShortBE (
        FILE * f,
        unsigned short data )
```

f data

Todo replace with embInt_read

18.34.1.8 emb_identify_format()

```
int emb_identify_format (
        const char * fileName )
```

fileName

Returns

int

18.34.1.9 embFormat_getExtension()

```
int embFormat_getExtension (
        const char * fileName,
        char * ending )
```

fileName ending

Returns

int

18.34.1.10 embPattern_read()

```
char embPattern_read (
        EmbPattern * pattern,
        const char * fileName,
        int format )
```

pattern fileName format

Returns

char

18.34.1.11 embPattern_readAuto()

```
char embPattern_readAuto (
        EmbPattern * pattern,
        const char * fileName )
```

pattern fileName

Returns

char

18.34.1.12 embPattern_write()

```
char embPattern_write (
        EmbPattern * pattern,
        const char * fileName,
        int format )
```

pattern fileName format

Returns

char

18.34.1.13 embPattern_writeAuto()

```
char embPattern_writeAuto (
        EmbPattern * pattern,
        const char * fileName )
```

pattern fileName

Returns

char

18.34.1.14 fpad()

```
void fpad (
        FILE * file,
        char c,
        int n )
```

f

Returns

int

18.34.1.15 fread_int16()

```
short fread_int16 (
        FILE * f )
```

f

Returns

short

18.34.1.16 fread_int32_be()

```
int fread_int32_be (
        FILE * f )
```

f

Returns

int

Todo replace with 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 `\textbackslash{}r\textbackslash{}n}.
Then the rest of the file is a comma seperated value list of all threads with 4 values per line: the index of the thread then the red, green and blue channels of the color in that order.

18.41.1.0.1 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 Seperated Values files aren't a universal system, here we aim to offer a broad support. The dialect is detected based on the opening lines, as each manufacturer should label their CSV files there.

18.43.1.0.1 Embroidermodder 2.0 CSV Dialect Our own version has the identifier comment line:
| Control Symbol | Type | Description | |—|—| | # | `COMMENT` | | | > | `VARIABLE` | To store records of a pattern's width, height etc. This means that data stored in the header of say a .dst file is preserved. | | $ | `THREAD` | | | ∗ | `STITCH` | | | ∗ | `JUMP` | | | ∗ | `COLOR` | To change a color: used for trim as well | | ∗ | `END` | To end a pattern. | | ∗ | `UNKNOWN` | For any feature that we can't identify.

18.43.1.0.2 EmBird CSV Dialect

18.43.2 Function Documentation

18.43.2.1 csvStitchFlagToStr() `char ∗ csvStitchFlagToStr (`
`int flags )`

18.43.2.2 csvStrToStitchFlag() `int csvStrToStitchFlag (`
`const char ∗ str )`

18.43.2.3 readCsv() `char readCsv (`
`EmbPattern ∗ pattern,`
`FILE ∗ file )`

18.43.2.4 writeCsv() `char writeCsv (`
`EmbPattern ∗ pattern,`
`FILE ∗ file )`

18.44 extern/libembroidery/src/formats/format_dat.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char readDat (EmbPattern ∗pattern, FILE ∗file)
- char writeDat (EmbPattern ∗pattern, FILE ∗file)

18.44.1 Function Documentation

18.44.1.1 readDat() `char readDat (`
`EmbPattern ∗ pattern,`
`FILE ∗ file )`

18.44.1.2 writeDat()

```
char writeDat (
        EmbPattern * pattern,
        FILE * file )
```

18.45 extern/libembroidery/src/formats/format_dem.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char readDem (EmbPattern ∗pattern, FILE ∗file)
- char writeDem (EmbPattern ∗pattern, FILE ∗file)

18.45.1 Detailed Description

The Melco Embroidery Format (.dem)
Stitch Only Format

18.45.2 Function Documentation

18.45.2.1 readDem()

```
char readDem (
        EmbPattern * pattern,
        FILE * file )
```

18.45.2.2 writeDem()

```
char writeDem (
        EmbPattern * pattern,
        FILE * file )
```

18.46 extern/libembroidery/src/formats/format_dsb.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char readDsb (EmbPattern ∗pattern, FILE ∗file)
- char writeDsb (EmbPattern ∗pattern, FILE ∗file)

18.46.1 Detailed Description

The Barudan Embroidery Format (.dsb)

- Stitch Only Format.
- [X] Basic Read Support
- [o] Basic Write Support
- [o] Well Tested Read
- [o] Well Tested Write

18.46.2 Function Documentation

18.46.2.1 readDsb()

```
char readDsb (
        EmbPattern * pattern,
        FILE * file )
```

18.46.2.2 writeDsb()

```
char writeDsb (
        EmbPattern * pattern,
        FILE * file )
```

18.47 extern/libembroidery/src/formats/format_dst.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Macros

- #define cci(c1, c2) (c1∗256+c2)

Functions

- int decode_record_flags (unsigned char b2)
- void encode_record (FILE ∗file, int x, int y, int flags)
- void set_dst_variable (EmbPattern ∗pattern, char ∗var, char ∗val)
- char readDst (EmbPattern ∗pattern, FILE ∗file)
- char writeDst (EmbPattern ∗pattern, FILE ∗file)

18.47.1 Detailed Description

.DST (Tajima) embroidery file read/write routines Format comments are thanks to tspilman@dalcoathletic.com who's notes appeared at http://www.wotsit.org under Tajima Format.

18.47.1.1 Tajima Embroidery Format (.dst)

Stitch Only Format. [X] Basic Read Support [X] Basic Write Support [ ] Well Tested Read [ ] Well Tested Write
.DST (Tajima) embroidery file read/write routines Format comments are thanks to tspilman@dalcoathletic.com who's notes appeared at http://www.wotsit.org under Tajima Format.
Other references: [2] , [1] .

18.47.1.1.1 Header

The header contains general information about the design. It is in lines of ASCII, so if you open a DST file as a text file, it's the only part that's easy to read. The line ending symbol is `0x0D}. The header is necessary for the file to be read by most softwares and hardwares.
The header is 125 bytes of data followed by padding spaces to make it 512 bytes in total.
The lines are as follows.

Label	*Size*	*Description*	*Example*
`LA:`	17	The design name with no path or extension. The space reserved is 16 characters, but the name must not be longer than 8 and be padded to 16 with spaces (0x20).	`"LA:Star "`
`ST:`	8	The stitch count. An integer in the format `%07d`, that is: a 7 digit number padded by leading zeros. This is the total accross all possible stitch flags.	

Label	***Size***	***Description***	***Example***
`CO:`	4	The number of color changes (not to be confused with thread count, an all black design we would have the record \textbf{000}). An integer in the format `%03d`, that is: a 3 digit number padded by leading zeros.	
`+X:`	6	The extent of the pattern in the postitive x direction in millimeters. An integer in the format `%05d`, that is: a 5 digit number padded by leading zeros.	
`-X:`	6	The extent of the pattern in the negative x direction in millimeters. An integer in the format `%05d`, that is: a 5 digit integer padded by leading zeros.	
`+Y:`	6	The extent of the pattern in the postitive y direction in millimeters. An integer in the format `%05d`, that is: a 5 digit integer padded by leading zeros.	
`-Y:`	6	The extent of the pattern in the negative y direction in millimeters. An integer in the format `%05d`, that is: a 5 digit integer padded by leading zeros.	
`AX:`	7	The difference of the end from the start in the x direction in 0.1mm, the first char should be the sign, followed by an integer in the format `%05d`, that is: a 5 digit integer padded by leading zeros.	
`AY:`	7	The difference of the end from the start in the y direction in 0.1mm, the first char should be the sign, followed by an integer in the format `%05d`, that is: a 5 digit integer padded by leading zeros.	
`MX:`	7	The x co-ordinate of the last point in the previous file should the design span multiple files. Like AX, it is the sign, followed by a 5 digit integer. If we have a one file design set it to zero.	
`MY:`	7	The y co-ordinate of the last point in the previous file should the design span multiple files. Like AY, it is the sign, followed by a 5 digit integer. If we have a one file design set it to zero.	
`PD:`	10	Information about multivolume designs.	

18.47.1.1.2 Stitch Data Uses 3 byte per stitch encoding with the format as follows:

Bit	***7***	***6***	***5***	***4***	***3***	***2***	***1***	***0***
Byte 0	y+1	y-1	y+9	y-9	x-9	x+9	x-1	x+1
Byte 1	y+3	y-3	y+27	y-27	x-27	x+27	x-3	x+3
Byte 2	jump	color change	y+81	y-81	x-81	x+81	set	set

T01 and Tap appear to use Tajima Ternary.
Where the stitch type is determined as:
Normal Stitch `0b00000011 0x03` Jump Stitch `0b10000011 0x83` Stop/Change Color `0b11000011 0xC3` 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 AudoDesk 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_eys.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char readEys (EmbPattern ∗pattern, FILE ∗file)
- char writeEys (EmbPattern ∗pattern, FILE ∗file)

18.54.1 Function Documentation

18.54.1.1 readEys()

```
char readEys (
        EmbPattern * pattern,
        FILE * file )
```

18.54.1.2 Sierra Expanded Embroidery Format (.eys)

Stitch Only Format. Smoothie G-Code Embroidery Format (.fxy)?

18.54.1.3 writeEys()

```
char writeEys (
        EmbPattern * pattern,
        FILE * file )
```

18.55 extern/libembroidery/src/formats/format_fxy.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char readFxy (EmbPattern *pattern, FILE *file)
- char writeFxy (EmbPattern *pattern, FILE *file)

18.55.1 Function Documentation

18.55.1.1 readFxy()

```
char readFxy (
        EmbPattern * pattern,
        FILE * file )
```

18.55.1.2 Embroidery Format (.fxy)

Stitch Only Format.

18.55.1.3 writeFxy()

```
char writeFxy (
        EmbPattern * pattern,
        FILE * file )
```

18.56 extern/libembroidery/src/formats/format_gc.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char readGc (EmbPattern *pattern, FILE *file)
- char writeGc (EmbPattern *pattern, FILE *file)

18.56.1 Function Documentation

18.56.1.1 readGc()

```
char readGc (
        EmbPattern * pattern,
        FILE * file )
```

Smoothie G-Code
Main Reference: Machinery's Handbook Guide A Guide to Tables, Formulas, & More in the 31st Edition by John Milton Amiss, Franklin D. Jones and Henry Ryffel

18.56.1.2 writeGc()

```
char writeGc (
        EmbPattern * pattern,
        FILE * file )
```

18.57 extern/libembroidery/src/formats/format_gnc.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char readGnc (EmbPattern *pattern, FILE *file)
- char writeGnc (EmbPattern *pattern, FILE *file)

18.57.1 Function Documentation

18.57.1.1 readGnc()

```
char readGnc (
        EmbPattern * pattern,
        FILE * file )
```

18.57.1.2 Great Notions Embroidery Format (.gnc)

Stitch Only Format.

18.57.1.3 writeGnc()

```
char writeGnc (
        EmbPattern * pattern,
        FILE * file )
```

18.58 extern/libembroidery/src/formats/format_gt.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char readGt (EmbPattern *pattern, FILE *file)
- char writeGt (EmbPattern *pattern, FILE *file)

18.58.1 Function Documentation

18.58.1.1 readGt()

```
char readGt (
        EmbPattern * pattern,
        FILE * file )
```

18.58.1.2 Gold Thread Embroidery Format (.gt) Stitch Only Format.

18.58.1.3 writeGt()

```
char writeGt (
    EmbPattern * pattern,
    FILE * file )
```

18.59 extern/libembroidery/src/formats/format_hus.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- int husDecodeStitchType (unsigned char b)
- unsigned char ∗ husDecompressData (unsigned char ∗input, int compressedInputLength, int decompressedContentLength)
- unsigned char ∗ husCompressData (unsigned char ∗input, int decompressedInputSize, int ∗compressedSize)
- int husDecodeByte (unsigned char b)
- unsigned char husEncodeByte (EmbReal f)
- unsigned char husEncodeStitchType (int st)
- char readHus (EmbPattern ∗pattern, FILE ∗file)
- char writeHus (EmbPattern ∗pattern, FILE ∗file)

18.59.1 Function Documentation

18.59.1.1 husCompressData()

```
unsigned char * husCompressData (
    unsigned char * input,
    int decompressedInputSize,
    int * compressedSize )
```

18.59.1.2 husDecodeByte()

```
int husDecodeByte (
    unsigned char b )
```

18.59.1.3 husDecodeStitchType()

```
int husDecodeStitchType (
    unsigned char b )
```

18.59.1.4 Husqvarna Viking Embroidery Format (.hus) Stitch Only Format.

18.59.1.5 husDecompressData()

```
unsigned char * husDecompressData (
    unsigned char * input,
    int compressedInputLength,
    int decompressedContentLength )
```

18.59.1.6 husEncodeByte()

```
unsigned char husEncodeByte (
    EmbReal f )
```

18.59.1.7 husEncodeStitchType()

```
unsigned char husEncodeStitchType (
        int st )
```

18.59.1.8 readHus()

```
char readHus (
        EmbPattern * pattern,
        FILE * file )
```

18.59.1.9 writeHus()

```
char writeHus (
        EmbPattern * pattern,
        FILE * file )
```

18.60 extern/libembroidery/src/formats/format_inb.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char readInb (EmbPattern ∗pattern, FILE ∗file)
- char writeInb (EmbPattern ∗pattern, FILE ∗file)

18.60.1 Function Documentation

18.60.1.1 readInb()

```
char readInb (
        EmbPattern * pattern,
        FILE * file )
```

18.60.1.2 Inbro Embroidery Format (.inb)

Stitch Only Format.

18.60.1.3 writeInb()

```
char writeInb (
        EmbPattern * pattern,
        FILE * file )
```

18.61 extern/libembroidery/src/formats/format_inf.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char readInf (EmbPattern ∗pattern, FILE ∗file)
- char writeInf (EmbPattern ∗pattern, FILE ∗file)

18.61.1 Function Documentation

18.61.1.1 readInf()

```
char readInf (
    EmbPattern * pattern,
    FILE * file )
```

18.61.1.2 Embroidery Color Format (.inf) Stitch Only Format.

18.61.1.3 writeInf()

```
char writeInf (
    EmbPattern * pattern,
    FILE * file )
```

18.62 extern/libembroidery/src/formats/format_jef.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Classes

- struct hoop_padding

Functions

- int jefGetHoopSize (int width, int height)
- char jefDecode (unsigned char inputByte)
- void jefSetHoopFromId (EmbPattern ∗pattern, int hoopCode)
- void read_hoop (FILE ∗file, struct hoop_padding ∗hoop, char ∗label)
- char readJef (EmbPattern ∗pattern, FILE ∗file)
- void jefEncode (unsigned char ∗b, char dx, char dy, int flags)
- char writeJef (EmbPattern ∗pattern, FILE ∗file)

18.62.1 Function Documentation

18.62.1.1 jefDecode()

```
char jefDecode (
    unsigned char inputByte )
```

18.62.1.2 jefEncode()

```
void jefEncode (
    unsigned char * b,
    char dx,
    char dy,
    int flags )
```

18.62.1.3 jefGetHoopSize()

```
int jefGetHoopSize (
    int width,
    int height )
```

18.62.1.4 Janome Embroidery Format (.jef) Stitch Only Format.

18.62.1.5 jefSetHoopFromId()

```
void jefSetHoopFromId (
    EmbPattern * pattern,
    int hoopCode )
```

18.62.1.6 read_hoop()

```
void read_hoop (
    FILE * file,
    struct hoop_padding * hoop,
    char * label )
```

18.62.1.7 readJef()

```
char readJef (
    EmbPattern * pattern,
    FILE * file )
```

18.62.1.8 writeJef()

```
char writeJef (
    EmbPattern * pattern,
    FILE * file )
```

18.63 extern/libembroidery/src/formats/format_ksm.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- void ksmEncode (unsigned char ∗b, char dx, char dy, int flags)
- char readKsm (EmbPattern ∗pattern, FILE ∗file)
- char writeKsm (EmbPattern ∗pattern, FILE ∗file)

18.63.1 Function Documentation

18.63.1.1 ksmEncode()

```
void ksmEncode (
    unsigned char * b,
    char dx,
    char dy,
    int flags )
```

18.63.1.2 Pfaff professional Design format (.ksm)

Stitch Only Format.

18.63.1.3 readKsm()

```
char readKsm (
    EmbPattern * pattern,
    FILE * file )
```

18.63.1.4 writeKsm()

```
char writeKsm (
    EmbPattern * pattern,
    FILE * file )
```

18.64 extern/libembroidery/src/formats/format_max.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char readMax (EmbPattern ∗pattern, FILE ∗file)
- char writeMax (EmbPattern ∗pattern, FILE ∗file)

Variables

- const unsigned char max_header [ ]

18.64.1 Function Documentation

18.64.1.1 readMax()

```
char readMax (
    EmbPattern * pattern,
    FILE * file )
```

18.64.1.2 writeMax()

```
char writeMax (
    EmbPattern * pattern,
    FILE * file )
```

18.64.2 Variable Documentation

18.64.2.1 max_header

```
const unsigned char max_header[]
```

Initial value:

```
= {
    0x56,0x43,0x53,0x4D,0xFC,0x03,0x00,0x00,0x01,0x00,0x00,0x00,0x01,0x00,0x00,0x00,
    0xF6,0x25,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x08,0x00,0x00,0x00,0x05,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x31,0x33,0x37,0x38,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x4D,0x61,0x64,0x65,0x69,0x72,0x61,0x20,
    0x52,0x61,0x79,0x6F,0x6E,0x20,0x34,0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x01,0x38,0x09,0x31,0x33,0x30,0x2F,0x37,0x30,0x35,0x20,0x48,0xFA,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00
}
```

18.64.2.2 Pfaff Embroidery Format (.max)

Stitch Only Format.

18.65 extern/libembroidery/src/formats/format_mit.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char readMit (EmbPattern ∗pattern, FILE ∗file)
- char writeMit (EmbPattern ∗pattern, FILE ∗file)

18.65.1 Function Documentation

18.65.1.1 readMit()

```
char readMit (
        EmbPattern * pattern,
        FILE * file )
```

18.65.1.2 Mitsubishi Embroidery Format (.mit)

Stitch Only Format.

18.65.1.3 writeMit()

```
char writeMit (
        EmbPattern * pattern,
        FILE * file )
```

18.66 extern/libembroidery/src/formats/format_new.c File Reference

```
#include <stdio.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char readNew (EmbPattern ∗pattern, FILE ∗file)
- char writeNew (EmbPattern ∗pattern, FILE ∗file)

18.66.1 Function Documentation

18.66.1.1 readNew()

```
char readNew (
        EmbPattern * pattern,
        FILE * file )
```

18.66.1.2 Ameco Embroidery Format (.new)

Stitch Only Format.

18.66.1.3 writeNew()

```
char writeNew (
        EmbPattern * pattern,
        FILE * file )
```

18.67 extern/libembroidery/src/formats/format_ofm.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char ∗ ofmReadLibrary (FILE ∗file)
- static int ofmReadClass (FILE ∗file)
- void ofmReadBlockHeader (FILE ∗file)
- void ofmReadColorChange (FILE ∗file, EmbPattern ∗pattern)
- void ofmReadThreads (FILE ∗file, EmbPattern ∗p)
- EmbReal ofmDecode (unsigned char b1, unsigned char b2)
- void ofmReadExpanded (FILE ∗file, EmbPattern ∗p)
- char readOfm (EmbPattern ∗pattern, FILE ∗fileCompound)
- char writeOfm (EmbPattern ∗pattern, FILE ∗file)

18.67.1 Function Documentation

18.67.1.1 ofmDecode()

```
EmbReal ofmDecode (
    unsigned char b1,
    unsigned char b2 )
```

18.67.1.2 ofmReadBlockHeader()

```
void ofmReadBlockHeader (
    FILE * file )
```

18.67.1.3 ofmReadClass()

```
static int ofmReadClass (
    FILE * file ) [static]
```

18.67.1.4 ofmReadColorChange()

```
void ofmReadColorChange (
    FILE * file,
    EmbPattern * pattern )
```

18.67.1.5 ofmReadExpanded()

```
void ofmReadExpanded (
    FILE * file,
    EmbPattern * p )
```

18.67.1.6 ofmReadLibrary()

```
char * ofmReadLibrary (
    FILE * file )
```

18.67.1.7 Melco Embroidery Format (.ofm) Stitch Only Format.

18.67.1.8 ofmReadThreads()

```
void ofmReadThreads (
    FILE * file,
    EmbPattern * p )
```

18.67.1.9 readOfm()

```
char readOfm (
    EmbPattern * pattern,
    FILE * fileCompound )
```

18.67.1.10 writeOfm()

```
char writeOfm (
    EmbPattern * pattern,
    FILE * file )
```

18.68 extern/libembroidery/src/formats/format_pcd.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "../embroidery_internal.h"
```

Functions

- char readPcd (EmbPattern ∗pattern, const char ∗fileName, FILE ∗file)
- char writePcd (EmbPattern ∗pattern, FILE ∗file)

18.68.1 Function Documentation

18.68.1.1 readPcd()

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

18.68.1.2 Pfaff PCD File Format (.pcd)

Stitch Only Format.
The format uses a signed 3 byte-length number type.
See the description here ([5](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 v1^{T} I_{2} v2 for our vectors *v1* and \v2 so a "component-wise product". The result is stored at the pointer *result*.
That is (1 0) (a) = (xa) (x y)(0 1) (b) (yb)

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 writeImage (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_difat_create()

```
bcf_file_difat * bcf_difat_create (
        FILE * file,
        unsigned int fatSectors,
        const unsigned int sectorSize )
```

file fatSectors sectorSize

Returns

bcf_file_difat*

18.109.2.2 bcf_directory_free()

```
void bcf_directory_free (
        bcf_directory ** dir )
```

dir

18.109.2.3 bcf_file_free()

```
void bcf_file_free (
        bcf_file * bcfFile )
```

bcfFile

18.109.2.4 bcfFile_read()

```
int bcfFile_read (
        FILE * file,
        bcf_file * bcfFile )
```

file bcfFile

Returns

int

18.109.2.5 bcfFileFat_create()

```
bcf_file_fat * bcfFileFat_create (
        const unsigned int sectorSize )
```

sectorSize

Returns

bcf_file_fat*

18.109.2.6 bcfFileHeader_read()

```
bcf_file_header bcfFileHeader_read (
        FILE * file )
```

file

Returns

bcf_file_header

18.109.2.7 binaryReadString()

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

file buffer maxLength

18.109.2.8 binaryReadUnicodeString()

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

file buffer stringLength

18.109.2.9 check_header_present()

```
int check_header_present (
        FILE * file,
        int minimum_header_length )
```

file minimum_header_length

Returns

int

Checks that there are enough bytes to interpret the header, stops possible segfaults when reading in the header bytes.
Returns 0 if there aren't enough, or the length of the file if there are.

18.109.2.10 CompoundFileDirectory()

```
bcf_directory * CompoundFileDirectory (
        const unsigned int maxNumberOfDirectoryEntries )
```

maxNumberOfDirectoryEntries

Returns

bcf_directory*

18.109.2.11 CompoundFileDirectoryEntry()

```
bcf_directory_entry * CompoundFileDirectoryEntry (
        FILE * file )
```

file

Returns

bcf_directory_entry*

18.109.2.12 copy_trim()

```
char * copy_trim (
        char const * s )
```

s

Returns

char*

Todo decription

18.109.2.13 emb_optOut()

```
char * emb_optOut (
            EmbReal num,
            char * str )
```

Optimizes the number (*num*) for output to a text file and returns it as a string (*str*).

num str

Returns

char∗

18.109.2.14 emb_readline()

```
int emb_readline (
            FILE * file,
            char * line,
            int maxLength )
```

file line maxLength

Returns

int

18.109.2.15 embArc_print()

```
void embArc_print (
            EmbArc arc )
```

arc

Todo move to arc.c

18.109.2.16 embColor_distance()

```
int embColor_distance (
            EmbColor a,
            EmbColor b )
```

a b

Returns

int

18.109.2.17 embColor_read()

```
void embColor_read (
            FILE * f,
            EmbColor * c,
            int toRead )
```

f c toRead

18.109.2.18 embColor_write()

```
void embColor_write (
            FILE * f,
            EmbColor c,
            int toWrite )
```

f c toWrite

18.109.2.19 embSatinOutline_generateSatinOutline()

```
void embSatinOutline_generateSatinOutline (
            EmbArray * lines,
            EmbReal thickness,
            EmbSatinOutline * result )
```

lines thickness result

18.109.2.20 embSatinOutline_renderStitches()

```
EmbArray * embSatinOutline_renderStitches (
        EmbSatinOutline * result,
        EmbReal density )
```

result density

Returns

EmbArray∗

18.109.2.21 embThread_findNearestColor()

```
int embThread_findNearestColor (
        EmbColor color,
        EmbColor * color_list,
        int n_colors )
```

Returns the closest color to the required color based on a list of available threads. The algorithm is a simple least squares search against the list. If the (square of) Euclidean 3-dimensional distance between the points in (red, green, blue) space is smaller then the index is saved and the remaining index is returned to the caller.
color The EmbColor color to match. *colors* The EmbThreadList pointer to start the search at. *mode* Is the argument an array of threads (0) or colors (1)?

Returns

closestIndex The entry in the ThreadList that matches.

18.109.2.22 embThread_findNearestThread()

```
int embThread_findNearestThread (
        EmbColor color,
        EmbThread * thread_list,
        int n_threads )
```

color thread_list n_threads

Returns

int

18.109.2.23 embThread_getRandom()

```
EmbThread embThread_getRandom (
        void )
```

Returns a random thread color, useful in filling in cases where the actual color of the thread doesn't matter but one needs to be declared to test or render a pattern.

Returns

c The resulting color.

18.109.2.24 embTime_initNow()

```
void embTime_initNow (
        EmbTime * t )
```

t

18.109.2.25 embTime_time()

```
EmbTime embTime_time (
        EmbTime * t )
```

t

Returns

EmbTime

18.109.2.26 embVector_print() `void embVector_print (`

```
EmbVector v,
char * label )
```

v label
move to vector.c

18.109.2.27 entriesInDifatSector() `unsigned int entriesInDifatSector (`

```
bcf_file_difat * fat )
```

fat

Returns

unsigned int

18.109.2.28 get_trim_bounds() `void get_trim_bounds (`

```
char const * s,
char const ** firstWord,
char const ** trailingSpace )
```

Get the trim bounds object.
s firstWord trailingSpace

18.109.2.29 GetFile() `FILE * GetFile (`

```
bcf_file * bcfFile,
FILE * file,
char * fileToFind )
```

Get the File object.
bcfFile file fileToFind

Returns

FILE∗

18.109.2.30 haveExtraDIFATSectors() `int haveExtraDIFATSectors (`

```
bcf_file * file )
```

file

Returns

int

18.109.2.31 loadFatFromSector() `void loadFatFromSector (`

```
bcf_file_fat * fat,
FILE * file )
```

fat file

18.109.2.32 parseDIFATSectors() `void parseDIFATSectors (`

```
FILE * file,
bcf_file * bcfFile )
```

file bcfFile

18.109.2.33 parseDirectoryEntryName() `void parseDirectoryEntryName (`

```
FILE * file,
bcf_directory_entry * dir )
```

file dir

18.109.2.34 parseTime()

```
EmbTime parseTime (
          FILE * file )
```

file

Returns

EmbTime

18.109.2.35 readFullSector()

```
unsigned int readFullSector (
          FILE * file,
          bcf_file_difat * bcfFile,
          unsigned int * difatEntriesToRead )
```

file bcfFile difatEntriesToRead

Returns

unsigned int

18.109.2.36 readNextSector()

```
void readNextSector (
          FILE * file,
          bcf_directory * dir )
```

file dir

18.109.2.37 sectorSize()

```
unsigned int sectorSize (
          bcf_file * bcfFile )
```

bcfFile

Returns

unsigned int

18.109.2.38 seekToSector()

```
int seekToSector (
          bcf_file * bcfFile,
          FILE * file,
          const unsigned int sector )
```

bcfFile file sector

Returns

int

18.109.2.39 stringInArray()

```
int stringInArray (
          const char * s,
          const char ** array )
```

Tests for the presence of a string *s* in the supplied *array*.
The end of the array is marked by an empty string.

Returns

0 if not present 1 if present.

18.109.2.40 write_24bit() `void write_24bit ( FILE * file, int x )`

file x

18.109.3 Variable Documentation

18.109.3.1 black_thread `EmbThread black_thread = { { 0, 0, 0 }, "Black", "Black" }`

18.109.3.2 difatEntriesInHeader `const unsigned int difatEntriesInHeader = 109`

18.109.3.3 emb_error `int emb_error = 0`

Error code storage for optional control flow blocking.

18.109.3.4 emb_verbose `int emb_verbose = 0`

Verbosity level.

18.109.3.5 embConstantPi `const EmbReal embConstantPi = 3.1415926535`

18.109.3.6 sizeOfChainingEntryAtEndOfDifatSector `const unsigned int sizeOfChainingEntryAtEndOfDifatSector = 4`

18.109.3.7 sizeOfDifatEntry `const unsigned int sizeOfDifatEntry = 4`

18.109.3.8 sizeOfDirectoryEntry `const unsigned int sizeOfDirectoryEntry = 128`

18.109.3.9 sizeOfFatEntry `const unsigned int sizeOfFatEntry = sizeof(unsigned int)`

18.109.3.10 WHITESPACE `char const WHITESPACE[] = " \t\n\r"`

18.110 extern/libembroidery/src/pattern.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "embroidery_internal.h"
```

Functions

- EmbPattern ∗ embPattern_create (void)
- void embPattern_hideStitchesOverLength (EmbPattern ∗p, int length)
- int embPattern_addThread (EmbPattern ∗pattern, EmbThread thread)
- void embPattern_fixColorCount (EmbPattern ∗p)

- void embPattern_copystitch_listToPolylines (EmbPattern ∗p)
- void embPattern_copyPolylinesTostitch_list (EmbPattern ∗p)
- void embPattern_movestitch_listToPolylines (EmbPattern ∗p)
- void embPattern_movePolylinesTostitch_list (EmbPattern ∗p)
- void embPattern_addStitchAbs (EmbPattern ∗p, EmbReal x, EmbReal y, int flags, int isAutoColorIndex)
- void embPattern_addStitchRel (EmbPattern ∗p, EmbReal dx, EmbReal dy, int flags, int isAutoColorIndex)
- void embPattern_changeColor (EmbPattern ∗p, int index)
- void embPattern_scale (EmbPattern ∗p, EmbReal scale)
- EmbRect embPattern_calcBoundingBox (EmbPattern ∗p)
- void embPattern_flipHorizontal (EmbPattern ∗p)
- void embPattern_flipVertical (EmbPattern ∗p)
- void embPattern_flip (EmbPattern ∗p, int horz, int vert)
- void embPattern_combineJumpStitches (EmbPattern ∗p)
- void embPattern_correctForMaxStitchLength (EmbPattern ∗p, EmbReal maxStitchLength, EmbReal max↩ JumpLength)
- 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