

Engrave Reference Manual

Generated by Doxygen 1.5.1

Wed Mar 28 00:04:49 2007

Contents

1	Engrave Library Documentation	1
1.1	Introduction	1
1.2	Conclusion	2
2	Engrave Module Index	3
2.1	Engrave Modules	3
3	Engrave Data Structure Index	5
3.1	Engrave Data Structures	5
4	Engrave File Index	7
4.1	Engrave File List	7
5	Engrave Module Documentation	9
5.1	Engrave_Canvas: Functions to work with the engrave canvas.	9
5.2	Engrave_Data: Functions to work with data blocks	11
5.3	Enums: Various Flags and Enumerations used in Engrave.	15
5.4	Engrave_File: Functions to work with the Engrave file object	18
5.5	Engrave_Font: Functions to work with engrave font objects	27
5.6	Engrave_Group: Functions to work with an Engrave_Group.	30
5.7	Engrave_Image: Functions to work with Engrave_Image blocks.	40
5.8	Engrave_Load: Contains the functions to load EDC	44
5.9	Functions needed to output the Engrave data into	46
5.10	Engrave_Part: Functions to work with part objects	48
5.11	Engrave_Part_State: Functions to work with the Engrave_Part_State objects	59
5.12	Engrave_Program: Functions to work with Engrave_Program objects.	95
5.13	Engrave_Spectrum: Functions to work with engrave spectrum objects	105
5.14	Engrave_Style: Functions to work with Engrave_Style blocks.	110
6	Engrave Data Structure Documentation	111

6.1	_Engrave_Data Struct Reference	111
6.2	_Engrave_File Struct Reference	112
6.3	_Engrave_Font Struct Reference	113
6.4	_Engrave_Group Struct Reference	114
6.5	_Engrave_Image Struct Reference	116
6.6	_Engrave_Part Struct Reference	117
6.7	_Engrave_Part_State Struct Reference	119
6.8	_Engrave_Program Struct Reference	125
6.9	_Engrave_Spectrum Struct Reference	127
6.10	_Engrave_Spectrum_Color Struct Reference	128
6.11	_Engrave_Style Struct Reference	129
6.12	_Engrave_Tag Struct Reference	130
7	Engrave File Documentation	131
7.1	Engrave.h File Reference	131
7.2	engrave_canvas.h File Reference	132
7.3	engrave_data.h File Reference	133
7.4	engrave_enums.h File Reference	135
7.5	engrave_file.h File Reference	137
7.6	engrave_font.h File Reference	140
7.7	engrave_group.h File Reference	141
7.8	engrave_image.h File Reference	144
7.9	engrave_load.h File Reference	146
7.10	engrave_out.h File Reference	147
7.11	engrave_part.h File Reference	148
7.12	engrave_part_state.h File Reference	151
7.13	engrave_program.h File Reference	159
7.14	engrave_spectrum.h File Reference	162
7.15	engrave_style.h File Reference	164

Chapter 1

Engrave Library Documentation

1.1 Introduction

Engrave is designed to allow you the ability to easily create and edit Edge EDJ files.

As an example of how easy Engrave is to work with, the following example will read in either an EDC file or an EDJ file and attempt to write out an EDJ and EDC version of the given file.

```
#include "Engrave.h"

int
main(int argc, char ** argv)
{
    Engrave_File *ef = NULL;

    if (argc < 2) {
        printf("need file\n");
        return 1;
    }

    if (strstr(argv[1], ".edj"))
        ef = engrave_load_edj(argv[1]);
    else {
        if (argc < 4) {
            printf("need img and font dirs with .edc file\n");
            return 1;
        }
        ef = engrave_load_edc(argv[1], argv[2], argv[3]);
    }

    if (!engrave_edj_output(ef, "test.edj"))
        printf("failed to write test.edj\n");

    if (!engrave_edc_output(ef, "test.out"))
        printf("failed to write test.out\n");

    return 0;
}
```

Compiling with the Engrave library is pretty simple, assuming you've named your app `engrave_test.c` the following command will do the trick:

```
gcc -o engrave_test 'engrave-config --cflags --libs' engrave_test.c
```

1.2 Conclusion

Engrave has been designed to make it easy to open and manipulate Edje files, be they EDJ or EDC.

If you have any questions or comments about Engrave please email enlightenment-devel at lists dot sourceforge dot net

Chapter 2

Engrave Module Index

2.1 Engrave Modules

Here is a list of all modules:

Engrave_Canvas: Functions to work with the engrave canvas.	9
Engrave_Data: Functions to work with data blocks	11
Enums: Various Flags and Enumerations used in Engrave.	15
Engrave_File: Functions to work with the Engrave file object	18
Engrave_Font: Functions to work with engrave font objects	27
Engrave_Group: Functions to work with an Engrave_Group.	30
Engrave_Image: Functions to work with Engrave_Image blocks.	40
Engrave_Load: Contains the functions to load EDC	44
Functions needed to output the Engrave data into	46
Engrave_Part: Functions to work with part objects	48
Engrave_Part_State: Functions to work with the Engrave_Part_State objects	59
Engrave_Program: Functions to work with Engrave_Program objects.	95
Engrave_Spectrum: Functions to work with engrave spectrum objects	105
Engrave_Style: Functions to work with Engrave_Style blocks.	110

Chapter 3

Engrave Data Structure Index

3.1 Engrave Data Structures

Here are the data structures with brief descriptions:

_Engrave_Data (Stores the information for the data object)	111
_Engrave_File (Structure to hold the file information)	112
_Engrave_Font (Stores the needed font information)	113
_Engrave_Group (Stores the needed group information)	114
_Engrave_Image (Contains the needed image values)	116
_Engrave_Part (All values needed to define a part)	117
_Engrave_Part_State (Stores the information for the Engrave_Part_State object) . .	119
_Engrave_Program (Contains all the info to define a program)	125
_Engrave_Spectrum (Stores the needed spectrum information)	127
_Engrave_Spectrum_Color (Stores the spectrum color information)	128
_Engrave_Style (Contains the needed style values)	129
_Engrave_Tag (Contains the tag information)	130

Chapter 4

Engrave File Index

4.1 Engrave File List

Here is a list of all documented files with brief descriptions:

Engrave.h (The file that should be included by any project using Engrave)	131
engrave_canvas.h (Engrave_Canvas object functions.Contains all of the functions related to the engrave canvas object)	132
engrave_data.h (Engrave_Data block object Contains all of the functions to manipulate Engrave_Data objects)	133
engrave_enums.h (Contains all of the enumerations used in Engrave)	135
engrave_file.h (Engrave_File object functions.Contains all of the functions related to the Engrave file object)	137
engrave_font.h (Engrave_Font block Contains all of the functions to manipulate Engrave_Font blocks)	140
engrave_group.h (Engrave_Group functions Contains all of the functions to manipulate Engrave_Group objects)	141
engrave_image.h (Engrave_Image object functions.Contains all of the functions to manipulate Engrave_Image objects)	144
engrave_load.h (Engrave loading functions Contains the functions to load either an EDC or EDJ into Engrave)	146
engrave_out.h (Functions to facilitate outputting the Engrave information.Provided the needed functions to output the Engrave information into various formats) . . .	147
engrave_part.h (Engrave_Part object functions.Contains all functions to manipulate the Engrave_Part object)	148
engrave_part_state.h (Engrave_Part_State object.Contains all of the functions to work with an Engrave_Part_State object)	151
engrave_private.h	??
engrave_program.h (Engrave_Program object functions.Contains all of the functions to mainuplulate Engrave_Program objects)	159
engrave_spectrum.h (Engrave_Spectrum block Contains all of the functions to manipulate Engrave_Spectrum blocks)	162
engrave_style.h (Engrave_Style object functions.Contains all of the functions to manipulate Engrave_Style objects)	164

Chapter 5

Engrave Module Documentation

5.1 Engrave_Canvas: Functions to work with the engrave canvas.

Functions

- EAPI Evas_Object * [engrave_canvas_new](#) (Evas *e)
engrave_canvas_new - create a new canvas
- EAPI void [engrave_canvas_file_set](#) (Evas_Object *o, [Engrave_File](#) *ef)
engrave_canvas_fil_set - set the file to display in this canvas
- EAPI void [engrave_canvas_current_group_set](#) (Evas_Object *o, [Engrave_Group](#) *eg)
engrave_canvas_current_group_set - set the group to display in the canvas

5.1.1 Function Documentation

5.1.1.1 EAPI void [engrave_canvas_current_group_set](#) (Evas_Object * o, [Engrave_Group](#) * eg)

[engrave_canvas_current_group_set](#) - set the group to display in the canvas

Parameters:

- o*,: The canvas object to display the group in
- eg*; The Engrave_Group to display

Returns:

Returns no value.

5.1.1.2 EAPI void engrave_canvas_file_set (Evas_Object * *o*, [Engrave_File](#) * *ef*)

engrave_canvas_fil_set - set the file to display in this canvas

Parameters:

o,: The evas object that is the canvas

ef,: The Engrave_File to display in the canvas

Returns:

Returns no value.

5.1.1.3 EAPI Evas_Object* engrave_canvas_new (Evas * *e*)

engrave_canvas_new - create a new canvas

Parameters:

e,: The Evas to create the canvas in

Returns:

Returns the Evas_Object that is the canvas or NULL on error.

5.2 Engrave_Data: Functions to work with data blocks

Data Structures

- struct [_Engrave_Data](#)
Stores the information for the data object.

Typedefs

- typedef [_Engrave_Data](#) [Engrave_Data](#)
The Engrave_Data typedef.

Functions

- EAPI [Engrave_Data](#) * [engrave_data_new](#) (const char *key, const char *value)
engrave_data_new - create a new data block
- EAPI void [engrave_data_free](#) ([Engrave_Data](#) *ed)
engrave_data_free - free the given data block
- EAPI void [engrave_data_parent_set](#) ([Engrave_Data](#) *ed, void *parent)
engrave_data_parent_set - set the parent pointer on this node This will set the parent pointer, in the case of Engrave_Data this can be either an Engrave_File or an Engrave_Group.
- EAPI void * [engrave_data_parent_get](#) ([Engrave_Data](#) *ed)
engrave_data_parent_get - get the parent pointer on this node
- EAPI const char * [engrave_data_key_get](#) ([Engrave_Data](#) *ed)
engrave_data_key_get - get the key for this data
- EAPI const char * [engrave_data_value_get](#) ([Engrave_Data](#) *ed)
engrave_data_value_get - get the string value of the data object
- EAPI int [engrave_data_int_value_get](#) ([Engrave_Data](#) *ed)
engrave_data_int_value_get - get the int value of the data object
- EAPI void [engrave_data_value_set](#) ([Engrave_Data](#) *ed, const char *value)
engrave_data_value_set - set the value of the data object
- EAPI void [engrave_data_int_value_set](#) ([Engrave_Data](#) *ed, int value)
engrave_data_int_value_set - set the int value of the data object

5.2.1 Function Documentation

5.2.1.1 EAPI void engrave_data_free ([Engrave_Data](#) * *ed*)

engrave_data_free - free the given data block

Parameters:

ed,: The Engrave_Data to free

Returns:

Returns no value.

5.2.1.2 EAPI int engrave_data_int_value_get ([Engrave_Data](#) * *ed*)

engrave_data_int_value_get - get the int value of the data object

Parameters:

ed,: The Engrave_Data to get the value from

Returns:

Returns the int value of the data or 0 if its not set.

5.2.1.3 EAPI void engrave_data_int_value_set ([Engrave_Data](#) * *ed*, int *value*)

engrave_data_int_value_set - set the int value of the data object

Parameters:

ed,: The Engrave_Data to set the value into

value,: The value to set

Returns:

Returns no value

5.2.1.4 EAPI const char* engrave_data_key_get ([Engrave_Data](#) * *ed*)

engrave_data_key_get - get the key for this data

Parameters:

ed,: The Engrave_Data to get the key from

Returns:

Returns the key for this data or NULL on failure.

5.2.1.5 EAPI [Engrave_Data](#)* engrave_data_new (const char * *key*, const char * *value*)

engrave_data_new - create a new data block

Parameters:

key,: the key to access the block

value,: the value to store in the block

Returns:

Returns a pointer to a newly allocated data block on success, NULL on failure.

5.2.1.6 EAPI void* engrave_data_parent_get ([Engrave_Data](#) * *ed*)

engrave_data_parent_get - get the parent pointer on this node

Parameters:

ed,: The Engrave_Data to get the pointer from

Returns:

Returns the parent pointer, or NULL if none set.

5.2.1.7 EAPI void engrave_data_parent_set ([Engrave_Data](#) * *ed*, void * *parent*)

engrave_data_parent_set - set the parent pointer on this node This will set the parent pointer, in the case of Engrave_Data this can be either an Engrave_File or an Engrave_Group.

Parameters:

ed,: The Engrave_Data to set the parent on

parent,: The parent pointer to set

Returns:

Returns no value.

5.2.1.8 EAPI const char* engrave_data_value_get (Engrave_Data * *ed*)

engrave_data_value_get - get the string value of the data object

Parameters:

ed,: The Engrave_Data to get the string value from.

Returns:

Returns the value of the data on success or NULL on failure.

5.2.1.9 EAPI void engrave_data_value_set (Engrave_Data * *ed*, const char * *value*)

engrave_data_value_set - set the value of the data object

Parameters:

ed,: The Engrave_Data to set the value into

value,: The value to set.

Returns:

Returns no value.

5.3 Enums: Various Flags and Enumerations used in Engrave.

Enumerations

- enum `_Engrave_Image_Type` {
`ENGRAVE_IMAGE_TYPE_RAW`,
`ENGRAVE_IMAGE_TYPE_COMP`,
`ENGRAVE_IMAGE_TYPE_LOSSY` }
- enum `_Engrave_Part_Type` {
`ENGRAVE_PART_TYPE_IMAGE`,
`ENGRAVE_PART_TYPE_TEXT`,
`ENGRAVE_PART_TYPE_TEXTBLOCK`,
`ENGRAVE_PART_TYPE_RECT`,
`ENGRAVE_PART_TYPE_SWALLOW`,
`ENGRAVE_PART_TYPE_GRADIENT` }
- enum `_Engrave_Text_Effect` {
`ENGRAVE_TEXT_EFFECT_NONE`,
`ENGRAVE_TEXT_EFFECT_PLAIN`,
`ENGRAVE_TEXT_EFFECT_OUTLINE`,
`ENGRAVE_TEXT_EFFECT_SOFT_OUTLINE`,
`ENGRAVE_TEXT_EFFECT_SHADOW`,
`ENGRAVE_TEXT_EFFECT_OUTLINE_SHADOW`,
`ENGRAVE_TEXT_EFFECT_SOFT_SHADOW`,
`ENGRAVE_TEXT_EFFECT_OUTLINE_SOFT_SHADOW` }
- enum `_Engrave_Action` {
`ENGRAVE_ACTION_STATE_SET`,
`ENGRAVE_ACTION_STOP`,
`ENGRAVE_ACTION_SIGNAL_EMIT`,
`ENGRAVE_ACTION_DRAG_VAL_SET`,
`ENGRAVE_ACTION_DRAG_VAL_STEP`,
`ENGRAVE_ACTION_DRAG_VAL_PAGE`,
`ENGRAVE_ACTION_SCRIPT` }
- enum `_Engrave_Transition` {
`ENGRAVE_TRANSITION_LINEAR`,
`ENGRAVE_TRANSITION_SINUSOIDAL`,
`ENGRAVE_TRANSITION_ACCELERATE`,
`ENGRAVE_TRANSITION_DECELERATE` }

- enum `_Engrave_Aspect_Preference` {
`ENGRAVE_ASPECT_PREFERENCE_NONE`,
`ENGRAVE_ASPECT_PREFERENCE_VERTICAL`,
`ENGRAVE_ASPECT_PREFERENCE_HORIZONTAL`,
`ENGRAVE_ASPECT_PREFERENCE_BOTH` }

5.3.1 Enumeration Type Documentation

5.3.1.1 enum `_Engrave_Action`

Enumerator:

ENGRAVE_ACTION_STATE_SET Set the state of a given part.
ENGRAVE_ACTION_STOP Stop the given action.
ENGRAVE_ACTION_SIGNAL_EMIT Emit the given signal.
ENGRAVE_ACTION_DRAG_VAL_SET Set the drag value of a given part.
ENGRAVE_ACTION_DRAG_VAL_STEP Set the drag step of a given part.
ENGRAVE_ACTION_DRAG_VAL_PAGE Set the drag page of a given part.
ENGRAVE_ACTION_SCRIPT Set implicitly if a script {} block is included.

5.3.1.2 enum `_Engrave_Aspect_Preference`

Enumerator:

ENGRAVE_ASPECT_PREFERENCE_NONE No aspect preference.
ENGRAVE_ASPECT_PREFERENCE_VERTICAL Vertical aspect preference.
ENGRAVE_ASPECT_PREFERENCE_HORIZONTAL Horizontal aspect preference.
ENGRAVE_ASPECT_PREFERENCE_BOTH Vertical and Horizontal aspect preference.

5.3.1.3 enum `_Engrave_Image_Type`

Enumerator:

ENGRAVE_IMAGE_TYPE_RAW No compression, no loss of detail.
ENGRAVE_IMAGE_TYPE_COMP Compress image, no loss of detail.
ENGRAVE_IMAGE_TYPE_LOSSY Lossy compression of image.

5.3.1.4 enum [__Engrave_Part_Type](#)

Enumerator:

ENGRAVE_PART_TYPE_IMAGE Part is an image.
ENGRAVE_PART_TYPE_TEXT Part is text.
ENGRAVE_PART_TYPE_TEXTBLOCK Part is textblock.
ENGRAVE_PART_TYPE_RECT Part is a rectangle.
ENGRAVE_PART_TYPE_SWALLOW Part is a swallow.
ENGRAVE_PART_TYPE_GRADIENT Part is a gradient.

5.3.1.5 enum [__Engrave_Text_Effect](#)

Enumerator:

ENGRAVE_TEXT_EFFECT_NONE No text effect.
ENGRAVE_TEXT_EFFECT_PLAIN Normal text.
ENGRAVE_TEXT_EFFECT_OUTLINE Outlined text.
ENGRAVE_TEXT_EFFECT_SOFT_OUTLINE Soft outlined text.
ENGRAVE_TEXT_EFFECT_SHADOW Shadowed text.
ENGRAVE_TEXT_EFFECT_OUTLINE_SHADOW Outlined and shadowed text.
ENGRAVE_TEXT_EFFECT_SOFT_SHADOW Soft shadowed text.
ENGRAVE_TEXT_EFFECT_OUTLINE_SOFT_SHADOW Outlined and soft shadowed text.

5.3.1.6 enum [__Engrave_Transition](#)

Enumerator:

ENGRAVE_TRANSITION_LINEAR A linear transtion.
ENGRAVE_TRANSITION_SINUSOIDAL A sinusoidal transition.
ENGRAVE_TRANSITION_ACCELERATE An accelerating transition.
ENGRAVE_TRANSITION_DECELERATE A decelerating transition.

5.4 Engrave_File: Functions to work with the Engrave file object

Data Structures

- struct [_Engrave_File](#)
Structure to hold the file information.

Typedefs

- typedef [_Engrave_File](#) [Engrave_File](#)
The Engrave_File typedef.

Functions

- EAPI [Engrave_File](#) * [engrave_file_new](#) (void)
engrave_file_new - create a new Engrave_File object
- EAPI void [engrave_file_free](#) ([Engrave_File](#) *ef)
engrave_file_free - free the file object
- EAPI void [engrave_file_image_dir_set](#) ([Engrave_File](#) *ef, const char *dir)
engrave_file_image_dir_set - set the image directory for the file
- EAPI void [engrave_file_font_dir_set](#) ([Engrave_File](#) *ef, const char *dir)
engrave_file_font_dir_set - set the font dir for the file
- EAPI const char * [engrave_file_image_dir_get](#) ([Engrave_File](#) *ef)
engrave_file_image_dir_get - get the image directory
- EAPI const char * [engrave_file_font_dir_get](#) ([Engrave_File](#) *ef)
engrave_file_font_dir_get - get the font directory
- EAPI void [engrave_file_font_add](#) ([Engrave_File](#) *e, [Engrave_Font](#) *ef)
engrave_file_font_add - add the font to the engrave file.
- EAPI void [engrave_file_spectrum_add](#) ([Engrave_File](#) *e, [Engrave_Spectrum](#) *es)
engrave_file_spectrum_add - add the spectrum to the engrave file.
- EAPI void [engrave_file_style_add](#) ([Engrave_File](#) *e, [Engrave_Style](#) *es)
engrave_file_style_add - add the style to the engrave file.
- EAPI void [engrave_file_image_add](#) ([Engrave_File](#) *ef, [Engrave_Image](#) *ei)
engrave_file_image_add - add the image to the engrave file.

- EAPI void `engrave_file_data_add` (`Engrave_File *ef`, `Engrave_Data *ed`)
engrave_file_data_add - add the data to the engrave file.
- EAPI void `engrave_file_group_add` (`Engrave_File *ef`, `Engrave_Group *eg`)
engrave_file_group_add - add the group to the given file
- EAPI `Engrave_Style *` `engrave_file_style_last_get` (`Engrave_File *ef`)
engrave_file_style_last_get - returns the last style in the file
- EAPI `Engrave_Spectrum *` `engrave_file_spectrum_last_get` (`Engrave_File *ef`)
engrave_file_spectrum_last_get - returns the last spectrum in the file
- EAPI `Engrave_Group *` `engrave_file_group_last_get` (`Engrave_File *ef`)
engrave_file_group_last_get - returns the last group in the file
- EAPI `Engrave_Group *` `engrave_file_group_by_name_find` (`Engrave_File *ef`, `const char *name`)
engrave_file_group_by_name_find - returns the Engrave_Group with the given name.
- EAPI `Engrave_Font *` `engrave_file_font_by_name_find` (`Engrave_File *ef`, `const char *name`)
engrave_file_font_by_name_find - find the Engrave_Font by name
- EAPI `Engrave_Image *` `engrave_file_image_by_name_find` (`Engrave_File *ef`, `const char *name`)
engrave_file_image_by_name_find - returns the Engrave_Image with the given name.
- EAPI `int` `engrave_file_images_count` (`Engrave_File *ef`)
engrave_file_images_count - get the number of images
- EAPI `int` `engrave_file_data_count` (`Engrave_File *ef`)
engrave_file_data_count - count data blocks in file
- EAPI `int` `engrave_file_groups_count` (`Engrave_File *ef`)
engrave_file_groups_count - count the groups in the file
- EAPI `int` `engrave_file_fonts_count` (`Engrave_File *ef`)
engrave_file_fonts_count - count the fonts in the file
- EAPI `int` `engrave_file_spectra_count` (`Engrave_File *ef`)
engrave_file_spectra_count - count the spectra in the file
- EAPI `int` `engrave_file_styles_count` (`Engrave_File *ef`)
engrave_file_styles_count - count the styles in the file
- EAPI `Engrave_Data *` `engrave_file_data_by_key_find` (`Engrave_File *ef`, `const char *key`)
engrave_file_data_by_key_find - find the Engrave_Data by key

5.4.1 Function Documentation

5.4.1.1 EAPI void engrave_file_data_add ([Engrave_File](#) * *ef*, [Engrave_Data](#) * *ed*)

engrave_file_data_add - add the data to the engrave file.

Parameters:

ef,: The [Engrave_File](#) to add the data too.

ed,: The [Engrave_Data](#) to add to the file.

Returns:

Returns no value.

5.4.1.2 EAPI [Engrave_Data](#)* engrave_file_data_by_key_find ([Engrave_File](#) * *ef*, const char * *key*)

engrave_file_data_by_key_find - find the [Engrave_Data](#) by key

Parameters:

ef,: The [Engrave_File](#) to search

key,: The key to search for

Returns:

Returns the [Engrave_Data](#) with the matching key or NULL if no such data exists.

5.4.1.3 EAPI int engrave_file_data_count ([Engrave_File](#) * *ef*)

engrave_file_data_count - count data blocks in file

Parameters:

ef,: The [Engrave_File](#) to check for data

Returns:

Returns number of data blocks, 0 otherwise

5.4.1.4 EAPI void engrave_file_font_add (Engrave_File * *e*, Engrave_Font * *ef*)

engrave_file_font_add - add the font to the engrave file.

Parameters:

e,: The Engrave_File to add the font too.

ef,: The Engrave_Font to add to the file.

Returns:

Returns no value.

5.4.1.5 EAPI Engrave_Font* engrave_file_font_by_name_find (Engrave_File * *ef*, const char * *name*)

engrave_file_font_by_name_find - find the Engrave_Font by name

Parameters:

ef,: The Engrave_File to search

name,: The name to search for

Returns:

Returns the Engrave_Font with the matching name or NULL if no such font exists.

5.4.1.6 EAPI const char* engrave_file_font_dir_get (Engrave_File * *ef*)

engrave_file_font_dir_get - get the font directory

Parameters:

ef,: The Engrave_File to get the directory from

Returns:

Returns the current font directory

5.4.1.7 EAPI void engrave_file_font_dir_set (Engrave_File * *ef*, const char * *dir*)

engrave_file_font_dir_set - set the font dir for the file

Parameters:

ef,: The Engrave_File to set the font dir on

dir,: The directory to set the font dir too

Returns:

Returns no value.

5.4.1.8 EAPI int engrave_file_fonts_count (Engrave_File * ef)

engrave_file_fonts_count - count the fonts in the file

Parameters:

ef,: The Engrave_File to check for fonts

Returns:

Returns the number of fonts in the file, 0 otherwise

5.4.1.9 EAPI void engrave_file_free (Engrave_File * ef)

engrave_file_free - free the file object

Parameters:

ef,: The Engrave_File to free

Returns:

Returns no value.

5.4.1.10 EAPI void engrave_file_group_add (Engrave_File * ef, Engrave_Group * eg)

engrave_file_group_add - add the group to the given file

Parameters:

ef,: The Engrave_File to add the group too.

eg,: The Engrave_Group to add to the file.

Returns:

Returns no value.

5.4.1.11 EAPI [Engrave_Group](#)* engrave_file_group_by_name_find
([Engrave_File](#) * *ef*, const char * *name*)

engrave_file_group_by_name_find - returns the Engrave_Group with the given name.

Parameters:

ef,: The Engrave_File to search for the group in.

name,: The name of the group to search for.

Returns:

Returns the Engrave_Group with the given *name* or NULL if no corresponding group can be found.

5.4.1.12 EAPI [Engrave_Group](#)* engrave_file_group_last_get (**[Engrave_File](#) * *ef***)

engrave_file_group_last_get - returns the last group in the file

Parameters:

ef,: The Engrave_File from which to retrieve the group

Returns:

Returns the last Engrave_Group in the engrave file *ef* or NULL if there are no available groups.

5.4.1.13 EAPI int engrave_file_groups_count (**[Engrave_File](#) * *ef***)

engrave_file_groups_count - count the groups in the file

Parameters:

ef,: The Engrave_File to check for groups

Returns:

Returns then number of groups, 0 otherwise

5.4.1.14 EAPI void engrave_file_image_add (**[Engrave_File](#) * *ef*, [Engrave_Image](#) * *ei***)

engrave_file_image_add - add the image to the engrave file.

Parameters:

ef,: The Engrave_File to add the image too.

ei,: The Engrave_Image to add to the file.

Returns:

Returns no value.

5.4.1.15 EAPI [Engrave_Image](#)* engrave_file_image_by_name_find ([Engrave_File](#) * *ef*, const char * *name*)

engrave_file_image_by_name_find - returns the Engrave_Image with the given name.

Parameters:

ef,: The Engrave_File to search for the image in.

name,: The name of the image to search for.

Returns:

Returns the Engrave_Image with the given *name* or NULL if no corresponding image can be found.

5.4.1.16 EAPI const char* engrave_file_image_dir_get ([Engrave_File](#) * *ef*)

engrave_file_image_dir_get - get the image directory

Parameters:

ef,: The Engrave_File to get the directory from

Returns:

Returns the current image directory or NULL on failure

5.4.1.17 EAPI void engrave_file_image_dir_set ([Engrave_File](#) * *ef*, const char * *dir*)

engrave_file_image_dir_set - set the image directory for the file

Parameters:

ef,: The Engrave_File to set the image dir into

dir,: The directory to set

Returns:

Returns no value.

5.4.1.18 EAPI int engrave_file_images_count (Engrave_File * ef)

engrave_file_images_count - get the number of images

Parameters:

ef,: The Engrave_File to check for images

Returns:

Returns the number of images, 0 otherwise

5.4.1.19 EAPI Engrave_File* engrave_file_new (void)

engrave_file_new - create a new Engrave_File object

Returns:

Returns a pointer to a newly allocated file object on success, NULL on failure.

5.4.1.20 EAPI int engrave_file_spectra_count (Engrave_File * ef)

engrave_file_spectra_count - count the spectra in the file

Parameters:

ef,: The Engrave_File to check for spectra

Returns:

Returns the number of spectra in the file, 0 otherwise

5.4.1.21 EAPI void engrave_file_spectrum_add (Engrave_File * e, Engrave_Spectrum * es)

engrave_file_spectrum_add - add the spectrum to the engrave file.

Parameters:

e,: The Engrave_File to add the font too.

es,: The Engrave_Spectrum to add to the file.

Returns:

Returns no value.

5.4.1.22 EAPI [Engrave_Spectrum*](#) engrave_file_spectrum_last_get ([Engrave_File](#) * *ef*)

engrave_file_spectrum_last_get - returns the last spectrum in the file

Parameters:

ef,: The Engrave_File from which to retrieve the group

Returns:

Returns the last Engrave_Spectrum in the engrave file *ef* or NULL

5.4.1.23 EAPI void engrave_file_style_add ([Engrave_File](#) * *e*, [Engrave_Style](#) * *es*)

engrave_file_style_add - add the style to the engrave file.

Parameters:

e,: The Engrave_File to add the style too.

es,: The Engrave_Style to add to the file.

Returns:

Returns no value.

5.4.1.24 EAPI [Engrave_Style*](#) engrave_file_style_last_get ([Engrave_File](#) * *ef*)

engrave_file_style_last_get - returns the last style in the file

Parameters:

ef,: The Engrave_File from which to retrieve the group

Returns:

Returns the last Engrave_Style in the engrave file *ef* or NULL if there are no available groups.

5.4.1.25 EAPI int engrave_file_styles_count ([Engrave_File](#) * *ef*)

engrave_file_styles_count - count the styles in the file

Parameters:

ef,: The Engrave_File to check for styles

Returns:

Returns the number of styles in the file, 0 otherwise

5.5 Engrave_Font: Functions to work with engrave font objects

Data Structures

- struct `_Engrave_Font`
Stores the needed font information.

Typedefs

- typedef `_Engrave_Font` `Engrave_Font`
The Engrave_Font typedef.

Functions

- EAPI `Engrave_Font * engrave_font_new` (const char *path, const char *name)
engrave_font_new - create a new Engrave_Font object.
- EAPI void `engrave_font_free` (`Engrave_Font *ef`)
engrave_font_free - free the memory
- EAPI void `engrave_font_parent_set` (`Engrave_Font *font`, void *ef)
engrave_font_parent_set - set the parent of the font node
- EAPI void * `engrave_font_parent_get` (`Engrave_Font *ef`)
engrave_font_parent_get - get the parent pointer
- EAPI const char * `engrave_font_name_get` (`Engrave_Font *ef`)
engrave_font_name_get - get the font name
- EAPI const char * `engrave_font_path_get` (`Engrave_Font *ef`)
engrave_font_path_get - get the font path

5.5.1 Function Documentation

5.5.1.1 EAPI void engrave_font_free (`Engrave_Font * ef`)

`engrave_font_free` - free the memory

Parameters:

ef,: The Engrave_Font to free

Returns:

Returns no value

5.5.1.2 EAPI const char* engrave_font_name_get (Engrave_Font * ef)

engrave_font_name_get - get the font name

Parameters:

ef,: The Engrave_Font to get the name from

Returns:

Returns the name of the font on success or NULL on failure.

5.5.1.3 EAPI Engrave_Font* engrave_font_new (const char * path, const char * name)

engrave_font_new - create a new Engrave_Font object.

Parameters:

path,: The path to the font

name,: The name for the font

Returns:

Returns a pointer to a newly allocated Engrave_Font on success, or NULL on failure.

5.5.1.4 EAPI void* engrave_font_parent_get (Engrave_Font * ef)

engrave_font_parent_get - get the parent pointer

Parameters:

ef,: The Engrave_Font to get the parent from

Returns:

Returns the pointer to the parent or NULL if none set

5.5.1.5 EAPI void engrave_font_parent_set (Engrave_Font * font, void * ef)

engrave_font_parent_set - set the parent of the font node

Parameters:

font,: The Engrave_Font to set the parent into

ef,: The Engrave_File to set as parent

Returns:

Retruns no value.

5.5.1.6 EAPI const char* engrave_font_path_get (Engrave_Font * ef)

engrave_font_path_get - get the font path

Parameters:

ef,: The Engrave_Font to get the path from

Returns:

Returns the path of the font on success or NULL on failure.

5.6 Engrave_Group: Functions to work with an Engrave_Group.

Data Structures

- struct [_Engrave_Group](#)
Stores the needed group information.

Typedefs

- typedef [_Engrave_Group](#) [Engrave_Group](#)
The Engrave_Group typedef.

Functions

- EAPI [Engrave_Group](#) * [engrave_group_new](#) (void)
engrave_group_new - create a new Engrave_Group object.
- EAPI void [engrave_group_free](#) ([Engrave_Group](#) *eg)
engrave_group_free - free the group data
- EAPI void [engrave_group_parent_set](#) ([Engrave_Group](#) *eg, void *ef)
engrave_group_parent_set - set the parent pointer
- EAPI void * [engrave_group_parent_get](#) ([Engrave_Group](#) *eg)
engrave_group_parent_get - get the parent pointer
- EAPI void [engrave_group_data_add](#) ([Engrave_Group](#) *eg, [Engrave_Data](#) *ed)
engrave_group_data_add - add the Engrave_Data to the group
- EAPI void [engrave_group_part_add](#) ([Engrave_Group](#) *eg, [Engrave_Part](#) *ep)
engrave_group_part_add - add the given part to the group
- EAPI void [engrave_group_program_add](#) ([Engrave_Group](#) *eg, [Engrave_Program](#) *ep)
engrave_group_program_add - add the program to the group
- EAPI void [engrave_group_script_set](#) ([Engrave_Group](#) *eg, const char *script)
engrave_group_script_set - attach the script to the given group
- EAPI void [engrave_group_name_set](#) ([Engrave_Group](#) *eg, const char *name)
engrave_group_name_set - set the name of the group to the given name.
- EAPI void [engrave_group_alias_set](#) ([Engrave_Group](#) *eg, const char *alias)
engrave_group_alias_set - set the alias of the group.

- EAPI void `engrave_group_min_size_set` (`Engrave_Group` *eg, int w, int h)
engrave_group_min_size_set - set the min size of the group.
- EAPI void `engrave_group_max_size_set` (`Engrave_Group` *eg, int w, int h)
engrave_group_max_size_set - set the max size of the group.
- EAPI `Engrave_Part` * `engrave_group_part_last_get` (`Engrave_Group` *eg)
engrave_group_part_last_get - retrieve the last part in the group.
- EAPI `Engrave_Program` * `engrave_group_program_last_get` (`Engrave_Group` *eg)
engrave_group_program_last_get - retrieve the last program in the group.
- EAPI const char * `engrave_group_name_get` (`Engrave_Group` *eg)
engrave_group_name_get - get the name attached to the group
- EAPI const char * `engrave_group_alias_get` (`Engrave_Group` *eg)
engrave_group_alias_get - get the alias attached to the group
- EAPI const char * `engrave_group_script_get` (`Engrave_Group` *eg)
engrave_group_script_get - get the script attached to the group
- EAPI void `engrave_group_min_size_get` (`Engrave_Group` *eg, int *w, int *h)
engrave_group_min_size_get - get the min size of the group
- EAPI void `engrave_group_max_size_get` (`Engrave_Group` *eg, int *w, int *h)
engrave_group_max_size_get - get the max size of the group
- EAPI int `engrave_group_data_count` (`Engrave_Group` *eg)
engrave_group_data_count - count the data blocks in the group
- EAPI int `engrave_group_parts_count` (`Engrave_Group` *eg)
engrave_group_parts_count - returns the number of parts in the group
- EAPI int `engrave_group_programs_count` (`Engrave_Group` *eg)
engrave_group_programs_count - get number of programs in the group
- EAPI void `engrave_group_data_foreach` (`Engrave_Group` *eg, void(*func)(`Engrave_Data` *, void *), void *data)
engrave_group_data_foreach - iterate over the data in the group
- EAPI void `engrave_group_parts_foreach` (`Engrave_Group` *eg, void(*func)(`Engrave_Part` *, void *), void *data)
engrave_group_parts_foreach - Iterate over the parts in the Engrave_group
- EAPI void `engrave_group_programs_foreach` (`Engrave_Group` *eg, void(*func)(`Engrave_Program` *, void *), void *data)
engrave_group_programs_foreach - Iterate over the groups programs

- EAPI `Engrave_Part * engrave_group_part_by_name_find (Engrave_Group *eg, const char *part)`
engrave_group_part_by_name_find - find the Engrave_Part by name
- EAPI `Engrave_Data * engrave_group_data_by_key_find (Engrave_Group *eg, const char *key)`
engrave_group_data_by_key_find - find the Engrave_Data by key

5.6.1 Function Documentation

5.6.1.1 EAPI `const char* engrave_group_alias_get (Engrave_Group * eg)`

`engrave_group_alias_get` - get the alias attached to the group

Parameters:

eg,: The `Engrave_Group` to get the alias from

Returns:

Returns the alias of the group or NULL on failure.

5.6.1.2 EAPI `void engrave_group_alias_set (Engrave_Group * eg, const char * alias)`

`engrave_group_alias_set` - set the alias of the group.

Parameters:

eg,: The `Engrave_Group` to attach the alias too.

alias,: The alias to attach to the group.

Returns:

Returns no value.

5.6.1.3 EAPI `void engrave_group_data_add (Engrave_Group * eg, Engrave_Data * ed)`

`engrave_group_data_add` - add the `Engrave_Data` to the group

Parameters:

eg,: The `Engrave_Group` to add the data too.

ed,: The Engrave_Data to add to the group.

Returns:

Returns no value.

5.6.1.4 EAPI Engrave_Data* engrave_group_data_by_key_find
(Engrave_Group * *eg*, const char * *key*)

engrave_group_data_by_key_find - find the Engrave_Data by key

Parameters:

eg,: The Engrave_Group to search

key,: The key to search for

Returns:

Returns the Engrave_Data with the matching key or NULL if no such data exists.

5.6.1.5 EAPI int engrave_group_data_count (Engrave_Group * *eg*)

engrave_group_data_count - count the data blocks in the group

Parameters:

eg,: The Engrave_Group to check if there is data

Returns:

Returns number of data blocks, 0 otherwise.

5.6.1.6 EAPI void engrave_group_data_foreach (Engrave_Group * *eg*,
void(*) (Engrave_Data *, void *) *func*, void * *data*)

engrave_group_data_foreach - iterate over the data in the group

Parameters:

eg,: The Engrave_Group to iterate over the data

func,: The function to call on each data block

data,: any user data to pass to the function

Returns:

Returns no value.

5.6.1.7 EAPI void engrave_group_free (Engrave_Group * eg)

engrave_group_free - free the group data

Parameters:

eg,: The Engrave_Group to free

Returns:

Returns no value

5.6.1.8 EAPI void engrave_group_max_size_get (Engrave_Group * eg, int * w, int * h)

engrave_group_max_size_get - get the max size of the group

Parameters:

eg,: The Engrave_Group to get the max size from

w,: Will be set to the max width of the group

h,: Will be set to the max height of the group

Returns:

Returns no value.

5.6.1.9 EAPI void engrave_group_max_size_set (Engrave_Group * eg, int w, int h)

engrave_group_max_size_set - set the max size of the group.

Parameters:

eg,: The Engrave_Group on which to set the max size.

w,: The max width to set on the group.

h,: The max height to set on the group.

Returns:

Returns no value.

5.6.1.10 EAPI void engrave_group_min_size_get (Engrave_Group * eg, int * w, int * h)

engrave_group_min_size_get - get the min size of the group

Parameters:

eg,: The Engrave_Group to get the min size from
w,: Will be set to the min width of the group
h,: Will be set to the min height of the group

Returns:

Returns no value.

5.6.1.11 EAPI void engrave_group_min_size_set (Engrave_Group * *eg*, int *w*, int *h*)

engrave_group_min_size_set - set the min size of the group.

Parameters:

eg,: The Engrave_Group on which to set the min size.
w,: The min width to set on the group.
h,: The min height to set on the group.

Returns:

Returns no value.

5.6.1.12 EAPI const char* engrave_group_name_get (Engrave_Group * *eg*)

engrave_group_name_get - get the name attached to the group

Parameters:

eg,: The Engrave_Group to get the name from

Returns:

Returns the name of the group or NULL on failure.

5.6.1.13 EAPI void engrave_group_name_set (Engrave_Group * *eg*, const char * *name*)

engrave_group_name_set - set the name of the group to the given name.

Parameters:

eg,: The Engrave_Group to attach the name too.
name,: The name to attach to the group.

Returns:

Returns no value.

5.6.1.14 EAPI [Engrave_Group](#)* engrave_group_new (void)

engrave_group_new - create a new Engrave_Group object.

Returns:

Returns a pointer to a newly allocated Engrave_Group object on success or NULL on failure.

5.6.1.15 EAPI void* engrave_group_parent_get ([Engrave_Group](#) * eg)

engrave_group_parent_get - get the parent pointer

Parameters:

eg,: The Engrave_Group to get the parent pointer from

Returns:

Returns the Engrave_File parent pointer or NULL if none set

5.6.1.16 EAPI void engrave_group_parent_set ([Engrave_Group](#) * eg, void * ef)

engrave_group_parent_set - set the parent pointer

Parameters:

eg,: The Engrave_Group to set the parent pointer into

ef,: The Engrave_File to set as the parent

Returns:

Returns no value.

5.6.1.17 EAPI void engrave_group_part_add ([Engrave_Group](#) * eg, [Engrave_Part](#) * ep)

engrave_group_part_add - add the given part to the group

Parameters:

eg,: The Engrave_Group to attach the part too.

ep,: The Engrave_Part to add too the group.

Returns:

Returns no value.

5.6.1.18 EAPI [Engrave_Part](#)* engrave_group_part_by_name_find
([Engrave_Group](#) * *eg*, const char * *part*)

engrave_group_part_by_name_find - find the Engrave_Part by name

Parameters:

eg,: The Engrave_Group to search

part,: The part name to search for

Returns:

Returns the Engrave_Part with the matching name or NULL if no such name exists.

5.6.1.19 EAPI [Engrave_Part](#)* engrave_group_part_last_get ([Engrave_Group](#) *
eg)

engrave_group_part_last_get - retrieve the last part in the group.

Parameters:

eg,: The Engrave_Group to retrieve the last part from.

Returns:

Returns the last Engrave_Part in the group or NULL if no such part exists.

5.6.1.20 EAPI int engrave_group_parts_count ([Engrave_Group](#) * *eg*)

engrave_group_parts_count - returns the number of parts in the group

Parameters:

eg,: The Engrave_Group to check if there are parts

Returns:

Returns the number of parts, 0 otherwise.

5.6.1.21 EAPI void engrave_group_parts_foreach ([Engrave_Group](#) * *eg*,
void(*)([Engrave_Part](#) *, void *) *func*, void * *data*)

engrave_group_parts_foreach - Iterate over the parts in the Engrave_group

Parameters:

eg,: The Engrave_Group to get the parts from

func,: the function to call for each part
data,: The user data to pass to the function

Returns:

Returns no value.

5.6.1.22 EAPI void engrave_group_program_add ([Engrave_Group](#) * *eg*,
[Engrave_Program](#) * *ep*)

engrave_group_program_add - add the program to the group

Parameters:

eg,: The Engrave_Group to add the program too.
ep,: The Engrave_Program to add to the group.

Returns:

Returns no value.

5.6.1.23 EAPI [Engrave_Program*](#) engrave_group_program_last_get
([Engrave_Group](#) * *eg*)

engrave_group_program_last_get - retrieve the last program in the group.

Parameters:

eg,: The Engrave_Group to retrieve the last program from.

Returns:

Returns the last Engrave_Program in the group or NULL if no such program exists.

5.6.1.24 EAPI int engrave_group_programs_count ([Engrave_Group](#) * *eg*)

engrave_group_programs_count - get number of programs in the group

Parameters:

eg,: The Engrave_Group to check if there are programs

Returns:

Returns number of programs, 0 otherwise.

5.6.1.25 EAPI void engrave_group_programs_foreach ([Engrave_Group](#) * *eg*, void (*)([Engrave_Program](#) *, void *) *func*, void * *data*)

engrave_group_programs_foreach - Iterate over the groups programs

Parameters:

eg,: The Engrave_Group to iterate over

func,: The function to call for each program

data,: User data

Returns:

Returns no value.

5.6.1.26 EAPI const char* engrave_group_script_get ([Engrave_Group](#) * *eg*)

engrave_group_script_get - get the script attached to the group

Parameters:

eg,: The Engrave_Group to get the script from

Returns:

Returns the script on success or NULL on failure.

5.6.1.27 EAPI void engrave_group_script_set ([Engrave_Group](#) * *eg*, const char * *script*)

engrave_group_script_set - attach the script to the given group

Parameters:

eg,: The Engrave_Group to attach the script too.

script,: The script to attach to the group.

Returns:

Returns no value.

5.7 Engrave_Image: Functions to work with Engrave_Image blocks.

Data Structures

- struct [_Engrave_Image](#)
Contains the needed image values.

Typedefs

- typedef [_Engrave_Image](#) [Engrave_Image](#)
The Engrave_Image typedef.

Functions

- EAPI [Engrave_Image](#) * [engrave_image_new](#) (const char *name, [Engrave_Image_Type](#) type, double value)
engrave_image_new - create a new Engrave_Image object.
- EAPI [Engrave_Image](#) * [engrave_image_dup](#) ([Engrave_Image](#) *from)
engrave_image_dup - duplicate an Engrave_Image
- EAPI void [engrave_image_free](#) ([Engrave_Image](#) *ef)
engrave_image_free - free the Engrave_Image
- EAPI void [engrave_image_parent_set](#) ([Engrave_Image](#) *ei, void *ef)
engrave_image_parent_set - set the image parent pointer
- EAPI void * [engrave_image_parent_get](#) ([Engrave_Image](#) *ei)
engrave_image_parent_get - get the parent of the image
- EAPI const char * [engrave_image_name_get](#) ([Engrave_Image](#) *ei)
engrave_image_name_get - get the image name
- EAPI [Engrave_Image_Type](#) [engrave_image_type_get](#) ([Engrave_Image](#) *ei)
engrave_image_type_get - get the type of the image
- EAPI double [engrave_image_compression_value_get](#) ([Engrave_Image](#) *ei)
engrave_image_compression_value_get - get the image compression value

5.7.1 Function Documentation

5.7.1.1 EAPI double engrave_image_compression_value_get (Engrave_Image * *ei*)

engrave_image_compression_value_get - get the image compression value

Parameters:

ei,: The Engrave_Image to get the value from

Returns:

Returns the compression value of the image or 0 if not set.

5.7.1.2 EAPI Engrave_Image* engrave_image_dup (Engrave_Image * *from*)

engrave_image_dup - duplicate an Engrave_Image

Parameters:

from,: The Engrave_Image to duplicate

Returns:

Returns a pointer to the newly allocated Engrave_Image on success or NULL on failure

5.7.1.3 EAPI void engrave_image_free (Engrave_Image * *ei*)

engrave_image_free - free the Engrave_Image

Parameters:

ei,: The Engrave_Image to free

Returns:

Returns no value.

5.7.1.4 EAPI const char* engrave_image_name_get (Engrave_Image * *ei*)

engrave_image_name_get - get the image name

Parameters:

ei,: The Engrave_Image to get the name from

Returns:

Returns the name of the image or NULL on failure.

5.7.1.5 EAPI **Engrave_Image*** engrave_image_new (const char * *name*, Engrave_Image_Type *type*, double *value*)

engrave_image_new - create a new Engrave_Image object.

Parameters:

- name*,: The name of the given image
- type*,: The Engrave_Image_Type of the given image.
- value*,: A compression value for the given image (if applicable)

Returns:

Returns a pointer to a newly allocated Engrave_Image object on success or NULL on failure.

5.7.1.6 EAPI void* engrave_image_parent_get (**Engrave_Image** * *ei*)

engrave_image_parent_get - get the parent of the image

Parameters:

- ei*,: The Engrave_Image to get the parent from

Returns:

Returns the pointer to the Engrave_File parent or NULL if none set

5.7.1.7 EAPI void engrave_image_parent_set (**Engrave_Image** * *ei*, void * *ef*)

engrave_image_parent_set - set the image parent pointer

Parameters:

- ei*,: The Engrave_Image to set the parent into
- ef*,: The Engrave_File to set as the parent

Returns:

Returns no value.

5.7.1.8 EAPI Engrave_Image_Type engrave_image_type_get (**Engrave_Image** * *ei*)

engrave_image_type_get - get the type of the image

Parameters:

- ei*,: The Engrave_Image to get the type from

Returns:

Returns the Engrave_Image_Type associated with *ei* or ENGRAVE_IMAGE_TYPE_NUM if the type is not set.

5.8 Engrave_Load: Contains the functions to load EDC

and EDJ files into Engrave.

Functions

- EAPI [Engrave_File](#) * [engrave_load_edc](#) (const char *file, const char *imdir, const char *fontdir)
engrave_load_edc - load the given edc file into memory.
- EAPI [Engrave_File](#) * [engrave_load_edj](#) (const char *filename)
engrave_load_edj - load the given EDJ file into memory.

5.8.1 Detailed Description

and EDJ files into Engrave.

5.8.2 Function Documentation

5.8.2.1 EAPI [Engrave_File](#)* [engrave_load_edc](#) (const char * *file*, const char * *imdir*, const char * *fontdir*)

engrave_load_edc - load the given edc file into memory.

Parameters:

- file*,: The EDC file to load.
- imdir*,: The image directory for the EDC file.
- fontdir*,: The font directory for the EDC file.

Returns:

Returns a pointer to a newly allocated [Engrave_File](#) object on success or NULL on failure.

5.8.2.2 EAPI [Engrave_File](#)* [engrave_load_edj](#) (const char * *filename*)

engrave_load_edj - load the given EDJ file into memory.

Parameters:

- filename*,: The filename of the EDJ file to load.

Returns:

Returns a pointer to a newly allocated Engrave_File object on success or NULL on failure.

5.9 Functions needed to output the Engrave data into

different file formats.

Functions

- EAPI int `engrave_edj_output` (`Engrave_File` *`engrave_file`, const char *`path`)
engrave_edj_output – Create an EDJ file from the in-memory data.
- EAPI int `engrave_edc_output` (`Engrave_File` *`engrave_file`, const char *`path`)
engrave_edc_output – Create an EDC file from the in-memory data.

5.9.1 Detailed Description

different file formats.

5.9.2 Function Documentation

5.9.2.1 EAPI int `engrave_edc_output` (`Engrave_File` * *engrave_file*, const char * *path*)

engrave_edc_output – Create an EDC file from the in-memory data.

Parameters:

engrave_file,: The `Engrave_File` to use to create the EDJ file.

path,: The filename to save the EDC file too.

Returns:

Returns 1 on success 0 otherwise.

5.9.2.2 EAPI int `engrave_edj_output` (`Engrave_File` * *engrave_file*, const char * *path*)

engrave_edj_output – Create an EDJ file from the in-memory data.

Parameters:

engrave_file,: The `Engrave_File` to use to create the EDJ file.

path,: The filename to save the EDJ file too.

Returns:

Returns 1 on success 0 otherwise.

5.10 Engrave_Part: Functions to work with part objects

Data Structures

- struct `_Engrave_Part`
All values needed to define a part.

Typedefs

- typedef `_Engrave_Part Engrave_Part`
The Engrave_Part typedef.

Functions

- EAPI `Engrave_Part * engrave_part_new` (`Engrave_Part_Type` type)
engrave_part_new - create a new part object.
- EAPI void `engrave_part_free` (`Engrave_Part *ep`)
engrave_part_free - free the part
- EAPI void `engrave_part_parent_set` (`Engrave_Part *ep`, void *eg)
engrave_part_parent_set - set the parent pointer in the part
- EAPI void * `engrave_part_parent_get` (`Engrave_Part *ep`)
engrave_part_parent_get - get the parent pointer for the part
- EAPI void `engrave_part_name_set` (`Engrave_Part *ep`, const char *name)
engrave_part_name_set - set the name of the part.
- EAPI void `engrave_part_mouse_events_set` (`Engrave_Part *ep`, int val)
engrave_part_mouse_events_set - set the mouse events flag for the part.
- EAPI void `engrave_part_repeat_events_set` (`Engrave_Part *ep`, int val)
engrave_part_repeat_events_set - set the repeat events flag for the part.
- EAPI void `engrave_part_type_set` (`Engrave_Part *ep`, `Engrave_Part_Type` type)
engrave_part_type_set - set the type of the part
- EAPI void `engrave_part_effect_set` (`Engrave_Part *ep`, `Engrave_Text_Effect` effect)
engrave_part_effect_set - set the effect on the given part.
- EAPI void `engrave_part_clip_to_set` (`Engrave_Part *ep`, const char *clip_to)
engrave_part_clip_set - set the clip of the given part.

- EAPI `Engrave_Part_State * engrave_part_state_last_get (Engrave_Part *ep)`
engrave_part_state_last_get - get the last state in the part.
- EAPI `void engrave_part_dragable_x_set (Engrave_Part *ep, int x, int step, int count)`
engrave_part_dragable_x_set - set the draggable x value on the part.
- EAPI `void engrave_part_dragable_y_set (Engrave_Part *ep, int y, int step, int count)`
engrave_part_dragable_y_set - set the draggable y value on the part.
- EAPI `void engrave_part_dragable_confine_set (Engrave_Part *ep, const char *confine)`
engrave_part_dragable_confine_set - set the confine of the parts draggable.
- EAPI `void engrave_part_state_add (Engrave_Part *ep, Engrave_Part_State *eps)`
engrave_part_state_add - add the state to the part.
- EAPI `Engrave_Part_State * engrave_part_state_by_name_value_find (Engrave_Part *ep, const char *name, double val)`
*engrave_part_state_by_name_value_find - Find the Engrave_Part State that * matches the given name and value in this part*
- EAPI `const char * engrave_part_name_get (Engrave_Part *ep)`
engrave_part_name_get - Get the name of the part
- EAPI `Engrave_Part_Type engrave_part_type_get (Engrave_Part *ep)`
engrave_part_type_get - Get the type of the part
- EAPI `int engrave_part_mouse_events_get (Engrave_Part *ep)`
engrave_part_mouse_events_get - Get the mouse events setting
- EAPI `int engrave_part_repeat_events_get (Engrave_Part *ep)`
engrave_part_repeat_events_get - Get the repeat events setting
- EAPI `Engrave_Text_Effect engrave_part_effect_get (Engrave_Part *ep)`
engrave_part_effect_get - Get the text effect on the part
- EAPI `const char * engrave_part_clip_to_get (Engrave_Part *ep)`
engrave_part_clip_to_get - Get the clip_to setting of the part
- EAPI `void engrave_part_dragable_x_get (Engrave_Part *ep, int *x, int *step, int *count)`
engrave_part_dragable_x_get - Get the draggable info in the x direction
- EAPI `void engrave_part_dragable_y_get (Engrave_Part *ep, int *y, int *step, int *count)`
engrave_part_dragable_y_get - Get the draggable info in the y direction
- EAPI `const char * engrave_part_dragable_confine_get (Engrave_Part *ep)`
engrave_part_dragable_confine_get - Get the draggable confine setting
- EAPI `int engrave_part_states_count (Engrave_Part *ep)`

engrave_part_states_count - get number of states in the part

- EAPI void `engrave_part_state_foreach` (`Engrave_Part` *ep, void(*func)(`Engrave_Part_State` *, `Engrave_Part` *, void *), void *data)
engrave_part_state_foreach - iterate over each of the states in the part
- EAPI `Evas_Object *` `engrave_part_evas_object_get` (`Engrave_Part` *ep)
engrave_part_evas_object_get - get the evas object for this part
- EAPI void `engrave_part_evas_object_set` (`Engrave_Part` *ep, `Evas_Object` *o)
engrave_part_evas_object_set - set the evas object for this part

5.10.1 Function Documentation

5.10.1.1 EAPI `const char* engrave_part_clip_to_get` (`Engrave_Part` * ep)

`engrave_part_clip_to_get` - Get the clip_to setting of the part

Parameters:

ep,: The `Engrave_Part` to get the clip_to from

Returns:

Returns the clip_to string for the part or NULL on failure.

5.10.1.2 EAPI void `engrave_part_clip_to_set` (`Engrave_Part` * ep, `const char *` clip_to)

`engrave_part_clip_set` - set the clip of the given part.

Parameters:

ep,: The `Engrave_Part` to set the clip on
clip_to,: The name of the part to clip too.

Returns:

Returns no value.

5.10.1.3 EAPI `const char* engrave_part_dragable_confine_get` (`Engrave_Part` * ep)

`engrave_part_dragable_confine_get` - Get the draggable confine setting

Parameters:

ep,: The Engrave_Part to get the confine setting from

Returns:

Returns the confine setting or NULL on failure.

5.10.1.4 EAPI void engrave_part_dragable_confine_set (Engrave_Part * ep, const char * confine)

engrave_part_dragable_confine_set - set the confine of the parts draggable.

Parameters:

ep,: The Engrave_Part to confine the draggable on.

confine,: The name of the part to confine the draggable to.

Returns:

Returns no value.

5.10.1.5 EAPI void engrave_part_dragable_x_get (Engrave_Part * ep, int * x, int * step, int * count)

engrave_part_dragable_x_get - Get the draggable info in the x direction

Parameters:

ep,: The Engrave_Part to get the draggable information from

x,: Where to place the x value

step,: Where to place the step value

count,: Where to place the count value

Returns:

Returns no value.

5.10.1.6 EAPI void engrave_part_dragable_x_set (Engrave_Part * ep, int x, int step, int count)

engrave_part_dragable_x_set - set the draggable x value on the part.

Parameters:

ep,: The Engrave_Part to set the x draggable on.

x,: The x value to set on the draggable.

step,: The step value to set on the draggable.

count,: The count value to set on the draggable.

Returns:

Returns no value.

5.10.1.7 EAPI void engrave_part_dragable_y_get (Engrave_Part * ep, int * y, int * step, int * count)

engrave_part_dragable_y_get - Get the draggable info in the y direction

Parameters:

ep,: The Engrave_Part to get the draggable information from

y,: Where to place the y value

step,: Where to place the step value

count,: Where to place the count value

Returns:

Returns no value.

5.10.1.8 EAPI void engrave_part_dragable_y_set (Engrave_Part * ep, int y, int step, int count)

engrave_part_dragable_y_set - set the draggable y value on the part.

Parameters:

ep,: The Engrave_Part to set the y draggable on.

y,: The y value to set on the draggable.

step,: The step value to set on the draggable.

count,: The count value to set on the draggable.

Returns:

Returns no value.

5.10.1.9 EAPI Engrave_Text_Effect engrave_part_effect_get (Engrave_Part * ep)

engrave_part_effect_get - Get the text effect on the part

Parameters:

ep,: The Engrave_Part to get the effect from

Returns:

Returns the Engrave_Text_Effect setting for the part or ENGRAVE_TEXT_EFFECT_NONE if non set

5.10.1.10 EAPI void engrave_part_effect_set (Engrave_Part * ep, Engrave_Text_Effect effect)

engrave_part_effect_set - set the effect on the given part.

Parameters:

ep,: The Engrave_Part to set the effect upon.

effect,: The Engrave_Text_Effect to set on the part.

Returns:

Returns no value.

5.10.1.11 EAPI Evas_Object* engrave_part_evas_object_get (Engrave_Part * ep)

engrave_part_evas_object_get - get the evas object for this part

Parameters:

ep,: The Engrave_Part to get the evas object from

Returns:

Returns the evas object that displays this part

5.10.1.12 EAPI void engrave_part_evas_object_set (Engrave_Part * ep, Evas_Object * o)

engrave_part_evas_object_set - set the evas object for this part

Parameters:

ep,: The Engrave_Part to set the evas object into

o,: The Evas_Object to set into the part

Returns:

Returns no value.

5.10.1.13 EAPI void engrave_part_free (Engrave_Part * ep)

engrave_part_free - free the part

Parameters:

ep,: The Engrave_Part to free

Returns:

Returns no value.

5.10.1.14 EAPI int engrave_part_mouse_events_get (Engrave_Part * ep)

engrave_part_mouse_events_get - Get the mouse events setting

Parameters:

ep,: The Engrave_Part to get the mouse events from

Returns:

Returns the mouse events setting for the part

5.10.1.15 EAPI void engrave_part_mouse_events_set (Engrave_Part * ep, int val)

engrave_part_mouse_events_set - set the mouse events flag for the part.

Parameters:

ep,: The Engrave_Part to set the mouse events flag on.

val,: The value to set the mouse events flag too.

Returns:

Returns no value.

5.10.1.16 EAPI const char* engrave_part_name_get (Engrave_Part * ep)

engrave_part_name_get - Get the name of the part

Parameters:

ep,: The Engrave_Part to retrieve the name from.

Returns:

Returns the name of the part, or NULL on failure.

5.10.1.17 EAPI void engrave_part_name_set (Engrave_Part * *ep*, const char * *name*)

engrave_part_name_set - set the name of the part.

Parameters:

ep,: The Engrave_Part to set the name off
name,: The name to attach to the part.

Returns:

Returns no value.

5.10.1.18 EAPI Engrave_Part* engrave_part_new (Engrave_Part_Type *type*)

engrave_part_new - create a new part object.

Parameters:

type,: The Engrave_Part_Type for the part.

Returns:

Returns a pointer to the newly allocated Engrave_Part on success or NULL on failure.

5.10.1.19 EAPI void* engrave_part_parent_get (Engrave_Part * *ep*)

engrave_part_parent_get - get the parent pointer for the part

Parameters:

ep,: The Engrave_Part to get the parent from

Returns:

Returns the Engrave_Group pointer if set, or NULL if none set.

5.10.1.20 EAPI void engrave_part_parent_set (Engrave_Part * *ep*, void * *eg*)

engrave_part_parent_set - set the parent pointer in the part

Parameters:

ep,: The Engrave_Part to set the pointer into
eg,: The Engrave_Group to set as parent

Returns:

Returns no value.

5.10.1.21 EAPI int engrave_part_repeat_events_get ([Engrave_Part](#) * *ep*)

engrave_part_repeat_events_get - Get the repeat events setting

Parameters:

ep,: The Engrave_Part to get the repeat event setting from

Returns:

Returns the repeat event setting of the part

5.10.1.22 EAPI void engrave_part_repeat_events_set ([Engrave_Part](#) * *ep*, int *val*)

engrave_part_repeat_events_set - set the repeat events flag for the part.

Parameters:

ep,: The Engrave_Part to set the repeat events flag on.

val,: The value to set the repeat events flag too.

Returns:

Returns no value.

5.10.1.23 EAPI void engrave_part_state_add ([Engrave_Part](#) * *ep*, [Engrave_Part_State](#) * *eps*)

engrave_part_state_add - add the state to the part.

Parameters:

ep,: The Engrave_Part to add the state too.

eps,: The Engrave_Part_State to add to the part.

Returns:

Returns no value.

5.10.1.24 EAPI [Engrave_Part_State](#)* engrave_part_state_by_name_value_find ([Engrave_Part](#) * *ep*, const char * *name*, double *val*)

engrave_part_state_by_name_value_find - Find the Engrave_Part State that * matches the given name and value in this part

Parameters:

ep,: The Engrave_Part to search.
name,: The name to search for
val,: The value to search for.

Returns:

Returns a pointer to the Engrave_Part_State matching the given criteria or NULL if not found.

5.10.1.25 EAPI void engrave_part_state_foreach (Engrave_Part * *ep*,
 void(*) (Engrave_Part_State *, Engrave_Part *, void *) *func*, void *
data)

engrave_part_state_foreach - iterate over each of the states in the part

Parameters:

ep,: The Engrave_Part to get the states from
func,: The function to call for each state
data,: The user data

Returns:

Returns no value.

5.10.1.26 EAPI Engrave_Part_State* engrave_part_state_last_get
 (Engrave_Part * *ep*)

engrave_part_state_last_get - get the last state in the part.

Parameters:

ep,: The Engrave_Part to retrieve the state from.

Returns:

Returns the last *Engrave_Part_State* value in the part or NULL if no such value exists.

5.10.1.27 EAPI int engrave_part_states_count (Engrave_Part * *ep*)

engrave_part_states_count - get number of states in the part

Parameters:

ep,: The Engrave_Part to check

Returns:

Returns number of states in the part, 0 otherwise.

5.10.1.28 EAPI Engrave_Part_Type engrave_part_type_get (Engrave_Part * *ep*)

engrave_part_type_get - Get the type of the part

Parameters:

ep,: The Engrave_Part to get the type from

Returns:

Returns the Engrave_Part_Type of the part

5.10.1.29 EAPI void engrave_part_type_set (Engrave_Part * *ep*, Engrave_Part_Type *type*)

engrave_part_type_set - set the type of the part

Parameters:

ep,: The Engrave_Part to set the type on.

type,: The Engrave_Part_Type to set on the part

Returns:

Returns no value.

5.11 Engrave_Part_State: Functions to work with the Engrave_Part_State objects

Data Structures

- struct `_Engrave_Part_State`
Stores the information for the Engrave_Part_State object.

Typedefs

- typedef `_Engrave_Part_State` `Engrave_Part_State`
The Engrave_Part_State typedef.

Functions

- EAPI `Engrave_Part_State * engrave_part_state_new` (void)
engrave_part_state_new - create a new Engrave_Part_State object.
- EAPI void `engrave_part_state_free` (Engrave_Part_State *eps)
engrave_part_state_free - free the state
- EAPI void `engrave_part_state_parent_set` (Engrave_Part_State *eps, void *ep)
engrave_part_state_parent_set - set the parent part on this state
- EAPI void * `engrave_part_state_parent_get` (Engrave_Part_State *eps)
engrave_part_state_parent_get - get the parent part of this state
- EAPI void `engrave_part_state_name_set` (Engrave_Part_State *eps, const char *name, double value)
engrave_part_state_name_set - Set the name of the state
- EAPI void `engrave_part_state_visible_set` (Engrave_Part_State *eps, int visible)
engrave_part_state_visible_set - Set the visibility of the state
- EAPI void `engrave_part_state_align_set` (Engrave_Part_State *eps, double x, double y)
engrave_part_state_align_set - Set the alignment of the state.
- EAPI void `engrave_part_state_step_set` (Engrave_Part_State *eps, double x, double y)
engrave_part_state_step_set - Set the step value of the state
- EAPI void `engrave_part_state_min_size_set` (Engrave_Part_State *eps, int w, int h)
engrave_part_state_min_size_set - Set the min size of the state.

- EAPI void `engrave_part_state_fixed_size_set` (`Engrave_Part_State *eps`, int w, int h)
engrave_part_state_fixed_size_set - Set the fixed size of the state.
- EAPI void `engrave_part_state_max_size_set` (`Engrave_Part_State *eps`, int w, int h)
engrave_part_state_max_size_set - Set the max size of the state.
- EAPI void `engrave_part_state_aspect_set` (`Engrave_Part_State *eps`, double w, double h)
engrave_part_state_aspect_set - Set the width and height of the aspect ratio
- EAPI void `engrave_part_state_aspect_preference_set` (`Engrave_Part_State *eps`, `Engrave_Aspect_Preference prefer`)
engrave_part_state_aspect_preference_set - Set the aspect preference.
- EAPI void `engrave_part_state_rel1_relative_set` (`Engrave_Part_State *eps`, double x, double y)
engrave_part_state_rel1_relative_set - Set the relative value of rel1.
- EAPI void `engrave_part_state_rel1_offset_set` (`Engrave_Part_State *eps`, int x, int y)
engrave_part_state_rel1_offset_set - Set the rel1 offset value.
- EAPI void `engrave_part_state_rel1_to_set` (`Engrave_Part_State *eps`, const char *to)
engrave_part_state_rel1_to_set - Set the rel1 to value for both x and y dirs.
- EAPI void `engrave_part_state_rel1_to_x_set` (`Engrave_Part_State *eps`, const char *to)
engrave_part_state_rel1_to_x_set - Set what rel1 is relative too in the x dir.
- EAPI void `engrave_part_state_rel1_to_y_set` (`Engrave_Part_State *eps`, const char *to)
engrave_part_state_rel1_to_y_set - Set what rel1 is relative too in the y dir.
- EAPI void `engrave_part_state_rel2_relative_set` (`Engrave_Part_State *eps`, double x, double y)
engrave_part_state_rel2_relative_set - Set the rel2 relative value
- EAPI void `engrave_part_state_rel2_offset_set` (`Engrave_Part_State *eps`, int x, int y)
engrave_part_state_rel2_offset_set - Set the rel2 offset value.
- EAPI void `engrave_part_state_rel2_to_set` (`Engrave_Part_State *eps`, const char *to)
engrave_part_state_rel2_to_set - Set the rel2 to value for both x and y dirs.
- EAPI void `engrave_part_state_rel2_to_x_set` (`Engrave_Part_State *eps`, const char *to)
engrave_part_state_rel2_to_x_set - Set what rel2 is relative too in the x dir.
- EAPI void `engrave_part_state_rel2_to_y_set` (`Engrave_Part_State *eps`, const char *to)
engrave_part_state_rel2_to_y_set - Set what rel2 is relative too in the y dir.

- EAPI void `engrave_part_state_image_normal_set` (`Engrave_Part_State *eps`, `Engrave_Image *im`)
engrave_part_state_image_normal_set - Set the image we are associated with.
- EAPI void `engrave_part_state_image_tween_add` (`Engrave_Part_State *eps`, `Engrave_Image *im`)
engrave_part_state_image_tween_add - Add a tween to the state.
- EAPI void `engrave_part_state_image_border_set` (`Engrave_Part_State *eps`, int l, int r, int t, int b)
engrave_part_state_image_border_set - Set the border on the image.
- EAPI void `engrave_part_state_image_middle_set` (`Engrave_Part_State *eps`, int middle)
engrave_part_state_image_middle_set - Set the middle on the image.
- EAPI void `engrave_part_state_color_class_set` (`Engrave_Part_State *eps`, const char *class)
engrave_part_state_color_class_set - Set the colour class on the state.
- EAPI void `engrave_part_state_color_set` (`Engrave_Part_State *eps`, int r, int g, int b, int a)
engrave_part_state_color_set - Set the colour of the state.
- EAPI void `engrave_part_state_color2_set` (`Engrave_Part_State *eps`, int r, int g, int b, int a)
engrave_part_state_color2_set - Set the colour2 of the state
- EAPI void `engrave_part_state_color3_set` (`Engrave_Part_State *eps`, int r, int g, int b, int a)
engrave_part_state_color3_set - Set the colour 3 of the state.
- EAPI void `engrave_part_state_fill_smooth_set` (`Engrave_Part_State *eps`, int smooth)
engrave_part_state_fill_smooth_set - Set the fill smooth value of the state.
- EAPI void `engrave_part_state_fill_origin_relative_set` (`Engrave_Part_State *state`, double x, double y)
engrave_part_state_fill_origin_relative_set - Set the relative value of the fill.
- EAPI void `engrave_part_state_fill_size_relative_set` (`Engrave_Part_State *eps`, double x, double y)
engrave_part_state_fill_size_relative_set - Set the fills relative size on the state.
- EAPI void `engrave_part_state_fill_origin_offset_set` (`Engrave_Part_State *eps`, int x, int y)
engrave_part_state_fill_origin_offset_set - Set the offset of the fill position
- EAPI void `engrave_part_state_fill_size_offset_set` (`Engrave_Part_State *eps`, int x, int y)
engrave_part_state_fill_size_offset_set - Set the fill size offset.

- EAPI void `engrave_part_state_text_text_set` (`Engrave_Part_State *eps`, const char *text)
engrave_part_state_text_text_set - Set the text string of the state.
- EAPI void `engrave_part_state_text_elipsis_set` (`Engrave_Part_State *eps`, double val)
engrave_part_state_text_elipsis_set - Set the text elipsis of the state.
- EAPI void `engrave_part_state_text_text_append` (`Engrave_Part_State *eps`, const char *text)
engrave_part_state_text_text_append - Appends the text string of the state.
- EAPI void `engrave_part_state_text_text_class_set` (`Engrave_Part_State *eps`, const char *text)
engrave_part_state_text_text_class_set - Set the text class of the state.
- EAPI void `engrave_part_state_text_style_set` (`Engrave_Part_State *eps`, const char *style)
engrave_part_state_text_style_set - Set the text style of the state.
- EAPI void `engrave_part_state_text_text_source_set` (`Engrave_Part_State *eps`, const char *source)
engrave_part_state_text_text_source_set - Set the text source of the state.
- EAPI void `engrave_part_state_text_source_set` (`Engrave_Part_State *eps`, const char *source)
engrave_part_state_text_source_set - Set the source of the text.
- EAPI void `engrave_part_state_text_font_set` (`Engrave_Part_State *eps`, const char *font)
engrave_part_state_text_font_set - Set the font for the text
- EAPI void `engrave_part_state_text_size_set` (`Engrave_Part_State *eps`, int size)
engrave_part_state_text_size_set - Set the text size.
- EAPI void `engrave_part_state_text_fit_set` (`Engrave_Part_State *eps`, int x, int y)
engrave_part_state_text_fit_set - Set the fit parameter of the text
- EAPI void `engrave_part_state_text_min_set` (`Engrave_Part_State *eps`, int x, int y)
engrave_part_state_text_min_set - Set the text min value.
- EAPI void `engrave_part_state_text_max_set` (`Engrave_Part_State *eps`, int x, int y)
engrave_part_state_text_max_set - Set the text max value.
- EAPI void `engrave_part_state_text_align_set` (`Engrave_Part_State *eps`, double x, double y)
engrave_part_state_text_align_set - Set the text alignment of the state.
- EAPI void `engrave_part_state_copy` (`Engrave_Part_State *from`, `Engrave_Part_State *to`)
engrave_part_state_copy - Copy the values from from to to

- EAPI const char * [engrave_part_state_name_get](#) (Engrave_Part_State *eps, double *val)
engrave_part_state_name_get - get the state name/value
- EAPI int [engrave_part_state_visible_get](#) (Engrave_Part_State *eps)
engrave_part_state_visible_get - get the visiblity flag
- EAPI void [engrave_part_state_align_get](#) (Engrave_Part_State *eps, double *x, double *y)
engrave_part_state_align_get - get the alignment of the state
- EAPI void [engrave_part_state_step_get](#) (Engrave_Part_State *eps, double *x, double *y)
engrave_part_state_step_get - get the stepping info for the state
- EAPI void [engrave_part_state_min_size_get](#) (Engrave_Part_State *eps, int *w, int *h)
engrave_part_state_min_size_get - get the minimum size of the state
- EAPI void [engrave_part_state_fixed_size_get](#) (Engrave_Part_State *eps, int *w, int *h)
engrave_part_state_fixed_size_get - get the fixed size of the state
- EAPI void [engrave_part_state_max_size_get](#) (Engrave_Part_State *eps, int *w, int *h)
engrave_part_state_max_size_get - get the maximum size of the state
- EAPI void [engrave_part_state_aspect_get](#) (Engrave_Part_State *eps, double *w, double *h)
engrave_part_state_aspect_get - get the aspect ratio for the state
- EAPI Engrave_Aspect_Preference [engrave_part_state_aspect_preference_get](#) (Engrave_Part_State *eps)
engrave_part_state_aspect_preference_get - get the aspect preference
- EAPI void [engrave_part_state_rel1_relative_get](#) (Engrave_Part_State *eps, double *x, double *y)
engrave_part_state_rel1_relative_get - get the rel1 relative values
- EAPI void [engrave_part_state_rel1_offset_get](#) (Engrave_Part_State *eps, int *x, int *y)
engrave_part_state_rel1_offset_get - get the rel1 offset values
- EAPI const char * [engrave_part_state_rel1_to_x_get](#) (Engrave_Part_State *eps)
engrave_part_state_rel1_to_x_get - get the to_x value for rel1
- EAPI const char * [engrave_part_state_rel1_to_y_get](#) (Engrave_Part_State *eps)
engrave_part_state_rel1_to_y_get - get the to_y value for rel1
- EAPI void [engrave_part_state_rel2_relative_get](#) (Engrave_Part_State *eps, double *x, double *y)

engrave_part_state_rel2_relative_get - get the rel2 relative values

- EAPI void `engrave_part_state_rel2_offset_get` (`Engrave_Part_State` *eps, int *x, int *y)

engrave_part_state_rel2_offset_get - get the rel2 offset values

- EAPI const char * `engrave_part_state_rel2_to_x_get` (`Engrave_Part_State` *eps)

engrave_part_state_rel2_to_x_get - get the to_x value for rel2

- EAPI const char * `engrave_part_state_rel2_to_y_get` (`Engrave_Part_State` *eps)

engrave_part_state_rel2_to_y_get - get the to_y value for rel2

- EAPI const char * `engrave_part_state_color_class_get` (`Engrave_Part_State` *eps)

engrave_part_state_color_class_get - get the color class for the state

- EAPI void `engrave_part_state_color_get` (`Engrave_Part_State` *eps, int *r, int *g, int *b, int *a)

engrave_part_state_color_get - get the color value from the state

- EAPI void `engrave_part_state_color2_get` (`Engrave_Part_State` *eps, int *r, int *g, int *b, int *a)

engrave_part_state_color2_get - get the color2 value from the state

- EAPI void `engrave_part_state_color3_get` (`Engrave_Part_State` *eps, int *r, int *g, int *b, int *a)

engrave_part_state_color3_get - get the color3 value from the state

- EAPI `Engrave_Image` * `engrave_part_state_image_normal_get` (`Engrave_Part_State` *eps)

engrave_part_state_image_normal_get - get the Engrave_Image for the normal setting

- EAPI void `engrave_part_state_image_border_get` (`Engrave_Part_State` *eps, int *l, int *r, int *t, int *b)

engrave_part_state_image_border_get - get the border settings

- EAPI int `engrave_part_state_image_middle_get` (`Engrave_Part_State` *eps)

engrave_part_state_image_middle_get - get the middle settings

- EAPI const char * `engrave_part_state_text_text_get` (`Engrave_Part_State` *eps)

engrave_part_state_text_text_get - get the text value

- EAPI const char * `engrave_part_state_text_text_class_get` (`Engrave_Part_State` *eps)

engrave_part_state_text_text_class_get - get the text class

- EAPI const char * `engrave_part_state_text_style_get` (`Engrave_Part_State` *eps)

engrave_part_state_text_style_get - get the text style

- EAPI const char * `engrave_part_state_text_text_source_get` (`Engrave_Part_State` *eps)

engrave_part_state_text_text_source_get - get the text text_source

- EAPI const char * [engrave_part_state_text_source_get](#) (Engrave_Part_State *eps)
engrave_part_state_text_source_get - get the text text_source
- EAPI const char * [engrave_part_state_text_font_get](#) (Engrave_Part_State *eps)
engrave_part_state_text_font_get - get the text font
- EAPI int [engrave_part_state_text_size_get](#) (Engrave_Part_State *eps)
engrave_part_state_text_size_get - get the font size
- EAPI void [engrave_part_state_text_fit_get](#) (Engrave_Part_State *eps, int *x, int *y)
engrave_part_state_text_fit_get - get the text fit settings
- EAPI double [engrave_part_state_text_elipsis_get](#) (Engrave_Part_State *eps)
engrave_part_state_text_elipsis_get - get the text elipsis settings
- EAPI void [engrave_part_state_text_min_get](#) (Engrave_Part_State *eps, int *x, int *y)
engrave_part_state_text_min_get - get the text min size
- EAPI void [engrave_part_state_text_align_get](#) (Engrave_Part_State *eps, double *x, double *y)
engrave_part_state_text_align_get - get the text alignment settings
- EAPI int [engrave_part_state_tweens_count](#) (Engrave_Part_State *eps)
engrave_part_state_tweens_count - get the number of tweens in the state
- EAPI void [engrave_part_state_tween_foreach](#) (Engrave_Part_State *eps, void(*func)([Engrave_Image](#) *, void *), void *data)
engrave_part_state_tween_foreach - Iterate over the tween list
- EAPI void [engrave_part_state_gradient_spectrum_set](#) (Engrave_Part_State *eps, const char *spec)
engrave_part_state_gradient_spectrum_set - set the spectrum of the gradient on this state
- EAPI const char * [engrave_part_state_gradient_spectrum_get](#) (Engrave_Part_State *eps)
engrave_part_state_gradient_spectrum_get - get the spectrum of the gradient on this state
- EAPI void [engrave_part_state_gradient_type_set](#) (Engrave_Part_State *eps, const char *type)
engrave_part_state_gradient_type - set the type of the gradient on this state
- EAPI const char * [engrave_part_state_gradient_type_get](#) (Engrave_Part_State *eps)
engrave_part_state_gradient_type_get - get the type of the gradient on this state

5.11.1 Function Documentation

5.11.1.1 EAPI void engrave_part_state_align_get ([Engrave_Part_State](#) * *eps*,
double * *x*, double * *y*)

engrave_part_state_align_get - get the alignment of the state

Parameters:

eps,: The Engrave_Part_State to get the info from

x,: Where to store the x value or 0.5 if not set

y,: Where to store the y value or 0.5 if not set

Returns:

Returns no value.

5.11.1.2 EAPI void engrave_part_state_align_set ([Engrave_Part_State](#) * *eps*,
double *x*, double *y*)

engrave_part_state_align_set - Set the alignment of the state.

Parameters:

eps,: The Engrave_Part_State to set the value too.

x,: The x alignment value to set

y,: The y alignment value to set

Returns:

Returns no value

5.11.1.3 EAPI void engrave_part_state_aspect_get ([Engrave_Part_State](#) * *eps*,
double * *w*, double * *h*)

engrave_part_state_aspect_get - get the aspect ratio for the state

Parameters:

eps,: The Engrave_Part_State to get the aspect info from

h,: Where to store the w value

w,: Where to store the h value

Returns:

Returns no value

5.11.1.4 EAPI Engrave_Aspect_Preference engrave_part_state_aspect_preference_get (Engrave_Part_State * *eps*)

engrave_part_state_aspect_preference_get - get the aspect preference

Parameters:

eps,: The Engrave_Part_State to get the aspect preference from

Returns:

Returns the Engrave_Aspect_Preference for the state

5.11.1.5 EAPI void engrave_part_state_aspect_preference_set (Engrave_Part_State * *eps*, Engrave_Aspect_Preference *prefer*)

engrave_part_state_aspect_preference_set - Set the aspect preference.

Parameters:

eps,: The Engrave_Part_State to set the value too.

prefer,: The Engrave_Aspect_Preference value to set on the state.

Returns:

Returns no value

5.11.1.6 EAPI void engrave_part_state_aspect_set (Engrave_Part_State * *eps*, double *w*, double *h*)

engrave_part_state_aspect_set - Set the width and height of the aspect ratio

Parameters:

eps,: The Engrave_Part_State to set the value too.

w,: The width value to set.

h,: The height value to set.

Returns:

Returns no value

5.11.1.7 EAPI void engrave_part_state_color2_get ([Engrave_Part_State](#) * *eps*, int * *r*, int * *g*, int * *b*, int * *a*)

engrave_part_state_color2_get - get the color2 value from the state

Parameters:

eps,: The Engrave_Part_State to get the values from
r,: Where to store the r value
g,: Where to store the g value
b,: Where to store the b value
a,: Where to store the a value

Returns:

Returns no value.

5.11.1.8 EAPI void engrave_part_state_color2_set ([Engrave_Part_State](#) * *eps*, int *r*, int *g*, int *b*, int *a*)

engrave_part_state_color2_set - Set the colour2 of the state

Parameters:

eps,: The Engrave_Part_State to set the value too.
r,: The red value (set -1 to keep the current value).
g,: The green value (set -1 to keep the current value).
b,: the blue value (set -1 to keep the current value).
a,: The alpha value (set -1 to keep the current value).

Returns:

Returns no value

5.11.1.9 EAPI void engrave_part_state_color3_get ([Engrave_Part_State](#) * *eps*, int * *r*, int * *g*, int * *b*, int * *a*)

engrave_part_state_color3_get - get the color3 value from the state

Parameters:

eps,: The Engrave_Part_State to get the values from
r,: Where to store the r value
g,: Where to store the g value
b,: Where to store the b value
a,: Where to store the a value

Returns:

Returns no value.

5.11.1.10 EAPI void engrave_part_state_color3_set ([Engrave_Part_State](#) * *eps*, int *r*, int *g*, int *b*, int *a*)

engrave_part_state_color3_set - Set the colour 3 of the state.

Parameters:

eps,: The Engrave_Part_State to set the value too.
r,: The red value (set -1 to keep the current value).
g,: The green value (set -1 to keep the current value).
b,: the blue value (set -1 to keep the current value).
a,: The alpha value (set -1 to keep the current value).

Returns:

Returns no value

5.11.1.11 EAPI const char* engrave_part_state_color_class_get ([Engrave_Part_State](#) * *eps*)

engrave_part_state_color_class_get - get the color class for the state

Parameters:

eps,: The Engrave_Part_State to get the colour class from

Returns:

Returns the colour class on success or NULL on failure.

5.11.1.12 EAPI void engrave_part_state_color_class_set ([Engrave_Part_State](#) * *eps*, const char * *class*)

engrave_part_state_color_class_set - Set the colour class on the state.

Parameters:

eps,: The Engrave_Part_State to set the value too.
class,: The colour class to associate with the state.

Returns:

Returns no value

5.11.1.13 EAPI void engrave_part_state_color_get ([Engrave_Part_State](#) * *eps*, int * *r*, int * *g*, int * *b*, int * *a*)

engrave_part_state_color_get - get the color value from the state

Parameters:

eps,: The Engrave_Part_State to get the values from
r,: Where to store the r value
g,: Where to store the g value
b,: Where to store the b value
a,: Where to store the a value

Returns:

Returns no value.

5.11.1.14 EAPI void engrave_part_state_color_set ([Engrave_Part_State](#) * *eps*, int *r*, int *g*, int *b*, int *a*)

engrave_part_state_color_set - Set the colour of the state.

Parameters:

eps,: The Engrave_Part_State to set the value too.
r,: The red value (set -1 to keep the current value).
g,: The green value (set -1 to keep the current value).
b,: the blue value (set -1 to keep the current value).
a,: The alpha value (set -1 to keep the current value).

Returns:

Returns no value

5.11.1.15 EAPI void engrave_part_state_copy ([Engrave_Part_State](#) * *from*, [Engrave_Part_State](#) * *to*)

engrave_part_state_copy - Copy the values from *from* to *to*

Parameters:

from,: The Engrave_Part_State from which to retrieve the values
to,: The Engrave_Part_State to place the values into

Returns:

Returns no value.

5.11.1.16 EAPI void engrave_part_state_fill_origin_offset_set
(Engrave_Part_State * *eps*, int *x*, int *y*)

engrave_part_state_fill_origin_offset_set - Set the offset of the fill position

Parameters:

eps,: The Engrave_Part_State to set the value too.

x,: The x value.

y,: The y value.

Returns:

Returns no value

5.11.1.17 EAPI void engrave_part_state_fill_origin_relative_set
(Engrave_Part_State * *eps*, double *x*, double *y*)

engrave_part_state_fill_origin_relative_set - Set the relative value of the fill.

Parameters:

eps,: The Engrave_Part_State to set the value too.

x,: The x value to set

y,: The y value to set

Returns:

Returns no value

5.11.1.18 EAPI void engrave_part_state_fill_size_offset_set
(Engrave_Part_State * *eps*, int *x*, int *y*)

engrave_part_state_fill_size_offset_set - Set the fill size offset.

Parameters:

eps,: The Engrave_Part_State to set the value too.

x,: The x value.

y,: The y value.

Returns:

Returns no value

5.11.1.19 EAPI void engrave_part_state_fill_size_relative_set ([Engrave_Part_State](#) * *eps*, double *x*, double *y*)

engrave_part_state_fill_size_relative_set - Set the fills relative size on the state.

Parameters:

eps,: The Engrave_Part_State to set the value too.

x,: The x value.

y,: The y value.

Returns:

Returns no value

5.11.1.20 EAPI void engrave_part_state_fill_smooth_set ([Engrave_Part_State](#) * *eps*, int *smooth*)

engrave_part_state_fill_smooth_set - Set the fill smooth value of the state.

Parameters:

eps,: The Engrave_Part_State to set the value too.

smooth,: The smooth value to set.

Returns:

Returns no value

5.11.1.21 EAPI void engrave_part_state_fixed_size_get ([Engrave_Part_State](#) * *eps*, int * *w*, int * *h*)

engrave_part_state_fixed_size_get - get the fixed size of the state

Parameters:

eps,: The Engrave_Part_State to get the min from

w,: Where to store the w value

h,: Where to store the h value

Returns:

Returns no value.

5.11.1.22 EAPI void engrave_part_state_fixed_size_set (Engrave_Part_State * *eps*, int *w*, int *h*)

engrave_part_state_fixed_size_set - Set the fixed size of the state.

Parameters:

eps,: The Engrave_Part_State to set the value too.

w,: for the width

h,: for the height

Returns:

Returns no value

5.11.1.23 EAPI void engrave_part_state_free (Engrave_Part_State * *eps*)

engrave_part_state_free - free the state

Parameters:

eps,: The Engrave_Part_State to free

Returns:

Returns no value.

5.11.1.24 EAPI const char* engrave_part_state_gradient_spectrum_get (Engrave_Part_State * *eps*)

engrave_part_state_gradient_spectrum_get - get the spectrum of the gradient on this state

Parameters:

eps,: The Engrave_Part_State to get the spectrum for

Returns:

Returns the name of the spectrum used

5.11.1.25 EAPI void engrave_part_state_gradient_spectrum_set (Engrave_Part_State * *eps*, const char * *spec*)

engrave_part_state_gradient_spectrum_set - set the spectrum of the gradient on this state

Parameters:

eps,: The Engrave_Part_State to set the spectrum for

spec,: The name of the spectrum

Returns:

Returns no value.

5.11.1.26 EAPI const char* engrave_part_state_gradient_type_get
([Engrave_Part_State](#) * *eps*)

engrave_part_state_gradient_type_get - get the type of the gradient on this state

Parameters:

eps,: The Engrave_Part_State to get the type for

Returns:

Returns the type of the gradient

5.11.1.27 EAPI void engrave_part_state_gradient_type_set
([Engrave_Part_State](#) * *eps*, const char * *type*)

engrave_part_state_gradient_type - set the type of the gradient on this state

Parameters:

eps,: The Engrave_Part_State to set the type for

type,: The type of the spectrum

Returns:

Returns no value.

5.11.1.28 EAPI void engrave_part_state_image_border_get
([Engrave_Part_State](#) * *eps*, int * *l*, int * *r*, int * *t*, int * *b*)

engrave_part_state_image_border_get - get the border settings

Parameters:

eps,: The Engrave_Part_State to get the settings from

l,: Where to store the left value

r,: Where to store the right value

t,: Where to store the top value

b,: Where to store the bottom value

Returns:

Returns no value.

5.11.1.29 EAPI void engrave_part_state_image_border_set
(**Engrave_Part_State** * *eps*, int *l*, int *r*, int *t*, int *b*)

engrave_part_state_image_border_set - Set the border on the image.

Parameters:

eps,: The Engrave_Part_State to set the value too.

l,: The left border value.

r,: The right border value.

t,: The top border value.

b,: The bottom border value.

Returns:

Returns no value k

5.11.1.30 EAPI int engrave_part_state_image_middle_get
(**Engrave_Part_State** * *eps*)

engrave_part_state_image_middle_get - get the middle settings

Parameters:

eps,: The Engrave_Part_State to get the settings from

Returns:

Returns the middle value.

5.11.1.31 EAPI void engrave_part_state_image_middle_set
(**Engrave_Part_State** * *eps*, int *middle*)

engrave_part_state_image_middle_set - Set the middle on the image.

Parameters:

eps,: The Engrave_Part_State to set the value too.

middle,: The middle value.

Returns:

Returns no value k

5.11.1.32 EAPI `Engrave_Image*` `engrave_part_state_image_normal_get` (`Engrave_Part_State` * *eps*)

`engrave_part_state_image_normal_get` - get the `Engrave_Image` for the normal setting

Parameters:

eps,: The `Engrave_Part_State` to get the image from

Returns:

Returns the `Engrave_Image` for the normal setting or NULL on failure.

5.11.1.33 EAPI `void` `engrave_part_state_image_normal_set` (`Engrave_Part_State` * *eps*, `Engrave_Image` * *im*)

`engrave_part_state_image_normal_set` - Set the image we are associated with.

Parameters:

eps,: The `Engrave_Part_State` to set the value too.

im,: The `Engrave_Image` attached to this state.

Returns:

Returns no value

5.11.1.34 EAPI `void` `engrave_part_state_image_tween_add` (`Engrave_Part_State` * *eps*, `Engrave_Image` * *im*)

`engrave_part_state_image_tween_add` - Add a tween to the state.

Parameters:

eps,: The `Engrave_Part_State` to set the value too.

im,: The `Engrave_Image` to add to the tween list.

Returns:

Returns no value

5.11.1.35 EAPI `void` `engrave_part_state_max_size_get` (`Engrave_Part_State` * *eps*, `int` * *w*, `int` * *h*)

`engrave_part_state_max_size_get` - get the maximum size of the state

Parameters:

eps,: The Engrave_Part_State to get the max from

w,: Where to store the w value

h,: Where to store the h value

Returns:

Returns no value.

**5.11.1.36 EAPI void engrave_part_state_max_size_set (Engrave_Part_State *
eps, int w, int h)**

engrave_part_state_max_size_set - Set the max size of the state.

Parameters:

eps,: The Engrave_Part_State to set the value too.

w,: The maximum width of the state.

h,: The maximum height of the state.

Returns:

Returns no value

**5.11.1.37 EAPI void engrave_part_state_min_size_get (Engrave_Part_State *
eps, int * w, int * h)**

engrave_part_state_min_size_get - get the minimum size of the state

Parameters:

eps,: The Engrave_Part_State to get the min from

w,: Where to store the w value

h,: Where to store the h value

Returns:

Returns no value.

**5.11.1.38 EAPI void engrave_part_state_min_size_set (Engrave_Part_State *
eps, int w, int h)**

engrave_part_state_min_size_set - Set the min size of the state.

Parameters:

eps,: The Engrave_Part_State to set the value too.

w,: The minimum width to set

h,: The minimum height to set

Returns:

Returns no value

5.11.1.39 EAPI `const char* engrave_part_state_name_get (Engrave_Part_State * eps, double * val)`

engrave_part_state_name_get - get the state name/value

Parameters:

eps,: The Engrave_Part_State to get the values from

val,: Where to put the value setting

Returns:

Returns the state name, or NULL on failure.

5.11.1.40 EAPI `void engrave_part_state_name_set (Engrave_Part_State * eps, const char * name, double value)`

engrave_part_state_name_set - Set the name of the state

Parameters:

eps,: The Engrave_Part_State to set the value too.

name,: The name to set to the state.

value,: The value to set to the state.

Returns:

Returns no value

5.11.1.41 EAPI `Engrave_Part_State* engrave_part_state_new (void)`

engrave_part_state_new - create a new Engrave_Part_State object.

Returns:

Returns a pointer to a newly allocated Engrave_Part_State object on success or NULL on failure

**5.11.1.42 EAPI void* engrave_part_state_parent_get (Engrave_Part_State *
eps)**

engrave_part_state_parent_get - get the parent part of this state

Parameters:

eps,: The Engrave_Part_State to get the parent from

Returns:

Returns the Engrave_Part parent pointer, or NULL if none set.

**5.11.1.43 EAPI void engrave_part_state_parent_set (Engrave_Part_State *
eps, void * ep)**

engrave_part_state_parent_set - set the parent part on this state

Parameters:

eps,: The Engrave_Part_State to set the parent into

ep,: The Engrave_Part to set as the parent

Returns:

Returns no value.

**5.11.1.44 EAPI void engrave_part_state_rell_offset_get (Engrave_Part_State
* eps, int * x, int * y)**

engrave_part_state_rell_offset_get - get the rell offset values

Parameters:

eps,: The Engrave_Part_State to get the values from

x,: Where to store the x value

y,: Where to store the y value

Returns:

Returns no value.

**5.11.1.45 EAPI void engrave_part_state_rell_offset_set (Engrave_Part_State
* eps, int x, int y)**

engrave_part_state_rell_offset_set - Set the rell offset value.

Parameters:

eps,: The Engrave_Part_State to set the value too.

x,: The x value to set.

y,: The y value to set.

Returns:

Returns no value

5.11.1.46 EAPI void engrave_part_state_rell_relative_get (Engrave_Part_State * *eps*, double * *x*, double * *y*)

engrave_part_state_rell_relative_get - get the rell relative values

Parameters:

eps,: The Engrave_Part_State to get the values from

x,: Where to store the x value

y,: Where to store the y value

Returns:

Returns no value.

5.11.1.47 EAPI void engrave_part_state_rell_relative_set (Engrave_Part_State * *eps*, double *x*, double *y*)

engrave_part_state_rell_relative_set - Set the relative value of rell.

Parameters:

eps,: The Engrave_Part_State to set the value too.

x,: The x value to set.

y,: The y value to set.

Returns:

Returns no value

5.11.1.48 EAPI void engrave_part_state_rell_to_set (Engrave_Part_State * *eps*, const char * *to*)

engrave_part_state_rell_to_set - Set the rell to value for both x and y dirs.

Parameters:

eps,: The Engrave_Part_State to set the value too.

to,: The name of the part we are relative too.

Returns:

Returns no value

5.11.1.49 EAPI const char* engrave_part_state_rell_to_x_get
(**Engrave_Part_State** * *eps*)

engrave_part_state_rell_to_x_get - get the to_x value for rell

Parameters:

eps,: The Engrave_Part_State to get the value from

Returns:

Returns the to_x value or NULL on failure.

5.11.1.50 EAPI void engrave_part_state_rell_to_x_set (**Engrave_Part_State** *
eps, const char * *to*)

engrave_part_state_rell_to_x_set - Set what rell is relative too in the x dir.

Parameters:

eps,: The Engrave_Part_State to set the value too.

to,: The name of the part we are relative too.

Returns:

Returns no value

5.11.1.51 EAPI const char* engrave_part_state_rell_to_y_get
(**Engrave_Part_State** * *eps*)

engrave_part_state_rell_to_y_get - get the to_y value for rell

Parameters:

eps,: The Engrave_Part_State to get the value from

Returns:

Returns the to_y value or NULL on failure.

5.11.1.52 EAPI void engrave_part_state_rel1_to_y_set ([Engrave_Part_State](#) * *eps*, const char * *to*)

engrave_part_state_rel1_to_y_set - Set what rel1 is relative too in the y dir.

Parameters:

eps,: The Engrave_Part_State to set the value too.

to,: The name of the part we are relative too.

Returns:

Returns no value

5.11.1.53 EAPI void engrave_part_state_rel2_offset_get ([Engrave_Part_State](#) * *eps*, int * *x*, int * *y*)

engrave_part_state_rel2_offset_get - get the rel2 offset values

Parameters:

eps,: The Engrave_Part_State to get the values from

x,: Where to store the x value

y,: Where to store the y value

Returns:

Returns no value.

5.11.1.54 EAPI void engrave_part_state_rel2_offset_set ([Engrave_Part_State](#) * *eps*, int *x*, int *y*)

engrave_part_state_rel2_offset_set - Set the rel2 offset value.

Parameters:

eps,: The Engrave_Part_State to set the value too.

x,: The x value to set.

y,: The y value to set.

Returns:

Returns no value

5.11.1.55 EAPI void engrave_part_state_rel2_relative_get
(**Engrave_Part_State** * *eps*, double * *x*, double * *y*)

engrave_part_state_rel2_relative_get - get the rel2 relative values

Parameters:

eps,: The Engrave_Part_State to get the values from

x,: Where to store the x value

y,: Where to store the y value

Returns:

Returns no value.

5.11.1.56 EAPI void engrave_part_state_rel2_relative_set
(**Engrave_Part_State** * *eps*, double *x*, double *y*)

engrave_part_state_rel2_relative_set - Set the rel2 relative value

Parameters:

eps,: The Engrave_Part_State to set the value too.

x,: The x value to set.

y,: The y value to set.

Returns:

Returns no value

5.11.1.57 EAPI void engrave_part_state_rel2_to_set (**Engrave_Part_State** *
eps, const char * *to*)

engrave_part_state_rel2_to_set - Set the rel2 to value for both x and y dirs.

Parameters:

eps,: The Engrave_Part_State to set the value too.

to,: The name of the part we are relative too.

Returns:

Returns no value

5.11.1.58 EAPI const char* engrave_part_state_rel2_to_x_get
([Engrave_Part_State](#) * *eps*)

engrave_part_state_rel2_to_x_get - get the to_x value for rel2

Parameters:

eps,: The Engrave_Part_State to get the value from

Returns:

Returns the to_x value or NULL on failure.

5.11.1.59 EAPI void engrave_part_state_rel2_to_x_set ([Engrave_Part_State](#) *
eps, const char * *to*)

engrave_part_state_rel2_to_x_set - Set what rel2 is relative too in the x dir.

Parameters:

eps,: The Engrave_Part_State to set the value too.

to,: The name of the part we are relative too.

Returns:

Returns no value

5.11.1.60 EAPI const char* engrave_part_state_rel2_to_y_get
([Engrave_Part_State](#) * *eps*)

engrave_part_state_rel2_to_y_get - get the to_y value for rel2

Parameters:

eps,: The Engrave_Part_State to get the value from

Returns:

Returns the to_y value or NULL on failure.

5.11.1.61 EAPI void engrave_part_state_rel2_to_y_set ([Engrave_Part_State](#) *
eps, const char * *to*)

engrave_part_state_rel2_to_y_set - Set what rel2 is relative too in the y dir.

Parameters:

eps,: The Engrave_Part_State to set the value too.

to,: The name of the part we are relative too.

Returns:

Returns no value

5.11.1.62 EAPI void engrave_part_state_step_get (Engrave_Part_State * *eps*, double * *x*, double * *y*)

engrave_part_state_step_get - get the stepping info for the state

Parameters:

eps,: The Engrave_Part_State to get the info from

x,: Where to store the x value

y,: Where to store the y value

Returns:

Returns no value.

5.11.1.63 EAPI void engrave_part_state_step_set (Engrave_Part_State * *eps*, double *x*, double *y*)

engrave_part_state_step_set - Set the step value of the state

Parameters:

eps,: The Engrave_Part_State to set the value too.

x,: The x step value to set

y,: The y step value to set

Returns:

Returns no value

5.11.1.64 EAPI void engrave_part_state_text_align_get (Engrave_Part_State * *eps*, double * *x*, double * *y*)

engrave_part_state_text_align_get - get the text alignment settings

Parameters:

eps,: The Engrave_Part_State to get the align from

x,: Where to store the x value

y,: Where to store the y value

Returns:

Returns no value.

5.11.1.65 EAPI void engrave_part_state_text_align_set ([Engrave_Part_State](#) * *eps*, double *x*, double *y*)

engrave_part_state_text_align_set - Set the text alignment of the state.

Parameters:

eps,: The Engrave_Part_State to set the value too.

x,: The x alignment setting.

y,: the y alignment setting.

Returns:

Returns no value

5.11.1.66 EAPI double engrave_part_state_text_elipsis_get ([Engrave_Part_State](#) * *eps*)

engrave_part_state_text_elipsis_get - get the text elipsis settings

Parameters:

eps,: The Engrave_Part_State to get the values from

Returns:

The elipsis value

5.11.1.67 EAPI void engrave_part_state_text_elipsis_set ([Engrave_Part_State](#) * *eps*, double *val*)

engrave_part_state_text_elipsis_set - Set the text elipsis of the state.

Parameters:

eps,: The Engrave_Part_State to set the value too.

val,: The value to set to the state.

Returns:

Returns no value

**5.11.1.68 EAPI void engrave_part_state_text_fit_get (Engrave_Part_State *
eps, int * *x*, int * *y*)**

engrave_part_state_text_fit_get - get the text fit settings

Parameters:

eps,: The Engrave_Part_State to get the values from

x,: Where to store the x value

y,: Where to store the y value

Returns:

Returns no value.

**5.11.1.69 EAPI void engrave_part_state_text_fit_set (Engrave_Part_State *
eps, int *x*, int *y*)**

engrave_part_state_text_fit_set - Set the fit parameter of the text

Parameters:

eps,: The Engrave_Part_State to set the value too.

x,: The x value to set.

y,: The y value to set.

Returns:

Returns no value

**5.11.1.70 EAPI const char* engrave_part_state_text_font_get
(Engrave_Part_State * *eps*)**

engrave_part_state_text_font_get - get the text font

Parameters:

eps,: The Engrave_Part_State to get the font from

Returns:

Returns the font on success or NULL on failure.

5.11.1.71 EAPI void engrave_part_state_text_font_set ([Engrave_Part_State](#) *
eps, const char * *font*)

engrave_part_state_text_font_set - Set the font for the text

Parameters:

eps,: The Engrave_Part_State to set the value too.

font,: The font to apply to the text.

Returns:

Returns no value

5.11.1.72 EAPI void engrave_part_state_text_max_set ([Engrave_Part_State](#) *
eps, int *x*, int *y*)

engrave_part_state_text_max_set - Set the text max value.

Parameters:

eps,: The Engrave_Part_State to set the value too.

x,: The x value to set.

y,: The y value to set.

Returns:

Returns no value

5.11.1.73 EAPI void engrave_part_state_text_min_get ([Engrave_Part_State](#) *
eps, int * *x*, int * *y*)

engrave_part_state_text_min_get - get the text min size

Parameters:

eps,: The Engrave_Part_State to get the values from

x,: Where to store the x value

y,: Where to store the y value

Returns:

Returns no value.

5.11.1.74 EAPI void engrave_part_state_text_min_set ([Engrave_Part_State](#) * *eps*, int *x*, int *y*)

engrave_part_state_text_min_set - Set the text min value.

Parameters:

eps,: The Engrave_Part_State to set the value too.

x,: The x value to set.

y,: The y value to set.

Returns:

Returns no value

5.11.1.75 EAPI int engrave_part_state_text_size_get ([Engrave_Part_State](#) * *eps*)

engrave_part_state_text_size_get - get the font size

Parameters:

eps,: The Engrave_Part_State to get the font size from

Returns:

Returns the font size.

5.11.1.76 EAPI void engrave_part_state_text_size_set ([Engrave_Part_State](#) * *eps*, int *size*)

engrave_part_state_text_size_set - Set the text size.

Parameters:

eps,: The Engrave_Part_State to set the value too.

size,: The font size to set.

Returns:

Returns no value

5.11.1.77 EAPI const char* engrave_part_state_text_source_get ([Engrave_Part_State](#) * *eps*)

engrave_part_state_text_source_get - get the text text_source

Parameters:

eps,: The Engrave_Part_State to get the value from

Returns:

Returns the source or NULL on failure.

5.11.1.78 EAPI void engrave_part_state_text_source_set ([Engrave_Part_State](#) * *eps*, const char * *source*)

engrave_part_state_text_source_set - Set the source of the text.

Parameters:

eps,: The Engrave_Part_State to set the value too.

source,: The text source to set to the state.

Returns:

Returns no value

5.11.1.79 EAPI const char* engrave_part_state_text_style_get ([Engrave_Part_State](#) * *eps*)

engrave_part_state_text_style_get - get the text style

Parameters:

eps,: The Engrave_Part_State to get the value from

Returns:

Returns the text style or NULL on failure.

5.11.1.80 EAPI void engrave_part_state_text_style_set ([Engrave_Part_State](#) * *eps*, const char * *text_style*)

engrave_part_state_text_style_set - Set the text style of the state.

Parameters:

eps,: The Engrave_Part_State to set the value too.

text_style,: The text style to set to the state.

Returns:

Returns no value

5.11.1.81 EAPI void engrave_part_state_text_text_append
(**Engrave_Part_State** * *eps*, const char * *text*)

engrave_part_state_text_text_append - Appends the text string of the state.

Parameters:

eps,: The Engrave_Part_State to set the value too.

text,: The text string to append.

Returns:

Returns no value

5.11.1.82 EAPI const char* engrave_part_state_text_text_class_get
(**Engrave_Part_State** * *eps*)

engrave_part_state_text_text_class_get - get the text class

Parameters:

eps,: The Engrave_Part_State to get the value from

Returns:

Returns the text class or NULL on failure.

5.11.1.83 EAPI void engrave_part_state_text_text_class_set
(**Engrave_Part_State** * *eps*, const char * *text_class*)

engrave_part_state_text_text_class_set - Set the text class of the state.

Parameters:

eps,: The Engrave_Part_State to set the value too.

text_class,: The text class to set to the state.

Returns:

Returns no value

5.11.1.84 EAPI const char* engrave_part_state_text_text_get
(**Engrave_Part_State** * *eps*)

engrave_part_state_text_text_get - get the text value

Parameters:

eps,: The Engrave_Part_State to get the text from

Returns:

Returns the text value on success or NULL on failure.

5.11.1.85 EAPI void engrave_part_state_text_text_set (Engrave_Part_State * *eps*, const char * *text*)

engrave_part_state_text_text_set - Set the text string of the state.

Parameters:

eps,: The Engrave_Part_State to set the value too.

text,: The text string to set.

Returns:

Returns no value

5.11.1.86 EAPI const char* engrave_part_state_text_text_source_get (Engrave_Part_State * *eps*)

engrave_part_state_text_text_source_get - get the text text_source

Parameters:

eps,: The Engrave_Part_State to get the value from

Returns:

Returns the text source or NULL on failure.

5.11.1.87 EAPI void engrave_part_state_text_text_source_set (Engrave_Part_State * *eps*, const char * *text_source*)

engrave_part_state_text_text_source_set - Set the text source of the state.

Parameters:

eps,: The Engrave_Part_State to set the value too.

text_source,: The text source to set to the state.

Returns:

Returns no value

5.11.1.88 EAPI void engrave_part_state_tween_foreach ([Engrave_Part_State](#) * *eps*, void (*)([Engrave_Image](#) *, void *) *func*, void * *data*)

engrave_part_state_tween_foreach - Iterate over the tween list

Parameters:

eps,: The Engrave_Part_State to get the tweens from

func,: The function to call for each tween

data,: User data to pass to the function

Returns:

Returns no value.

5.11.1.89 EAPI int engrave_part_state_tweens_count ([Engrave_Part_State](#) * *eps*)

engrave_part_state_tweens_count - get the number of tweens in the state

Parameters:

eps,: The Engrave_Part_State to check for tweens

Returns:

Returns number of tweens, 0 otherwise

5.11.1.90 EAPI int engrave_part_state_visible_get ([Engrave_Part_State](#) * *eps*)

engrave_part_state_visible_get - get the visibility flag

Parameters:

eps,: The Engrave_Part_State to get the visibility from

Returns:

Returns the visibility of the state

5.11.1.91 EAPI void engrave_part_state_visible_set ([Engrave_Part_State](#) * *eps*, int *visible*)

engrave_part_state_visible_set - Set the visibility of the state

Parameters:

eps,: The Engrave_Part_State to set the value too.

visible,: The visibility value to set to the state

Returns:

Returns no value

5.12 Engrave_Program: Functions to work with Engrave_Program objects.

Data Structures

- struct [_Engrave_Program](#)
Contains all the info to define a program.

Typedefs

- typedef [_Engrave_Program](#) [Engrave_Program](#)
The Engrave_Program typedef.

Functions

- EAPI [Engrave_Program *](#) [engrave_program_new](#) (void)
engrave_program_new - create a new Engrave_Program object.
- EAPI void [engrave_program_free](#) ([Engrave_Program *](#)ep)
engrave_program_free - free the program
- EAPI void [engrave_program_parent_set](#) ([Engrave_Program *](#)ep, void *eg)
engrave_program_parent_set - set the parent of the program
- EAPI void * [engrave_program_parent_get](#) ([Engrave_Program *](#)ep)
engrave_program_parent_get - get the parent pointer
- EAPI void [engrave_program_script_set](#) ([Engrave_Program *](#)ep, const char *script)
engrave_program_script_set - attach a script to the program.
- EAPI void [engrave_program_name_set](#) ([Engrave_Program *](#)ep, const char *name)
engrave_program_name_set - set the name of the program.
- EAPI void [engrave_program_signal_set](#) ([Engrave_Program *](#)ep, const char *signal)
engrave_program_signal_set - set the given signal on the program
- EAPI void [engrave_program_source_set](#) ([Engrave_Program *](#)ep, const char *source)
engrave_program_source_set - set the source of the program.
- EAPI void [engrave_program_target_add](#) ([Engrave_Program *](#)ep, const char *target)
engrave_program_target_add - set the target on the program.
- EAPI void [engrave_program_after_add](#) ([Engrave_Program *](#)ep, const char *after)
engrave_program_after_add - add the after to the program.

- EAPI void `engrave_program_in_set` (`Engrave_Program` *ep, double from, double range)
engrave_program_in_set - set the in value of the program.
- EAPI void `engrave_program_action_set` (`Engrave_Program` *ep, `Engrave_Action` action, const char *state, const char *state2, double value, double value2)
engrave_program_action_set - set the action for the program.
- EAPI void `engrave_program_transition_set` (`Engrave_Program` *ep, `Engrave_Transition` trans, double duration)
engrave_program_transition_set - set the transition on the program
- EAPI const char * `engrave_program_name_get` (`Engrave_Program` *ep)
engrave_program_name_get - Get the program name
- EAPI const char * `engrave_program_signal_get` (`Engrave_Program` *ep)
engrave_program_signal_get - Get the program signal
- EAPI const char * `engrave_program_source_get` (`Engrave_Program` *ep)
engrave_program_source_get - Get the program source
- EAPI void `engrave_program_action_get` (`Engrave_Program` *ep, `Engrave_Action` *action, char *state, char *state2, int state_len, int state2_len, double *value, double *value2)
engrave_program_action_get - Get the action information for the program
- EAPI void `engrave_program_transition_get` (`Engrave_Program` *ep, `Engrave_Transition` *trans, double *duration)
engrave_program_transition_get - Get the programs transition information
- EAPI const char * `engrave_program_script_get` (`Engrave_Program` *ep)
engrave_program_script_get - Get the script from the program
- EAPI void `engrave_program_in_get` (`Engrave_Program` *ep, double *from, double *range)
engrave_program_in_get - Get the in data for the program
- EAPI int `engrave_program_afters_count` (`Engrave_Program` *ep)
engrave_program_afters_count - get the number of afters
- EAPI int `engrave_program_targets_count` (`Engrave_Program` *ep)
engrave_program_targets_count - count the number of targets
- EAPI void `engrave_program_target_foreach` (`Engrave_Program` *ep, void(*func)(const char *, void *), void *data)
engrave_program_target_foreach - Iterate over each target
- EAPI void `engrave_program_after_foreach` (`Engrave_Program` *ep, void(*func)(const char *, void *), void *data)
engrave_program_after_foreach - Iterate over each after

5.12.1 Function Documentation

5.12.1.1 EAPI void engrave_program_action_get ([Engrave_Program](#) * *ep*,
Engrave_Action * *action*, char * *state*, char * *state2*, int *state_len*, int
state2_len, double * *value*, double * *value2*)

engrave_program_action_get - Get the action information for the program

Parameters:

ep,: The Engrave_Program to get the action information from
action,: Where to store the action setting
state,: Buffer to put state into
state2,: Buffer to put state2 into
state_len,: The length of the state buffer
state2_len,: The length of the state2 buffer
value,: Where to put the value setting
value2,: Where to put the value2 setting

Returns:

Returns no value.

5.12.1.2 EAPI void engrave_program_action_set ([Engrave_Program](#) * *ep*,
Engrave_Action *action*, const char * *state*, const char * *state2*, double
value, double *value2*)

engrave_program_action_set - set the action for the program.

Parameters:

ep,: The Engrave_Program to set the action on.
action,: The Engrave_Action to set on the program.
state,: The state value to set.
state2,: The state2 value to set.
value,: The value value to set.
value2,: The value2 value to set.

Returns:

Returns no value.

5.12.1.3 EAPI void engrave__program__after__add ([Engrave__Program](#) * *ep*, const char * *after*)

engrave__program__after__add - add the after to the program.

Parameters:

ep,: The Engrave__Program to add the after too.

after,: The after to add to the program.

Returns:

Returns no value.

5.12.1.4 EAPI void engrave__program__after__foreach ([Engrave__Program](#) * *ep*, void(*) (const char *, void *) *func*, void * *data*)

engrave__program__after__foreach - Iterate over each after

Parameters:

ep,: The Engrave__Program to get the afters from

func,: The function to call for each after

data,: User data

Returns:

Returns no value.

5.12.1.5 EAPI int engrave__program__afters__count ([Engrave__Program](#) * *ep*)

engrave__program__afters__count - get the number of afters

Parameters:

ep,: The Engrave__Program to check for afters

Returns:

Returns number of afters, 0 otherwise

5.12.1.6 EAPI void engrave__program__free ([Engrave__Program](#) * *ep*)

engrave__program__free - free the program

Parameters:

ep,: The Engrave__Program to free

Returns:

Returns no value.

5.12.1.7 EAPI void engrave_program_in_get (Engrave_Program * ep, double * from, double * range)

engrave_program_in_get - Get the in data for the program

Parameters:

ep,: The Engrave_Program to get the in data from

from,: The place to put the from value

range,: The place to put the range value

Returns:

Returns no value.

5.12.1.8 EAPI void engrave_program_in_set (Engrave_Program * ep, double from, double range)

engrave_program_in_set - set the in value of the program.

Parameters:

ep,: The Engrave_Program to set the in value on.

from,: The from value to set.

range,: The range value to set.

Returns:

Returns no value.

5.12.1.9 EAPI const char* engrave_program_name_get (Engrave_Program * ep)

engrave_program_name_get - Get the program name

Parameters:

ep,: The Engrave_Program to get the name from

Returns:

Returns the name of the program or NULL on failure.

5.12.1.10 EAPI void engrave_program_name_set (Engrave_Program * *ep*, const char * *name*)

engrave_program_name_set - set the name of the program.

Parameters:

ep,: The Engrave_Program to attach the name too.
name,: The name to attach to the program.

Returns:

Returns no value.

5.12.1.11 EAPI Engrave_Program* engrave_program_new (void)

engrave_program_new - create a new Engrave_Program object.

Returns:

Returns a pointer to the newly allocated Engrave_Program object on success or NULL on failure.

5.12.1.12 EAPI void* engrave_program_parent_get (Engrave_Program * *ep*)

engrave_program_parent_get - get the parent pointer

Parameters:

ep,: The Engrave_Program to get the parent from

Returns:

Returns a pointer to the Engrave_Group parent, or NULL if none set

5.12.1.13 EAPI void engrave_program_parent_set (Engrave_Program * *ep*, void * *eg*)

engrave_program_parent_set - set the parent of the program

Parameters:

ep,: The Engrave_Program to set the parent into
eg,: The Engrave_Group to set as parent

Returns:

Returns no value.

5.12.1.14 EAPI const char* engrave_program_script_get (Engrave_Program * *ep*)

engrave_program_script_get - Get the script from the program

Parameters:

ep,: The Engrave_Program to get the script from

Returns:

Returns the script or NULL on failure.

5.12.1.15 EAPI void engrave_program_script_set (Engrave_Program * *ep*, const char * *script*)

engrave_program_script_set - attach a script to the program.

Parameters:

ep,: The Engrave_Program to attach the script too.

script,: The script to attach to the program.

Returns:

Returns no value.

5.12.1.16 EAPI const char* engrave_program_signal_get (Engrave_Program * *ep*)

engrave_program_signal_get - Get the program signal

Parameters:

ep,: The Engrave_Program to get the signal from

Returns:

Returns the signal of the program or NULL on failure.

5.12.1.17 EAPI void engrave_program_signal_set (Engrave_Program * *ep*, const char * *signal*)

engrave_program_signal_set - set the given signal on the program

Parameters:

ep,: The Engrave_Program to attach the signal too.

signal,: The signal to attach to the program.

Returns:

Returns no value.

5.12.1.18 EAPI `const char* engrave_program_source_get (Engrave_Program * ep)`

`engrave_program_source_get` - Get the program source

Parameters:

ep,: The Engrave_Program to get the source from

Returns:

Returns the source of the program or NULL on failure.

5.12.1.19 EAPI `void engrave_program_source_set (Engrave_Program * ep, const char * source)`

`engrave_program_source_set` - set the source of the program.

Parameters:

ep,: The Engrave_Program to attach the source too.

source,: The name to attach to the group.

Returns:

Returns no value.

5.12.1.20 EAPI `void engrave_program_target_add (Engrave_Program * ep, const char * target)`

`engrave_program_target_add` - set the target on the program.

Parameters:

ep,: The Engrave_Program to set the target on.

target,: The target to set on the program.

Returns:

Returns no value.

5.12.1.21 EAPI void engrave_program_target_foreach (Engrave_Program * *ep*, void(*)(const char *, void *) *func*, void * *data*)

engrave_program_target_foreach - Iterate over each target

Parameters:

ep,: The Engrave_Program to get the targets from

func,: The function to call for each target

data,: User data

Returns:

Returns no value.

5.12.1.22 EAPI int engrave_program_targets_count (Engrave_Program * *ep*)

engrave_program_targets_count - count the number of targets

Parameters:

ep,: The Engrave_Program to check for targets

Returns:

Returns number of targets, 0 otherwise

5.12.1.23 EAPI void engrave_program_transition_get (Engrave_Program * *ep*, Engrave_Transition * *trans*, double * *duration*)

engrave_program_transition_get - Get the programs transition information

Parameters:

ep,: The Engrave_Program to get the transition information from

trans,: Where to store the transition setting

duration,: Where to store the duration setting

Returns:

Returns no value.

5.12.1.24 EAPI void engrave_program_transition_set (Engrave_Program * *ep*, Engrave_Transition *trans*, double *duration*)

engrave_program_transition_set - set the transition on the program

Parameters:

- ep,:* The Engrave_Program to set the transition on.
- trans,:* The Engrave_Transition to set on the program.
- duration,:* The duration of the given transition.

Returns:

Returns no value.

5.13 Engrave_Spectrum: Functions to work with engrave spectrum objects

Data Structures

- struct `_Engrave_Spectrum`
Stores the needed spectrum information.
- struct `_Engrave_Spectrum_Color`
Stores the spectrum color information.

Typedefs

- typedef `_Engrave_Spectrum` `Engrave_Spectrum`
The Engrave_Spectrum typedef.

Functions

- EAPI `Engrave_Spectrum * engrave_spectrum_new` ()
engrave_spectrum_new - create a new Engrave_Spectrum object
- EAPI void `engrave_spectrum_free` (`Engrave_Spectrum *es`)
engrave_spectrum_free - frees a Engrave_Spectrum object
- EAPI `Engrave_Spectrum_Color * engrave_spectrum_color_new` (int r, int g, int b, int a, int d)
engrave_spectrum_color_new - Creates an Engrave_Spectrum_Color object
- EAPI void `engrave_spectrum_color_free` (`Engrave_Spectrum_Color *esc`)
engrave_spectrum_color_free - frees a Engrave_Spectrum_Color object
- EAPI void `engrave_spectrum_parent_set` (`Engrave_Spectrum *es`, void *parent)
engrave_spectrum_parent_set - sets the parent of the Engrave_Spectrum object
- EAPI void `engrave_spectrum_color_add` (`Engrave_Spectrum *spec`, `Engrave_Spectrum_Color *col`)
engrave_spectrum_color_add - Adds a color to a Engrave_Spectrum object
- EAPI void `engrave_spectrum_color_del` (`Engrave_Spectrum *spec`, `Engrave_Spectrum_Color *col`)
engrave_spectrum_color_del - Removes a color from a Engrave_Spectrum object
- EAPI const char * `engrave_spectrum_name_get` (`Engrave_Spectrum *es`)
engrave_spectrum_name_get - gets the name of the Engrave_Spectrum object

- EAPI void `engrave_spectrum_name_set` (`Engrave_Spectrum *es`, const char *name)
engrave_spectrum_name_set - sets the name of the Engrave_Spectrum object
- EAPI int `engrave_spectrum_color_count` (`Engrave_Spectrum *es`)
engrave_spectrum_color_count - returns the number of colors inside an Engrave_Spectrum object
- EAPI `Engrave_Spectrum_Color * engrave_spectrum_color_nth` (`Engrave_Spectrum *es`, int n)
engrave_spectrum_color_nth - gets the nth color of the Engrave_Spectrum object

5.13.1 Function Documentation

5.13.1.1 EAPI void `engrave_spectrum_color_add` (`Engrave_Spectrum * es`, `Engrave_Spectrum_Color * col`)

`engrave_spectrum_color_add` - Adds a color to a `Engrave_Spectrum` object

Parameters:

- es*,: The spectrum object to add to
col,: The `Engrave_Spectrum_Color` object to add

Returns:

Returns no value

5.13.1.2 EAPI int `engrave_spectrum_color_count` (`Engrave_Spectrum * es`)

`engrave_spectrum_color_count` - returns the number of colors inside an `Engrave_Spectrum` object

Parameters:

- es*,: The spectrum object

Returns:

Returns the number of colors

5.13.1.3 EAPI void engrave_spectrum_color_del (Engrave_Spectrum * *es*, Engrave_Spectrum_Color * *col*)

engrave_spectrum_color_del - Removes a color from a Engrave_Spectrum object

Parameters:

- es*,: The spectrum object to remove from
- col*,: The Engrave_Spectrum_Color object to remove

Returns:

Returns no value

5.13.1.4 EAPI void engrave_spectrum_color_free (Engrave_Spectrum_Color * *esc*)

engrave_spectrum_color_free - frees a Engrave_Spectrum_Color object

Parameters:

- esc*,: The spectrum color object to free

Returns:

Returns no value

5.13.1.5 EAPI Engrave_Spectrum_Color* engrave_spectrum_color_new (int *r*, int *g*, int *b*, int *a*, int *d*)

engrave_spectrum_color_new - Creates an Engrave_Spectrum_Color object

Parameters:

- r*,: The red value
- g*,: The green value
- b*,: The blue value
- a*,: The alpha value
- d*,: TBD

Returns:

Returns a new Engrave_Spectrum_Color object

**5.13.1.6 EAPI `Engrave_Spectrum_Color*` `engrave_spectrum_color_nth`
(`Engrave_Spectrum` * *es*, int *n*)**

`engrave_spectrum_color_nth` - gets the *n*th color of the `Engrave_Spectrum` object

Parameters:

es,: The spectrum object

n,: The *n*th color to get

Returns:

Returns the *n*th `Engrave_Spectrum_Color` of the object

5.13.1.7 EAPI void `engrave_spectrum_free` (`Engrave_Spectrum` * *es*)

`engrave_spectrum_free` - frees a `Engrave_Spectrum` object

Parameters:

es,: The spectrum object to free

Returns:

Returns no value

**5.13.1.8 EAPI const char* `engrave_spectrum_name_get` (`Engrave_Spectrum` *
es)**

`engrave_spectrum_name_get` - gets the name of the `Engrave_Spectrum` object

Parameters:

es,: The spectrum object

Returns:

Returns the name of the object

**5.13.1.9 EAPI void `engrave_spectrum_name_set` (`Engrave_Spectrum` * *es*, const
char * *name*)**

`engrave_spectrum_name_set` - sets the name of the `Engrave_Spectrum` object

Parameters:

es,: The spectrum object

name,: The name to set

Returns:

Returns no value

5.13.1.10 EAPI [Engrave_Spectrum](#)* engrave_spectrum_new ()

engrave_spectrum_new - create a new Engrave_Spectrum object

Returns:

Returns a pointer to a newly allocated object on success, or NULL

5.13.1.11 EAPI void engrave_spectrum_parent_set ([Engrave_Spectrum](#) * *es*, void * *parent*)

engrave_spectrum_parent_set - sets the parent of the Engrave_Spectrum object

Parameters:

es,: The spectrum object

parent,: The parent of the spectrum object

Returns:

Returns no value

5.14 Engrave_Style: Functions to work with Engrave_Style blocks.

Data Structures

- struct [_Engrave_Style](#)
Contains the needed style values.
- struct [_Engrave_Tag](#)
Contains the tag information.

Typedefs

- typedef [_Engrave_Style](#) [Engrave_Style](#)
The Engrave_Style typedef.

Functions

- EAPI [Engrave_Style](#) * [engrave_style_new](#) ()
engrave_style_new - create a new Engrave_Style object.

5.14.1 Function Documentation

5.14.1.1 EAPI [Engrave_Style](#)* [engrave_style_new](#) ()

[engrave_style_new](#) - create a new [Engrave_Style](#) object.

Returns:

Returns a pointer to a newly allocated [Engrave_Style](#) object on success or NULL on failure.

Chapter 6

Engrave Data Structure Documentation

6.1 __Engrave__Data Struct Reference

Stores the information for the data object.

Data Fields

- char * [key](#)
The data key.
- char * [value](#)
The data string value.
- int [int_value](#)
The data int value.
- void * [parent](#)
A pointer to the parent.

6.1.1 Detailed Description

Stores the information for the data object.

6.2 `_Engrave_File` Struct Reference

Structure to hold the file information.

Data Fields

- `char * font_dir`
The font directory.
- `char * im_dir`
The image directory.
- `Evas_List * images`
The list of images in the file.
- `Evas_List * fonts`
The list of fonts in the file.
- `Evas_List * spectra`
The list of spectra in the file.
- `Evas_List * styles`
The list of styles in the file.
- `Evas_List * data`
The list of data items in the file.
- `Evas_List * groups`
The list of groups in the file.

6.2.1 Detailed Description

Structure to hold the file information.

6.3 `_Engrave_Font Struct` Reference

Stores the needed font information.

Data Fields

- `char * name`
The font alias.
- `char * path`
The font relative path.
- `void * parent`
Pointer to parent.

6.3.1 Detailed Description

Stores the needed font information.

6.4 `_Engrave_Group` Struct Reference

Stores the needed group information.

Data Fields

- `char * name`
The group name.
- `char * alias`
The group alias.
- ```
struct {
 int w
 The width.
 int h
 The height.
} min
```

  
*The max/min size values for the group.*
- ```
struct {  
    int w  
        The width.  
    int h  
        The height.  
} max
```


The max/min size values for the group.
- `Evas_List * parts`
The list of parts in this group.
- `Evas_List * programs`
The list of programs in this group.
- `Evas_List * data`
The list of data used in this group.
- `char * script`
The script attached to this group.
- `void * parent`
Pointer to parent.

6.4.1 Detailed Description

Stores the needed group information.

6.5 `_Engrave_Image` Struct Reference

Contains the needed image values.

Data Fields

- `char * name`
The image basename.
- `Engrave_Image_Type type`
The image storage type.
- `double value`
Compression value for image.
- `void * parent`
Pointer to parent.

6.5.1 Detailed Description

Contains the needed image values.

6.6 `_Engrave_Part` Struct Reference

All values needed to define a part.

Data Fields

- `char * name`
The part name.
- `Engrave_Part_Type type`
The part type.
- `Engrave_Text_Effect effect`
The parts text effect.
- `int mouse_events`
The mouse events flag.
- `int repeat_events`
The repeat events flag.
- `char * clip_to`
The part to clip too.
- ```
struct {
 signed char x
 Can drag in x direction.
 signed char y
 Can drag in y direction.
 struct {
 int x
 The x value.
 int y
 The y value.
 } step
 The step and count values for the draggable.
 struct {
 int x
 The x value.
 int y
 The y value.
 } count
 The step and count values for the draggable.
 char * confine
 The confine setting for the draggable.
} draggable
```

*The draggable settings for the part.*

- Evas\_List \* [states](#)  
*The list of states associated with the part.*
- Engrave\_Part\_State \* [current\\_state](#)  
*The current state.*
- Evas\_Object \* [object](#)  
*The evas object used to display this state.*
- void \* [parent](#)  
*Pointer to parent.*
- Evas\_Coord [x](#)  
*The x position to place the object.*
- Evas\_Coord [y](#)  
*The y position to place the object.*
- Evas\_Coord [w](#)  
*The width of the object.*
- Evas\_Coord [h](#)  
*The height of the object.*

### 6.6.1 Detailed Description

All values needed to define a part.

## 6.7 `_Engrave_Part_State` Struct Reference

Stores the information for the `Engrave_Part_State` object.

### Data Fields

- `char * name`  
*The name of the state.*
- `double value`  
*The value assigned to the state.*
- `unsigned char visible`  
*Flag for if the state is visible.*
- ```
struct {  
    double y  
        x value  
} align
```


Store the align and step information for the state.
- ```
struct {
 double y
 x value
} step
```

  
*Store the align and step information for the state.*
- ```
struct {  
    int w  
        width value  
    int h  
        height value  
} min
```


The min and max settings for the state.
- ```
struct {
 int w
 width value
 int h
 height value
```

```
} max
```

*The min and max settings for the state.*

- 

```
struct {
 int w
 width value
 int h
 height value
} fixed
```

*The min and max settings for the state.*

- 

```
struct {
 double w
 The width.
 double h
 The height.
} aspect
```

*The aspect ratio settings for the state.*

- 

```
struct {
 struct {
 double x
 The x position.
 double y
 The y position.
 } relative
 The relative positions.
 struct {
 int x
 The x value.
 int y
 The y value.
 } offset
 The offset values.
 char * to_x
 The x part relativity.
 char * to_y
 The y part relativity.
} rell
```

*The relative orientation settings for the state.*

- 

```
struct {
 struct {
 double x
```

```

 The x position.
double y
 The y position.
} relative
 The relative positions.
struct {
 int x
 The x value.
 int y
 The y value.
} offset
 The offset values.
char * to_x
 The x part relativity.
char * to_y
 The y part relativity.
} rel2

```

*The relative orientation settings for the state.*

•

```

struct {
 char * spectrum
 Name of the spectrum.
 char * type
 Type of the gradient.
} gradient

```

*gradient information for the state*

•

```

struct {
 Engrave_Image * normal
 The image itself.
 Evas_List * tween
 List of tween images.
 int middle
 The middle information.
 struct {
 int l
 Left value.
 int r
 Right value.
 int t
 Top value.
 int b
 Bottom value.
 } border
 The border information.
} image

```

*image information for the state*

- char \* `color_class`

*Colour class setting.*

- 

```
struct {
 int r
 Red setting.
 int g
 Green setting.
 int b
 Blue setting.
 int a
 Alpha setting.
} color
```

*Colour settings for the state.*

- 

```
struct {
 int r
 Red setting.
 int g
 Green setting.
 int b
 Blue setting.
 int a
 Alpha setting.
} color2
```

*Colour settings for the state.*

- 

```
struct {
 int r
 Red setting.
 int g
 Green setting.
 int b
 Blue setting.
 int a
 Alpha setting.
} color3
```

*Colour settings for the state.*

- 

```
struct {
 int smooth
 The smooth setting.
 struct {
```

```

 double x
 The x value.
 double y
 The y value.
} pos_rel
 relativity settings for the fill position
struct {
 double x
 The x value.
 double y
 The y value.
} rel
 relativity settings for the fill position
struct {
 int x
 The x value.
 int y
 The y value.
} pos_abs
 absolute settings for the fill position
struct {
 int x
 The x value.
 int y
 The y value.
} abs
 absolute settings for the fill position
} fill

```

*Fill information for the state.*

•

```

struct {
 char * text
 The text string.
 char * text_class
 The text class.
 char * text_source
 The text source.
 char * source
 The text source ?
 char * font
 The font to use.
 char * style
 The text style.
 int size
 Size to display as.
 double elipsis
 Elipsis.
 struct {
 int x
 The x value.
 int y

```

```

 The y value.
} fit
 Fit, max and min size information.
struct {
 int x
 The x value.
 int y
 The y value.
} min
 Fit, max and min size information.
struct {
 int x
 The x value.
 int y
 The y value.
} max
 Fit, max and min size information.
struct {
 double x
 The x value.
 double y
 The y value.
} align
 Alignment settings for the text.
} text

```

*State text settings.*

- void \* **parent**  
*Pointer to parent.*

### 6.7.1 Detailed Description

Stores the information for the Engrave\_Part\_State object.



## 6.8 `_Engrave_Program` Struct Reference

Contains all the info to define a program.

### Data Fields

- `char * name`  
*The program name.*
- `char * signal`  
*The signal to act on.*
- `char * source`  
*The souce to act on.*
- `Evas_List * targets`  
*The list of targets.*
- `Evas_List * afters`  
*The list of afters.*
- struct {  
        double `from`  
            *The from in setting.*  
        double `range`  
            *The range in setting.*  
    } `in`  
  
    *Stores the in settings for the program.*
- `Engrave_Action action`  
*The action associated with the program.*
- `char * state`  
*The state setting for the action.*
- `char * state2`  
*The state2 setting for the action.*
- `double value`  
*The value setting for the action.*
- `double value2`  
*The value2 setting for the action.*
- `Engrave_Transition transition`

*The transition type of the program.*

- double [duration](#)

*The duration of the transition.*

- char \* [script](#)

*The script associated with the program.*

- void \* [parent](#)

*Pointer to parent.*

### 6.8.1 Detailed Description

Contains all the info to define a program.

## 6.9 \_\_Engrave\_\_Spectrum Struct Reference

Stores the needed spectrum information.

### Data Fields

- char \* [name](#)  
*The spectrum alias.*
- void \* [parent](#)  
*The spectrum parent.*
- Evas\_List \* [colors](#)  
*The spectrum colors.*

### 6.9.1 Detailed Description

Stores the needed spectrum information.

## 6.10 `_Engrave_Spectrum_Color` Struct Reference

Stores the spectrum color information.

### 6.10.1 Detailed Description

Stores the spectrum color information.

## 6.11 \_\_Engrave\_\_ Style Struct Reference

Contains the needed style values.

### Data Fields

- char \* [name](#)  
*The style name.*
- char \* [base](#)  
*The style base.*
- Evas\_List \* [tags](#)  
*The style tags.*
- void \* [parent](#)  
*Pointer to parent.*

#### 6.11.1 Detailed Description

Contains the needed style values.

## 6.12 `_Engrave_Tag` Struct Reference

Contains the tag information.

### 6.12.1 Detailed Description

Contains the tag information.

## Chapter 7

# Engrave File Documentation

### 7.1 Engrave.h File Reference

The file that should be included by any project using Engrave.

#### 7.1.1 Detailed Description

The file that should be included by any project using Engrave.

It provides all the necessary headers and includes to work with Engrave.

## 7.2 `engrave_canvas.h` File Reference

Engrave\_Canvas object functions. Contains all of the functions related to the engrave canvas object.

### Functions

- EAPI Evas\_Object \* [engrave\\_canvas\\_new](#) (Evas \*e)  
*engrave\_canvas\_new - create a new canvas*
- EAPI void [engrave\\_canvas\\_file\\_set](#) (Evas\_Object \*o, [Engrave\\_File](#) \*ef)  
*engrave\_canvas\_fil\_set - set the file to display in this canvas*
- EAPI void [engrave\\_canvas\\_current\\_group\\_set](#) (Evas\_Object \*o, [Engrave\\_Group](#) \*eg)  
*engrave\_canvas\_current\_group\_set - set the group to display in the canvas*

### 7.2.1 Detailed Description

Engrave\_Canvas object functions. Contains all of the functions related to the engrave canvas object.



## 7.3 engrave\_data.h File Reference

Engrave\_Data block object Contains all of the functions to manipulate Engrave\_Data objects.

### Data Structures

- struct [\\_Engrave\\_Data](#)  
*Stores the information for the data object.*

### Typedefs

- typedef [\\_Engrave\\_Data](#) [Engrave\\_Data](#)  
*The Engrave\_Data typedef.*

### Functions

- EAPI [Engrave\\_Data](#) \* [engrave\\_data\\_new](#) (const char \*key, const char \*value)  
*engrave\_data\_new - create a new data block*
- EAPI void [engrave\\_data\\_free](#) ([Engrave\\_Data](#) \*ed)  
*engrave\_data\_free - free the given data block*
- EAPI void [engrave\\_data\\_parent\\_set](#) ([Engrave\\_Data](#) \*ed, void \*parent)  
*engrave\_data\_parent\_set - set the parent pointer on this node This will set the parent pointer, in the case of Engrave\_Data this can be either an Engrave\_File or an Engrave\_Group.*
- EAPI void \* [engrave\\_data\\_parent\\_get](#) ([Engrave\\_Data](#) \*ed)  
*engrave\_data\_parent\_get - get the parent pointer on this node*
- EAPI const char \* [engrave\\_data\\_key\\_get](#) ([Engrave\\_Data](#) \*ed)  
*engrave\_data\_key\_get - get the key for this data*
- EAPI const char \* [engrave\\_data\\_value\\_get](#) ([Engrave\\_Data](#) \*ed)  
*engrave\_data\_value\_get - get the string value of the data object*
- EAPI int [engrave\\_data\\_int\\_value\\_get](#) ([Engrave\\_Data](#) \*ed)  
*engrave\_data\_int\_value\_get - get the int value of the data object*
- EAPI void [engrave\\_data\\_value\\_set](#) ([Engrave\\_Data](#) \*ed, const char \*value)  
*engrave\_data\_value\_set - set the value of the data object*
- EAPI void [engrave\\_data\\_int\\_value\\_set](#) ([Engrave\\_Data](#) \*ed, int value)  
*engrave\_data\_int\_value\_set - set the int value of the data object*

### 7.3.1 Detailed Description

Engrave\_Data block object Contains all of the functions to manipulate Engrave\_Data objects.

## 7.4 engrave\_enums.h File Reference

Contains all of the enumerations used in Engrave.

### Enumerations

- enum `_Engrave_Image_Type` {  
    `ENGRAVE_IMAGE_TYPE_RAW`,  
    `ENGRAVE_IMAGE_TYPE_COMP`,  
    `ENGRAVE_IMAGE_TYPE_LOSSY` }
- enum `_Engrave_Part_Type` {  
    `ENGRAVE_PART_TYPE_IMAGE`,  
    `ENGRAVE_PART_TYPE_TEXT`,  
    `ENGRAVE_PART_TYPE_TEXTBLOCK`,  
    `ENGRAVE_PART_TYPE_RECT`,  
    `ENGRAVE_PART_TYPE_SWALLOW`,  
    `ENGRAVE_PART_TYPE_GRADIENT` }
- enum `_Engrave_Text_Effect` {  
    `ENGRAVE_TEXT_EFFECT_NONE`,  
    `ENGRAVE_TEXT_EFFECT_PLAIN`,  
    `ENGRAVE_TEXT_EFFECT_OUTLINE`,  
    `ENGRAVE_TEXT_EFFECT_SOFT_OUTLINE`,  
    `ENGRAVE_TEXT_EFFECT_SHADOW`,  
    `ENGRAVE_TEXT_EFFECT_OUTLINE_SHADOW`,  
    `ENGRAVE_TEXT_EFFECT_SOFT_SHADOW`,  
    `ENGRAVE_TEXT_EFFECT_OUTLINE_SOFT_SHADOW` }
- enum `_Engrave_Action` {  
    `ENGRAVE_ACTION_STATE_SET`,  
    `ENGRAVE_ACTION_STOP`,  
    `ENGRAVE_ACTION_SIGNAL_EMIT`,  
    `ENGRAVE_ACTION_DRAG_VAL_SET`,  
    `ENGRAVE_ACTION_DRAG_VAL_STEP`,  
    `ENGRAVE_ACTION_DRAG_VAL_PAGE`,  
    `ENGRAVE_ACTION_SCRIPT` }
- enum `_Engrave_Transition` {  
    `ENGRAVE_TRANSITION_LINEAR`,  
    `ENGRAVE_TRANSITION_SINUSOIDAL`,  
    `ENGRAVE_TRANSITION_ACCELERATE`,  
    `ENGRAVE_TRANSITION_DECELERATE` }

- `enum _Engrave_Aspect_Preference {`  
    `ENGRAVE_ASPECT_PREFERENCE_NONE,`  
    `ENGRAVE_ASPECT_PREFERENCE_VERTICAL,`  
    `ENGRAVE_ASPECT_PREFERENCE_HORIZONTAL,`  
    `ENGRAVE_ASPECT_PREFERENCE_BOTH }`

### 7.4.1 Detailed Description

Contains all of the enumerations used in Engrave.

## 7.5 engrave\_file.h File Reference

Engrave\_File object functions. Contains all of the functions related to the Engrave file object.

### Data Structures

- struct [\\_Engrave\\_File](#)  
*Structure to hold the file information.*

### Typedefs

- typedef [\\_Engrave\\_File](#) [Engrave\\_File](#)  
*The Engrave\_File typedef.*

### Functions

- EAPI [Engrave\\_File](#) \* [engrave\\_file\\_new](#) (void)  
*engrave\_file\_new - create a new Engrave\_File object*
- EAPI void [engrave\\_file\\_free](#) ([Engrave\\_File](#) \*ef)  
*engrave\_file\_free - free the file object*
- EAPI void [engrave\\_file\\_image\\_dir\\_set](#) ([Engrave\\_File](#) \*ef, const char \*dir)  
*engrave\_file\_image\_dir\_set - set the image directory for the file*
- EAPI void [engrave\\_file\\_font\\_dir\\_set](#) ([Engrave\\_File](#) \*ef, const char \*dir)  
*engrave\_file\_font\_dir\_set - set the font dir for the file*
- EAPI const char \* [engrave\\_file\\_image\\_dir\\_get](#) ([Engrave\\_File](#) \*ef)  
*engrave\_file\_image\_dir\_get - get the image directory*
- EAPI const char \* [engrave\\_file\\_font\\_dir\\_get](#) ([Engrave\\_File](#) \*ef)  
*engrave\_file\_font\_dir\_get - get the font directory*
- EAPI void [engrave\\_file\\_font\\_add](#) ([Engrave\\_File](#) \*e, [Engrave\\_Font](#) \*ef)  
*engrave\_file\_font\_add - add the font to the engrave file.*
- EAPI void [engrave\\_file\\_spectrum\\_add](#) ([Engrave\\_File](#) \*e, [Engrave\\_Spectrum](#) \*es)  
*engrave\_file\_spectrum\_add - add the spectrum to the engrave file.*
- EAPI void [engrave\\_file\\_style\\_add](#) ([Engrave\\_File](#) \*e, [Engrave\\_Style](#) \*es)  
*engrave\_file\_style\_add - add the style to the engrave file.*
- EAPI void [engrave\\_file\\_image\\_add](#) ([Engrave\\_File](#) \*ef, [Engrave\\_Image](#) \*ei)

*engrave\_file\_image\_add* - add the image to the engrave file.

- EAPI void [engrave\\_file\\_data\\_add](#) ([Engrave\\_File](#) \*ef, [Engrave\\_Data](#) \*ed)  
*engrave\_file\_data\_add* - add the data to the engrave file.
- EAPI void [engrave\\_file\\_group\\_add](#) ([Engrave\\_File](#) \*ef, [Engrave\\_Group](#) \*eg)  
*engrave\_file\_group\_add* - add the group to the given file
- EAPI [Engrave\\_Style](#) \* [engrave\\_file\\_style\\_last\\_get](#) ([Engrave\\_File](#) \*ef)  
*engrave\_file\_style\_last\_get* - returns the last style in the file
- EAPI [Engrave\\_Spectrum](#) \* [engrave\\_file\\_spectrum\\_last\\_get](#) ([Engrave\\_File](#) \*ef)  
*engrave\_file\_spectrum\_last\_get* - returns the last spectrum in the file
- EAPI [Engrave\\_Group](#) \* [engrave\\_file\\_group\\_last\\_get](#) ([Engrave\\_File](#) \*ef)  
*engrave\_file\_group\_last\_get* - returns the last group in the file
- EAPI [Engrave\\_Group](#) \* [engrave\\_file\\_group\\_by\\_name\\_find](#) ([Engrave\\_File](#) \*ef, const char \*name)  
*engrave\_file\_group\_by\_name\_find* - returns the *Engrave\_Group* with the given name.
- EAPI [Engrave\\_Font](#) \* [engrave\\_file\\_font\\_by\\_name\\_find](#) ([Engrave\\_File](#) \*ef, const char \*name)  
*engrave\_file\_font\_by\_name\_find* - find the *Engrave\_Font* by name
- EAPI [Engrave\\_Image](#) \* [engrave\\_file\\_image\\_by\\_name\\_find](#) ([Engrave\\_File](#) \*ef, const char \*name)  
*engrave\_file\_image\_by\_name\_find* - returns the *Engrave\_Image* with the given name.
- EAPI int [engrave\\_file\\_images\\_count](#) ([Engrave\\_File](#) \*ef)  
*engrave\_file\_images\_count* - get the number of images
- EAPI int [engrave\\_file\\_data\\_count](#) ([Engrave\\_File](#) \*ef)  
*engrave\_file\_data\_count* - count data blocks in file
- EAPI int [engrave\\_file\\_groups\\_count](#) ([Engrave\\_File](#) \*ef)  
*engrave\_file\_groups\_count* - count the groups in the file
- EAPI int [engrave\\_file\\_fonts\\_count](#) ([Engrave\\_File](#) \*ef)  
*engrave\_file\_fonts\_count* - count the fonts in the file
- EAPI int [engrave\\_file\\_spectra\\_count](#) ([Engrave\\_File](#) \*ef)  
*engrave\_file\_spectra\_count* - count the spectra in the file
- EAPI int [engrave\\_file\\_styles\\_count](#) ([Engrave\\_File](#) \*ef)  
*engrave\_file\_styles\_count* - count the styles in the file
- EAPI [Engrave\\_Data](#) \* [engrave\\_file\\_data\\_by\\_key\\_find](#) ([Engrave\\_File](#) \*ef, const char \*key)  
*engrave\_file\_data\_by\_key\_find* - find the *Engrave\_Data* by key

### 7.5.1 Detailed Description

Engrave\_File object functions. Contains all of the functions related to the Engrave file object.

## 7.6 engrave\_font.h File Reference

Engrave\_Font block Contains all of the functions to manipulate Engrave\_Font blocks.

### Data Structures

- struct [\\_Engrave\\_Font](#)  
*Stores the needed font information.*

### Typedefs

- typedef [\\_Engrave\\_Font](#) [Engrave\\_Font](#)  
*The Engrave\_Font typedef.*

### Functions

- EAPI [Engrave\\_Font](#) \* [engrave\\_font\\_new](#) (const char \*path, const char \*name)  
*engrave\_font\_new - create a new Engrave\_Font object.*
- EAPI void [engrave\\_font\\_free](#) ([Engrave\\_Font](#) \*ef)  
*engrave\_font\_free - free the memory*
- EAPI void [engrave\\_font\\_parent\\_set](#) ([Engrave\\_Font](#) \*font, void \*ef)  
*engrave\_font\_parent\_set - set the parent of the font node*
- EAPI void \* [engrave\\_font\\_parent\\_get](#) ([Engrave\\_Font](#) \*ef)  
*engrave\_font\_parent\_get - get the parent pointer*
- EAPI const char \* [engrave\\_font\\_name\\_get](#) ([Engrave\\_Font](#) \*ef)  
*engrave\_font\_name\_get - get the font name*
- EAPI const char \* [engrave\\_font\\_path\\_get](#) ([Engrave\\_Font](#) \*ef)  
*engrave\_font\_path\_get - get the font path*

#### 7.6.1 Detailed Description

Engrave\_Font block Contains all of the functions to manipulate Engrave\_Font blocks.



## 7.7 engrave\_group.h File Reference

Engrave\_Group functions Contains all of the functions to manipulate Engrave\_Group objects.

### Data Structures

- struct [\\_Engrave\\_Group](#)  
*Stores the needed group information.*

### Typedefs

- typedef [\\_Engrave\\_Group](#) [Engrave\\_Group](#)  
*The Engrave\_Group typedef.*

### Functions

- EAPI [Engrave\\_Group](#) \* [engrave\\_group\\_new](#) (void)  
*engrave\_group\_new - create a new Engrave\_Group object.*
- EAPI void [engrave\\_group\\_free](#) ([Engrave\\_Group](#) \*eg)  
*engrave\_group\_free - free the group data*
- EAPI void [engrave\\_group\\_parent\\_set](#) ([Engrave\\_Group](#) \*eg, void \*ef)  
*engrave\_group\_parent\_set - set the parent pointer*
- EAPI void \* [engrave\\_group\\_parent\\_get](#) ([Engrave\\_Group](#) \*eg)  
*engrave\_group\_parent\_get - get the parent pointer*
- EAPI void [engrave\\_group\\_data\\_add](#) ([Engrave\\_Group](#) \*eg, [Engrave\\_Data](#) \*ed)  
*engrave\_group\_data\_add - add the Engrave\_Data to the group*
- EAPI void [engrave\\_group\\_part\\_add](#) ([Engrave\\_Group](#) \*eg, [Engrave\\_Part](#) \*ep)  
*engrave\_group\_part\_add - add the given part to the group*
- EAPI void [engrave\\_group\\_program\\_add](#) ([Engrave\\_Group](#) \*eg, [Engrave\\_Program](#) \*ep)  
*engrave\_group\_program\_add - add the program to the group*
- EAPI void [engrave\\_group\\_script\\_set](#) ([Engrave\\_Group](#) \*eg, const char \*script)  
*engrave\_group\_script\_set - attach the script to the given group*
- EAPI void [engrave\\_group\\_name\\_set](#) ([Engrave\\_Group](#) \*eg, const char \*name)  
*engrave\_group\_name\_set - set the name of the group to the given name.*
- EAPI void [engrave\\_group\\_alias\\_set](#) ([Engrave\\_Group](#) \*eg, const char \*alias)

*engrave\_group\_alias\_set* - set the alias of the group.

- EAPI void `engrave_group_min_size_set` (`Engrave_Group` \*eg, int w, int h)  
*engrave\_group\_min\_size\_set* - set the min size of the group.
- EAPI void `engrave_group_max_size_set` (`Engrave_Group` \*eg, int w, int h)  
*engrave\_group\_max\_size\_set* - set the max size of the group.
- EAPI `Engrave_Part` \* `engrave_group_part_last_get` (`Engrave_Group` \*eg)  
*engrave\_group\_part\_last\_get* - retrieve the last part in the group.
- EAPI `Engrave_Program` \* `engrave_group_program_last_get` (`Engrave_Group` \*eg)  
*engrave\_group\_program\_last\_get* - retrieve the last program in the group.
- EAPI const char \* `engrave_group_name_get` (`Engrave_Group` \*eg)  
*engrave\_group\_name\_get* - get the name attached to the group
- EAPI const char \* `engrave_group_alias_get` (`Engrave_Group` \*eg)  
*engrave\_group\_alias\_get* - get the alias attached to the group
- EAPI const char \* `engrave_group_script_get` (`Engrave_Group` \*eg)  
*engrave\_group\_script\_get* - get the script attached to the group
- EAPI void `engrave_group_min_size_get` (`Engrave_Group` \*eg, int \*w, int \*h)  
*engrave\_group\_min\_size\_get* - get the min size of the group
- EAPI void `engrave_group_max_size_get` (`Engrave_Group` \*eg, int \*w, int \*h)  
*engrave\_group\_max\_size\_get* - get the max size of the group
- EAPI int `engrave_group_data_count` (`Engrave_Group` \*eg)  
*engrave\_group\_data\_count* - count the data blocks in the group
- EAPI int `engrave_group_parts_count` (`Engrave_Group` \*eg)  
*engrave\_group\_parts\_count* - returns the number of parts in the group
- EAPI int `engrave_group_programs_count` (`Engrave_Group` \*eg)  
*engrave\_group\_programs\_count* - get number of programs in the group
- EAPI void `engrave_group_data_foreach` (`Engrave_Group` \*eg, void(\*func)(`Engrave_Data` \*, void \*), void \*data)  
*engrave\_group\_data\_foreach* - iterate over the data in the group
- EAPI void `engrave_group_parts_foreach` (`Engrave_Group` \*eg, void(\*func)(`Engrave_Part` \*, void \*), void \*data)  
*engrave\_group\_parts\_foreach* - Iterate over the parts in the Engrave\_group
- EAPI void `engrave_group_programs_foreach` (`Engrave_Group` \*eg, void(\*func)(`Engrave_Program` \*, void \*), void \*data)  
*engrave\_group\_programs\_foreach* - Iterate over the groups programs

- EAPI [Engrave\\_Part](#) \* [engrave\\_group\\_part\\_by\\_name\\_find](#) ([Engrave\\_Group](#) \*eg, const char \*part)  
*engrave\_group\_part\_by\_name\_find - find the Engrave\_Part by name*
- EAPI [Engrave\\_Data](#) \* [engrave\\_group\\_data\\_by\\_key\\_find](#) ([Engrave\\_Group](#) \*eg, const char \*key)  
*engrave\_group\_data\_by\_key\_find - find the Engrave\_Data by key*

### 7.7.1 Detailed Description

[Engrave\\_Group](#) functions Contains all of the functions to manipulate [Engrave\\_Group](#) objects.

## 7.8 engrave\_image.h File Reference

Engrave\_Image object functions. Contains all of the functions to manipulate Engrave\_Image objects.

### Data Structures

- struct [\\_Engrave\\_Image](#)  
*Contains the needed image values.*

### Typedefs

- typedef [\\_Engrave\\_Image](#) [Engrave\\_Image](#)  
*The Engrave\_Image typedef.*

### Functions

- EAPI [Engrave\\_Image](#) \* [engrave\\_image\\_new](#) (const char \*name, Engrave\_Image\_Type type, double value)  
*engrave\_image\_new - create a new Engrave\_Image object.*
- EAPI [Engrave\\_Image](#) \* [engrave\\_image\\_dup](#) ([Engrave\\_Image](#) \*from)  
*engrave\_image\_dup - duplicate an Engrave\_Image*
- EAPI void [engrave\\_image\\_free](#) ([Engrave\\_Image](#) \*ef)  
*engrave\_image\_free - free the Engrave\_Image*
- EAPI void [engrave\\_image\\_parent\\_set](#) ([Engrave\\_Image](#) \*ei, void \*ef)  
*engrave\_image\_parent\_set - set the image parent pointer*
- EAPI void \* [engrave\\_image\\_parent\\_get](#) ([Engrave\\_Image](#) \*ei)  
*engrave\_image\_parent\_get - get the parent of the image*
- EAPI const char \* [engrave\\_image\\_name\\_get](#) ([Engrave\\_Image](#) \*ei)  
*engrave\_image\_name\_get - get the image name*
- EAPI Engrave\_Image\_Type [engrave\\_image\\_type\\_get](#) ([Engrave\\_Image](#) \*ei)  
*engrave\_image\_type\_get - get the type of the image*
- EAPI double [engrave\\_image\\_compression\\_value\\_get](#) ([Engrave\\_Image](#) \*ei)  
*engrave\_image\_compression\_value\_get - get the image compression value*

### 7.8.1 Detailed Description

Engrave\_Image object functions. Contains all of the functions to manipulate Engrave\_Image objects.

## 7.9 engrave\_load.h File Reference

Engrave loading functions Contains the functions to load either an EDC or EDJ into Engrave.

### Functions

- EAPI [Engrave\\_File](#) \* [engrave\\_load\\_edc](#) (const char \*file, const char \*imdir, const char \*fontdir)  
*engrave\_load\_edc* - load the given edc file into memory.
- EAPI [Engrave\\_File](#) \* [engrave\\_load\\_edj](#) (const char \*filename)  
*engrave\_load\_edj* - load the given EDJ file into memory.

### 7.9.1 Detailed Description

Engrave loading functions Contains the functions to load either an EDC or EDJ into Engrave.

## 7.10 engrave\_out.h File Reference

Functions to facilitate outputting the Engrave information. Provided the needed functions to output the Engrave information into various formats.

### Functions

- EAPI int [engrave\\_edj\\_output](#) ([Engrave\\_File](#) \*engrave\_file, const char \*path)  
*engrave\_edj\_output* – Create an EDJ file from the in-memory data.
- EAPI int [engrave\\_edc\\_output](#) ([Engrave\\_File](#) \*engrave\_file, const char \*path)  
*engrave\_edc\_output* – Create an EDC file from the in-memory data.

### 7.10.1 Detailed Description

Functions to facilitate outputting the Engrave information. Provided the needed functions to output the Engrave information into various formats.

## 7.11 engrave\_part.h File Reference

Engrave\_Part object functions. Contains all functions to manipulate the Engrave\_Part object.

### Data Structures

- struct [\\_Engrave\\_Part](#)  
*All values needed to define a part.*

### Typedefs

- typedef [\\_Engrave\\_Part](#) [Engrave\\_Part](#)  
*The Engrave\_Part typedef.*

### Functions

- EAPI [Engrave\\_Part](#) \* [engrave\\_part\\_new](#) ([Engrave\\_Part\\_Type](#) type)  
*engrave\_part\_new - create a new part object.*
- EAPI void [engrave\\_part\\_free](#) ([Engrave\\_Part](#) \*ep)  
*engrave\_part\_free - free the part*
- EAPI void [engrave\\_part\\_parent\\_set](#) ([Engrave\\_Part](#) \*ep, void \*eg)  
*engrave\_part\_parent\_set - set the parent pointer in the part*
- EAPI void \* [engrave\\_part\\_parent\\_get](#) ([Engrave\\_Part](#) \*ep)  
*engrave\_part\_parent\_get - get the parent pointer for the part*
- EAPI void [engrave\\_part\\_name\\_set](#) ([Engrave\\_Part](#) \*ep, const char \*name)  
*engrave\_part\_name\_set - set the name of the part.*
- EAPI void [engrave\\_part\\_mouse\\_events\\_set](#) ([Engrave\\_Part](#) \*ep, int val)  
*engrave\_part\_mouse\_events\_set - set the mouse events flag for the part.*
- EAPI void [engrave\\_part\\_repeat\\_events\\_set](#) ([Engrave\\_Part](#) \*ep, int val)  
*engrave\_part\_repeat\_events\_set - set the repeat events flag for the part.*
- EAPI void [engrave\\_part\\_type\\_set](#) ([Engrave\\_Part](#) \*ep, [Engrave\\_Part\\_Type](#) type)  
*engrave\_part\_type\_set - set the type of the part*
- EAPI void [engrave\\_part\\_effect\\_set](#) ([Engrave\\_Part](#) \*ep, [Engrave\\_Text\\_Effect](#) effect)  
*engrave\_part\_effect\_set - set the effect on the given part.*
- EAPI void [engrave\\_part\\_clip\\_to\\_set](#) ([Engrave\\_Part](#) \*ep, const char \*clip\_to)



*engrave\_part\_clip\_set* - set the clip of the given part.

- EAPI [Engrave\\_Part\\_State \\*](#) [engrave\\_part\\_state\\_last\\_get](#) ([Engrave\\_Part \\*](#)ep)  
*engrave\_part\_state\_last\_get* - get the last state in the part.
- EAPI void [engrave\\_part\\_dragable\\_x\\_set](#) ([Engrave\\_Part \\*](#)ep, int x, int step, int count)  
*engrave\_part\_dragable\_x\_set* - set the draggable x value on the part.
- EAPI void [engrave\\_part\\_dragable\\_y\\_set](#) ([Engrave\\_Part \\*](#)ep, int y, int step, int count)  
*engrave\_part\_dragable\_y\_set* - set the draggable y value on the part.
- EAPI void [engrave\\_part\\_dragable\\_confine\\_set](#) ([Engrave\\_Part \\*](#)ep, const char \*confine)  
*engrave\_part\_dragable\_confine\_set* - set the confine of the parts draggable.
- EAPI void [engrave\\_part\\_state\\_add](#) ([Engrave\\_Part \\*](#)ep, [Engrave\\_Part\\_State \\*](#)eps)  
*engrave\_part\_state\_add* - add the state to the part.
- EAPI [Engrave\\_Part\\_State \\*](#) [engrave\\_part\\_state\\_by\\_name\\_value\\_find](#) ([Engrave\\_Part \\*](#)ep, const char \*name, double val)  
*engrave\_part\_state\_by\_name\_value\_find* - Find the Engrave\_Part State that \* matches the given name and value in this part
- EAPI const char \* [engrave\\_part\\_name\\_get](#) ([Engrave\\_Part \\*](#)ep)  
*engrave\_part\_name\_get* - Get the name of the part
- EAPI [Engrave\\_Part\\_Type](#) [engrave\\_part\\_type\\_get](#) ([Engrave\\_Part \\*](#)ep)  
*engrave\_part\_type\_get* - Get the type of the part
- EAPI int [engrave\\_part\\_mouse\\_events\\_get](#) ([Engrave\\_Part \\*](#)ep)  
*engrave\_part\_mouse\_events\_get* - Get the mouse events setting
- EAPI int [engrave\\_part\\_repeat\\_events\\_get](#) ([Engrave\\_Part \\*](#)ep)  
*engrave\_part\_repeat\_events\_get* - Get the repeat events setting
- EAPI [Engrave\\_Text\\_Effect](#) [engrave\\_part\\_effect\\_get](#) ([Engrave\\_Part \\*](#)ep)  
*engrave\_part\_effect\_get* - Get the text effect on the part
- EAPI const char \* [engrave\\_part\\_clip\\_to\\_get](#) ([Engrave\\_Part \\*](#)ep)  
*engrave\_part\_clip\_to\_get* - Get the clip\_to setting of the part
- EAPI void [engrave\\_part\\_dragable\\_x\\_get](#) ([Engrave\\_Part \\*](#)ep, int \*x, int \*step, int \*count)  
*engrave\_part\_dragable\_x\_get* - Get the draggable info in the x direction
- EAPI void [engrave\\_part\\_dragable\\_y\\_get](#) ([Engrave\\_Part \\*](#)ep, int \*y, int \*step, int \*count)  
*engrave\_part\_dragable\_y\_get* - Get the draggable info in the y direction
- EAPI const char \* [engrave\\_part\\_dragable\\_confine\\_get](#) ([Engrave\\_Part \\*](#)ep)  
*engrave\_part\_dragable\_confine\_get* - Get the draggable confine setting

- EAPI int [engrave\\_part\\_states\\_count](#) ([Engrave\\_Part](#) \*ep)  
*engrave\_part\_states\_count - get number of states in the part*
- EAPI void [engrave\\_part\\_state\\_foreach](#) ([Engrave\\_Part](#) \*ep,  
void(\*func)([Engrave\\_Part\\_State](#) \*, [Engrave\\_Part](#) \*, void \*), void \*data)  
*engrave\_part\_state\_foreach - iterate over each of the states in the part*
- EAPI Evas\_Object \* [engrave\\_part\\_evas\\_object\\_get](#) ([Engrave\\_Part](#) \*ep)  
*engrave\_part\_evas\_object\_get - get the evas object for this part*
- EAPI void [engrave\\_part\\_evas\\_object\\_set](#) ([Engrave\\_Part](#) \*ep, Evas\_Object \*o)  
*engrave\_part\_evas\_object\_set - set the evas object for this part*

### 7.11.1 Detailed Description

[Engrave\\_Part](#) object functions. Contains all functions to manipulate the [Engrave\\_Part](#) object.

## 7.12 engrave\_part\_state.h File Reference

Engrave\_Part\_State object. Contains all of the functions to work with an Engrave\_Part\_State object.

### Data Structures

- struct [\\_Engrave\\_Part\\_State](#)  
*Stores the information for the Engrave\_Part\_State object.*

### Typedefs

- typedef [\\_Engrave\\_Part\\_State](#) [Engrave\\_Part\\_State](#)  
*The Engrave\_Part\_State typedef.*

### Functions

- EAPI [Engrave\\_Part\\_State \\* engrave\\_part\\_state\\_new](#) (void)  
*engrave\_part\_state\_new - create a new Engrave\_Part\_State object.*
- EAPI void [engrave\\_part\\_state\\_free](#) ([Engrave\\_Part\\_State](#) \*eps)  
*engrave\_part\_state\_free - free the state*
- EAPI void [engrave\\_part\\_state\\_parent\\_set](#) ([Engrave\\_Part\\_State](#) \*eps, void \*ep)  
*engrave\_part\_state\_parent\_set - set the parent part on this state*
- EAPI void \* [engrave\\_part\\_state\\_parent\\_get](#) ([Engrave\\_Part\\_State](#) \*eps)  
*engrave\_part\_state\_parent\_get - get the parent part of this state*
- EAPI void [engrave\\_part\\_state\\_name\\_set](#) ([Engrave\\_Part\\_State](#) \*eps, const char \*name, double value)  
*engrave\_part\_state\_name\_set - Set the name of the state*
- EAPI void [engrave\\_part\\_state\\_visible\\_set](#) ([Engrave\\_Part\\_State](#) \*eps, int visible)  
*engrave\_part\_state\_visible\_set - Set the visibility of the state*
- EAPI void [engrave\\_part\\_state\\_align\\_set](#) ([Engrave\\_Part\\_State](#) \*eps, double x, double y)  
*engrave\_part\_state\_align\_set - Set the alignment of the state.*
- EAPI void [engrave\\_part\\_state\\_step\\_set](#) ([Engrave\\_Part\\_State](#) \*eps, double x, double y)  
*engrave\_part\_state\_step\_set - Set the step value of the state*
- EAPI void [engrave\\_part\\_state\\_min\\_size\\_set](#) ([Engrave\\_Part\\_State](#) \*eps, int w, int h)

*engrave\_part\_state\_min\_size\_set* - Set the min size of the state.

- EAPI void `engrave_part_state_fixed_size_set` (`Engrave_Part_State` \*eps, int w, int h)  
*engrave\_part\_state\_fixed\_size\_set* - Set the fixed size of the state.
- EAPI void `engrave_part_state_max_size_set` (`Engrave_Part_State` \*eps, int w, int h)  
*engrave\_part\_state\_max\_size\_set* - Set the max size of the state.
- EAPI void `engrave_part_state_aspect_set` (`Engrave_Part_State` \*eps, double w, double h)  
*engrave\_part\_state\_aspect\_set* - Set the width and height of the aspect ratio
- EAPI void `engrave_part_state_aspect_preference_set` (`Engrave_Part_State` \*eps, `Engrave_Aspect_Preference` prefer)  
*engrave\_part\_state\_aspect\_preference\_set* - Set the aspect preference.
- EAPI void `engrave_part_state_rel1_relative_set` (`Engrave_Part_State` \*eps, double x, double y)  
*engrave\_part\_state\_rel1\_relative\_set* - Set the relative value of rel1.
- EAPI void `engrave_part_state_rel1_offset_set` (`Engrave_Part_State` \*eps, int x, int y)  
*engrave\_part\_state\_rel1\_offset\_set* - Set the rel1 offset value.
- EAPI void `engrave_part_state_rel1_to_set` (`Engrave_Part_State` \*eps, const char \*to)  
*engrave\_part\_state\_rel1\_to\_set* - Set the rel1 to value for both x and y dirs.
- EAPI void `engrave_part_state_rel1_to_x_set` (`Engrave_Part_State` \*eps, const char \*to)  
*engrave\_part\_state\_rel1\_to\_x\_set* - Set what rel1 is relative too in the x dir.
- EAPI void `engrave_part_state_rel1_to_y_set` (`Engrave_Part_State` \*eps, const char \*to)  
*engrave\_part\_state\_rel1\_to\_y\_set* - Set what rel1 is relative too in the y dir.
- EAPI void `engrave_part_state_rel2_relative_set` (`Engrave_Part_State` \*eps, double x, double y)  
*engrave\_part\_state\_rel2\_relative\_set* - Set the rel2 relative value
- EAPI void `engrave_part_state_rel2_offset_set` (`Engrave_Part_State` \*eps, int x, int y)  
*engrave\_part\_state\_rel2\_offset\_set* - Set the rel2 offset value.
- EAPI void `engrave_part_state_rel2_to_set` (`Engrave_Part_State` \*eps, const char \*to)  
*engrave\_part\_state\_rel2\_to\_set* - Set the rel2 to value for both x and y dirs.
- EAPI void `engrave_part_state_rel2_to_x_set` (`Engrave_Part_State` \*eps, const char \*to)  
*engrave\_part\_state\_rel2\_to\_x\_set* - Set what rel2 is relative too in the x dir.
- EAPI void `engrave_part_state_rel2_to_y_set` (`Engrave_Part_State` \*eps, const char \*to)

*engrave\_part\_state\_rel2\_to\_y\_set* - Set what rel2 is relative too in the y dir.

- EAPI void [engrave\\_part\\_state\\_image\\_normal\\_set](#) (Engrave\_Part\_State \*eps, Engrave\_Image \*im)

*engrave\_part\_state\_image\_normal\_set* - Set the image we are associated with.

- EAPI void [engrave\\_part\\_state\\_image\\_tween\\_add](#) (Engrave\_Part\_State \*eps, Engrave\_Image \*im)

*engrave\_part\_state\_image\_tween\_add* - Add a tween to the state.

- EAPI void [engrave\\_part\\_state\\_image\\_border\\_set](#) (Engrave\_Part\_State \*eps, int l, int r, int t, int b)

*engrave\_part\_state\_image\_border\_set* - Set the border on the image.

- EAPI void [engrave\\_part\\_state\\_image\\_middle\\_set](#) (Engrave\_Part\_State \*eps, int middle)

*engrave\_part\_state\_image\_middle\_set* - Set the middle on the image.

- EAPI void [engrave\\_part\\_state\\_color\\_class\\_set](#) (Engrave\_Part\_State \*eps, const char \*class)

*engrave\_part\_state\_color\_class\_set* - Set the colour class on the state.

- EAPI void [engrave\\_part\\_state\\_color\\_set](#) (Engrave\_Part\_State \*eps, int r, int g, int b, int a)

*engrave\_part\_state\_color\_set* - Set the colour of the state.

- EAPI void [engrave\\_part\\_state\\_color2\\_set](#) (Engrave\_Part\_State \*eps, int r, int g, int b, int a)

*engrave\_part\_state\_color2\_set* - Set the colour2 of the state

- EAPI void [engrave\\_part\\_state\\_color3\\_set](#) (Engrave\_Part\_State \*eps, int r, int g, int b, int a)

*engrave\_part\_state\_color3\_set* - Set the colour 3 of the state.

- EAPI void [engrave\\_part\\_state\\_fill\\_smooth\\_set](#) (Engrave\_Part\_State \*eps, int smooth)

*engrave\_part\_state\_fill\_smooth\_set* - Set the fill smooth value of the state.

- EAPI void [engrave\\_part\\_state\\_fill\\_origin\\_relative\\_set](#) (Engrave\_Part\_State \*state, double x, double y)

*engrave\_part\_state\_fill\_origin\_relative\_set* - Set the relative value of the fill.

- EAPI void [engrave\\_part\\_state\\_fill\\_size\\_relative\\_set](#) (Engrave\_Part\_State \*eps, double x, double y)

*engrave\_part\_state\_fill\_size\_relative\_set* - Set the fills relative size on the state.

- EAPI void [engrave\\_part\\_state\\_fill\\_origin\\_offset\\_set](#) (Engrave\_Part\_State \*eps, int x, int y)

*engrave\_part\_state\_fill\_origin\_offset\_set* - Set the offset of the fill position

- EAPI void [engrave\\_part\\_state\\_fill\\_size\\_offset\\_set](#) (Engrave\_Part\_State \*eps, int x, int y)

*engrave\_part\_state\_fill\_size\_offset\_set* - Set the fill size offset.

- EAPI void `engrave_part_state_text_text_set` (`Engrave_Part_State` \*eps, const char \*text)

*engrave\_part\_state\_text\_text\_set* - Set the text string of the state.

- EAPI void `engrave_part_state_text_elipsis_set` (`Engrave_Part_State` \*eps, double val)

*engrave\_part\_state\_text\_elipsis\_set* - Set the text elipsis of the state.

- EAPI void `engrave_part_state_text_text_append` (`Engrave_Part_State` \*eps, const char \*text)

*engrave\_part\_state\_text\_text\_append* - Appends the text string of the state.

- EAPI void `engrave_part_state_text_text_class_set` (`Engrave_Part_State` \*eps, const char \*text)

*engrave\_part\_state\_text\_text\_class\_set* - Set the text class of the state.

- EAPI void `engrave_part_state_text_style_set` (`Engrave_Part_State` \*eps, const char \*style)

*engrave\_part\_state\_text\_style\_set* - Set the text style of the state.

- EAPI void `engrave_part_state_text_text_source_set` (`Engrave_Part_State` \*eps, const char \*source)

*engrave\_part\_state\_text\_text\_source\_set* - Set the text source of the state.

- EAPI void `engrave_part_state_text_source_set` (`Engrave_Part_State` \*eps, const char \*source)

*engrave\_part\_state\_text\_source\_set* - Set the source of the text.

- EAPI void `engrave_part_state_text_font_set` (`Engrave_Part_State` \*eps, const char \*font)

*engrave\_part\_state\_text\_font\_set* - Set the font for the text

- EAPI void `engrave_part_state_text_size_set` (`Engrave_Part_State` \*eps, int size)

*engrave\_part\_state\_text\_size\_set* - Set the text size.

- EAPI void `engrave_part_state_text_fit_set` (`Engrave_Part_State` \*eps, int x, int y)

*engrave\_part\_state\_text\_fit\_set* - Set the fit parameter of the text

- EAPI void `engrave_part_state_text_min_set` (`Engrave_Part_State` \*eps, int x, int y)

*engrave\_part\_state\_text\_min\_set* - Set the text min value.

- EAPI void `engrave_part_state_text_max_set` (`Engrave_Part_State` \*eps, int x, int y)

*engrave\_part\_state\_text\_max\_set* - Set the text max value.

- EAPI void `engrave_part_state_text_align_set` (`Engrave_Part_State` \*eps, double x, double y)

*engrave\_part\_state\_text\_align\_set* - Set the text alignment of the state.

- EAPI void `engrave_part_state_copy` (`Engrave_Part_State` \*from, `Engrave_Part_State` \*to)

*engrave\_part\_state\_copy* - Copy the values from from to to

- EAPI const char \* [engrave\\_part\\_state\\_name\\_get](#) (Engrave\_Part\_State \*eps, double \*val)  
*engrave\_part\_state\_name\_get* - get the state name/value
- EAPI int [engrave\\_part\\_state\\_visible\\_get](#) (Engrave\_Part\_State \*eps)  
*engrave\_part\_state\_visible\_get* - get the visibility flag
- EAPI void [engrave\\_part\\_state\\_align\\_get](#) (Engrave\_Part\_State \*eps, double \*x, double \*y)  
*engrave\_part\_state\_align\_get* - get the alignment of the state
- EAPI void [engrave\\_part\\_state\\_step\\_get](#) (Engrave\_Part\_State \*eps, double \*x, double \*y)  
*engrave\_part\_state\_step\_get* - get the stepping info for the state
- EAPI void [engrave\\_part\\_state\\_min\\_size\\_get](#) (Engrave\_Part\_State \*eps, int \*w, int \*h)  
*engrave\_part\_state\_min\_size\_get* - get the minimum size of the state
- EAPI void [engrave\\_part\\_state\\_fixed\\_size\\_get](#) (Engrave\_Part\_State \*eps, int \*w, int \*h)  
*engrave\_part\_state\_fixed\_size\_get* - get the fixed size of the state
- EAPI void [engrave\\_part\\_state\\_max\\_size\\_get](#) (Engrave\_Part\_State \*eps, int \*w, int \*h)  
*engrave\_part\_state\_max\_size\_get* - get the maximum size of the state
- EAPI void [engrave\\_part\\_state\\_aspect\\_get](#) (Engrave\_Part\_State \*eps, double \*w, double \*h)  
*engrave\_part\_state\_aspect\_get* - get the aspect ratio for the state
- EAPI Engrave\_Aspect\_Preference [engrave\\_part\\_state\\_aspect\\_preference\\_get](#) (Engrave\_Part\_State \*eps)  
*engrave\_part\_state\_aspect\_preference\_get* - get the aspect preference
- EAPI void [engrave\\_part\\_state\\_rel1\\_relative\\_get](#) (Engrave\_Part\_State \*eps, double \*x, double \*y)  
*engrave\_part\_state\_rel1\_relative\_get* - get the rel1 relative values
- EAPI void [engrave\\_part\\_state\\_rel1\\_offset\\_get](#) (Engrave\_Part\_State \*eps, int \*x, int \*y)  
*engrave\_part\_state\_rel1\_offset\_get* - get the rel1 offset values
- EAPI const char \* [engrave\\_part\\_state\\_rel1\\_to\\_x\\_get](#) (Engrave\_Part\_State \*eps)  
*engrave\_part\_state\_rel1\_to\_x\_get* - get the to\_x value for rel1
- EAPI const char \* [engrave\\_part\\_state\\_rel1\\_to\\_y\\_get](#) (Engrave\_Part\_State \*eps)  
*engrave\_part\_state\_rel1\_to\_y\_get* - get the to\_y value for rel1

- EAPI void `engrave_part_state_rel2_relative_get` (`Engrave_Part_State` \*eps, double \*x, double \*y)  
*engrave\_part\_state\_rel2\_relative\_get - get the rel2 relative values*
- EAPI void `engrave_part_state_rel2_offset_get` (`Engrave_Part_State` \*eps, int \*x, int \*y)  
*engrave\_part\_state\_rel2\_offset\_get - get the rel2 offset values*
- EAPI const char \* `engrave_part_state_rel2_to_x_get` (`Engrave_Part_State` \*eps)  
*engrave\_part\_state\_rel2\_to\_x\_get - get the to\_x value for rel2*
- EAPI const char \* `engrave_part_state_rel2_to_y_get` (`Engrave_Part_State` \*eps)  
*engrave\_part\_state\_rel2\_to\_y\_get - get the to\_y value for rel2*
- EAPI const char \* `engrave_part_state_color_class_get` (`Engrave_Part_State` \*eps)  
*engrave\_part\_state\_color\_class\_get - get the color class for the state*
- EAPI void `engrave_part_state_color_get` (`Engrave_Part_State` \*eps, int \*r, int \*g, int \*b, int \*a)  
*engrave\_part\_state\_color\_get - get the color value from the state*
- EAPI void `engrave_part_state_color2_get` (`Engrave_Part_State` \*eps, int \*r, int \*g, int \*b, int \*a)  
*engrave\_part\_state\_color2\_get - get the color2 value from the state*
- EAPI void `engrave_part_state_color3_get` (`Engrave_Part_State` \*eps, int \*r, int \*g, int \*b, int \*a)  
*engrave\_part\_state\_color3\_get - get the color3 value from the state*
- EAPI `Engrave_Image` \* `engrave_part_state_image_normal_get` (`Engrave_Part_State` \*eps)  
*engrave\_part\_state\_image\_normal\_get - get the Engrave\_Image for the normal setting*
- EAPI void `engrave_part_state_image_border_get` (`Engrave_Part_State` \*eps, int \*l, int \*r, int \*t, int \*b)  
*engrave\_part\_state\_image\_border\_get - get the border settings*
- EAPI int `engrave_part_state_image_middle_get` (`Engrave_Part_State` \*eps)  
*engrave\_part\_state\_image\_middle\_get - get the middle settings*
- EAPI const char \* `engrave_part_state_text_text_get` (`Engrave_Part_State` \*eps)  
*engrave\_part\_state\_text\_text\_get - get the text value*
- EAPI const char \* `engrave_part_state_text_text_class_get` (`Engrave_Part_State` \*eps)  
*engrave\_part\_state\_text\_text\_class\_get - get the text class*
- EAPI const char \* `engrave_part_state_text_style_get` (`Engrave_Part_State` \*eps)  
*engrave\_part\_state\_text\_style\_get - get the text style*



- EAPI const char \* [engrave\\_part\\_state\\_text\\_text\\_source\\_get](#) (Engrave\_Part\_State \*eps)  
*engrave\_part\_state\_text\_text\_source\_get - get the text text\_source*
- EAPI const char \* [engrave\\_part\\_state\\_text\\_source\\_get](#) (Engrave\_Part\_State \*eps)  
*engrave\_part\_state\_text\_source\_get - get the text text\_source*
- EAPI const char \* [engrave\\_part\\_state\\_text\\_font\\_get](#) (Engrave\_Part\_State \*eps)  
*engrave\_part\_state\_text\_font\_get - get the text font*
- EAPI int [engrave\\_part\\_state\\_text\\_size\\_get](#) (Engrave\_Part\_State \*eps)  
*engrave\_part\_state\_text\_size\_get - get the font size*
- EAPI void [engrave\\_part\\_state\\_text\\_fit\\_get](#) (Engrave\_Part\_State \*eps, int \*x, int \*y)  
*engrave\_part\_state\_text\_fit\_get - get the text fit settings*
- EAPI double [engrave\\_part\\_state\\_text\\_elipsis\\_get](#) (Engrave\_Part\_State \*eps)  
*engrave\_part\_state\_text\_elipsis\_get - get the text elipsis settings*
- EAPI void [engrave\\_part\\_state\\_text\\_min\\_get](#) (Engrave\_Part\_State \*eps, int \*x, int \*y)  
*engrave\_part\_state\_text\_min\_get - get the text min size*
- EAPI void [engrave\\_part\\_state\\_text\\_align\\_get](#) (Engrave\_Part\_State \*eps, double \*x, double \*y)  
*engrave\_part\_state\_text\_align\_get - get the text alignment settings*
- EAPI int [engrave\\_part\\_state\\_tweens\\_count](#) (Engrave\_Part\_State \*eps)  
*engrave\_part\_state\_tweens\_count - get the number of tweens in the state*
- EAPI void [engrave\\_part\\_state\\_tween\\_foreach](#) (Engrave\_Part\_State \*eps, void(\*func)(Engrave\_Image \*, void \*), void \*data)  
*engrave\_part\_state\_tween\_foreach - Iterate over the tween list*
- EAPI void [engrave\\_part\\_state\\_gradient\\_spectrum\\_set](#) (Engrave\_Part\_State \*eps, const char \*spec)  
*engrave\_part\_state\_gradient\_spectrum\_set - set the spectrum of the gradient on this state*
- EAPI const char \* [engrave\\_part\\_state\\_gradient\\_spectrum\\_get](#) (Engrave\_Part\_State \*eps)  
*engrave\_part\_state\_gradient\_spectrum\_get - get the spectrum of the gradient on this state*
- EAPI void [engrave\\_part\\_state\\_gradient\\_type\\_set](#) (Engrave\_Part\_State \*eps, const char \*type)  
*engrave\_part\_state\_gradient\_type - set the type of the gradient on this state*
- EAPI const char \* [engrave\\_part\\_state\\_gradient\\_type\\_get](#) (Engrave\_Part\_State \*eps)  
*engrave\_part\_state\_gradient\_type\_get - get the type of the gradient on this state*

### 7.12.1 Detailed Description

Engrave\_Part\_State object. Contains all of the functions to work with an Engrave\_Part\_State object.

## 7.13 engrave\_program.h File Reference

Engrave\_Program object functions. Contains all of the functions to mainuplulate Engrave\_Program objects.

### Data Structures

- struct [\\_Engrave\\_Program](#)  
*Contains all the info to define a program.*

### Typedefs

- typedef [\\_Engrave\\_Program](#) [Engrave\\_Program](#)  
*The Engrave\_Program typedef.*

### Functions

- EAPI [Engrave\\_Program \\*](#) [engrave\\_program\\_new](#) (void)  
*engrave\_program\_new - create a new Engrave\_Program object.*
- EAPI void [engrave\\_program\\_free](#) ([Engrave\\_Program](#) \*ep)  
*engrave\_program\_free - free the program*
- EAPI void [engrave\\_program\\_parent\\_set](#) ([Engrave\\_Program](#) \*ep, void \*eg)  
*engrave\_program\_parent\_set - set the parent of the program*
- EAPI void \* [engrave\\_program\\_parent\\_get](#) ([Engrave\\_Program](#) \*ep)  
*engrave\_program\_parent\_get - get the parent pointer*
- EAPI void [engrave\\_program\\_script\\_set](#) ([Engrave\\_Program](#) \*ep, const char \*script)  
*engrave\_program\_script\_set - attach a script to the program.*
- EAPI void [engrave\\_program\\_name\\_set](#) ([Engrave\\_Program](#) \*ep, const char \*name)  
*engrave\_program\_name\_set - set the name of the program.*
- EAPI void [engrave\\_program\\_signal\\_set](#) ([Engrave\\_Program](#) \*ep, const char \*signal)  
*engrave\_program\_signal\_set - set the given signal on the program*
- EAPI void [engrave\\_program\\_source\\_set](#) ([Engrave\\_Program](#) \*ep, const char \*source)  
*engrave\_program\_source\_set - set the source of the program.*
- EAPI void [engrave\\_program\\_target\\_add](#) ([Engrave\\_Program](#) \*ep, const char \*target)  
*engrave\_program\_target\_add - set the target on the program.*

- EAPI void [engrave\\_program\\_after\\_add](#) ([Engrave\\_Program](#) \*ep, const char \*after)  
*engrave\_program\_after\_add - add the after to the program.*
- EAPI void [engrave\\_program\\_in\\_set](#) ([Engrave\\_Program](#) \*ep, double from, double range)  
*engrave\_program\_in\_set - set the in value of the program.*
- EAPI void [engrave\\_program\\_action\\_set](#) ([Engrave\\_Program](#) \*ep, [Engrave\\_Action](#) action, const char \*state, const char \*state2, double value, double value2)  
*engrave\_program\_action\_set - set the action for the program.*
- EAPI void [engrave\\_program\\_transition\\_set](#) ([Engrave\\_Program](#) \*ep, [Engrave\\_Transition](#) trans, double duration)  
*engrave\_program\_transition\_set - set the transition on the program*
- EAPI const char \* [engrave\\_program\\_name\\_get](#) ([Engrave\\_Program](#) \*ep)  
*engrave\_program\_name\_get - Get the program name*
- EAPI const char \* [engrave\\_program\\_signal\\_get](#) ([Engrave\\_Program](#) \*ep)  
*engrave\_program\_signal\_get - Get the program signal*
- EAPI const char \* [engrave\\_program\\_source\\_get](#) ([Engrave\\_Program](#) \*ep)  
*engrave\_program\_source\_get - Get the program source*
- EAPI void [engrave\\_program\\_action\\_get](#) ([Engrave\\_Program](#) \*ep, [Engrave\\_Action](#) \*action, char \*state, char \*state2, int state\_len, int state2\_len, double \*value, double \*value2)  
*engrave\_program\_action\_get - Get the action information for the program*
- EAPI void [engrave\\_program\\_transition\\_get](#) ([Engrave\\_Program](#) \*ep, [Engrave\\_Transition](#) \*trans, double \*duration)  
*engrave\_program\_transition\_get - Get the programs transition information*
- EAPI const char \* [engrave\\_program\\_script\\_get](#) ([Engrave\\_Program](#) \*ep)  
*engrave\_program\_script\_get - Get the script from the program*
- EAPI void [engrave\\_program\\_in\\_get](#) ([Engrave\\_Program](#) \*ep, double \*from, double \*range)  
*engrave\_program\_in\_get - Get the in data for the program*
- EAPI int [engrave\\_program\\_afters\\_count](#) ([Engrave\\_Program](#) \*ep)  
*engrave\_program\_afters\_count - get the number of afters*
- EAPI int [engrave\\_program\\_targets\\_count](#) ([Engrave\\_Program](#) \*ep)  
*engrave\_program\_targets\_count - count the number of targets*
- EAPI void [engrave\\_program\\_target\\_foreach](#) ([Engrave\\_Program](#) \*ep, void(\*func)(const char \*, void \*), void \*data)  
*engrave\_program\_target\_foreach - Iterate over each target*
- EAPI void [engrave\\_program\\_after\\_foreach](#) ([Engrave\\_Program](#) \*ep, void(\*func)(const char \*, void \*), void \*data)  
*engrave\_program\_after\_foreach - Iterate over each after*

### 7.13.1 Detailed Description

Engrave\_Program object functions. Contains all of the functions to mainuplulate Engrave\_Program objects.

## 7.14 engrave\_spectrum.h File Reference

Engrave\_Spectrum block Contains all of the functions to manipulate Engrave\_Spectrum blocks.

### Data Structures

- struct [\\_Engrave\\_Spectrum](#)  
*Stores the needed spectrum information.*
- struct [\\_Engrave\\_Spectrum\\_Color](#)  
*Stores the spectrum color information.*

### Typedefs

- typedef [\\_Engrave\\_Spectrum](#) [Engrave\\_Spectrum](#)  
*The Engrave\_Spectrum typedef.*

### Functions

- EAPI [Engrave\\_Spectrum](#) \* [engrave\\_spectrum\\_new](#) ()  
*engrave\_spectrum\_new - create a new Engrave\_Spectrum object*
- EAPI void [engrave\\_spectrum\\_free](#) ([Engrave\\_Spectrum](#) \*es)  
*engrave\_spectrum\_free - frees a Engrave\_Spectrum object*
- EAPI [Engrave\\_Spectrum\\_Color](#) \* [engrave\\_spectrum\\_color\\_new](#) (int r, int g, int b, int a, int d)  
*engrave\_spectrum\_color\_new - Creates an Engrave\_Spectrum\_Color object*
- EAPI void [engrave\\_spectrum\\_color\\_free](#) ([Engrave\\_Spectrum\\_Color](#) \*esc)  
*engrave\_spectrum\_color\_free - frees a Engrave\_Spectrum\_Color object*
- EAPI void [engrave\\_spectrum\\_parent\\_set](#) ([Engrave\\_Spectrum](#) \*es, void \*parent)  
*engrave\_spectrum\_parent\_set - sets the parent of the Engrave\_Spectrum object*
- EAPI void [engrave\\_spectrum\\_color\\_add](#) ([Engrave\\_Spectrum](#) \*spec, [Engrave\\_Spectrum\\_Color](#) \*col)  
*engrave\_spectrum\_color\_add - Adds a color to a Engrave\_Spectrum object*
- EAPI void [engrave\\_spectrum\\_color\\_del](#) ([Engrave\\_Spectrum](#) \*spec, [Engrave\\_Spectrum\\_Color](#) \*col)  
*engrave\_spectrum\_color\_del - Removes a color from a Engrave\_Spectrum object*
- EAPI const char \* [engrave\\_spectrum\\_name\\_get](#) ([Engrave\\_Spectrum](#) \*es)

*engrave\_spectrum\_name\_get* - gets the name of the *Engrave\_Spectrum* object

- EAPI void [engrave\\_spectrum\\_name\\_set](#) ([Engrave\\_Spectrum](#) \*es, const char \*name)

*engrave\_spectrum\_name\_set* - sets the name of the *Engrave\_Spectrum* object

- EAPI int [engrave\\_spectrum\\_color\\_count](#) ([Engrave\\_Spectrum](#) \*es)

*engrave\_spectrum\_color\_count* - returns the number of colors inside an *Engrave\_Spectrum* object

- EAPI [Engrave\\_Spectrum\\_Color](#) \* [engrave\\_spectrum\\_color\\_nth](#) ([Engrave\\_Spectrum](#) \*es, int n)

*engrave\_spectrum\_color\_nth* - gets the *n*th color of the *Engrave\_Spectrum* object

### 7.14.1 Detailed Description

[Engrave\\_Spectrum](#) block Contains all of the functions to manipulate [Engrave\\_Spectrum](#) blocks.

## 7.15 engrave\_style.h File Reference

Engrave\_Style object functions. Contains all of the functions to manipulate Engrave\_Style objects.

### Data Structures

- struct [\\_Engrave\\_Style](#)  
*Contains the needed style values.*
- struct [\\_Engrave\\_Tag](#)  
*Contains the tag information.*

### Typedefs

- typedef [\\_Engrave\\_Style](#) [Engrave\\_Style](#)  
*The Engrave\_Style typedef.*

### Functions

- EAPI [Engrave\\_Style](#) \* [engrave\\_style\\_new](#) ()  
*engrave\_style\_new - create a new Engrave\_Style object.*

#### 7.15.1 Detailed Description

Engrave\_Style object functions. Contains all of the functions to manipulate Engrave\_Style objects.



# Index

- `_Engrave_Action`
    - `Engrave_Enums`, [16](#)
  - `_Engrave_Aspect_Preference`
    - `Engrave_Enums`, [16](#)
  - `_Engrave_Data`, [111](#)
  - `_Engrave_File`, [112](#)
  - `_Engrave_Font`, [113](#)
  - `_Engrave_Group`, [114](#)
  - `_Engrave_Image`, [116](#)
  - `_Engrave_Image_Type`
    - `Engrave_Enums`, [16](#)
  - `_Engrave_Part`, [117](#)
  - `_Engrave_Part_State`, [119](#)
  - `_Engrave_Part_Type`
    - `Engrave_Enums`, [16](#)
  - `_Engrave_Program`, [125](#)
  - `_Engrave_Spectrum`, [127](#)
  - `_Engrave_Spectrum_Color`, [128](#)
  - `_Engrave_Style`, [129](#)
  - `_Engrave_Tag`, [130](#)
  - `_Engrave_Text_Effect`
    - `Engrave_Enums`, [17](#)
  - `_Engrave_Transition`
    - `Engrave_Enums`, [17](#)
- `Engrave.h`, [131](#)
- `ENGRAVE_ACTION_DRAG_VAL_PAGE`
  - `Engrave_Enums`, [16](#)
- `ENGRAVE_ACTION_DRAG_VAL_SET`
  - `Engrave_Enums`, [16](#)
- `ENGRAVE_ACTION_DRAG_VAL_STEP`
  - `Engrave_Enums`, [16](#)
- `ENGRAVE_ACTION_SCRIPT`
  - `Engrave_Enums`, [16](#)
- `ENGRAVE_ACTION_SIGNAL_EMIT`
  - `Engrave_Enums`, [16](#)
- `ENGRAVE_ACTION_STATE_SET`
  - `Engrave_Enums`, [16](#)
- `ENGRAVE_ACTION_STOP`
  - `Engrave_Enums`, [16](#)
- `ENGRAVE_ASPECT_PREFERENCE_-`
  - `BOTH`
    - `Engrave_Enums`, [16](#)
  - `HORIZONTAL`
    - `Engrave_Enums`, [16](#)
  - `NONE`
    - `Engrave_Enums`, [16](#)
  - `VERTICAL`
    - `Engrave_Enums`, [16](#)
- `Engrave_Canvas`
  - `engrave_canvas_current_group_set`, [9](#)
  - `engrave_canvas_file_set`, [10](#)
  - `engrave_canvas_new`, [10](#)
- `engrave_canvas.h`, [132](#)
- `Engrave_Canvas: Functions to work with the`
  - `engrave_canvas.`, [9](#)
- `engrave_canvas_current_group_set`
  - `Engrave_Canvas`, [9](#)
- `engrave_canvas_file_set`
  - `Engrave_Canvas`, [10](#)
- `engrave_canvas_new`
  - `Engrave_Canvas`, [10](#)
- `Engrave_Data`
  - `engrave_data_free`, [12](#)
  - `engrave_data_int_value_get`, [12](#)
  - `engrave_data_int_value_set`, [12](#)
  - `engrave_data_key_get`, [12](#)
  - `engrave_data_new`, [13](#)
  - `engrave_data_parent_get`, [13](#)
  - `engrave_data_parent_set`, [13](#)
  - `engrave_data_value_get`, [13](#)
  - `engrave_data_value_set`, [14](#)
- `engrave_data.h`, [133](#)
- `Engrave_Data: Functions to work with data`
  - `blocks`, [11](#)
- `engrave_data_free`
  - `Engrave_Data`, [12](#)
- `engrave_data_int_value_get`
  - `Engrave_Data`, [12](#)
- `engrave_data_int_value_set`
  - `Engrave_Data`, [12](#)
- `engrave_data_key_get`
  - `Engrave_Data`, [12](#)
- `engrave_data_new`
  - `Engrave_Data`, [13](#)
- `engrave_data_parent_get`
  - `Engrave_Data`, [13](#)

- engrave\_data\_parent\_set
  - Engrave\_Data, [13](#)
- engrave\_data\_value\_get
  - Engrave\_Data, [13](#)
- engrave\_data\_value\_set
  - Engrave\_Data, [14](#)
- engrave\_edc\_output
  - Engrave\_Out, [46](#)
- engrave\_edj\_output
  - Engrave\_Out, [46](#)
- Engrave\_Enums
  - ENGRAVE\_ACTION\_DRAG\_VAL\_-PAGE, [16](#)
  - ENGRAVE\_ACTION\_DRAG\_VAL\_-SET, [16](#)
  - ENGRAVE\_ACTION\_DRAG\_VAL\_-STEP, [16](#)
  - ENGRAVE\_ACTION\_SCRIPT, [16](#)
  - ENGRAVE\_ACTION\_SIGNAL\_EMIT, [16](#)
  - ENGRAVE\_ACTION\_STATE\_SET, [16](#)
  - ENGRAVE\_ACTION\_STOP, [16](#)
  - ENGRAVE\_ASPECT\_-PREFERENCE\_BOTH, [16](#)
  - ENGRAVE\_ASPECT\_-PREFERENCE\_HORIZONTAL, [16](#)
  - ENGRAVE\_ASPECT\_-PREFERENCE\_NONE, [16](#)
  - ENGRAVE\_ASPECT\_-PREFERENCE\_VERTICAL, [16](#)
  - ENGRAVE\_IMAGE\_TYPE\_COMP, [16](#)
  - ENGRAVE\_IMAGE\_TYPE\_LOSSY, [16](#)
  - ENGRAVE\_IMAGE\_TYPE\_RAW, [16](#)
  - ENGRAVE\_PART\_TYPE\_-GRADIENT, [17](#)
  - ENGRAVE\_PART\_TYPE\_IMAGE, [17](#)
  - ENGRAVE\_PART\_TYPE\_RECT, [17](#)
  - ENGRAVE\_PART\_TYPE\_SWALLOW, [17](#)
  - ENGRAVE\_PART\_TYPE\_TEXT, [17](#)
  - ENGRAVE\_PART\_TYPE\_-TEXTBLOCK, [17](#)
  - ENGRAVE\_TEXT\_EFFECT\_NONE, [17](#)
  - ENGRAVE\_TEXT\_EFFECT\_-OUTLINE, [17](#)
  - ENGRAVE\_TEXT\_EFFECT\_-OUTLINE\_SHADOW, [17](#)
  - ENGRAVE\_TEXT\_EFFECT\_-OUTLINE\_SOFT\_SHADOW, [17](#)
  - ENGRAVE\_TEXT\_EFFECT\_PLAIN, [17](#)
  - ENGRAVE\_TEXT\_EFFECT\_-SHADOW, [17](#)
  - ENGRAVE\_TEXT\_EFFECT\_SOFT\_-OUTLINE, [17](#)
  - ENGRAVE\_TEXT\_EFFECT\_SOFT\_-SHADOW, [17](#)
  - ENGRAVE\_TRANSITION\_-ACCELERATE, [17](#)
  - ENGRAVE\_TRANSITION\_-DECELERATE, [17](#)
  - ENGRAVE\_TRANSITION\_LINEAR, [17](#)
  - ENGRAVE\_TRANSITION\_-SINUSOIDAL, [17](#)
- Engrave\_Enums
  - \_Engrave\_Action, [16](#)
  - \_Engrave\_Aspect\_Preference, [16](#)
  - \_Engrave\_Image\_Type, [16](#)
  - \_Engrave\_Part\_Type, [16](#)
  - \_Engrave\_Text\_Effect, [17](#)
  - \_Engrave\_Transition, [17](#)
- engrave\_enums.h, [135](#)
- Engrave\_File
  - engrave\_file\_data\_add, [20](#)
  - engrave\_file\_data\_by\_key\_find, [20](#)
  - engrave\_file\_data\_count, [20](#)
  - engrave\_file\_font\_add, [20](#)
  - engrave\_file\_font\_by\_name\_find, [21](#)
  - engrave\_file\_font\_dir\_get, [21](#)
  - engrave\_file\_font\_dir\_set, [21](#)
  - engrave\_file\_fonts\_count, [22](#)
  - engrave\_file\_free, [22](#)
  - engrave\_file\_group\_add, [22](#)
  - engrave\_file\_group\_by\_name\_find, [22](#)
  - engrave\_file\_group\_last\_get, [23](#)
  - engrave\_file\_groups\_count, [23](#)
  - engrave\_file\_image\_add, [23](#)
  - engrave\_file\_image\_by\_name\_find, [24](#)
  - engrave\_file\_image\_dir\_get, [24](#)
  - engrave\_file\_image\_dir\_set, [24](#)
  - engrave\_file\_images\_count, [24](#)
  - engrave\_file\_new, [25](#)
  - engrave\_file\_spectra\_count, [25](#)
  - engrave\_file\_spectrum\_add, [25](#)
  - engrave\_file\_spectrum\_last\_get, [25](#)
  - engrave\_file\_style\_add, [26](#)
  - engrave\_file\_style\_last\_get, [26](#)
  - engrave\_file\_styles\_count, [26](#)
- engrave\_file.h, [137](#)
- Engrave\_File: Functions to work with the Engrave file object, [18](#)
- engrave\_file\_data\_add
  - Engrave\_File, [20](#)
- engrave\_file\_data\_by\_key\_find
  - Engrave\_File, [20](#)

- engrave\_file\_data\_count
  - Engrave\_File, [20](#)
- engrave\_file\_font\_add
  - Engrave\_File, [20](#)
- engrave\_file\_font\_by\_name\_find
  - Engrave\_File, [21](#)
- engrave\_file\_font\_dir\_get
  - Engrave\_File, [21](#)
- engrave\_file\_font\_dir\_set
  - Engrave\_File, [21](#)
- engrave\_file\_fonts\_count
  - Engrave\_File, [22](#)
- engrave\_file\_free
  - Engrave\_File, [22](#)
- engrave\_file\_group\_add
  - Engrave\_File, [22](#)
- engrave\_file\_group\_by\_name\_find
  - Engrave\_File, [22](#)
- engrave\_file\_group\_last\_get
  - Engrave\_File, [23](#)
- engrave\_file\_groups\_count
  - Engrave\_File, [23](#)
- engrave\_file\_image\_add
  - Engrave\_File, [23](#)
- engrave\_file\_image\_by\_name\_find
  - Engrave\_File, [24](#)
- engrave\_file\_image\_dir\_get
  - Engrave\_File, [24](#)
- engrave\_file\_image\_dir\_set
  - Engrave\_File, [24](#)
- engrave\_file\_images\_count
  - Engrave\_File, [24](#)
- engrave\_file\_new
  - Engrave\_File, [25](#)
- engrave\_file\_spectra\_count
  - Engrave\_File, [25](#)
- engrave\_file\_spectrum\_add
  - Engrave\_File, [25](#)
- engrave\_file\_spectrum\_last\_get
  - Engrave\_File, [25](#)
- engrave\_file\_style\_add
  - Engrave\_File, [26](#)
- engrave\_file\_style\_last\_get
  - Engrave\_File, [26](#)
- engrave\_file\_styles\_count
  - Engrave\_File, [26](#)
- Engrave\_Font
  - engrave\_font\_free, [27](#)
  - engrave\_font\_name\_get, [28](#)
  - engrave\_font\_new, [28](#)
  - engrave\_font\_parent\_get, [28](#)
  - engrave\_font\_parent\_set, [28](#)
  - engrave\_font\_path\_get, [29](#)
- engrave\_font.h, [140](#)
- Engrave\_Font: Functions to work with engrave font objects, [27](#)
- engrave\_font\_free
  - Engrave\_Font, [27](#)
- engrave\_font\_name\_get
  - Engrave\_Font, [28](#)
- engrave\_font\_new
  - Engrave\_Font, [28](#)
- engrave\_font\_parent\_get
  - Engrave\_Font, [28](#)
- engrave\_font\_parent\_set
  - Engrave\_Font, [28](#)
- engrave\_font\_path\_get
  - Engrave\_Font, [29](#)
- Engrave\_Group
  - engrave\_group\_alias\_get, [32](#)
  - engrave\_group\_alias\_set, [32](#)
  - engrave\_group\_data\_add, [32](#)
  - engrave\_group\_data\_by\_key\_find, [33](#)
  - engrave\_group\_data\_count, [33](#)
  - engrave\_group\_data\_foreach, [33](#)
  - engrave\_group\_free, [33](#)
  - engrave\_group\_max\_size\_get, [34](#)
  - engrave\_group\_max\_size\_set, [34](#)
  - engrave\_group\_min\_size\_get, [34](#)
  - engrave\_group\_min\_size\_set, [35](#)
  - engrave\_group\_name\_get, [35](#)
  - engrave\_group\_name\_set, [35](#)
  - engrave\_group\_new, [35](#)
  - engrave\_group\_parent\_get, [36](#)
  - engrave\_group\_parent\_set, [36](#)
  - engrave\_group\_part\_add, [36](#)
  - engrave\_group\_part\_by\_name\_find, [36](#)
  - engrave\_group\_part\_last\_get, [37](#)
  - engrave\_group\_parts\_count, [37](#)
  - engrave\_group\_parts\_foreach, [37](#)
  - engrave\_group\_program\_add, [38](#)
  - engrave\_group\_program\_last\_get, [38](#)
  - engrave\_group\_programs\_count, [38](#)
  - engrave\_group\_programs\_foreach, [38](#)
  - engrave\_group\_script\_get, [39](#)
  - engrave\_group\_script\_set, [39](#)
- engrave\_group.h, [141](#)
- Engrave\_Group: Functions to work with an Engrave\_Group., [30](#)
- engrave\_group\_alias\_get
  - Engrave\_Group, [32](#)
- engrave\_group\_alias\_set
  - Engrave\_Group, [32](#)
- engrave\_group\_data\_add
  - Engrave\_Group, [32](#)
- engrave\_group\_data\_by\_key\_find
  - Engrave\_Group, [33](#)
- engrave\_group\_data\_count

- Engrave\_Group, [33](#)
- engrave\_group\_data\_foreach
  - Engrave\_Group, [33](#)
- engrave\_group\_free
  - Engrave\_Group, [33](#)
- engrave\_group\_max\_size\_get
  - Engrave\_Group, [34](#)
- engrave\_group\_max\_size\_set
  - Engrave\_Group, [34](#)
- engrave\_group\_min\_size\_get
  - Engrave\_Group, [34](#)
- engrave\_group\_min\_size\_set
  - Engrave\_Group, [35](#)
- engrave\_group\_name\_get
  - Engrave\_Group, [35](#)
- engrave\_group\_name\_set
  - Engrave\_Group, [35](#)
- engrave\_group\_new
  - Engrave\_Group, [35](#)
- engrave\_group\_parent\_get
  - Engrave\_Group, [36](#)
- engrave\_group\_parent\_set
  - Engrave\_Group, [36](#)
- engrave\_group\_part\_add
  - Engrave\_Group, [36](#)
- engrave\_group\_part\_by\_name\_find
  - Engrave\_Group, [36](#)
- engrave\_group\_part\_last\_get
  - Engrave\_Group, [37](#)
- engrave\_group\_parts\_count
  - Engrave\_Group, [37](#)
- engrave\_group\_parts\_foreach
  - Engrave\_Group, [37](#)
- engrave\_group\_program\_add
  - Engrave\_Group, [38](#)
- engrave\_group\_program\_last\_get
  - Engrave\_Group, [38](#)
- engrave\_group\_programs\_count
  - Engrave\_Group, [38](#)
- engrave\_group\_programs\_foreach
  - Engrave\_Group, [38](#)
- engrave\_group\_script\_get
  - Engrave\_Group, [39](#)
- engrave\_group\_script\_set
  - Engrave\_Group, [39](#)
- Engrave\_Image
  - engrave\_image\_compression\_value\_get, [40](#)
  - engrave\_image\_dup, [41](#)
  - engrave\_image\_free, [41](#)
  - engrave\_image\_name\_get, [41](#)
  - engrave\_image\_new, [41](#)
  - engrave\_image\_parent\_get, [42](#)
  - engrave\_image\_parent\_set, [42](#)
  - engrave\_image\_type\_get, [42](#)
- engrave\_image.h, [144](#)
- Engrave\_Image: Functions to work with Engrave\_Image blocks., [40](#)
- engrave\_image\_compression\_value\_get
  - Engrave\_Image, [40](#)
- engrave\_image\_dup
  - Engrave\_Image, [41](#)
- engrave\_image\_free
  - Engrave\_Image, [41](#)
- engrave\_image\_name\_get
  - Engrave\_Image, [41](#)
- engrave\_image\_new
  - Engrave\_Image, [41](#)
- engrave\_image\_parent\_get
  - Engrave\_Image, [42](#)
- engrave\_image\_parent\_set
  - Engrave\_Image, [42](#)
- ENGRAVE\_IMAGE\_TYPE\_COMP
  - Engrave\_Enums, [16](#)
- engrave\_image\_type\_get
  - Engrave\_Image, [42](#)
- ENGRAVE\_IMAGE\_TYPE\_LOSSY
  - Engrave\_Enums, [16](#)
- ENGRAVE\_IMAGE\_TYPE\_RAW
  - Engrave\_Enums, [16](#)
- Engrave\_Load
  - engrave\_load\_edc, [44](#)
  - engrave\_load\_edj, [44](#)
- engrave\_load.h, [146](#)
- Engrave\_Load: Contains the functions to load EDC, [44](#)
- engrave\_load\_edc
  - Engrave\_Load, [44](#)
- engrave\_load\_edj
  - Engrave\_Load, [44](#)
- Engrave\_Out
  - engrave\_edc\_output, [46](#)
  - engrave\_edj\_output, [46](#)
- engrave\_out.h, [147](#)
- Engrave\_Part
  - engrave\_part\_clip\_to\_get, [50](#)
  - engrave\_part\_clip\_to\_set, [50](#)
  - engrave\_part\_dragable\_confine\_get, [50](#)
  - engrave\_part\_dragable\_confine\_set, [51](#)
  - engrave\_part\_dragable\_x\_get, [51](#)
  - engrave\_part\_dragable\_x\_set, [51](#)
  - engrave\_part\_dragable\_y\_get, [52](#)
  - engrave\_part\_dragable\_y\_set, [52](#)
  - engrave\_part\_effect\_get, [52](#)
  - engrave\_part\_effect\_set, [53](#)
  - engrave\_part\_evas\_object\_get, [53](#)
  - engrave\_part\_evas\_object\_set, [53](#)
  - engrave\_part\_free, [53](#)

- engrave\_part\_mouse\_events\_get, [54](#)
- engrave\_part\_mouse\_events\_set, [54](#)
- engrave\_part\_name\_get, [54](#)
- engrave\_part\_name\_set, [54](#)
- engrave\_part\_new, [55](#)
- engrave\_part\_parent\_get, [55](#)
- engrave\_part\_parent\_set, [55](#)
- engrave\_part\_repeat\_events\_get, [55](#)
- engrave\_part\_repeat\_events\_set, [56](#)
- engrave\_part\_state\_add, [56](#)
- engrave\_part\_state\_by\_name\_value\_-  
find, [56](#)
- engrave\_part\_state\_foreach, [57](#)
- engrave\_part\_state\_last\_get, [57](#)
- engrave\_part\_states\_count, [57](#)
- engrave\_part\_type\_get, [58](#)
- engrave\_part\_type\_set, [58](#)
- engrave\_part.h, [148](#)
- Engrave\_Part: Functions to work with part ob-  
jects, [48](#)
- engrave\_part\_clip\_to\_get  
Engrave\_Part, [50](#)
- engrave\_part\_clip\_to\_set  
Engrave\_Part, [50](#)
- engrave\_part\_dragable\_confine\_get  
Engrave\_Part, [50](#)
- engrave\_part\_dragable\_confine\_set  
Engrave\_Part, [51](#)
- engrave\_part\_dragable\_x\_get  
Engrave\_Part, [51](#)
- engrave\_part\_dragable\_x\_set  
Engrave\_Part, [51](#)
- engrave\_part\_dragable\_y\_get  
Engrave\_Part, [52](#)
- engrave\_part\_dragable\_y\_set  
Engrave\_Part, [52](#)
- engrave\_part\_effect\_get  
Engrave\_Part, [52](#)
- engrave\_part\_effect\_set  
Engrave\_Part, [53](#)
- engrave\_part\_evas\_object\_get  
Engrave\_Part, [53](#)
- engrave\_part\_evas\_object\_set  
Engrave\_Part, [53](#)
- engrave\_part\_free  
Engrave\_Part, [53](#)
- engrave\_part\_mouse\_events\_get  
Engrave\_Part, [54](#)
- engrave\_part\_mouse\_events\_set  
Engrave\_Part, [54](#)
- engrave\_part\_name\_get  
Engrave\_Part, [54](#)
- engrave\_part\_name\_set  
Engrave\_Part, [54](#)
- engrave\_part\_new  
Engrave\_Part, [55](#)
- engrave\_part\_parent\_get  
Engrave\_Part, [55](#)
- engrave\_part\_parent\_set  
Engrave\_Part, [55](#)
- engrave\_part\_repeat\_events\_get  
Engrave\_Part, [55](#)
- engrave\_part\_repeat\_events\_set  
Engrave\_Part, [56](#)
- Engrave\_Part\_State
  - engrave\_part\_state\_align\_get, [65](#)
  - engrave\_part\_state\_align\_set, [66](#)
  - engrave\_part\_state\_aspect\_get, [66](#)
  - engrave\_part\_state\_aspect\_preference\_-  
get, [66](#)
  - engrave\_part\_state\_aspect\_preference\_-  
set, [67](#)
  - engrave\_part\_state\_aspect\_set, [67](#)
  - engrave\_part\_state\_color2\_get, [67](#)
  - engrave\_part\_state\_color2\_set, [68](#)
  - engrave\_part\_state\_color3\_get, [68](#)
  - engrave\_part\_state\_color3\_set, [69](#)
  - engrave\_part\_state\_color\_class\_get, [69](#)
  - engrave\_part\_state\_color\_class\_set, [69](#)
  - engrave\_part\_state\_color\_get, [69](#)
  - engrave\_part\_state\_color\_set, [70](#)
  - engrave\_part\_state\_copy, [70](#)
  - engrave\_part\_state\_fill\_origin\_offset\_-  
set, [70](#)
  - engrave\_part\_state\_fill\_origin\_-  
relative\_set, [71](#)
  - engrave\_part\_state\_fill\_size\_offset\_set,  
[71](#)
  - engrave\_part\_state\_fill\_size\_relative\_-  
set, [71](#)
  - engrave\_part\_state\_fill\_smooth\_set, [72](#)
  - engrave\_part\_state\_fixed\_size\_get, [72](#)
  - engrave\_part\_state\_fixed\_size\_set, [72](#)
  - engrave\_part\_state\_free, [73](#)
  - engrave\_part\_state\_gradient\_-  
spectrum\_get, [73](#)
  - engrave\_part\_state\_gradient\_-  
spectrum\_set, [73](#)
  - engrave\_part\_state\_gradient\_type\_get,  
[74](#)
  - engrave\_part\_state\_gradient\_type\_set,  
[74](#)
  - engrave\_part\_state\_image\_border\_get,  
[74](#)
  - engrave\_part\_state\_image\_border\_set,  
[74](#)
  - engrave\_part\_state\_image\_middle\_get,  
[75](#)

- engrave\_part\_state\_image\_middle\_set, 75
- engrave\_part\_state\_image\_normal\_get, 75
- engrave\_part\_state\_image\_normal\_set, 76
- engrave\_part\_state\_image\_tween\_add, 76
- engrave\_part\_state\_max\_size\_get, 76
- engrave\_part\_state\_max\_size\_set, 77
- engrave\_part\_state\_min\_size\_get, 77
- engrave\_part\_state\_min\_size\_set, 77
- engrave\_part\_state\_name\_get, 78
- engrave\_part\_state\_name\_set, 78
- engrave\_part\_state\_new, 78
- engrave\_part\_state\_parent\_get, 78
- engrave\_part\_state\_parent\_set, 79
- engrave\_part\_state\_rell\_offset\_get, 79
- engrave\_part\_state\_rell\_offset\_set, 79
- engrave\_part\_state\_rell\_relative\_get, 80
- engrave\_part\_state\_rell\_relative\_set, 80
- engrave\_part\_state\_rell\_to\_set, 80
- engrave\_part\_state\_rell\_to\_x\_get, 81
- engrave\_part\_state\_rell\_to\_x\_set, 81
- engrave\_part\_state\_rell\_to\_y\_get, 81
- engrave\_part\_state\_rell\_to\_y\_set, 81
- engrave\_part\_state\_rel2\_offset\_get, 82
- engrave\_part\_state\_rel2\_offset\_set, 82
- engrave\_part\_state\_rel2\_relative\_get, 82
- engrave\_part\_state\_rel2\_relative\_set, 83
- engrave\_part\_state\_rel2\_to\_set, 83
- engrave\_part\_state\_rel2\_to\_x\_get, 83
- engrave\_part\_state\_rel2\_to\_x\_set, 84
- engrave\_part\_state\_rel2\_to\_y\_get, 84
- engrave\_part\_state\_rel2\_to\_y\_set, 84
- engrave\_part\_state\_step\_get, 85
- engrave\_part\_state\_step\_set, 85
- engrave\_part\_state\_text\_align\_get, 85
- engrave\_part\_state\_text\_align\_set, 86
- engrave\_part\_state\_text\_elipsis\_get, 86
- engrave\_part\_state\_text\_elipsis\_set, 86
- engrave\_part\_state\_text\_fit\_get, 86
- engrave\_part\_state\_text\_fit\_set, 87
- engrave\_part\_state\_text\_font\_get, 87
- engrave\_part\_state\_text\_font\_set, 87
- engrave\_part\_state\_text\_max\_set, 88
- engrave\_part\_state\_text\_min\_get, 88
- engrave\_part\_state\_text\_min\_set, 88
- engrave\_part\_state\_text\_size\_get, 89
- engrave\_part\_state\_text\_size\_set, 89
- engrave\_part\_state\_text\_source\_get, 89
- engrave\_part\_state\_text\_source\_set, 90
- engrave\_part\_state\_text\_style\_get, 90
- engrave\_part\_state\_text\_style\_set, 90
- engrave\_part\_state\_text\_text\_append, 90
- engrave\_part\_state\_text\_text\_class\_get, 91
- engrave\_part\_state\_text\_text\_class\_set, 91
- engrave\_part\_state\_text\_text\_get, 91
- engrave\_part\_state\_text\_text\_set, 92
- engrave\_part\_state\_text\_text\_source\_get, 92
- engrave\_part\_state\_text\_text\_source\_set, 92
- engrave\_part\_state\_tween\_foreach, 92
- engrave\_part\_state\_tweens\_count, 93
- engrave\_part\_state\_visible\_get, 93
- engrave\_part\_state\_visible\_set, 93
- engrave\_part\_state.h, 151
- Engrave\_Part\_State: Functions to work with the Engrave\_Part\_State objects, 59
- engrave\_part\_state\_add
  - Engrave\_Part, 56
- engrave\_part\_state\_align\_get
  - Engrave\_Part\_State, 65
- engrave\_part\_state\_align\_set
  - Engrave\_Part\_State, 66
- engrave\_part\_state\_aspect\_get
  - Engrave\_Part\_State, 66
- engrave\_part\_state\_aspect\_preference\_get
  - Engrave\_Part\_State, 66
- engrave\_part\_state\_aspect\_preference\_set
  - Engrave\_Part\_State, 67
- engrave\_part\_state\_aspect\_set
  - Engrave\_Part\_State, 67
- engrave\_part\_state\_by\_name\_value\_find
  - Engrave\_Part, 56
- engrave\_part\_state\_color2\_get
  - Engrave\_Part\_State, 67
- engrave\_part\_state\_color2\_set
  - Engrave\_Part\_State, 68
- engrave\_part\_state\_color3\_get
  - Engrave\_Part\_State, 68
- engrave\_part\_state\_color3\_set
  - Engrave\_Part\_State, 69
- engrave\_part\_state\_color\_class\_get
  - Engrave\_Part\_State, 69
- engrave\_part\_state\_color\_class\_set
  - Engrave\_Part\_State, 69
- engrave\_part\_state\_color\_get
  - Engrave\_Part\_State, 69
- engrave\_part\_state\_color\_set

Engrave\_Part\_State, 70  
 engrave\_part\_state\_copy  
     Engrave\_Part\_State, 70  
 engrave\_part\_state\_fill\_origin\_offset\_set  
     Engrave\_Part\_State, 70  
 engrave\_part\_state\_fill\_origin\_relative\_set  
     Engrave\_Part\_State, 71  
 engrave\_part\_state\_fill\_size\_offset\_set  
     Engrave\_Part\_State, 71  
 engrave\_part\_state\_fill\_size\_relative\_set  
     Engrave\_Part\_State, 71  
 engrave\_part\_state\_fill\_smooth\_set  
     Engrave\_Part\_State, 72  
 engrave\_part\_state\_fixed\_size\_get  
     Engrave\_Part\_State, 72  
 engrave\_part\_state\_fixed\_size\_set  
     Engrave\_Part\_State, 72  
 engrave\_part\_state\_foreach  
     Engrave\_Part, 57  
 engrave\_part\_state\_free  
     Engrave\_Part\_State, 73  
 engrave\_part\_state\_gradient\_spectrum\_get  
     Engrave\_Part\_State, 73  
 engrave\_part\_state\_gradient\_spectrum\_set  
     Engrave\_Part\_State, 73  
 engrave\_part\_state\_gradient\_type\_get  
     Engrave\_Part\_State, 74  
 engrave\_part\_state\_gradient\_type\_set  
     Engrave\_Part\_State, 74  
 engrave\_part\_state\_image\_border\_get  
     Engrave\_Part\_State, 74  
 engrave\_part\_state\_image\_border\_set  
     Engrave\_Part\_State, 74  
 engrave\_part\_state\_image\_middle\_get  
     Engrave\_Part\_State, 75  
 engrave\_part\_state\_image\_middle\_set  
     Engrave\_Part\_State, 75  
 engrave\_part\_state\_image\_normal\_get  
     Engrave\_Part\_State, 75  
 engrave\_part\_state\_image\_normal\_set  
     Engrave\_Part\_State, 76  
 engrave\_part\_state\_image\_tween\_add  
     Engrave\_Part\_State, 76  
 engrave\_part\_state\_last\_get  
     Engrave\_Part, 57  
 engrave\_part\_state\_max\_size\_get  
     Engrave\_Part\_State, 76  
 engrave\_part\_state\_max\_size\_set  
     Engrave\_Part\_State, 77  
 engrave\_part\_state\_min\_size\_get  
     Engrave\_Part\_State, 77  
 engrave\_part\_state\_min\_size\_set  
     Engrave\_Part\_State, 77  
 engrave\_part\_state\_name\_get  
     Engrave\_Part\_State, 78  
 engrave\_part\_state\_name\_set  
     Engrave\_Part\_State, 78  
 engrave\_part\_state\_new  
     Engrave\_Part\_State, 78  
 engrave\_part\_state\_parent\_get  
     Engrave\_Part\_State, 78  
 engrave\_part\_state\_parent\_set  
     Engrave\_Part\_State, 79  
 engrave\_part\_state\_rel1\_offset\_get  
     Engrave\_Part\_State, 79  
 engrave\_part\_state\_rel1\_offset\_set  
     Engrave\_Part\_State, 79  
 engrave\_part\_state\_rel1\_relative\_get  
     Engrave\_Part\_State, 80  
 engrave\_part\_state\_rel1\_relative\_set  
     Engrave\_Part\_State, 80  
 engrave\_part\_state\_rel1\_to\_set  
     Engrave\_Part\_State, 80  
 engrave\_part\_state\_rel1\_to\_x\_get  
     Engrave\_Part\_State, 81  
 engrave\_part\_state\_rel1\_to\_x\_set  
     Engrave\_Part\_State, 81  
 engrave\_part\_state\_rel1\_to\_y\_get  
     Engrave\_Part\_State, 81  
 engrave\_part\_state\_rel1\_to\_y\_set  
     Engrave\_Part\_State, 81  
 engrave\_part\_state\_rel2\_offset\_get  
     Engrave\_Part\_State, 82  
 engrave\_part\_state\_rel2\_offset\_set  
     Engrave\_Part\_State, 82  
 engrave\_part\_state\_rel2\_relative\_get  
     Engrave\_Part\_State, 82  
 engrave\_part\_state\_rel2\_relative\_set  
     Engrave\_Part\_State, 83  
 engrave\_part\_state\_rel2\_to\_set  
     Engrave\_Part\_State, 83  
 engrave\_part\_state\_rel2\_to\_x\_get  
     Engrave\_Part\_State, 83  
 engrave\_part\_state\_rel2\_to\_x\_set  
     Engrave\_Part\_State, 84  
 engrave\_part\_state\_rel2\_to\_y\_get  
     Engrave\_Part\_State, 84  
 engrave\_part\_state\_rel2\_to\_y\_set  
     Engrave\_Part\_State, 84  
 engrave\_part\_state\_step\_get  
     Engrave\_Part\_State, 85  
 engrave\_part\_state\_step\_set  
     Engrave\_Part\_State, 85  
 engrave\_part\_state\_text\_align\_get  
     Engrave\_Part\_State, 85  
 engrave\_part\_state\_text\_align\_set  
     Engrave\_Part\_State, 86  
 engrave\_part\_state\_text\_elipsis\_get

- Engrave\_Part\_State, [86](#)
- engrave\_part\_state\_text\_elipsis\_set
  - Engrave\_Part\_State, [86](#)
- engrave\_part\_state\_text\_fit\_get
  - Engrave\_Part\_State, [86](#)
- engrave\_part\_state\_text\_fit\_set
  - Engrave\_Part\_State, [87](#)
- engrave\_part\_state\_text\_font\_get
  - Engrave\_Part\_State, [87](#)
- engrave\_part\_state\_text\_font\_set
  - Engrave\_Part\_State, [87](#)
- engrave\_part\_state\_text\_max\_set
  - Engrave\_Part\_State, [88](#)
- engrave\_part\_state\_text\_min\_get
  - Engrave\_Part\_State, [88](#)
- engrave\_part\_state\_text\_min\_set
  - Engrave\_Part\_State, [88](#)
- engrave\_part\_state\_text\_size\_get
  - Engrave\_Part\_State, [89](#)
- engrave\_part\_state\_text\_size\_set
  - Engrave\_Part\_State, [89](#)
- engrave\_part\_state\_text\_source\_get
  - Engrave\_Part\_State, [89](#)
- engrave\_part\_state\_text\_source\_set
  - Engrave\_Part\_State, [90](#)
- engrave\_part\_state\_text\_style\_get
  - Engrave\_Part\_State, [90](#)
- engrave\_part\_state\_text\_style\_set
  - Engrave\_Part\_State, [90](#)
- engrave\_part\_state\_text\_text\_append
  - Engrave\_Part\_State, [90](#)
- engrave\_part\_state\_text\_text\_class\_get
  - Engrave\_Part\_State, [91](#)
- engrave\_part\_state\_text\_text\_class\_set
  - Engrave\_Part\_State, [91](#)
- engrave\_part\_state\_text\_text\_get
  - Engrave\_Part\_State, [91](#)
- engrave\_part\_state\_text\_text\_set
  - Engrave\_Part\_State, [92](#)
- engrave\_part\_state\_text\_text\_source\_get
  - Engrave\_Part\_State, [92](#)
- engrave\_part\_state\_text\_text\_source\_set
  - Engrave\_Part\_State, [92](#)
- engrave\_part\_state\_tween\_foreach
  - Engrave\_Part\_State, [92](#)
- engrave\_part\_state\_tweens\_count
  - Engrave\_Part\_State, [93](#)
- engrave\_part\_state\_visible\_get
  - Engrave\_Part\_State, [93](#)
- engrave\_part\_state\_visible\_set
  - Engrave\_Part\_State, [93](#)
- engrave\_part\_states\_count
  - Engrave\_Part, [57](#)
- engrave\_part\_type\_get
  - Engrave\_Part, [58](#)
- ENGRAVE\_PART\_TYPE\_GRADIENT
  - Engrave\_Enums, [17](#)
- ENGRAVE\_PART\_TYPE\_IMAGE
  - Engrave\_Enums, [17](#)
- ENGRAVE\_PART\_TYPE\_RECT
  - Engrave\_Enums, [17](#)
- engrave\_part\_type\_set
  - Engrave\_Part, [58](#)
- ENGRAVE\_PART\_TYPE\_SWALLOW
  - Engrave\_Enums, [17](#)
- ENGRAVE\_PART\_TYPE\_TEXT
  - Engrave\_Enums, [17](#)
- ENGRAVE\_PART\_TYPE\_TEXTBLOCK
  - Engrave\_Enums, [17](#)
- Engrave\_Program
  - engrave\_program\_action\_get, [97](#)
  - engrave\_program\_action\_set, [97](#)
  - engrave\_program\_after\_add, [97](#)
  - engrave\_program\_after\_foreach, [98](#)
  - engrave\_program\_afters\_count, [98](#)
  - engrave\_program\_free, [98](#)
  - engrave\_program\_in\_get, [99](#)
  - engrave\_program\_in\_set, [99](#)
  - engrave\_program\_name\_get, [99](#)
  - engrave\_program\_name\_set, [99](#)
  - engrave\_program\_new, [100](#)
  - engrave\_program\_parent\_get, [100](#)
  - engrave\_program\_parent\_set, [100](#)
  - engrave\_program\_script\_get, [100](#)
  - engrave\_program\_script\_set, [101](#)
  - engrave\_program\_signal\_get, [101](#)
  - engrave\_program\_signal\_set, [101](#)
  - engrave\_program\_source\_get, [102](#)
  - engrave\_program\_source\_set, [102](#)
  - engrave\_program\_target\_add, [102](#)
  - engrave\_program\_target\_foreach, [102](#)
  - engrave\_program\_targets\_count, [103](#)
  - engrave\_program\_transition\_get, [103](#)
  - engrave\_program\_transition\_set, [103](#)
- engrave\_program.h, [159](#)
- Engrave\_Program: Functions to work with
  - Engrave\_Program objects., [95](#)
- engrave\_program\_action\_get
  - Engrave\_Program, [97](#)
- engrave\_program\_action\_set
  - Engrave\_Program, [97](#)
- engrave\_program\_after\_add
  - Engrave\_Program, [97](#)
- engrave\_program\_after\_foreach
  - Engrave\_Program, [98](#)
- engrave\_program\_afters\_count
  - Engrave\_Program, [98](#)
- engrave\_program\_free



- Engrave\_Program, [98](#)
- engrave\_program\_in\_get
  - Engrave\_Program, [99](#)
- engrave\_program\_in\_set
  - Engrave\_Program, [99](#)
- engrave\_program\_name\_get
  - Engrave\_Program, [99](#)
- engrave\_program\_name\_set
  - Engrave\_Program, [99](#)
- engrave\_program\_new
  - Engrave\_Program, [100](#)
- engrave\_program\_parent\_get
  - Engrave\_Program, [100](#)
- engrave\_program\_parent\_set
  - Engrave\_Program, [100](#)
- engrave\_program\_script\_get
  - Engrave\_Program, [100](#)
- engrave\_program\_script\_set
  - Engrave\_Program, [101](#)
- engrave\_program\_signal\_get
  - Engrave\_Program, [101](#)
- engrave\_program\_signal\_set
  - Engrave\_Program, [101](#)
- engrave\_program\_source\_get
  - Engrave\_Program, [102](#)
- engrave\_program\_source\_set
  - Engrave\_Program, [102](#)
- engrave\_program\_target\_add
  - Engrave\_Program, [102](#)
- engrave\_program\_target\_foreach
  - Engrave\_Program, [102](#)
- engrave\_program\_targets\_count
  - Engrave\_Program, [103](#)
- engrave\_program\_transition\_get
  - Engrave\_Program, [103](#)
- engrave\_program\_transition\_set
  - Engrave\_Program, [103](#)
- Engrave\_Spectrum
  - engrave\_spectrum\_color\_add, [106](#)
  - engrave\_spectrum\_color\_count, [106](#)
  - engrave\_spectrum\_color\_del, [106](#)
  - engrave\_spectrum\_color\_free, [107](#)
  - engrave\_spectrum\_color\_new, [107](#)
  - engrave\_spectrum\_color\_nth, [107](#)
  - engrave\_spectrum\_free, [108](#)
  - engrave\_spectrum\_name\_get, [108](#)
  - engrave\_spectrum\_name\_set, [108](#)
  - engrave\_spectrum\_new, [109](#)
  - engrave\_spectrum\_parent\_set, [109](#)
- engrave\_spectrum.h, [162](#)
- Engrave\_Spectrum: Functions to work with engrave spectrum objects, [105](#)
- engrave\_spectrum\_color\_add
  - Engrave\_Spectrum, [106](#)
- engrave\_spectrum\_color\_count
  - Engrave\_Spectrum, [106](#)
- engrave\_spectrum\_color\_del
  - Engrave\_Spectrum, [106](#)
- engrave\_spectrum\_color\_free
  - Engrave\_Spectrum, [107](#)
- engrave\_spectrum\_color\_new
  - Engrave\_Spectrum, [107](#)
- engrave\_spectrum\_color\_nth
  - Engrave\_Spectrum, [107](#)
- engrave\_spectrum\_free
  - Engrave\_Spectrum, [108](#)
- engrave\_spectrum\_name\_get
  - Engrave\_Spectrum, [108](#)
- engrave\_spectrum\_name\_set
  - Engrave\_Spectrum, [108](#)
- engrave\_spectrum\_new
  - Engrave\_Spectrum, [109](#)
- engrave\_spectrum\_parent\_set
  - Engrave\_Spectrum, [109](#)
- Engrave\_Style
  - engrave\_style\_new, [110](#)
- engrave\_style.h, [164](#)
- Engrave\_Style: Functions to work with Engrave\_Style blocks., [110](#)
- engrave\_style\_new
  - Engrave\_Style, [110](#)
- ENGRAVE\_TEXT\_EFFECT\_NONE
  - Engrave\_Enums, [17](#)
- ENGRAVE\_TEXT\_EFFECT\_OUTLINE
  - Engrave\_Enums, [17](#)
- ENGRAVE\_TEXT\_EFFECT\_OUTLINE\_SHADOW
  - Engrave\_Enums, [17](#)
- ENGRAVE\_TEXT\_EFFECT\_OUTLINE\_SOFT\_SHADOW
  - Engrave\_Enums, [17](#)
- ENGRAVE\_TEXT\_EFFECT\_PLAIN
  - Engrave\_Enums, [17](#)
- ENGRAVE\_TEXT\_EFFECT\_SHADOW
  - Engrave\_Enums, [17](#)
- ENGRAVE\_TEXT\_EFFECT\_SOFT\_OUTLINE
  - Engrave\_Enums, [17](#)
- ENGRAVE\_TEXT\_EFFECT\_SOFT\_SHADOW
  - Engrave\_Enums, [17](#)
- ENGRAVE\_TRANSITION\_ACCELERATE
  - Engrave\_Enums, [17](#)
- ENGRAVE\_TRANSITION\_DECELERATE
  - Engrave\_Enums, [17](#)
- ENGRAVE\_TRANSITION\_LINEAR
  - Engrave\_Enums, [17](#)
- ENGRAVE\_TRANSITION\_SINUSOIDAL

Engrave\_Enums, [17](#)

Enums: Various Flags and Enumerations used  
in Engrave., [15](#)

Functions needed to output the Engrave data  
into, [46](#)