



Prologue

The world is

You can effectively scroll around the canvas simply by changing the viewport location, and you can zoom in and out by simply changing the viewport size (a

Evas Programmers Guide

By Carsten Haitzler, Tjeerd Veenstra (Tje)(2,8,30,34,62) Raja (9) Raja (9) Raja (5,9,10) Raja (9,10,15) Raja (1,5) Raja (1,5) Raja (1,5)

Evas Programmers Guide

Evas Programmers Guide

By Carsten Haitzler, Copyright © 2005-2007 EFL Group LLC

E

solid colour, but do not want to go to the expense of needing an image for it. They can be used to highlight areas with lighting and shadow effects and many other things.

Creating and managing these objects, like almost everything else, is also child's play. Take a look at this:

```
object = evas_add_gradient_box(evas);
evas_move(evas, object, 150.0, 100.0);
evas_resize(evas, object, 200.0, 120.0);
gradient = evas_gradient_new();
evas_gradient_add_color(gradient, 255, 255, 255, 255, 10);
evas_gradient_add_color(gradient, 255, 255, 0, 255, 10);
evas_gradient_add_color(gradient, 255, 0, 0, 255, 10);
evas_gradient_add_color(gradient, 0, 0, 128, 255, 10);
evas_gradient_add_color(gradient, 0, 0, 128, 0, 10);
evas_set_gradient(evas, object, gradient);
evas_gradient_free(gradient);
evas_set_angle(evas, object, 290.0);
evas_show(evas, object);
```

Now the program we are developing looks now something like  Tj 5.04004 0 Td(s)j 6 0 Td(g)Tj 6 0 Td(r)Td(i)T

Evas Programmers Guide

By Ca.52002 0 Td(r)Tj 3.35999 03999 0 Td(t)Tj 2.85 0 Td(e)Tj 5.52 0 Td(n)Tj 5.52 0 Td()Tj 2.85 0 Td(H)Tj 7.2 0 Td(

goes IDLE, or you ABSOLUTELY NEED a rendering pass done – i.e. no more events to process and no state to change. Doing otherwise will impact performance badly.

Remember e

Evas Programmers Guide

By Carsten Haitzler, Copyright © 2001 Carsten Ha

Evas

Evas Programmers Guide

By Carsten Haitzler, Copyright © 2001 Carsten Haitzler (The Rasterman)

they are required again in the future.

image_cache

This specifies to Evas how many bytes of memory are to be used to cache image data that isn't currently required for rendering of the visible Evas area. This works just like the font cache, but for image pixel data.



```
void evas_free(Evas e);
```

This function will free all objects in the Evas canvas passed to it, free the canvas and all memory used by that canvas. If Evas created the window for you for this canvas with `evas_new_all()`, this window will also be destroyed.

Arguments:

`e`

A valid Evas canvas handle.


```
Visual *evas_get_visual(Evas e);
```

This function returns the visual currently being used by the Evas passed in.

Return:

The return value is the currentl.03999 0 Td()Tj 2.87998 08 0 Td(u)Tj 5.51997 0 Td(s)Tj 5.04004 0 Td


```
int evas_get_color$all
```


Evas Programmers Guide
By Carsten Haitzha

Eva a aa

Evva E

E E

```
void evas_render(Evas e);
```

This function flushes all changes and updates that have been queued whilst objects have been created and destroyed in the Evas canvas, and properties of the objects have been set.

```
Visual *evas_get_optimal_visual(Evas e, Display *disp);
```

This function returns the optimal visual to use for the destination window for Evas output if used on the specified display. It is NOT valid to call this before evas_set_ri0u5.52002603999 0 Td(_)

Evas Programmers Guide

By Carsten Höglund 5.52 0 Td(i)Tj 2.15999

void evas

Evas Programmers Guide

By Carsten Haitzler, Copyright (C) 2005-2008 EFL Group 0 Td(C)Tj 2.48 0 Td(z)T,

Ev

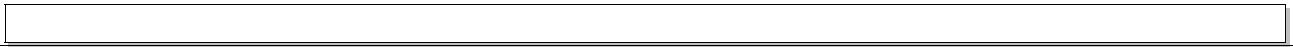
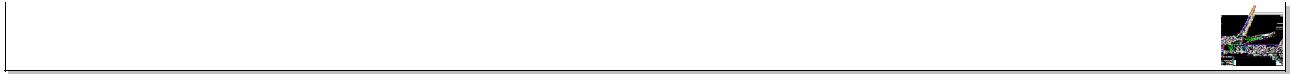
Eva

Evas Programme

Evas Programmers Guide

By

Evas Programmers Gui



o

The Evas object handle `�mObj` is `1052475025215`

Evas Programm

Evas Programmers Guide

By Carsten Haitzler, Copyright © 2001 Carsten Haitzler (The Rasterman)

```
void evas_set_line_xy(Evas e, Evas_Object o, double x1, double y1,  
                      double x2, double y2);
```

This fun 6.23999 0 Td(h)Tj 5d h

Evas

Evas

```
void evas_set_font_cache(Evas e, int size);
```

This function sets the size of the font cache for the


```
void evas_font_add_path(Evas e, char *path);
```

This function adds the path to the font's internal font path.

Eva

Ev

Evas Programmers Guide

By Carsten Haitzler, Copyright © 2001 Carsten Haitzler (The Rasterman)

```
void es_gr0ad6c0lTd0grd6Tjd6j66TdtdTj6600Td4etdj66Td40d1j)6j06Td(6  
i b, i a, i dis);
```

r to the specified gradient asla specified "distance" from the colour

. The first colour 0 Td(e)Tj 5.52002 0 Td(d)Tj 5.52002 0 Td(d)Tj 5.52 0 Td(e)Tj 5.51997 0 Td(d)Tj 5.52002 0 Td(0 Td

Evas Programmers Guide

By Carsten Haitzler, Copyright © 2001 Carsten Haitzler


```
char *evas_object_get_name(Evas e, Evas_Object o);
```

This function will return the name set on a specific object, if there is one, or NULL if there is not.

Return:

```
Evas_List evas_get_points(Evas e, Evas_Object o);
```

This function returns a list of hoints in a holygon object. If the holygon object has no hoints, or the object handle is not a polygon object, NULL is returned.

Evas Programmers Guide

By Carsten Haitzler, Copyright © 2001 Carsn OHaocso Td(1)Tj 5.52 0 T0 Td(i27.2 0 Td(ah0 Td(e)T43999 0 T)Tj 5.52

Evas Programmers Guide

By Carsten Haitzler, Copyright © 2001 Carsten Haitzler (The Rasterman)

ImlH

Evas

```
double evas_screen_x_to_world(Evas e, int x);
```

This function is part of the `Font` interface in `0x0000000000000002`. `0 Td()Tj 2.87998p`


```
char *evas_get_text_string(Evas e, Evas_Object o);
```

This function returns a pointer to the internal string of the specified object in the canvas specified. This is only valid as long as the object is not destroyed and ed

Ev

```
double evas_get_text_height(Evas e, Evas_Object o);
```

This function returns the total height of a text object in canvas units. If the object specified is not a text object, 0 is returned.

Return:

A double precision float value that

Evas Programmers Guide

By Carsten Haitzler, Copyright © 2001 Carsten Haitzler (The Rasterman)

```
void evas_text_at(Evas e, Evas_Object o, int index,  
                  double *char_x, double *char_y,
```


Evas Programmers Guide

By Arst na itler, copyright 2001 Arst na itler The Rasterman

```
void evas_text_get_max_ascent_descent(Evas e, Evas_Object o,
                                      double *ascent, double *descent);
```

This function operates the same way `evas_text_get_ascent_descent()` works, but returns the maximum extent for ~~text~~ 0.


```
Evas_Object *evas_get_object_under_mouse(Evas *e);
```

This function returns a handle to the object that is currently under the moj 5.5210205710000001f


```
void *evas_remove_data(Evas e, Evas_Object o, char *key);
```

This function removes the data pointer attached to the specified object and returned the pointer. If no data was attached NULL is returned.

Return:

A pointer to data attached under the key t

Ev

Evas Programmers Guide

By Carsten Haitzler, Copyright © 2001 Carsten Haitzler (The Rasterman)  `es6001 0 Td(m)6001 0 _d(m)6001`

Eva

Evas Programmers Guide

By Carsten Haitzler, Copyright © 2001 Carsten Haitzler TextFontj 3.35999 T Textd(B)2j 5.03999 h Textd(a)Tj 5.52002

Evas Programm



Eva

Evas Programmers Guide

By Carsten Haitzler, Copyright © 2001 Carsten Haitzler (The Rasterman)

```
/* also creates the window for us in the right colormap & visual */
evas = evas_new_all(e_d
```


Demo Example:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <sys/time.h>
#include <unistd.h>
#include <Evas.h>
#include <Ecore.h>

#define MAX
```



```
    e = (Ev_Mouse_Down *)ev->event;
    evas_event_button_down(evas, e->x, e->y, e->button);
}

static void
e_mouse_up(Event * ev)
{
    Ev_Mouse_Up      *e;

    e = (Ev_Mouse_Up *)ev->event;
    evas_event_button_up(evas, e->x, e->y, e->button)
```