



# EGL

*- Embedded Graphic Library -*

**Ver 1.00**  
**December 31, 2012**

**Advanced Digital Chips Inc.**

## History

2012-12-31

Created

## adStarSDK Reference Manual

---

*©Advanced Digital Chips Inc.*

All right reserved.

No part of this document may be reproduced in any form without written permission from Advanced Digital Chips Inc.

Advanced Digital Chips Inc. reserves the right to change in its products or product specification to improve function or design at any time, without notice.

### ***Office***

22F, Bldg A, KeumkangPentarium IT Tower,  
810, Gwanyang\_dong,Dongan-Gu, Anyang-si, Gyeonggi-do, 431-060, Korea  
Tel : +82-31-463-7500  
Fax : +82-31-463-7588  
URL : <http://www.adc.co.kr>

— Table of Contents —

<b>EGL Library</b> .....	<b>10</b>
<b>Window</b> .....	<b>13</b>
▶ <b>egl_create_window</b> function .....	14
▶ <b>egl_window_show</b> function.....	15
▶ <b>egl_window_isshow</b> function.....	16
▶ <b>egl_window_invalidate</b> function .....	17
▶ <b>egl_window_invalidate_rect</b> function .....	18
▶ <b>egl_window_redraw_rect</b> function .....	19
▶ <b>egl_window_set_bg</b> function .....	20
▶ <b>egl_window_get_bg</b> function.....	21
▶ <b>egl_window_add_object</b> function.....	22
▶ <b>egl_window_set_callback</b> function.....	23
▶ <b>egl_window_get_active</b> function.....	24
▶ <b>egl_user_touch_input</b> function .....	25
▶ <b>egl_draw</b> function.....	26
▶ <b>Window Example</b> .....	27
<b>Button</b> .....	<b>29</b>
▶ <b>egl_create_button</b> function .....	30
▶ <b>egl_button_callback</b> function .....	31
▶ <b>Button Example</b> .....	32
<b>Image button</b> .....	<b>33</b>
▶ <b>egl_create_image_button</b> function.....	34
▶ <b>egl_image_button_callback</b> function.....	35
▶ <b>Image Button Example</b> .....	36
<b>Toggle image button</b> .....	<b>37</b>

▶ <b>egl_create_toggle_image</b> function.....	38
▶ <b>egl_toggle_image_callback</b> function.....	39
▶ <b>egl_toggle_image_set_on</b> function.....	40
▶ <b>Toggle Image Button Example.</b> .....	41
<b>Label.....</b>	<b>42</b>
▶ <b>egl_create_label</b> function.....	43
▶ <b>egl_label_set_text</b> function .....	44
▶ <b>egl_label_set_redraw_bg</b> function.....	45
▶ <b>egl_label_set_color</b> function.....	46
▶ <b>Label Example.</b> .....	47
<b>Check button / Radio button .....</b>	<b>48</b>
▶ <b>egl_create_checkbutton</b> function.....	49
▶ <b>egl_checkbutton_callback</b> function.....	50
▶ <b>egl_checkbutton_set_check</b> function.....	51
▶ <b>egl_checkbutton_get_check</b> function.....	52
▶ <b>egl_checkbutton_set_style</b> function.....	53
▶ <b>Check / Radio button Example.</b> .....	54
<b>Progress Bar .....</b>	<b>56</b>
▶ <b>egl_create_progressbar</b> function.....	57
▶ <b>egl_progressbar_set_barcolor</b> function.....	59
▶ <b>egl_progressbar_set_bgcolor</b> function .....	60
▶ <b>egl_progressbar_set_textcolor</b> function.....	61
▶ <b>egl_progressbar_set_text</b> function.....	62
▶ <b>egl_progressbar_set_pos</b> function .....	63
▶ <b>egl_progressbar_get_pos</b> function.....	64
▶ <b>Progress Bar Example.</b> .....	65

<b>Scroll Bar</b> .....	<b>66</b>
▶ <b>egl_create_scrollbar</b> function.....	68
▶ <b>egl_scrollbar_callback</b> function .....	70
▶ <b>egl_scroll_set_position</b> function.....	71
▶ <b>egl_scroll_get_position</b> function .....	72
▶ <b>egl_scroll_set_totalcount</b> function .....	73
▶ <b>egl_scroll_get_totalcount</b> function .....	74
▶ <b>egl_scroll_set_viewcount</b> function .....	75
▶ <b>egl_scroll_get_viewcount</b> function.....	76
▶ <b>egl_scroll_set_upcount</b> function.....	77
▶ <b>egl_scroll_get_upcount</b> function.....	78
▶ <b>egl_scroll_set_bgcolor</b> function.....	79
▶ <b>egl_scroll_set_size</b> function.....	80
▶ <b>Scroll Bar Example</b> .....	81
<b>Slider</b> .....	<b>82</b>
▶ <b>egl_create_slider</b> function .....	84
▶ <b>egl_slider_callback</b> function .....	86
▶ <b>egl_slider_set_pos</b> function.....	87
▶ <b>egl_slider_get_pos</b> function.....	88
▶ <b>egl_slider_set_range</b> function.....	89
▶ <b>egl_slider_get_range</b> function .....	90
▶ <b>egl_slider_stepit</b> function .....	91
▶ <b>egl_slider_set_tick_frequency</b> function.....	92
▶ <b>egl_slider_set_tick_style</b> function.....	93
▶ <b>egl_slider_set_thumb_size</b> function.....	94
▶ <b>egl_slider_get_thumb_size</b> function .....	95
▶ <b>egl_slider_set_barcolor</b> function .....	96

▶ <b>egl_slider_set_tickcolor</b> function.....	97
▶ <b>egl_slider_set_transparent</b> function .....	98
▶ <b>Slider Example</b> .....	99
<b>List Box</b> .....	<b>100</b>
▶ <b>egl_create_listbox</b> function.....	102
▶ <b>egl_listbox_callback</b> function.....	103
▶ <b>egl_listbox_additem</b> function .....	104
▶ <b>egl_listbox_delitem</b> function.....	105
▶ <b>egl_listbox_delitem_text</b> function .....	107
▶ <b>egl_listbox_alldelitem</b> function.....	109
▶ <b>egl_listbox_get_all_itemlist</b> function .....	110
▶ <b>egl_listbox_get_sel_item</b> function.....	111
▶ <b>egl_listbox_get_multiple_sel_itemlist</b> function.....	112
▶ <b>egl_listbox_set_bgcolor</b> function .....	113
▶ <b>egl_listbox_set_selbgcolor</b> function .....	114
▶ <b>egl_listbox_set_textcolor</b> function.....	115
▶ <b>egl_listbox_set_seltextcolor</b> function.....	116
▶ <b>egl_listbox_set_scrollwidth</b> function .....	117
▶ <b>List box Example</b> .....	118
<b>Circle Gauge</b> .....	<b>121</b>
▶ <b>egl_create_circle_gauge</b> function .....	122
▶ <b>egl_circle_gauge_set_value</b> function.....	124
▶ <b>egl_circle_gauge_get_value</b> function .....	125
▶ <b>Circle Gauge Example</b> .....	126
<b>Bar Gauge</b> .....	<b>128</b>
▶ <b>egl_create_bar_gauge</b> function.....	129
▶ <b>egl_bar_gauge_set_value</b> function .....	131

▶ <b>egl_bar_gauge_get_value</b> function .....	132
▶ <b>Bar Gauge Example.</b> .....	133
<b>Animation .....</b>	<b>135</b>
▶ <b>egl_create_animation</b> function .....	136
▶ <b>Animation Example.</b> .....	138
<b>Custom Object.....</b>	<b>139</b>
▶ <b>egl_create_custom_object</b> function.....	140
▶ <b>Custom Example.</b> .....	142
<b>MessageBox.....</b>	<b>144</b>
▶ <b>egl_show_messagebox</b> function .....	145
▶ <b>MessageBox Example.</b> .....	147
<b>EGL Font.....</b>	<b>149</b>
▶ <b>egl_get_font</b> function .....	150
▶ <b>egl_set_font</b> function .....	151
▶ <b>egl_font_set_bkmode</b> function .....	152
▶ <b>egl_font_get_bk_color</b> function.....	153
▶ <b>egl_font_set_bk_color</b> function.....	154
▶ <b>egl_font_get_color</b> function .....	155
▶ <b>egl_font_set_color</b> function .....	156
▶ <b>create_bitfont</b> function.....	157
▶ <b>release_bitfont</b> function.....	158
▶ <b>create_bmpfont</b> function.....	159
▶ <b>bmfont_release</b> function.....	160
▶ <b>draw_text</b> function.....	161
▶ <b>draw_text_pivot</b> function.....	162
▶ <b>draw_text_len</b> function .....	163

▶ <b>draw_text_in_box</b> function.....	164
▶ <b>text_width</b> function .....	166
<b>EGL Primitives .....</b>	<b>167</b>
▶ <b>draw_line</b> function.....	169
▶ <b>draw_hline</b> function.....	170
▶ <b>draw_vline</b> function .....	171
▶ <b>draw_thickline</b> function.....	172
▶ <b>draw_rect</b> function .....	173
▶ <b>draw_rectfill</b> function.....	174
▶ <b>draw_rectfill_gradient</b> function .....	175
▶ <b>draw_rectfill_h_gradient</b> function.....	176
▶ <b>draw_rectfill_v_gradient</b> function.....	177
▶ <b>draw_roundrect</b> function.....	178
▶ <b>draw_roundrectfill</b> function .....	179
▶ <b>draw_arc</b> function.....	180
▶ <b>draw_pie</b> function.....	181
▶ <b>draw_piefill</b> function .....	182
▶ <b>draw_ellipse</b> function.....	183
▶ <b>draw_ellipsefill</b> function.....	184
▶ <b>draw_circle</b> function.....	185
▶ <b>draw_circlefill</b> function.....	186
▶ <b>draw_bezier</b> function .....	187
▶ <b>draw_polyline</b> function .....	189
▶ <b>draw_polygon</b> function .....	190
▶ <b>draw_polygonfill</b> function .....	191
<b>EGL etc.....</b>	<b>192</b>
▶ <b>egl_init</b> function .....	193

▶ <b>egl_show_object</b> function.....	194
▶ <b>egl_object_set_redraw</b> function.....	195

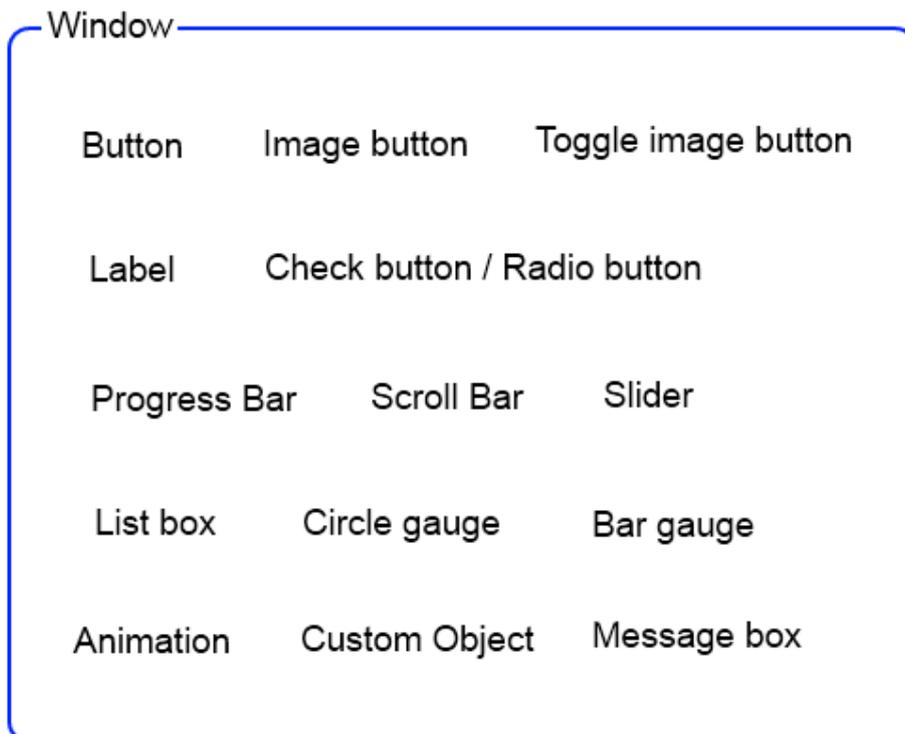
# EGL Library

## 1) 소개

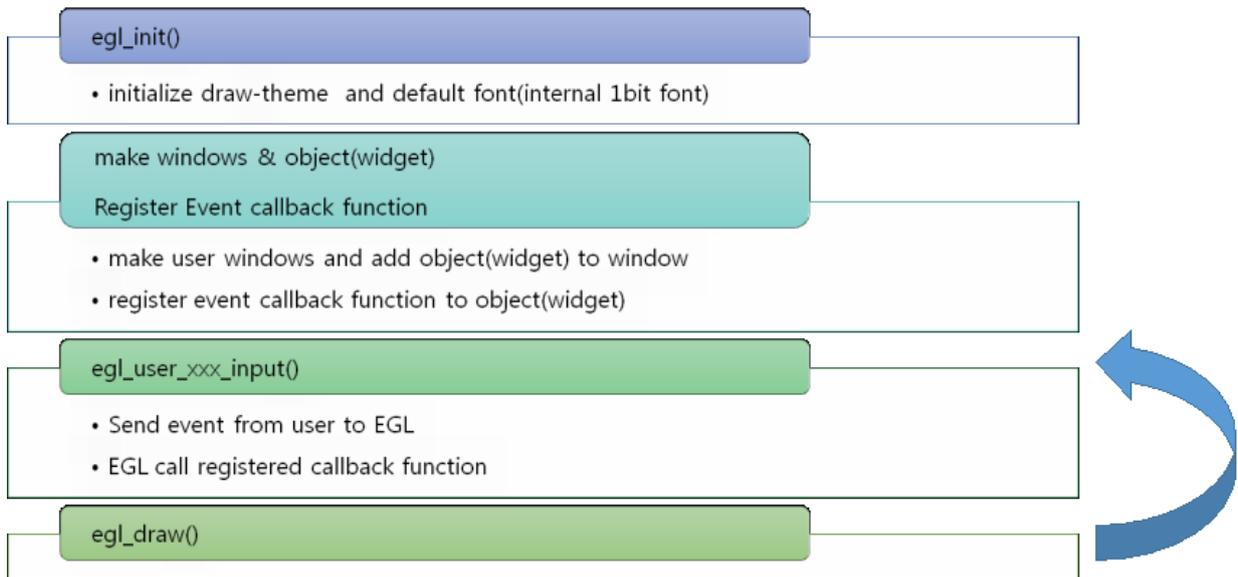
EGL(Embedded Graphic Library)은 graphic application 개발시간을 단축하고 다양한 object를 편리하게 사용할 수 있도록 제작 되었다. EGL은 다양한 object를 간편하게 사용할 수 있도록 함수를 제공하며, event에 대한 처리와 object 관리를 library내에서 하도록 되어 있다. Event에 대한 추가 동작은 각 object별 사용자 callback 함수를 두어, 개발자가 원하는 동작의 callback 함수를 작성하여 사용할 수 있다.

## 2) 구성

EGL은 다음의 object 들로 구성되어 있다.



### 3) Program 기본 구조



< Source Code >

```
#include "adstar.h"

static void btn_callback(EGL_HANDLE h, int event) // button callback function.
{
    /* 사용자 call back 함수 */
}

extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    EGL_HANDLE btn;
    egl_init();
    hWin = egl_create_window("Main Window"); // create window
    btn = egl_create_button(100,100,200,50,"Button 1"); // create button obj
    egl_btn_callback(btn, btn_callback);
    egl_window_add_object(hWin, btn); // button obj add.
    egl_window_show(hWin); // window show
```

```
while(1)
{
    if( process_touch( &touchdown, &touch_pt ) )    // touch 입력.
        egl_user_touch_input( touchdown, &touch_pt ); // touch 처리, msg 처리.

    egl_draw();
}
...
}
```

# Window

Function	Description
<a href="#"><u>egl_create_window</u></a>	Window를 생성한다.
<a href="#"><u>egl_window_show</u></a>	해당 window를 화면에 출력할지를 결정한다.
<a href="#"><u>egl_window_isshow</u></a>	해당 window가 현재 보여지고 있는지를 확인한다.
<a href="#"><u>egl_window_invalidate</u></a>	현재 window안의 object를 다시 draw한다.
<a href="#"><u>egl_window_invalidate_rect</u></a>	현재 window의 지정한 영역의 object를 다시 draw한다.
<a href="#"><u>egl_window_redraw_rect</u></a>	현재 window의 지정한 영역을 다시 draw한다.
<a href="#"><u>egl_window_set_bg</u></a>	Window의 배경 이미지를 설정한다.
<a href="#"><u>egl_window_get_bg</u></a>	Window의 배경 이미지를 가져온다.
<a href="#"><u>egl_window_add_object</u></a>	Window에 object를 추가한다.
<a href="#"><u>egl_window_set_callback</u></a>	Window 이벤트가 발생했을 때 호출될 callback함수를 등록한다.
<a href="#"><u>egl_window_get_active</u></a>	현재 보이는 window의 handle값을 반환한다.
<a href="#"><u>egl_user_touch_input</u></a>	Msg_handler로 touch message를 전달한다.
<a href="#"><u>egl_draw</u></a>	현재 windows의 object를 draw한다.

## ▶ egl\_create\_window function

```
EGL_HANDLE egl_create_window(  
    const char* title  
);
```

### Overview

Window를 생성한다. Window의 크기는 사용하는 해상도와 같다.

### Parameter

const char\* title                      Window의 title.

### Return Value

생성된 Window의 handle.

### Example

```
EGL_HANDLE hWin;  
  
hWin = egl_create_window("Main Window");
```

## ▶ egl\_window\_show function

```
BOOL egl_window_show(  
    EGL_HANDLE hWin,  
    BOOL bShow  
);
```

### Overview

해당 window를 화면에 출력할지를 결정한다.

### Parameter

EGL_HANDLE hWin	출력을 결정 할 window handle.
BOOL bShow	출력 여부 결정.

### Return Value

TRUE or FALSE

### Example

```
EGL_HANDLE hWin;  
hWin = egl_create_window("Main Window");  
...  
egl_window_show(hWin, TRUE);
```

## ▶ egl\_window\_isshow function

```
BOOL egl_window_isshow(  
    EGL_HANDLE hWin  
);
```

### Overview

해당 window가 현재 보여지고 있는지를 확인한다.

### Parameter

EGL\_HANDLE hWin                      출력을 상태를 확인 할 window handle.

### Return Value

TRUE or FALSE

### Example

```
EGL_HANDLE hWin;  
hWin = egl_create_window("Main Window");  
...  
if(egl_window_isshow(hWin))  
{  
...  
}
```

## ▶ **egl\_window\_invalidate** function

```
void egl_window_invalidate( void );
```

### Overview

현재 window안의 object를 다시 draw한다.

### Parameter

없음.

### Return Value

없음.

### Example

```
EGL_HANDLE hWin;  
hWin = egl_create_window("Main Window");  
...  
egl_window_invalidate( );
```

## ▶ egl\_window\_invalidate\_rect function

```
void egl_window_invalidate_rect(  
    EGL_RECT* pRect  
);
```

### Overview

현재 window의 지정한 영역의 object를 다시 draw한다.

### Parameter

EGL\_RECT\* pRect                      지정 영역 value.

```
typedef Struct _tag_RECT{  
    int x;  
    int y;  
    int w;  
    int h;  
}EGL_RECT
```

### Return Value

없음.

### Example

```
EGL_HANDLE hWin;  
EGL_RECT pRect;  
hWin = egl_create_window("Main Window");  
...  
pRect.x = 100;  
pRect.y = 200;  
pRect.w = 150;  
pRect.h = 150;  
egl_window_invalidate_rect(&pRect );
```

## ▶ egl\_window\_redraw\_rect function

```
Void egl_window_redraw_rect(  
    EGL_RECT* pRect  
);
```

### Overview

현재 window의 지정한 영역을 다시 draw한다.

### Parameter

EGL\_RECT\* pRect                      지정 영역 value.

```
typedef Struct _tag_RECT{  
    int x;  
    int y;  
    int w;  
    int h;  
}EGL_RECT
```

### Return Value

없음.

### Example

```
EGL_HANDLE hWin;  
EGL_RECT pRect;  
hWin = egl_create_window("Main Window");  
...  
pRect.x = 100;  
pRect.y = 200;  
pRect.w = 150;  
pRect.h = 150;  
egl_window_redraw_rect(&pRect);
```

## ▶ egl\_window\_set\_bg function

```
BOOL egl_window_set_bg(  
    EGL_HANDLE hWin;  
    SURFACE* Img;  
);
```

### Overview

Window의 배경 이미지를 설정한다.

### Parameter

EGL_HANDLE hWin	배경 이미지를 설정할 Window handle.
SURFACE* Img	배경 이미지 SURFACE

### Return Value

TRUE or FALSE

### Example

```
EGL_HANDLE hWin;  
SURFACE* surface = loadbmp("windowbg.bmp");  
hWin = egl_create_window("Main Window");  
egl_window_set_bg(hWin, surface);
```

## ▶ egl\_window\_get\_bg function

```
SURFACE* egl_window_get_bg(  
    EGL_HANDLE hWin;  
);
```

### Overview

Window의 배경 이미지를 가져온다.

### Parameter

EGL\_HANDLE hWin                      배경 이미지를 가져 올 Window handle.

### Return Value

해당 window의 image.

### Example

```
EGL_HANDLE hWin;  
SURFACE* getimage;  
hWin = egl_create_window("Main Window");  
...  
getimage = egl_window_get_bg(hWin);
```

## ▶ egl\_window\_add\_object function

```
BOOL egl_window_add_object(  
    EGL_HANDLE hWin,  
    EGL_HANDLE hObj  
);
```

### Overview

Window에 object를 추가한다.

### Parameter

EGL_HANDLE hWin	Object를 추가 할 window handle.
EGL_HANDLE hObj	Window에 추가 할 object handle.

### Return Value

TRUE or FALSE

### Example

```
EGL_HANDLE hWin;  
EGL_HANDLE btn;  
hWin = egl_create_window("Main Window");  
btn = egl_create_button(100, 100, 100, 50, "Button");  
egl_window_add_object(hWin, btn);
```

## ▶ egl\_window\_set\_callback function

```
void egl_window_set_callback(  
    EGL_HANDLE hWin,  
    EVENT_CALLBACK cb  
);
```

### Overview

Window 이벤트가 발생했을 때 호출 되는 callback 함수를 등록한다.

### Parameter

EGL_HANDLE hWin	Callback 함수가 등록되는 Window handle.
EVENT_CALLBACK cb	Event 발생 시 호출되는 callback 함수.

### Return Value

없음.

### Example

```
Void window_callback(EGL_HANDLE h, int event)  
{  
    ...  
}  
int main()  
{  
    ...  
    EGL_HANDLE hWin;  
    hWin = egl_create_window("Main Window");  
    egl_window_callback(hWin, window_callback);  
    ...  
}
```

## ▶ egl\_window\_get\_active function

```
EGL_HANDLE egl_window_get_active( void );
```

### Overview

현재 출력되고 있는 window의 handle값을 반환한다.

### Parameter

없음.

### Return Value

현재 출력되고 있는 window handle.

### Example

```
EGL_HANDLE hWin;  
EGL_HANDLE active_win;  
hWin = egl_create_window("Main Window");  
...  
active_win = egl_window_get_active( );
```

## ▶ egl\_user\_touch\_input function

```
void egl_user_touch_input(  
    BOOL bPressed,  
    EGL_POINT* pt  
);
```

### Overview

Msg\_handler로 touch message를 전달한다.

### Parameter

BOOL bPressed	Touch가 눌린 상태.
EGL_POINT* pt	Touch 된 좌표.

```
typedef struct _tagPOINT  
{  
    int x;  
    int y;  
} EGL_POINT;
```

### Return Value

없음.

### Example

```
EGL_POINT touch_pt;  
BOOL touchdown = FALSE;  
...  
if(process_touch(&touchdown, &touch_pt))  
    egl_user_touch_input(touchdown, &touch_pt);
```

## ▶ egl\_draw function

```
void egl_draw( void );
```

### Overview

현재 windows의 object를 draw한다.

### Parameter

없음.

### Return Value

없음.

### Example

```
if(process_touch(&touchdown, &touch_pt))
    egl_user_touch_input(touchdown, &touch_pt);
egl_draw();
```

## ► Window Example.

### Example

```
#include "adstar.h"

static void btn1_callback(EGL_HANDLE h, int event)
{
    if( event == BTN_CLICKED )
    {
        debugprintf("btn1 clicked\n");
        egl_window_show( hWin2, TRUE );
    }
}

static void btn2_callback(EGL_HANDLE h, int event)
{
    if( event == BTN_CLICKED )
    {
        debugprintf("btn2 clicked\n");
        egl_window_show( hWin1, TRUE );
    }
}

extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin1;
    EGL_HANDLE hWin2;
    EGL_HANDLE btn1;
    EGL_HANDLE btn2;
    SURFACE* bg = loadbmp("bg.bmp");
    egl_init();
    hWin1 = egl_create_window("First Window");
    egl_windows_set_bg(hWin1, bg);
    hWin2 = egl_create_window("Second Window");
    btn1 = egl_create_button(100, 100, 200, 50, "Second Window");
```

```
egl_btn_callback(btn1, btn1_callback);
btn2 = egl_create_button(100, 200, 200, 50, "First Window");
egl_btn_callback(btn2, btn2_callback);
egl_window_add_object(hWin1, btn1);
egl_window_add_object(hWin2, btn2);
egl_window_show(hWin1);
egl_draw();

while(1)
{
    if( process_touch(&touchdown, &touch_pt ) )
        egl_user_touch_input(touchdown, &touch_pt );

    egl_draw();
}
}
```

# Button



Function	Description
----------	-------------

[egl\\_create\\_button](#)

Button을 생성한다.

[egl\\_button\\_callback](#)

Button 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.

Define
--------

BUTTON\_EVENT

```
typedef enum
{
    BTN_CLICKED = 0,
    BTN_PRESSED,
    BTN_MAX,
}BUTTON_EVENT
```

## ▶ egl\_create\_button function

```
EGL_HANDLE egl_create_button(  
    int x,  
    int y,  
    int w,  
    int h,  
    const char* text  
);
```

### Overview

Button을 생성한다.

### Parameter

int x	생성될 Button의 x좌표.
int y	생성될 Button의 y좌표.
int w	생성될 Button의 가로 크기.
int h	생성될 Button의 세로 크기.
const char* text	Button에 출력될 text.

### Return Value

생성된 Button의 handle.

### Example

```
EGL_HANDLE btn[2];  
  
btn[0] = egl_create_button(100, 100, 100, 50, "Button1");  
  
btn[1] = egl_create_button(300, 100, 100, 50, "Button2");
```

## ▶ egl\_button\_callback function

```
BOOL egl_button_callback(  
    EGL_HANDLE hObj,  
    EVENT_CALLBACK cb  
);
```

### Overview

Button 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.

### Parameter

EGL_HANDLE hObj	Callback 함수가 등록되는 Button handle.
EVENT_CALLBACK cb	Event가 발생 시 호출되는 callback 함수.

### Return Value

TRUE or FALSE

### Example

```
void btn_callback(EGL_HANDLE h, int event)  
{  
    If(event == BTN_CLICKED)  
    {  
        ...  
    }  
}  
int main()  
{  
    ...  
    EGL_HANDLE btn;  
    btn = egl_create_button(100, 100, 100, 50, "Button1");  
    egl_button_callback(btn,btn_callback);  
    ...  
}
```

## ► Button Example.

### Example

```
void btn_callback(EGL_HANDLE h, int event)
{
    if(event == BTN_CLICKED)
    {
        debugprintf("buttton clicked.");
    }
}

extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    EGL_HANDLE btn;
    egl_init();
    btn = egl_create_button(100, 100, 100, 50, "Button Ex");
    egl_button_callback(btn, btn_callback);
    egl_window_add_object(hWin, btn);
    egl_window_show(hWin);
    egl_draw();

    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        egl_draw();
    }
}
```

# Image button



< img >



< pressed\_img >

Function	Description
<a href="#">egl_create_image_button</a>	Image button을 생성한다.
<a href="#">egl_image_button_callback</a>	Image button 이벤트가 발생했을 때 호출될 callback 함수를 등록한다

## ▶ egl\_create\_image\_button function

```
EGL_HANDLE egl_create_image_button(  
    SURFACE* img,  
    SURFACE* pressed_img,  
    int x,  
    int y,  
    int w,  
    int h  
);
```

### Overview

Image button을 생성한다.

### Parameter

SURFACE* img	Image button의 표시 될 image.
SURFACE* pressed_img	눌렸을 때 표시 될 image.
int x	Image button의 x좌표.
int y	Image button의 y좌표.
int w	Image button의 가로 크기.
int h	Image button의 세로 크기.

### Return Value

생성 된 Image button handle.

### Example

```
EGL_HANDLE imagebtn;  
SURFACE* img = loadbmp("btnimg.bmp");  
SURFACE* pressed_img = loadbmp("pressedimg.bmp");  
imagebtn = egl_create_image_button(img, pressed_img, 100, 100, 150, 50);
```

## ▶ egl\_image\_button\_callback function

```
BOOL egl_image_button_callback(  
    EGL_HANDLE hObj,  
    EVENT_CALLBACK cb  
);
```

### Overview

Image button 이벤트가 발생했을 때 호출될 callback 함수를 등록한다

### Parameter

EGL_HANDLE hObj	Callback 함수가 등록되는 Image button handle.
EVENT_CALLBACK cb	Event가 발생 시 호출되는 callback 함수.

### Return Value

TRUE or FALSE

### Example

```
void imgbtn_callback(EGL_HANDLE h, int event)  
{  
    ...  
}  
int main()  
{  
    ...  
    EGL_HANDLE imagebtn;  
    SURFACE* img = loadbmp("btnimg.bmp");  
    SURFACE* pressed_img = loadbmp("pressedimg.bmp");  
    imagebtn = egl_create_image_button(img, pressed_img, 100, 100, 150, 50);  
    egl_image_button_callback(imagebtn, imgbtn_callback);  
    ...  
}
```

## ► Image Button Example.

### Example

```
void imgbtn_callback(EGLError h, int event)
{
    if(event == BTN_CLICKED)
    {
        debugprintf("image button clicked.");
    }
}

extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL  touchdown = FALSE;
    EGL_HANDLE hWin;
    EGL_HANDLE imgbtn;
    SURFACE* img = loadbmp("btnimg.bmp");
    SURFACE* pressed_img = loadbmp("pressedimg.bmp");
    egl_init();
    imgbtn = egl_create_image_button(img, pressed_img, 100, 100, 150, 50);
    egl_button_callback(imgbtn, imgbtn_callback);
    egl_window_add_object(hWin, imgbtn);
    egl_window_show(hWin);
    egl_draw();

    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        egl_draw();
    }
}
```

# Toggle image button



< surf\_on >



< surf\_off >

Function	Description
<a href="#">egl_create_toggle_image</a>	Toggle image button을 생성한다.
<a href="#">egl_toggle_image_callback</a>	Toggle image button 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.
<a href="#">egl_toggle_image_set_on</a>	Toggle image button의 상태를 설정한다.

Define	
TOGGLE_IMAGE_EVENT	<pre>typedef enum {     TOGGLE_IMAGE_ON = 0,     TOGGLE_IMAGE_OFF, } TOGGLE_IMAGE_EVENT;</pre>

## ▶ egl\_create\_toggle\_image function

```
EGL_HANDLE egl_create_toggle_image(  
    SURFACE* surf_off,  
    SURFACE* surf_on,  
    int x,  
    int y,  
    int w,  
    int h  
);
```

### Overview

Toggle image button을 생성한다.

### Parameter

SURFACE* surf_off	Toggle image 중 off시에 출력 할 image. ( default상태 : Off )
SURFACE* surf_on	Toggle image 중 on시에 출력 할 image.
int x	Toggle image의 x 좌표.
int y	Toggle image의 y 좌표.
int w	Toggle image의 가로 크기.
int h	Toggle image의 세로 크기.

### Return Value

생성 된 toggle image handle.

### Example

```
EGL_HANDLE toggleimg;  
SURFACE* surf_off = loadpng("off.png");  
SURFACE* surf_on = loadpng("on.png");  
toggleimg = egl_create_toggle_image(surf_off, surf_on, 100, 100, 150, 50);
```

## ▶ egl\_toggle\_image\_callback function

```
BOOL egl_toggle_image_callback(  
    EGL_HANDLE hObj,  
    EVENT_CALLBACK cb  
);
```

### Overview

Toggle image button 이벤트가 발생했을 때 호출될 callback 함수를 등록한다

### Parameter

EGL_HANDLE hObj	Callback 함수가 등록되는 toggle image handle.
EVENT_CALLBACK cb	Event가 발생 시 호출되는 callback 함수.

### Return Value

TRUE or FALSE

### Example

```
void toggleimg_callback(EGL_HANDLE h, int event)  
{  
    if(event == TOGGLE_IMAGE_ON)  
        ...  
    else  
        ...  
}  
int main()  
{  
    EGL_HANDLE toggleimage;  
    ...  
    toggleimage = egl_create_toggle_image(surf_off, surf_on, 100, 100, 150, 50);  
    egl_toggle_image_callback(toggleimage, toggleimg_callback);  
    ...  
}
```

## ▶ egl\_toggle\_image\_set\_on function

```
BOOL egl_toggle_image_set_on(  
    EGL_HANDLE hObj,  
    BOOL b  
);
```

### Overview

Toggle image button의 상태를 설정한다.

### Parameter

EGL_HANDLE hObj	상태를 설정 할 toggle image handle.
BOOL b	설정 값. TRUE => On FALSE => Off

### Return Value

TRUE or FALSE

### Example

```
EGL_HANDLE toggleimg;  
SURFACE* surf_off = loadpng("off.png");  
SURFACE* surf_on = loadpng("on.png");  
toggleimg = egl_create_toggle_image(surf_off, surf_on, 100, 100, 150, 50);  
egl_toggle_image_set_on(toggleimg, TRUE);
```

## ► Toggle Image Button Example.

### Example

```
void toggleimg_callback(EGL_HANDLE h, int event)
{
    if(event == TOGGLE_IMAGE_ON)
        debugprintf(" toggle image on %r%rn");
    else
        debugprintf(" toggle image off %r%rn");
}

extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL  touchdown = FALSE;
    EGL_HANDLE  hWin;
    EGL_HANDLE  toggleimg;
    SURFACE* surf_off = loadpng("off.png");;
    SURFACE* surf_on = loadpng("on.png");
    egl_init();
    toggleimg = egl_create_toggle_image(surf_off, surf_on, 100, 100, 150, 50);
    egl_toggle_image_callback(toggleimg, toggleimg_callback);
    egl_toggle_image_set_on(toggleimg, TRUE);
    egl_window_add_object(hWin, toggleimg);
    egl_window_show(hWin);
    egl_draw();

    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        egl_draw();
    }
}
```

# Label

Function	Description
<a href="#"><u>egl_create_label</u></a>	Label object를 생성한다.
<a href="#"><u>egl_label_set_text</u></a>	Label text를 설정한다.
<a href="#"><u>egl_label_set_redraw_bg</u></a>	Label 배경을 draw할지를 결정한다.
<a href="#"><u>egl_label_set_color</u></a>	Label text color를 설정한다.

## ▶ egl\_create\_label function

```
EGL_HANDLE egl_create_label(  
    int x,  
    int y,  
    int w,  
    int h,  
    const char* text  
);
```

### Overview

Label object를 생성한다.

### Parameter

int x	label의 x 좌표.
int y	label의 y 좌표.
int w	label의 가로 크기.
int h	label의 세로 크기.
const char* text	label의 text.

### Return Value

생성 된 label handle.

### Example

```
EGL_HANDLE label;  
label = egl_create_label( 100, 100, 150, 50, "label test");
```

## ▶ egl\_label\_set\_text function

```
BOOL egl_label_set_text(  
    EGL_HANDLE h,  
    char* text  
);
```

### Overview

Label의 text를 변경한다.

### Parameter

EGL_HANDLE h	Text를 변경 할 label handle.
char text	변경 할 text.

### Return Value

TRUE or FALSE

### Example

```
EGL_HANDLE label;  
label = egl_create_label( 100, 100, 150, 50, "label test" );  
egl_label_set_text( label, "change text");
```

## ▶ egl\_label\_set\_redraw\_bg function

```
void egl_label_set_redraw_bg(  
    EGL_HANDLE h,  
    BOOL b  
);
```

### Overview

Label 배경을 draw할지를 결정한다. label을 단독 object로 사용할 경우 TRUE로 설정해야 한다.

### Parameter

EGL_HANDLE hObj	상태를 설정 할 label handle.
BOOL b	설정 값. TRUE or FALSE

### Return Value

없음.

### Example

```
EGL_HANDLE label;  
label = egl_create_label( 100, 100, 150, 50, "label test" );  
egl_label_set_redraw_bg(label, TRUE);
```

## ▶ egl\_label\_set\_color function

```
void egl_label_set_color(  
    EGL_HANDLE h,  
    EGL_COLOR clr  
);
```

### Overview

Label text color를 설정한다.

### Parameter

EGL_HANDLE h	Color를 변경 할 label handle.
EGL_COLOR clr	Color value. MAKE_COLORREF(r, g, b) 매크로 함수를 사용하면 color 지정을 쉽게 할 수 있다.

### Return Value

없음.

### Example

```
EGL_HANDLE label;  
label = egl_create_label( 100, 100, 150, 50, "label test" );  
egl_label_set_color( label, MAKE_COLORREF( 0, 255, 0 ) );
```

## ► Label Example.

### Example

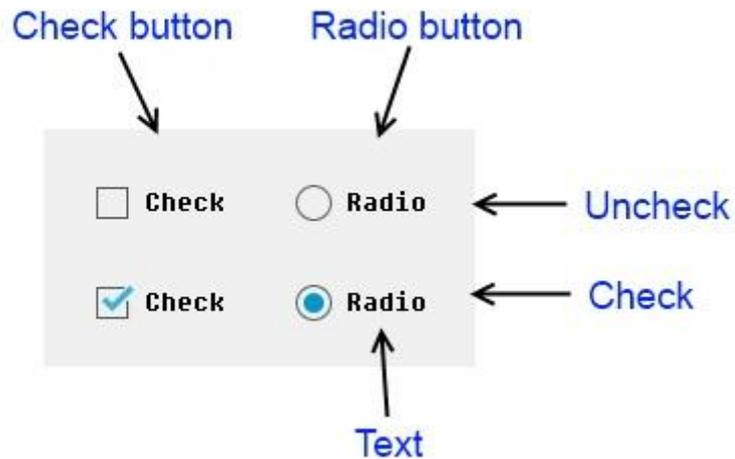
```
extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    EGL_HANDLE label;
    egl_init();
    label = egl_create_label(50, 50, 100, 30, "label");
    egl_label_set_redraw_bg(label, TRUE);
    egl_label_set_color(label, MAKE_COLORREF(0, 0, 255));
    egl_window_add_object(hWin, label);
    egl_window_show(hWin);
    egl_draw();

    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        egl_draw();
    }
}
```

# Check button / Radio button



Function	Description
<a href="#">egl_create_checkbutton</a>	Check / Radio button을 생성한다.
<a href="#">egl_checkbutton_callback</a>	Check / Radio button 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.
<a href="#">egl_checkbutton_set_check</a>	Check / Radio button을 check / uncheck 상태를 설정한다.
<a href="#">egl_checkbutton_get_check</a>	Check / Radio button의 현재 check 상태 값을 가져온다.
<a href="#">egl_checkbutton_set_style</a>	Check / Radio button의 style을 변경한다.

Define	
CHECK_EVENT	<pre>typedef enum {     CHECK_CHECKED = 0,     CHECK_UNCHECKED,, } CHECK_EVENT;</pre>

## ▶ egl\_create\_checkbutton function

```
EGL_HANDLE egl_create_checkbutton(  
    int x,  
    int y,  
    int w,  
    int h,  
    const char* text,  
    CHECK_STYLE style  
);
```

### Overview

Check / Radio button을 생성한다.

### Parameter

int x	생성될 Check / Radio button의 x좌표.
int y	생성될 Check / Radio button의 y좌표.
int w	생성될 Check / Radio button의 가로 크기.
int h	생성될 Check / Radio button의 세로 크기.
const char* text	Check / Radio button에 출력될 text.
CHECK_STLE style	Check / Radio button의 style. Check button ->CHECK_STYLE_CHECKBUTTON. Radio button ->CHECK_STYLE_RADIOBUTTON.

### Return Value

생성된 Check / Radio button의 handle.

### Example

```
EGL_HANDLE check;  
EGL_HANDLE radio;  
check = egl_create_checkbutton(100,100, 100, 32, "Check", CHECK_STYLE_CHECKBUTTON);  
radio = egl_create_checkbutton(250, 100, 100, 32, "Radio", CHECK_STYLE_RADIOBUTTON);
```

## ▶ egl\_checkbutton\_callback function

```
BOOL egl_button_callback(  
    EGL_HANDLE hObj,  
    EVENT_CALLBACK cb  
);
```

### Overview

Check / Radio button 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.

### Parameter

EGL_HANDLE hObj	Callback 함수가 등록되는 Check / Radio button handle.
EVENT_CALLBACK cb	Event 발생 시 호출되는 callback 함수.

### Return Value

TRUE or FALSE

### Example

```
Void check_callback(EGL_HANDLE h, int event)  
{  
    if(event == CHECK_CHECKED)  
    ...  
        else if(event == CHECK_UNCHECKED)  
    ...  
}  
int main()  
{  
    ...  
    EGL_HANDLE check;  
    check = egl_create_checkbutton(100, 100, 100, 32, "Check", STYLE_CHECKBUTTON);  
    egl_checkbutton_callback(check, check_callback);  
    ...  
}
```

## ▶ egl\_checkbutton\_set\_check function

```
void egl_checkbutton_set_check(  
    EGL_HANDLE hObj,  
    BOOL b  
);
```

### Overview

Check / Radio button을 check / uncheck 상태를 설정한다.

### Parameter

EGL_HANDLE hObj	Check 상태를 설정할 Check / Radio button handle.
BOOL b	TRUE = Check. FALSE = Uncheck.

### Return Value

없음.

### Example

```
EGL_HANDLE check;  
  
EGL_HANDLE radio;  
  
check = egl_create_checkbutton(100, 100, 100, 32, "Check", STYLE_CHECKBUTTON);  
  
radio = egl_create_checkbutton(200, 100, 100, 32, "Radio", STYLE_RADIOBUTTON);  
  
egl_checkbutton_set_check(check, TRUE);  
  
egl_checkbutton_set_check(radio, FASLE);
```

## ▶ egl\_checkbutton\_get\_check function

```
BOOL egl_checkbutton_get_check(  
    EGL_HANDLE hObj,  
);
```

### Overview

Check / Radio button의 현재 check 상태 값을 가져온다.

### Parameter

EGL\_HANDLE hObj                      Check 상태를 가지고 올 Check / Radio button handle.

### Return Value

Check 상태를 반환. TRUE = check.  
                  FALSE = uncheck.

### Example

```
EGL_HANDLE check;  
  
EGL_HANDLE radio;  
  
BOOL check_status = 0;  
  
BOOL radio_status = 0;  
  
check = egl_create_checkbutton(100, 100, 100, 32, "Check", STYLE_CHECKBUTTON);  
  
radio = egl_create_checkbutton(200, 100, 100, 32, "Radio", STYLE_RADIOBUTTON);  
  
check_status = egl_checkbutton_get_check(check);  
  
radio_status = egl_checkbutton_get_check(radio);
```

## ▶ egl\_checkbutton\_set\_style function

```
void egl_checkbutton_set_style(  
    EGL_HANDLE hObj,  
    CHECK_STYLE style  
);
```

### Overview

Check / Radio button의 style을 변경한다.

### Parameter

EGL_HANDLE hObj	Style을 변경할 Check / Radio button handle.
CHECK_STYLE style	Check / Radio button의 style. Check button ->CHECK_STYLE_CHECKBUTTON. Radio button ->CHECK_STYLE_RADIOBUTTON.

### Return Value

없음

### Example

```
EGL_HANDLE check;  
  
EGL_HANDLE radio;  
  
check = egl_create_checkbutton(100, 100, 100, 32, "Check", CHECK_STYLE_CHECKBUTTON);  
  
radio = egl_create_checkbutton(200, 100, 100, 32, "Radio", CHECK_STYLE_RADIOBUTTON);  
  
egl_checkbutton_set_style(check, CHECK_STYLE_RADIOBUTTON);  
  
egl_checkbutton_set_style(radio, CHECK_STYLE_CHECKBUTTON);
```

## ► Check / Radio button Example.

### Example

```
EGL_HANDLE check;
EGL_HANDLE radio1;
EGL_HANDLE radio2;

void check_callback(EGL_HANDLE h, int event)
{
    if(event == CHECK_CHECKED)
    {
        if(egl_checkbutton_get_check())
        {
            debugprintf("status : checked\r\n");
        }
        else
        {
            debugprintf("status : unchecked\r\n");
        }
    }
}

void radio1_callback(EGL_HANDLE h, int event)
{
    if(event == CHECK_CHECKED)
    {
        debugprintf("radio1 select\r\n");
        egl_checkbutton_set_check(radio2, FALSE);
    }
}

void radio2_callback(EGL_HANDLE h, int event)
{
    if(event == CHECK_CHECKED)
    {
        debugprintf("radio2 select\r\n");
        egl_checkbutton_set_check(radio1, FALSE);
    }
}
```

```

}

extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;

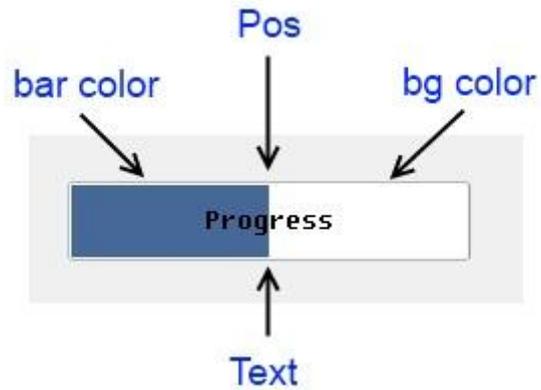
    egl_init();
    check = egl_create_checkbutton(100, 100, 100, 32, "Check", CHECK_STYLE_CHECKBUTTON);
    egl_checkbutton_callback(check, check_callback);
    radio1 = egl_create_checkbutton(250, 100, 100, 32, "Radio1", CHECK_STYLE_RADIOBUTTON);
    egl_checkbutton_callback(radio1, radio1_callback);
    radio2 = egl_create_checkbutton(250, 160, 100, 32, "Radio2", CHECK_STYLE_RADIOBUTTON);
    egl_checkbutton_callback(radio2, radio2_callback);
    egl_window_add_object(hWin, check);
    egl_window_add_object(hWin, radio1);
    egl_window_add_object(hWin, radio2);
    egl_window_show(hWin);
    egl_draw();

    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        egl_draw();
    }
}

```

# Progress Bar



Function	Description
<a href="#">egl_create_progressbar</a>	Progress bar를 생성한다.
<a href="#">egl_progressbar_set_barcolor</a>	Progress bar의 bar color를 설정한다.
<a href="#">egl_progressbar_set_bgcolor</a>	Progress bar의 back ground color를 설정한다.
<a href="#">egl_progressbar_set_textcolor</a>	Progress bar의 text color를 설정한다.
<a href="#">egl_progressbar_set_text</a>	Progress bar의 text를 설정한다.
<a href="#">egl_progressbar_set_pos</a>	Progress bar의 bar position을 설정한다.
<a href="#">egl_progressbar_get_pos</a>	Progress bar의 현재 bar position value를 가져온다.

## ▶ egl\_create\_progressbar function

```
 EGL_HANDLE egl_create_progressbar(  
     int x,  
     int y,  
     int w,  
     int h,  
     const char* text,  
     BOOL style  
     BOOL bVertical  
 );
```

### Overview

Progress bar를 생성한다.

### Parameter

int x	생성될 Progress bar의 x좌표.
int y	생성될 Progress bar의 y좌표
int w	생성될 Progress bar의 가로 크기
int h	생성될 Progress bar의 세로 크기
const char* text	Progress bar에 출력 될 text.
BOOL style	Progress bar의 style. Normal = STYLE_PGBAR_NOMAL. Divide bar = STYLE_PGVAR_DIV.
BOOL bVertical	Progress의 출력 형태. TRUE = Vertical. FALSE = Horizontal.

### Return Value

생성 된 Progress bar의 handle.

## Example

```
EGL_HANDLE pgbar;
```

```
pgbar = egl_create_progressbar(100, 100, 200, 40, "Progress", STYLE_PGBAR_NOMAL, FALSE);
```

## ▶ egl\_progressbar\_set\_barcolor function

```
void egl_progressbar_set_barcolor(  
    EGL_HANDLE hObj,  
    unsigned char r,  
    unsigned char g,  
    unsigned char b  
);
```

### Overview

Progress bar의 bar color를 설정한다.

### Parameter

EGL_HANDLE hObj	Bar color를 설정할 Progress bar handle.
unsigned char r	RGB 중 Red 값.
unsigned char g	RGB 중 Green 값.
unsigned char b	RGB 중 Blue 값.

### Return Value

없음.

### Example

```
EGL_HANDLE pgbars;  
  
pgbars = egl_create_progressbar(100,100,200,40,"Progress", STYLE_PGBAR_NOMAL, FALSE);  
  
egl_progressbar_set_barcolor(pgbars, 0, 0, 0xff);
```

## ▶ egl\_progressbar\_set\_bgcolor function

```
void egl_progressbar_set_bgcolor(  
    EGL_HANDLE hObj,  
    unsigned char r,  
    unsigned char g,  
    unsigned char b  
);
```

### Overview

Progress bar의 back ground color를 설정한다.

### Parameter

EGL_HANDLE hObj	Back ground color를 설정 할 Progress bar handle.
unsigned char r	RGB 중 Red 값.
unsigned char g	RGB 중 Green 값.
unsigned char b	RGB 중 Blue 값.

### Return Value

없음.

### Example

```
EGL_HANDLE pgbar;  
  
pgbar = egl_create_progressbar(100,100,200,40,"Progress", STYLE_PG BAR_NOMAL, FALSE);  
  
egl_progressbar_set_bgcolor(pgbar, 0, 0xff, 0);
```

## ▶ egl\_progressbar\_set\_textcolor function

```
void egl_progressbar_set_textcolor(  
    EGL_HANDLE hObj,  
    unsigned char r,  
    unsigned char g,  
    unsigned char b  
);
```

### Overview

Progress bar의 text color를 설정한다.

### Parameter

EGL_HANDLE hObj	Text color를 설정 할 Progress bar handle.
unsigned char r	RGB 중 Red 값.
unsigned char g	RGB 중 Green 값.
unsigned char b	RGB 중 Blue 값.

### Return Value

없음.

### Example

```
EGL_HANDLE pgbars;  
  
pgbars = egl_create_progressbar(100,100,200,40,"Progress", STYLE_PGBAR_NOMAL, FALSE);  
  
egl_progressbar_set_textcolor(pgbars, 0xff, 0, 0xff);
```

## ▶ egl\_progressbar\_set\_text function

```
void egl_progressbar_set_text(  
    EGL_HANDLE hObj,  
    const char* text  
);
```

### Overview

Progress bar의 text를 설정한다.

### Parameter

EGL_HANDLE hObj	Text를 설정할 Progress bar handle.
const char* text	설정 할 Text.

### Return Value

없음.

### Example

```
EGL_HANDLE pgbar;  
  
pgbar = egl_create_progressbar(100,100,200,40,"Progress", STYLE_PG_BAR_NORMAL, FALSE);  
  
egl_progressbar_set_text(pgbar, "text1");
```

## ▶ egl\_progressbar\_set\_pos function

```
void egl_progressbar_set_pos(  
    EGL_HANDLE hObj,  
    int value  
);
```

### Overview

Progress bar의 bar position을 설정한다.

### Parameter

EGL_HANDLE hObj	Bar position을 변경 할 Progress bar handle.
int value	Bar position 값 ( 0 ~ 100% )

### Return Value

없음.

### Example

```
EGL_HANDLE pgbar;  
  
pgbar = egl_create_progressbar(100,100,200,40,"Progress", STYLE_PG BAR_NOMAL, FALSE);  
  
egl_progressbar_set_pos(pgbar, 50);
```

## ▶ egl\_progressbar\_get\_pos function

```
int egl_progressbar_get_pos(  
    EGL_HANDLE hObj  
);
```

### Overview

Progress bar의 현재 bar position value를 가져온다.

### Parameter

EGL\_HANDLE hObj                      Bar position value를 가져 올 Progress bar handle.

### Return Value

Bar position value. ( 0 ~ 100%)

### Example

```
EGL_HANDLE pgbars;  
  
int pgbars_pos;  
  
pgbars = egl_create_progressbar(100,100,200,40,"Progress", STYLE_PGBAR_NOMAL, FALSE);  
  
pgbars_pos = egl_progressbar_get_pos(pgbars);
```

## ► Progress Bar Example.

### Example

```
EGL_HANDLE pgbars;
int value = 0;

void pgbars_callback(EGL_HANDLE h, int event)
{
    value++;
    if(value >100)
        value = 0;
    egl_progressbar_set_pos(pgbars, value);
}

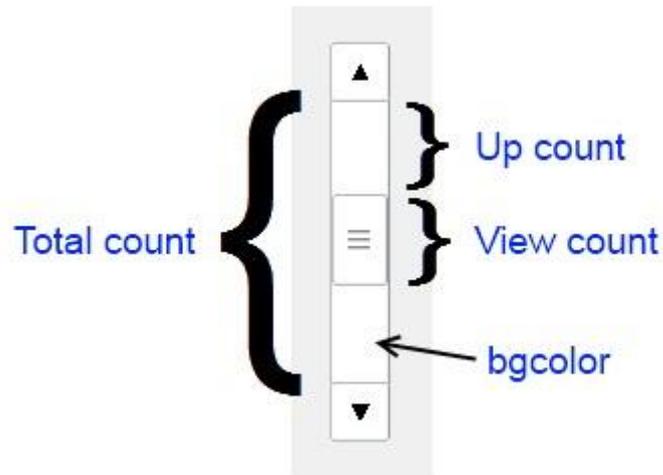
extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    egl_init();
    pgbars = egl_create_progressbar(100, 100, 200, 40, "progress", STYLE_PGBAR_NOMAL,FALSE);
    egl_progress_callback(pgbars, pgbars_callback);
    egl_window_add_object(hWin, pgbars);
    egl_window_show(hWin);
    egl_draw();

    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        egl_draw();
    }
}
```

# Scroll Bar



Function	Description
<a href="#">egl_create_scrollbar</a>	Scroll bar를 생성한다.
<a href="#">egl_scrollbar_callback</a>	Scroll bar 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.
<a href="#">egl_scroll_set_position</a>	Scroll bar의 thumb position을 설정한다.
<a href="#">egl_scroll_get_position</a>	Scroll bar의 thumb position을 가져온다.
<a href="#">egl_scroll_set_totalcount</a>	Scroll bar의 total count를 설정한다.
<a href="#">egl_scroll_get_totalcount</a>	Scroll bar의 total count를 가져온다.
<a href="#">egl_scroll_set_view_count</a>	Scroll bar의 view count를 설정한다.
<a href="#">egl_scroll_get_view_count</a>	Scroll bar의 view count를 가져온다.
<a href="#">egl_scroll_set_upcount</a>	Scroll bar의 up count를 설정한다.
<a href="#">egl_scroll_get_upcount</a>	Scroll bar의 up count를 가져온다.
<a href="#">egl_scroll_set_bgcolor</a>	Scroll bar의 back ground color를 설정한다.
<a href="#">egl_scroll_set_size</a>	Scrollbar의 크기를 설정한다.

## Define

```
SCROLLBAR_EVENT          typedef enum
                          {
                            SCBAR_CLICKED = 0,
                          } SCROLLBAR_EVENT;
```

## ▶ egl\_create\_scrollbar function

```
EGL_HANDLE egl_create_scrollbar(  
    int x,  
    int y,  
    int w,  
    int h,  
    int totalcount,  
    int viewcount,  
    BOOL bVertical  
);
```

### Overview

Scroll bar를 생성한다.

### Parameter

int x	생성 될 Scroll bar의 x좌표.
int y	생성 될 Scroll bar의 y좌표.
int w	생성 될 Scroll bar의 가로 크기.
int h	생성 될 Scroll bar의 세로 크기.
int totalcount	Scroll bar가 관리할 항목의 전체 크기.
int viewcount	Scroll bar의 thumb가 표시할 항목의 크기
BOOL bVertical	Scroll bar의 출력 형태. TRUE = Vertical. FALSE = Horizontal.

### Return Value

생성 된 Scroll bar handle.

## Example

```
EGL_HANDLE scroll;
```

```
scroll = egl_create_scrollbar(100, 100, 30, 200, 30, 10, TRUE);
```

## ▶ egl\_scrollbar\_callback function

```
BOOL egl_scrollbar_callback(  
    EGL_HANDLE hObj,  
    EVENT_CALLBACK cb  
);
```

### Overview

Scroll bar 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.

### Parameter

EGL_HANDLE hObj	Callback 함수를 등록 할 Scroll bar handle.
EVENT_CALLBACK cb	호출 될 callback 함수.

### Return Value

TRUE or FALSE

### Example

```
void scroll_callback(EGL_HANDLE h, int event)  
{  
  
}  
int main()  
{  
    ...  
    EGL_HANDLE scroll;  
    scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
    egl_scroll_callback(scroll, scroll_callback);  
    ...  
}
```

## ▶ egl\_scroll\_set\_position function

```
void egl_scroll_set_position(  
    EGL_HANDLE hObj,  
    int totalcount,  
    int viewcount,  
    int upcount  
);
```

### Overview

Scroll bar의 thumb position을 설정한다.

### Parameter

EGL_HANDLE hObj	Thumb position을 설정 할 Scroll bar handle.
int totalcount	Scroll bar가 관리 할 항목의 전체 크기.
int viewcount	Scroll bar의 thumb가 표시할 항목의 크기.
int upcount	Scroll bar의 thumb 이전 항목의 크기.

### Return Value

없음.

### Example

```
EGL_HANDLE scroll;  
  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
  
egl_scroll_set_position(scroll, 25, 5, 10);
```

## ▶ egl\_scroll\_get\_position function

```
void egl_scroll_get_position(  
    EGL_HANDLE hObj,  
    int* totalcount,  
    int* viewcount,  
    int* upcount  
);
```

### Overview

Scroll bar의 thumb position을 가져온다.

### Parameter

EGL_HANDLE hObj	Thumb position을 가져 올Scroll bar handle.
int* totalcount	Total count를 저장 할 포인터.
int* viewcount	View count를 저장 할 포인터.
int* upcount	Up count를 저장 할 포인터.

### Return Value

없음.

### Example

```
EGL_HANDLE scroll;  
int totalcount = 0;  
int viewcount = 0;  
int upcount = 0;  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
egl_scroll_get_position(scroll, &totalcount, &viewcount, &upcount);
```

## ▶ egl\_scroll\_set\_totalcount function

```
void egl_scroll_set_totalcount(  
    EGL_HANDLE hObj,  
    int totalcount  
);
```

### Overview

Scroll bar의 total count를 설정한다.

### Parameter

EGL_HANDLE hObj	Total count를 설정 할 Scroll bar handle.
int totalcount	Scroll bar가 관리 할 항목의 전체 크기.

### Return Value

없음.

### Example

```
EGL_HANDLE scroll;  
  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
  
egl_scroll_set_totalcount(scroll, 45);
```

## ▶ egl\_scroll\_get\_totalcount function

```
int egl_scroll_get_totalcount(  
    EGL_HANDLE hObj  
);
```

### Overview

Scroll bar의 total count를 가져온다.

### Parameter

EGL\_HANDLE hObj                      Total count를 가져 올 Scroll bar handle.

### Return Value

Total count value.

### Example

```
EGL_HANDLE scroll;  
int totalcount = 0;  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
totalcount = egl_scroll_get_totalcount(scroll);
```

## ▶ egl\_scroll\_set\_viewcount function

```
void egl_scroll_set_viewcount(  
    EGL_HANDLE hObj,  
    int viewcount  
);
```

### Overview

Scroll bar의 view count를 설정한다.

### Parameter

EGL_HANDLE hObj	View count를 설정 할 Scroll bar handle.
int viewcount	Scroll bar의 thumb가 표시할 항목의 크기.

### Return Value

없음.

### Example

```
EGL_HANDLE scroll;  
  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
  
egl_scroll_set_viewcount(scroll, 5);
```

## ▶ `egl_scroll_get_viewcount` function

```
int egl_scroll_get_viewcount(  
    EGL_HANDLE hObj  
);
```

### Overview

Scroll bar의 view count를 가져온다.

### Parameter

EGL\_HANDLE hObj                      View count를 가져 올 Scroll bar handle.

### Return Value

View count value.

### Example

```
EGL_HANDLE scroll;  
int viewcount = 0;  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
viewcount = egl_scroll_get_viewcount(scroll);
```

## ▶ egl\_scroll\_set\_upcount function

```
void egl_scroll_set_upcount(  
    EGL_HANDLE hObj,  
    int upcount  
);
```

### Overview

Scroll bar의 up count를 설정한다.

### Parameter

EGL_HANDLE hObj	Up count를 설정 할 Scroll bar handle.
int upcount	Scroll bar의 thumb 이전 항목의 크기.

### Return Value

없음.

### Example

```
EGL_HANDLE scroll;  
  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
  
egl_scroll_set_upcount(scroll, 10);
```

## ▶ egl\_scroll\_get\_upcount function

```
int egl_scroll_get_upcount(  
    EGL_HANDLE hObj  
);
```

### Overview

Scroll bar의 up count를 가져온다.

### Parameter

EGL\_HANDLE hObj                      Up count를 가져 올 Scroll bar handle.

### Return Value

Up count value.

### Example

```
EGL_HANDLE scroll;  
int upcount = 0;  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
upcount = egl_scroll_get_upcount(scroll);
```

## ▶ egl\_scroll\_set\_bgcolor function

```
void egl_scroll_set_bgcolor(  
    EGL_HANDLE hObj,  
    unsigned char r,  
    unsigned char g,  
    unsigned char b  
)
```

### Overview

Scroll bar의 back ground color를 설정한다.

### Parameter

EGL_HANDLE hObj	Back ground color를 설정 할 Scroll bar handle.
unsigned char r	RGB 중 Red 값.
unsigned char g	RGB 중 Green 값.
unsigned char b	RGB 중 Blue 값.

### Return Value

없음.

### Example

```
EGL_HANDLE scroll;  
  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
  
egl_scroll_set_bgcolor(scroll, 0, 0, 0xff);
```

## ▶ egl\_scroll\_set\_size function

```
void egl_scroll_set_size(  
    EGL_HANDLE hObj,  
    int w,  
    int h  
)
```

### Overview

Scroll bar의 크기를 설정한다.

### Parameter

EGL_HANDLE hObj	크기를 설정 할 Scroll bar handle.
int w	가로 크기.
int h	세로 크기.

### Return Value

없음.

### Example

```
EGL_HANDLE scroll;  
scroll = egl_create_scrollbar(100,100, 30, 200, 30, 10, TRUE);  
...  
egl_scroll_set_size(scroll, 40, 200);
```

## ► Scroll Bar Example.

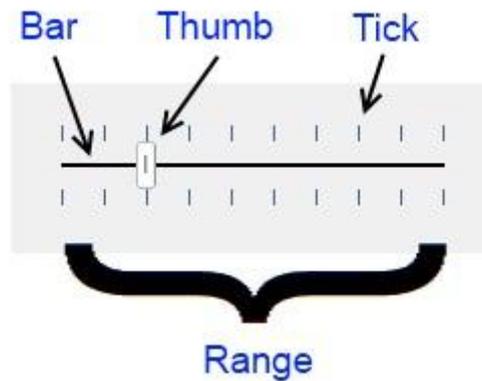
### Example

```
EGL_HANDLE scroll;
void scroll_callback(EGL_HANDLE h, int event)
{
    int totalcount = 0;
    int viewcount = 0;
    int upcount = 0;

    if(event == SCBAR_CLICKED)
    {
        egl_scroll_get_position(scroll, &totalcount, &viewcount, &upcount);
        debugprintf("%d, %d, %d\n", totalcount, viewcount, upcount);
    }
}
extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);
int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    egl_init();
    scroll = egl_create_scrollbar(100, 100, 30, 200, 30, 10, TRUE);
    egl_scroll_set_totalcount(scroll, 40);
    egl_scroll_callback(scroll, scroll_callback);
    egl_window_add_object(hWin, scroll);
    egl_window_show(hWin);
    egl_draw();
    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        egl_draw();
    }
}
```

# Slider



Function	Description
<a href="#">egl_create_slider</a>	Slider를 생성한다.
<a href="#">egl_slider_callback</a>	Slider 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.
<a href="#">egl_slider_set_pos</a>	Slider의 thumb position을 설정한다.
<a href="#">egl_slider_get_pos</a>	Slider의 thumb position 값을 가져온다.
<a href="#">egl_slider_set_range</a>	Slider의 range를 설정한다.
<a href="#">egl_slider_get_range</a>	Slider의 range를 가져온다.
<a href="#">egl_slider_stepit</a>	Slider를 전후로 한 step이동한다.
<a href="#">egl_slider_set_tick_frequency</a>	Slider의 tick 빈도 수를 설정한다.
<a href="#">egl_slider_set_tick_style</a>	Slider의 tick style을 설정한다.
<a href="#">egl_slider_set_thumb_size</a>	Slider의 thumb size를 설정한다.
<a href="#">egl_slider_get_thumb_size</a>	Slider의 thumb size를 가져온다.
<a href="#">egl_slider_set_barcolor</a>	Slider의 bar color를 설정한다.
<a href="#">egl_slider_set_tickcolor</a>	Slider의 tick color를 설정한다.

Function	Description
<a href="#">egl_slider_set_transparent</a>	Slider의 배경을 투명하게 할지를 결정한다.

## define

```
SLIDER_EVENT          typedef enum
                       {
                           SLR_CLICKED = 0,
                       } SLIDER_EVENT;
```

## ▶ egl\_create\_slider function

```
EGL_HANDLE egl_create_slider(  
    int x,  
    int y,  
    int w,  
    int h,  
    int range,  
    TICKSTYLE style,  
    BOOL bVertical  
);
```

### Overview

Slider를 생성한다.

### Parameter

int x	생성 될 Slider의 x좌표.
int y	생성 될 Slider의 y좌표.
int w	생성 될 Slider의 가로 크기.
int h	생성 될 Slider의 세로 크기.
int range	생성 될 Slider의 Range.
TICKSTYLE style	생성 될 Slider의 Tick style TICK_NONE : 눈금 표시 없음. TICK_TOPLEFT : 눈금 표시가 위쪽 또는 좌측에 표시. TICK_BOTTOMRIGHT : 눈금 표시가 아래쪽 또는 우측에 표시. TICK_BOTH : 눈금 표시가 양쪽에 표시.
BOOL bVertical	Slider의 출력 형태. TRUE = Vertical. FALSE = Horizontal.

### Return Value

생성 된 Slider handle.

## Example

```
EGL_HANDLE slider;
```

```
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);
```

## ▶ egl\_slider\_callback function

```
BOOL egl_slider_callback(  
    EGL_HANDLE hObj,  
    EVENT_CALLBACK cb  
);
```

### Overview

Slider 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.

### Parameter

EGL_HANDLE hObj	Callback 함수를 등록 할 Slider handle.
EVENT_CALLBACK cb	호출 될 callback 함수.

### Return Value

TRUE or FALSE

### Example

```
Void slider_callback(EGL_HANDLE h, int event)  
{  
  
}  
int main()  
{  
    ...  
    EGL_HANDLE slider;  
    slider = egl_create_slider(100,100, 200, 40, 10, TICK_BOTH, FALSE);  
    egl_slider_callback(check, slider_callback);  
    ...  
}
```

## ▶ egl\_slider\_set\_pos function

```
void egl_slider_set_pos(  
    EGL_HANDLE hObj,  
    int nPos  
);
```

### Overview

Slider의 thumb position을 설정한다.

### Parameter

EGL_HANDLE hObj	Position을 설정 할 Slider handle.
int nPos	Position 위치. (nPos는 정수로 0 ~ range)

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
slider = egl_create_slider(100,100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_set_pos(slider, 2);
```

## ▶ egl\_slider\_get\_pos function

```
int egl_slider_get_pos(  
    EGL_HANDLE hObj  
);
```

### Overview

Slider의 thumb position 값을 가져온다.

### Parameter

EGL\_HANDLE hObj                      Position을 가져 올 Slider handle.

### Return Value

Position value.

### Example

```
EGL_HANDLE slider;  
int slider_pos = 0;  
slider = egl_create_slider(100,100, 200, 40, 10, TICK_BOTH, FALSE);  
slider_pos = egl_slider_get_pos(slider);
```

## ▶ egl\_slider\_set\_range function

```
void egl_slider_set_range(  
    EGL_HANDLE hObj,  
    int nMinPos,  
    int nMaxPos  
);
```

### Overview

Slider의 range를 설정한다.

### Parameter

EGL_HANDLE hObj	Range를 설정 할 Slider handle.
int nMinPos	Min position 값. ( range = Max position – Min position ).
int nMaxPos	Max position 값. ( range = Max position – Min position).

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
slider = egl_create_slider(100,100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_set_range(slider, 0, 15);
```

## ▶ egl\_slider\_get\_range function

```
void egl_slider_get_range(  
    EGL_HANDLE hObj,  
    int* IpMinPos,  
    int* IpMaxPos  
);
```

### Overview

Slider의 range를 가져온다.

### Parameter

EGL_HANDLE hObj	Range 값을 가져 올 Slider handle.
int* IpMinPos	Min position 값을 저장 할 포인터.
int* IpMaxPos	Max position 값을 저장 할 포인터.

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
int max_pos = 0;  
int min_pos = 0;  
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_get_range(slider, &min_pos, &max_pos);
```

## ▶ egl\_slider\_stepit function

```
void egl_slider_stepit(  
    EGL_HANDLE hObj,  
    BOOL inc  
);
```

### Overview

Slider를 전후로 한 step이동한다.

### Parameter

EGL_HANDLE hObj	Step 이동 할 Slider handle.
BOOL inc	Step의 증가/감소를 결정. TRUE = 증가. FALSE = 감소.

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_stepit(slider, TRUE);  
//egl_slider_stepit(slider, FALSE);
```

## ▶ egl\_slider\_set\_tick\_frequency function

```
void egl_slider_set_tick_frequency(  
    EGL_HANDLE hObj,  
    int freq  
);
```

### Overview

Slider의 tick 빈도 수를 설정한다.

### Parameter

EGL_HANDLE hObj	Tick 빈도 수를 설정 할 Slider handle.
int freq	Tick 출력 빈도 수.

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_set_tick_frequency(slider, 2);
```

## ▶ egl\_slider\_set\_tick\_style function

```
void egl_slider_set_tick_frequency(  
    EGL_HANDLE hObj,  
    TICKSTYLE style  
);
```

### Overview

Slider의 tick style을 설정한다.

### Parameter

EGL\_HANDLE hObj  
TICKSTYLE style

Tick style을 설정 할 Slider handle.  
TICK style.

TICK\_NONE : 눈금 표시 없음.

TICK\_TOPLEFT : 눈금 표시가 위쪽 또는 좌측에 표시.

TICK\_BOTTOMRIGHT : 눈금 표시가 아래쪽 또는 우측에 표시.

TICK\_BOTH : 눈금 표시가 양쪽에 표시.

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_set_tick_style(slider, TICK_BOTTOMRIGHT);
```

## ▶ egl\_slider\_set\_thumb\_size function

```
void egl_slider_set_thumb_size(  
    EGL_HANDLE hObj,  
    int width,  
    int height  
);
```

### Overview

Slider의 thumb size를 설정한다.

### Parameter

EGL_HANDLE hObj	Thumb size를 설정 할 Slider handle.
int width	Thumb의 가로 크기.
int height	Thumb의 세로 크기.

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_set_thumb_size(slider, 10, 30);
```

## ▶ egl\_slider\_get\_thumb\_size function

```
void egl_slider_get_thumb_size(  
    EGL_HANDLE hObj,  
    int* w,  
    int* h  
)
```

### Overview

Slider의 thumb size를 가져온다.

### Parameter

EGL_HANDLE hObj	Thumb size를 가져 올 Slider handle.
int* w	Thumb 가로 크기를 저장할 포인터.
int* h	Thumb 세로 크기를 저장할 포인터.

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
int w = 0;  
int h = 0;  
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_get_thumb_size(slider, &w, &h);
```

## ▶ egl\_slider\_set\_barcolor function

```
void egl_slider_set_barcolor(  
    EGL_HANDLE hObj,  
    unsigned char r,  
    unsigned char g,  
    unsigned char b  
);
```

### Overview

Slider의 bar color를 설정한다.

### Parameter

EGL_HANDLE hObj	Bar color를 설정 할 Slider handle.
unsigned char r	RGB 중 Red 값.
unsigned char g	RGB 중 Green 값.
unsigned char b	RGB 중 Blue 값.

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_set_barcolor(slider, 0, 0xff, 0);
```

## ▶ egl\_slider\_set\_tickcolor function

```
void egl_slider_set_tickcolor(  
    EGL_HANDLE hObj,  
    unsigned char r,  
    unsigned char g,  
    unsigned char b  
);
```

### Overview

Slider의 tick color를 설정한다.

### Parameter

EGL_HANDLE hObj	Tick color를 설정 할 Slider handle.
unsigned char r	RGB 중 Red 값.
unsigned char g	RGB 중 Green 값.
unsigned char b	RGB 중 Blue 값.

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_set_tickcolor(slider, 0xff, 0, 0);
```

## ▶ egl\_slider\_set\_transparent function

```
void egl_slider_set_transparent(  
    EGL_HANDLE hObj,  
    BOOL bflag  
);
```

### Overview

Slider의 배경을 투명하게 할지를 결정한다.

### Parameter

EGL_HANDLE hObj	배경을 투명하게 설정 할 Slider handle.
BOOL bflag	TRUE => 배경 투명. FALSE => 배경 불투명.

### Return Value

없음.

### Example

```
EGL_HANDLE slider;  
slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);  
egl_slider_set_transparent(slider, TRUE);
```

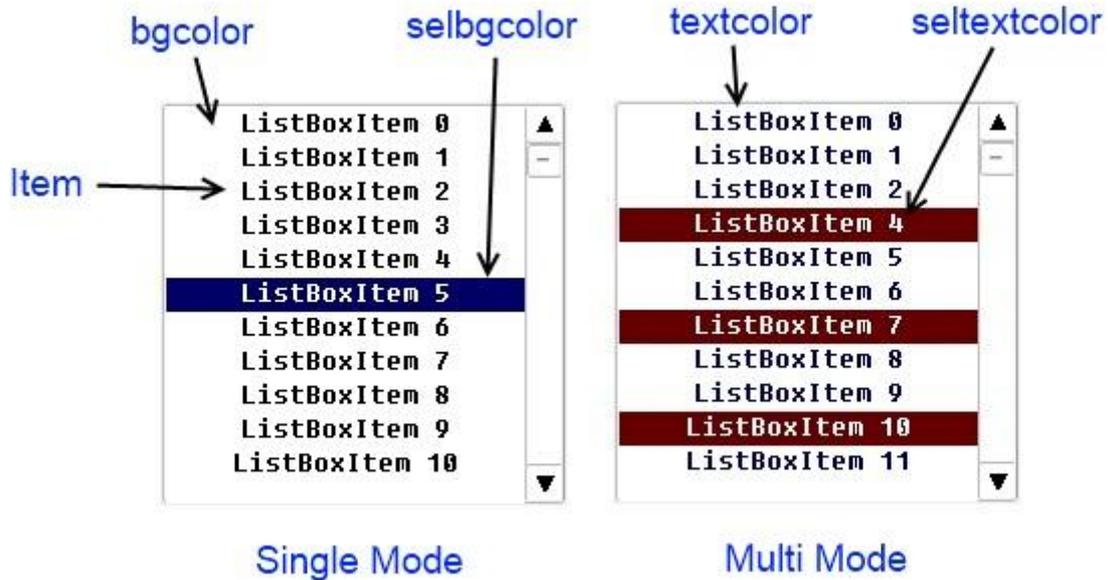
## ► Slider Example.

### Example

```
EGL_HANDLE slider;
void slider_callback(EGL_HANDLE h, int event)
{
    if(event == SLR_CLICKED)
    {
        debugprintf("position = %dWrWn",egl_slider_get_pos());
    }
}
extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);
int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    egl_init();
    slider = egl_create_slider(100, 100, 200, 40, 10, TICK_BOTH, FALSE);
    egl_sliderl_callback(slider, slider_callback);
    egl_window_add_object(hWin, slider);
    egl_window_show(hWin);
    egl_draw();
    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        egl_draw();
    }
}
```

# List Box



Function	Description
<a href="#">egl_create_listbox</a>	List box를 생성한다.
<a href="#">egl_listbox_callback</a>	List box이벤트가 발생했을 때 호출될 callback 함수를 등록한다.
<a href="#">egl_listbox_additem</a>	List box에 Item을 추가한다.
<a href="#">egl_listbox_delitem</a>	List box의 마지막 Item을 제거한다.
<a href="#">egl_listbox_delitem_text</a>	List box에서 text에 해당되는 Item을 제거한다.
<a href="#">egl_listbox_alldelitem</a>	List box의 모든 Item을 제거한다.
<a href="#">egl_listbox_get_all_itemlist</a>	List box의 모든 Item들의 text목록을 반환한다.
<a href="#">egl_listbox_get_sel_item</a>	List box가 Single Mode일 때 선택된 Item의 text를 반환한다.
<a href="#">egl_listbox_get_multiple_sel_itemlist</a>	List box가 Multiple Mode일 때 선택된 Item들의 text목록을 반환한다.
<a href="#">egl_listbox_set_bgcolor</a>	List box의 배경색을 지정한다.
<a href="#">egl_listbox_set_selbgcolor</a>	List box에서 선택된 Item의 배경색을 지정한다.

Function	Description
<a href="#">egl_listbox_set_textcolor</a>	List box의 글자 색을 지정한다.
<a href="#">egl_listbox_set_seltexcolor</a>	List box에서 선택된 item의 글자 색을 지정한다.
<a href="#">egl_listbox_set_scrollwidth</a>	List box의 scrollbar의 가로 크기를 설정한다.

Define	
LIST_EVENT	<pre>typedef enum {     LIST_CHANGED = 0, } LIST_EVENT;</pre>

## ▶ egl\_create\_listbox function

```
EGL_HANDLE egl_create_listbox(  
    int x,  
    int y,  
    int w,  
    int h,  
    bool bMultiple  
);
```

### Overview

List box를 생성한다.

### Parameter

int x	생성될 List box의 x좌표.
int y	생성될 List box의 y좌표.
int w	생성될 List box의 가로 크기.
int h	생성될 List box의 세로 크기.
Bool bMultiple	생성될 List box에서 복수 item 선택 여부를 설정한다.

### Return Value

생성된 List box의 pointer

### Example

```
EGL_HANDLE listbox[2];  
  
listbox [0] = egl_create_button (100, 100, 200, 200, TRUE);  
  
listbox [1] = egl_create_button (400, 100, 200, 200, FALSE);
```

## ▶ egl\_listbox\_callback function

```
void egl_listbox_evnet_callback (  
    EGL_HANDLE hObj,  
    EVENT_CALLBACK cb  
);
```

### Overview

List box 이벤트가 발생했을 때 호출될 callback 함수를 등록한다.

### Parameter

EGL_HANDLE hObj	이벤트가 발생하는 List box의 handle
EVENT_CALLBACK cb	이벤트가 발생했을 때 호출될 callback 함수

### Return Value

없음.

### Example

```
static void listbox_callback(EGL_HANDLE h, int event)  
{  
    if(event == LIST_CHANGED)  
        debugprintf("LIST_CHANGED");  
}  
  
void main(void)  
{  
    EGL_HANDLE listbox;  
  
    egl_listbox_event_callback(listbox, listbox_callback);  
}
```

## ▶ egl\_listbox\_additem function

```
void egl_listbox_additem (  
    EGL_HANDLE hObj,  
    const char* text  
);
```

### Overview

List box에 Item을 추가한다.

### Parameter

EGL_HANDLE hObj	Item을 추가할 List box의 handle
const char* text	추가할 Item의 text

### Return Value

없음.

### Example

```
EGL_HANDLE listbox;  
  
egl_listbox_additem(listbox, "ListboxItem 1");
```

## ▶ egl\_listbox\_delitem function

```
void egl_listbox_delitem (  
    EGL_HANDLE hObj  
);
```

### Overview

List box의 마지막 Item을 제거한다.

### Parameter

EGL\_HANDLE hObj                      Item을 제거할 List box의 handle

### Return Value

없음.

### Example

```
EGL_HANDLE listbox;  
  
egl_listbox_delitem(listbox);
```

## ▶ egl\_listbox\_delitem\_textfunction

```
BOOL egl_listbox_delitem_text (  
    EGL_HANDLE hObj,  
    const char* text  
);
```

### Overview

List box에서 text에 해당되는 Item을 제거한다.

### Parameter

EGL_HANDLE hObj	Item을 제거할 List box의 handle
const char* text	List box에서 제거할 Item의 text

### Return Value

Item 제거의 성공 여부를 반환

### Example

```
EGL_HANDLE listbox;  
  
if(egl_listbox_delitem_text(listbox, "ListboxItem 1") == TRUE)  
    debugprintf("egl_listbox_delitem_text complete\n");
```

## ▶ egl\_listbox\_delitem\_text function

```
void egl_listbox_delitem_text (  
    EGL_HANDLE hObj,  
    const char* text  
);
```

### Overview

List box에서 text에 해당되는 Item을 제거한다.

### Parameter

EGL_HANDLE hObj	Item을 제거할 List box의 handle
const char* text	제거할 item의 text.

### Return Value

없음.

### Example

```
EGL_HANDLE listbox;  
  
egl_listbox_delitem_text(listbox, "item1");
```

## ▶ egl\_listbox\_delitem\_textfunction

```
BOOL egl_listbox_delitem_text (  
    EGL_HANDLE hObj,  
    const char* text  
);
```

### Overview

List box에서 text에 해당되는 Item을 제거한다.

### Parameter

EGL_HANDLE hObj	Item을 제거할 List box의 handle
const char* text	List box에서 제거할 Item의 text

### Return Value

Item 제거의 성공 여부를 반환

### Example

```
EGL_HANDLE listbox;  
  
if(egl_listbox_delitem_text(listbox, "ListboxItem 1") == TRUE)  
    debugprintf("egl_listbox_delitem_text complete\n");
```

## ▶ egl\_listbox\_alldelitem function

```
void egl_listbox_alldelitem (  
    EGL_HANDLE hObj,  
);
```

### Overview

List box의 모든 Item을 제거한다.

### Parameter

EGL\_HANDLE hObj                      Item을 제거할 List box의 handle

### Return Value

없음

### Example

```
EGL_HANDLE listbox;  
  
egl_listbox_alldelitem(listbox);
```

## ▶ egl\_listbox\_get\_all\_itemlist function

```
const char** egl_listbox_get_all_itemlist (  
    EGL_HANDLE hObj,  
    int* itemcnt  
);
```

### Overview

List box의 모든 Item들의 text목록을 반환한다.

### Parameter

EGL_HANDLE hObj	Item의 text 목록을 반환 받을 List box의 handle
int* itemcnt	List box의 Item개수를 반환 받을 변수의 pointer

### Return Value

모든 Item의 text list pointer

※ 주의 : 반드시 return 받은 list pointer의 메모리를 해제하여야 한다.

### Example

```
EGL_HANDLE listbox;  
int ItemCnt;  
const char** ItemTextList;  
  
ItemTextList = egl_listbox_all_itemlist(listbox, &ItemCnt);  
  
free(ItemTextList);
```

## ▶ egl\_listbox\_get\_sel\_item function

```
const char* egl_listbox_get_sel_item (  
    EGL_HANDLE hObj,  
    int* index  
);
```

### Overview

List box가 Single Mode 일 때 선택된 Item의 Text를 반환한다

### Parameter

EGL_HANDLE hObj	선택된 Item의 text를 반환 받을 List box의 handle
int* index	선택된 Item의 순번을 반환 받을 변수의 pointer

### Return Value

선택된 Item의 text pointer

### Example

```
EGL_HANDLE listbox;  
int selectItemIndex;  
const char* selectItemText;  
  
selectItemText = egl_listbox_get_sel_item(listbox, &selectItem);
```

## ▶ egl\_listbox\_get\_multiple\_sel\_itemlist function

```
const char** egl_listbox_get_multiple_sel_itemlist (  
    EGL_HANDLE hObj,  
    int* selcnt  
);
```

### Overview

List box가 Multiple Mode일 때 선택된 Item의 text들의 목록을 반환한다.

### Parameter

EGL_HANDLE hObj	선택된 Item들의 text를 반환 받을 List box의 handle
int* selcnt	선택된 Item의 개수를 반환 받을 변수의 pointer

### Return Value

선택된 Item들의 text list pointer

※ 주의 : 반드시 return 받은 list pointer의 메모리를 해제하여야 한다.

### Example

```
EGL_HANDLE listbox;  
int selectItemCnt;  
const char** selectItemTextList;  
  
selectItemTextList = egl_listbox_get_sel_itemlist (listbox, & selectItemCnt);  
  
free(selectItemTextList);
```

## ▶ egl\_listbox\_set\_bgcolor function

```
void egl_listbox_set_bgcolor (  
    EGL_HANDLE  hObj,  
    unsigned char r,  
    unsigned char g,  
    unsigned char b  
);
```

### Overview

List box의 배경색을 지정한다.

### Parameter

EGL_HANDLE  hObj	배경색을 지정할 List box의 handle
unsigned char r	지정될 배경색의 RGB값 중 Red값
unsigned char g	지정될 배경색의 RGB값 중 Green값
unsigned char b	지정될 배경색의 RGB값 중 Blue값

### Return Value

없음

### Example

```
EGL_HANDLE listbox;  
  
egl_listbox_set_bgcolor(listbox, 0x0, 0x0, 0xFF); // Blue
```

## ▶ egl\_listbox\_set\_selbgcolor function

```
void egl_listbox_set_selbgcolor (  
    EGL_HANDLE  hObj,  
    unsigned char r,  
    unsigned char g,  
    unsigned char b  
);
```

### Overview

List box에서 선택된 Item의 배경색을 지정한다.

### Parameter

EGL_HANDLE  hObj	선택된 Item의 배경색을 지정할 List box의 handle
unsigned char r	지정될 배경색의 RGB값 중 Red값
unsigned char g	지정될 배경색의 RGB값 중 Green값
unsigned char b	지정될 배경색의 RGB값 중 Blue값

### Return Value

없음

### Example

```
EGL_HANDLE listbox;  
  
egl_listbox_set_selbgcolor(listbox, 0xFF, 0x0, 0x0); // Red
```

## ▶ egl\_listbox\_set\_textcolor function

```
void egl_listbox_set_textcolor (  
    EGL_HANDLE  hObj,  
    unsigned char r,  
    unsigned char g,  
    unsigned char b  
);
```

### Overview

List box의 글자 색을 지정한다.

### Parameter

EGL_HANDLE  hObj	글자 색을 지정할 List box의 handle
unsigned char r	지정될 글자 색의 RGB값 중 Red값
unsigned char g	지정될 글자 색의 RGB값 중 Green값
unsigned char b	지정될 글자 색의 RGB값 중 Blue값

### Return Value

없음

### Example

```
EGL_HANDLE listbox;  
  
egl_listbox_set_textcolor(listbox, 0x0, 0x0, 0x0); // Black
```

## ▶ egl\_listbox\_set\_seltextrcolor function

```
void egl_listbox_set_seltextrcolor (  
    EGL_HANDLE  hObj,  
    unsigned char r,  
    unsigned char g,  
    unsigned char b  
);
```

### Overview

List box에서 선택된 Item의 글자 색을 지정한다.

### Parameter

EGL_HANDLE hObj	선택된 Item의 글자 색을 지정할 List box의 handle
unsigned char r	지정될 글자 색의 RGB값 중 Red값
unsigned char g	지정될 글자 색의 RGB값 중 Green값
unsigned char b	지정될 글자 색의 RGB값 중 Blue값

### Return Value

없음

### Example

```
EGL_HANDLE listbox;  
  
egl_listbox_set_seltextrcolor(listbox, 0xFF, 0xFF, 0xFF); // White
```

## ▶ egl\_listbox\_set\_scrollwidth function

```
void egl_listbox_set_scrollwidth (  
    EGL_HANDLE hObj,  
    int width  
);
```

### Overview

List box의 scrollbar의 가로 크기를 설정한다.

### Parameter

EGL_HANDLE hObj	Scrollbar의 가로크기를 설정 할 List box의 handle
int width	Scrollbar의 가로 크기.

### Return Value

없음

### Example

```
EGL_HANDLE listbox;  
  
egl_listbox_set_scrollwidth(listbox, 40);
```

## ► List box Example.

### Example

```
#include "adStar.h"
extern BOOL process_touch(BOOL* touchdown,EGL_POINT* pPoint);

static void listbox_cb1(EGL_HANDLE h, int event)
{
    int itemcnt, index;
    const char **itemlist;
    const char *selitem;
    int i;

    if(event == LIST_CHANGED) {
        debugprintf("LIST_CHANGED\r\n");
        // Get all item list
        itemlist = egl_listbox_get_all_itemlist(h, &itemcnt);
        debugprintf("All Item %d\r\n", itemcnt);
        for(i = 0; i < itemcnt; i++)
            debugprintf("Item : [%s]\r\n", itemlist[i]);
        // Get selected item list
        selitem = egl_listbox_get_sel_item(h, &index);
        debugprintf("Selected Item [%d] : [%s]\r\n", index, selitem);
    }
    free(itemlist);
}

static void listbox_cb2(EGL_HANDLE h, int event)
{
    int selcnt;
    const char **selitemlist;
    int i;

    if(event == LIST_CHANGED) {
        debugprintf("LIST_CHANGED\r\n");
        selitemlist = egl_listbox_get_multiple_sel_itemlist(h, &selcnt);
        debugprintf("Selected Items %d \r\n", selcnt);
        for(i = 0; i < selcnt; i++)
            debugprintf("Selected Item : [%s]\r\n", selitemlist[i]);
    }
}
```

```

    free(selitemlist);
}
void user_main()
{
    float addpos=1.0f;
    EGL_POINT touch_pt;
    BOOL touchdown=FALSE;
    EGL_HANDLE hWin;
    EGL_HANDLE listbox[2];

    egl_init();

    hWin = egl_create_window("Main Window");

    listbox[0] = egl_create_listbox(100, 100, 200, 200, FALSE);
    listbox[1] = egl_create_listbox(400, 100, 200, 200, TRUE);

    egl_listbox_callback(listbox[0], listbox_cb1);
    egl_listbox_callback(listbox[1], listbox_cb2);

    {
        int i;
        char str_text[16];
        for(i=0;i<100;i++)
        {
            sprintf(str_text,"ListBoxItem %d",i);
            egl_listbox_additem(listbox[0],str_text);
            egl_listbox_additem(listbox[1],str_text);
        }
    }
    // Delete last item
    egl_listbox_delitem(listbox[0]);
    // Delete "ListBoxItem 3" item
    egl_listbox_delitem_text(listbox[1], "ListBoxItem 3");

    // Set background color
    egl_listbox_set_bgcolor(listbox[1], 0xff, 0xff, 0xff);
    // Set selected background color
    egl_listbox_set_selbgcolor(listbox[1], 0x66, 0x0, 0x0);
    // Set text color
    egl_listbox_set_textcolor(listbox[1], 0, 0, 0x66);
}

```

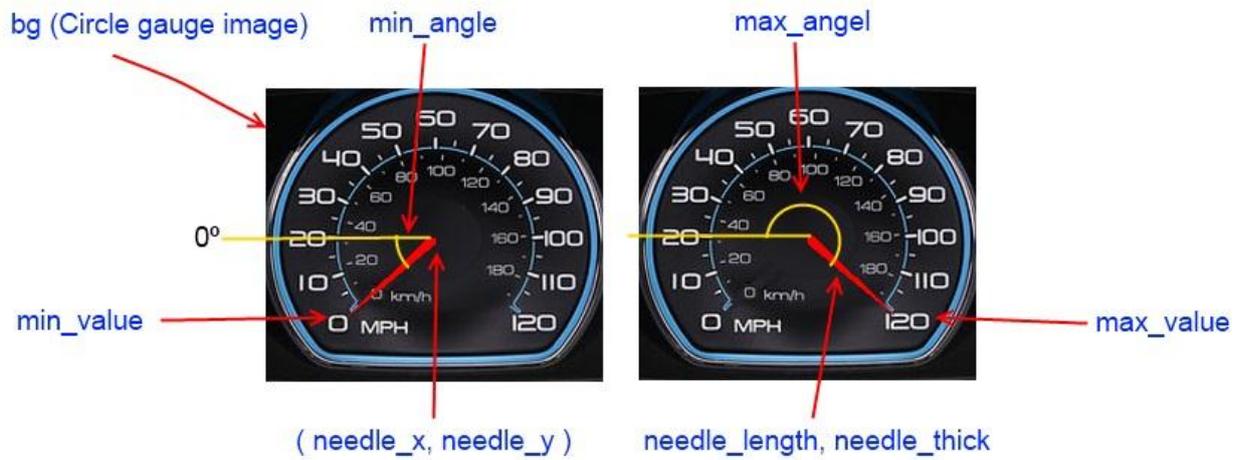
```
// Set selected text color
egl_listbox_set_seltextrcolor(listbox[1], 0xff, 0xff, 0xff);
egl_window_add_object(hWin, listbox[0]);
egl_window_add_object(hWin, listbox[1]);

egl_window_show(hWin, TRUE);
egl_draw();

while(1)
{
    BOOL bEvent = TRUE;

    if(process_touch(&touchdown, &touch_pt))
        egl_user_touch_input(touchdown, &touch_pt);
    else
        bEvent = FALSE;
    egl_draw();
}
}
```

# Circle Gauge



Function	Description
<a href="#">egl_create_circle_gauge</a>	Circle gauge를 생성한다.
<a href="#">egl_circle_gauge_set_value</a>	Circle gauge 값을 설정한다.
<a href="#">egl_circle_gauge_get_value</a>	Circle gauge 값을 가져온다.

## ▶ egl\_create\_circle\_gauge function

```
EGL_HANDLE egl_create_circle_gauge(  
    SURFACE* bg,  
    int x,  
    int y,  
    EGL_CIRCLE_GAUGE_INFO* pInfo  
);
```

### Overview

Circle gauge를 생성한다.

### Parameter

SURFACE* bg	Circle gauge image.
int x	Circle gauge x좌표.
int y	Circle gauge y좌표.
EGL_CIRCLE_GAUGE_INFO* pInfo	Circle gauge 정보.

```
typedef struct _tagCIRCLE_GAUGE_INFO  
{  
    int needle_x;        // gauge needle x좌표.  
    int needle_y;        // gauge needle y좌표.  
    int needle_length;   // gauge needle 길이.  
    int needle_thick;    // gauge needle 두께.  
    int min_value;       // gauge의 최소 값.  
    int max_value;       // gauge의 최대 값.  
    int min_angle;       // gauge의 최소 각.  
    int max_angle;       // gauge의 최대 각.  
}EGL_CIRCLE_GAUGE_INFO;
```

### Return Value

생성 된 circle gauge handle

## Example

```
SURFACE* gauge_bg;
EGL_HANDLE c_gauge;
EGL_CIRCLE_GAUGE_INFO CGInfo;
CGInfo.needle_x = 113;
CGInfo.needle_y = 101;
CGInfo.min_angle = -42;
CGInfo.max_angle = 222;
CGInfo.min_value = 0;
CGInfo.max_value = 120;
CGInfo.needle_length = 80;
CGInfo.needle_thick = 3;
gauge_bg = loadbmp("gauge.bmp");
c_gauge = egl_create_circle_gauge(gauge_bg, 286, 37, &CGInfo);
```

## ▶ egl\_circle\_gauge\_set\_value function

```
BOOL egl_circle_gauge_set_value(  
    EGL_HANDLE h,  
    int value  
);
```

### Overview

Circle gauge 값을 설정한다.

### Parameter

EGL_HANDLE h	Gauge 값을 설정 할 circle gauge handle.
int vaule	설정 gauge 값.

### Return Value

TRUE or FALSE

### Example

```
...  
c_gauge = egl_create_circle_gauge(gauge_bg, 286,37,&CGInfo);  
egl_circle_gauge_set_value(c_gauge, 80);  
...
```

## ▶ egl\_circle\_gauge\_get\_value function

```
int egl_circle_gauge_get_value(  
    EGL_HANDLE h  
);
```

### Overview

Circle gauge 값을 가져온다.

### Parameter

EGL\_HANDLE h                      Gauge 값을 가져 올 circle gauge handle.

### Return Value

Gauge 값.

### Example

```
...  
c_gauge = egl_create_circle_gauge(gauge_bg, 286,37,&CGInfo);  
int gauge_value = egl_circle_gauge_get_value(c_gauge);  
...
```

## ► Circle Gauge Example.

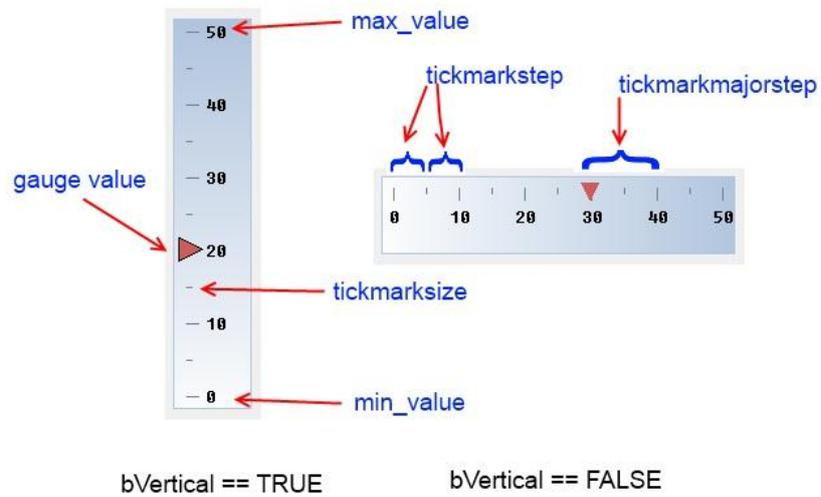
### Example

```
extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);
int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    SURFACE* gauge_bg;
    EGL_HANDLE c_gauge;
    EGL_CIRCLE_GAUGE_INFO CGInfo;
    egl_init();
    CGInfo.needle_x = 113;
    CGInfo.needle_y = 101;
    CGInfo.min_angle = -42;
    CGInfo.max_angle = 222;
    CGInfo.min_value = 0;
    CGInfo.max_value = 120;
    CGInfo.needle_length = 80;
    CGInfo.needle_thick = 3;
    gauge_bg = loadbmp("gauge.bmp");
    c_gauge = egl_create_circle_gauge(gauge_bg, 286, 37, &CGInfo);
    egl_window_add_object(hWin, c_gauge);
    egl_window_show(hWin);
    egl_draw();
    int gauge_value = 0;
    int gauge_flag = 0;
    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        if(gauge_flag)
        {
            gauge_value++;
        }
    }
}
```

```
    if(gauge_value > 120)
    {
        gauge_value = 120;
        gauge_flag = 0;
    }
}
else
{
    gauge_value--;
    if(gauge_value < 0)
    {
        gauge_value = 0;
        gauge_flag = 1;
    }
}
egl_circle_gauge_set_valut(c_gauge, gauge_value);
egl_draw();
}
}
```

# Bar Gauge



Function	Description
<a href="#">egl_create_bar_gauge</a>	bar gauge를 생성한다.
<a href="#">egl_bar_gauge_set_value</a>	bar gauge 값을 설정한다.
<a href="#">egl_bar_gauge_get_value</a>	bar gauge 값을 가져온다.

## ▶ egl\_create\_bar\_gauge function

```
EGL_HANDLE egl_create_bar_gauge(  
    int x,  
    int y,  
    int w,  
    int h,  
    EGL_BAR_GAUGE_INFO* pInfo  
);
```

### Overview

bar gauge를 생성한다.

### Parameter

int x	Bar gauge x좌표.
int y	Bar gauge y좌표.
int w	Bar gauge 가로 크기.
int h	Bar gauge 세로 크기.
EGL_BAR_GAUGE_INFO* pInfo	Bar gauge 정보

```
typedef struct _tagBAR_GAUGE_INFO  
{  
    int min_value;           // gauge 최소 값.  
    int max_value;          // gauge 최대 값.  
    int tickmarksize;       // tick size. major tick = tick size * 2  
    int tickmarkstep;       // tick step size.  
    int tickmarkmajorstep;  // major tick step size.  
    BOOL bVertical;         // vertical mode select.  
    SURFACE* bg;           // back ground image.  
} EGL_BAR_GAUGE_INFO;
```

### Return Value

생성 된 bar gauge handle.

## Example

```
EGL_HANDLE b_gauge;
EGL_BAR_GAUGE_INFO BGInfo;
BGInfo.min_value = 0;
BGInfo.max_value = 50;
BGInfo.bVertical = TRUE;
BGInfo.tickmarksize = 5;
BGInfo.tickmarkstep = 5;
BGInfo.tickmarkmajorstep = 10;
BGInfo.bg = NULL;
b_gauge = egl_create_bar_gauge(10, 40, 60, 300, &BGInfo);
```

## ▶ egl\_bar\_gauge\_set\_value function

```
BOOL egl_bar_gauge_set_value(  
    EGL_HANDLE h,  
    int value  
);
```

### Overview

bar gauge 값을 설정한다.

### Parameter

EGL_HANDLE h	값을 설정 할 bar gauge handle.
int value	설정 gauge 값.

### Return Value

TRUE or FALSE

### Example

```
...  
b_gauge = egl_create_bar_gauge(10, 40, 60, 300, &BGInfo);  
egl_bar_gauge_set_value(b_gauge, 20);  
...
```

## ▶ egl\_bar\_gauge\_get\_value function

```
int egl_bar_gauge_get_value(  
    EGL_HANDLE h  
);
```

### Overview

bar gauge 값을 가져온다.

### Parameter

EGL\_HANDLE h                      Gauge 값을 가져 올 bar gauge handle.

### Return Value

Gauge 값

### Example

```
...  
b_gauge = egl_create_bar_gauge(10, 40, 60, 300, &BGInfo);  
int gauge = egl_bar_gauge_get_value(b_gauge);  
...
```

## ► Bar Gauge Example.

### Example

```
extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);
int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    EGL_HANDLE b_gauge;
    EGL_BAR_GAUGE_INFO BGInfo;
    egl_init();
    BGInfo.min_value = 0;
    BGInfo.max_value = 50;
    BGInfo.bVertical = TRUE;
    BGInfo.tickmarksize = 5;
    BGInfo.tickmarkstep = 5;
    BGInfo.tickmarkmajorstep = 10;
    BGInfo.bg = NULL;
    b_gauge = egl_create_bar_gauge(10,40,60,300,&BGInfo);
    egl_window_add_object(hWin, b_gauge);
    egl_window_show(hWin);
    egl_draw();
    int gauge_value = 0;
    int gauge_flag = 0;
    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        if(gauge_flag)
        {
            gauge_value++;
            if(gauge_value > 50)
            {
                gauge_value = 50;
                gauge_flag = 0;
            }
        }
    }
}
```

```
    }  
  }  
  else  
  {  
    gauge_value--;  
    if(gauge_value < 0)  
    {  
      gauge_value = 0;  
      gauge_flag = 1;  
    }  
  }  
  egl_bar_gauge_set_valut(b_gauge, gauge_value);  
  egl_draw();  
}  
}
```

# Animation

Function	Description
<a href="#">egl_create_animation</a>	Animation object를 생성한다.

## ▶ egl\_create\_animation function

```
EGL_HANDLE egl_create_animation(  
    int x,  
    int y,  
    int w,  
    int h,  
    SURFACE** surflist,  
    int surfcnt,  
    int delaycnt  
);
```

### Overview

Animation object를 생성한다. Animation object는 연속 이미지를 번갈아 가며 draw한다.

### Parameter

int x	Animation object x좌표.
int y	Animation object y좌표.
int w	Animation object 가로 크기.
int h	Animation object 세로 크기.
SURFACE** surflist	Animation image list.
int surfcnt	Animation image count.
int delaycnt	Animation image 전환 delay.

### Return Value

생성 된 animation object handle.

## Example

```
EGL_HANDLE btn_ani;
SURFACE* surf_ani[10];
char fname[12];
int i;
for(i=0;i<10;i++)
{
    sprintf(fname, "Frame%d.bmp",i);
    surf_ani[i] = loadbmp(fname);
}
btn_ani = egl_create_animation(350, 150, 128, 128, surf_ani, 10, 0);
```

## ► Animation Example.

### Example

```
extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);
int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    EGL_HANDLE btn_ani;
    SURFACE* surf_ani[10];
    char fname[12];
    egl_init();
    int i;
    for(i=0;i<10;i++)
    {
        sprintf(fname, "Frame%d.bmp",i);
        surf_ani[i] = loadbmp(fname);
    }
    btn_ani = egl_create_animation(350, 150, 128, 128, surf_ani, 10, 0);
    egl_window_add_object(hWin, btn_ani);
    egl_window_show(hWin);
    egl_draw();
    while(1)
    {
        if( process_touch( &touchdown, &touch_pt ) )
            egl_user_touch_input( touchdown, &touch_pt );

        egl_draw();
    }
}
```

# Custom Object

Function	Description
<a href="#">egl_create_custom_object</a>	Custom object를 생성한다.

## Define

```
EGL_MSG_ID          typedef enum enumMSG
                    {
                        EGL_MSG_DRAW = 0,
                        EGL_MSG_DELETE,
                        EGL_MSG_FOCUS,
                        EGL_MSG_UNFOCUS,
                        EGL_MSG_KEY_UP,
                        EGL_MSG_KEY_DOWN,
                        EGL_MSG_TOUCHED,
                        EGL_MSG_UNTOUCHED,
                        EGL_MSG_MOVE,
                        EGL_MSG_TIMETICK
                    } EGL_MSG_ID;

EGL_MSG             typedef struct _tagMessage
                    {
                        EGL_MSGID msgID;
                        EGL_HANDLE hObj;
                        EGL_HANDLE hWin;
                        Union
                        {
                            EGL_POINT point;
                            U32 key;
                        } param;
                    } EGL_MSG;
```

## ▶ egl\_create\_custom\_object function

```
EGL_HANDLE egl_create_custom_object(  
    int x,  
    int y,  
    int w,  
    int h,  
    void* (*msg_handler)(EGL_MSG* pMsg)  
);
```

### Overview

Custom object를 생성한다.

### Parameter

int x	Custom object x좌표.
int y	Custom object y좌표.
int w	Custom object 가로 크기.
int h	Custom object 세로 크기.
void* (*msg_handler)(EGL_MSG* pMsg)	Custom object msg 처리 함수.

### Return Value

생성 된 custom object handle.

## Example

```
Static void* custom_obj_msghandler(EGL_MSG* pMsg)
{
    Switch(pMsg->msgID)
    {
        Case EGL_MSG_DRAW:
            /* draw */
            Break;
    }
    Return NULL;
}
...
Int main()
{
    ...
    EGL_HANDLE custom_obj;
    custom_obj = egl_create_custom(350, 240, 100, 30, custom_obj_msghandler);
    ...
}
```

## ► Custom Example.

### Example

```
static char speed_str[16];
static EGL_HANDLE hSpeed;

static void speed_draw(EGL_OBJECT_PTR pObj)
{
    draw_text_in_box(pObj->pFont, &pObj->rect, speed_str, EGL_ALIGN_CENTER);
}

static void* speed_msghandler(EGL_MSG* pMsg)
{
    switch(pMsg->msgID)
    {
        case EGL_MSG_DRAW:
            speed_draw(EGL_HANDLE_TO_OBJECT(pMsg->hObj));
            break;
    }
}

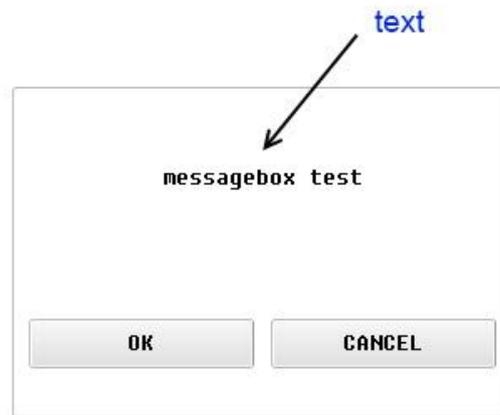
extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL  touchdown = FALSE;
    EGL_HANDLE hWin;
    EGL_HANDLE hSpeed;
    EGL_FONT* myfont;
    egl_init();
    myfont = egl_create_default_font();
    hSpeed = egl_create_custom_object(350, 240, 100, 30, speed_msghandler);
    egl_window_add_object(hWin, hSpeed);
    egl_set_font(hSpeed, myfont);
    strcpy(speed_str, "  0 Mile/h");
    egl_window_show(hWin);
    egl_draw();
}
```

```
while(1)
{
    if( process_touch( &touchdown, &touch_pt ) )
        egl_user_touch_input( touchdown, &touch_pt );

    egl_draw();
}
}
```

# MessageBox



`flags == MB_OKCANCEL`

Function	Description
<a href="#">egl_show_messagebox</a>	MessageBox를 출력한다.

## ▶ egl\_show\_messagebox function

```
int egl_show_messagebox(  
    const char* text,  
    int flags,  
    BOOL (*user_input_function)(EGL_MSG* pMsg)  
);
```

### Overview

Messagebox를 출력한다.

### Parameter

const char\* text

Messagebox에 출력할 문자.

int flags

Messagebox type.

MB\_OK

MB\_OKCANCEL

MB\_YESNO

MB\_YESNOCANCEL

BOOL (\*user\_input\_function)(EGL\_MSG\* pMsg)

사용자 입력 함수.

Messagebox 안의 어떤 버튼이 눌렸는지 확인을 위한 입력 함수.

### Return Value

Message box 안의 눌린 버튼의 type 값을 반환.

```
enum  
{  
    IDOK = 0,  
    IDCANCEL,  
    IDABORT,  
    IDRETRY,  
    IDIGNORE,  
    IDYES,  
    IDNO,  
    IDCONTINUE,  
}
```

## Example

```
BOOL my_touchinput(EG_MSG* pMsg)
{
    BOOL touchdown;
    EGL_POINT touch_pt;
    if(process_touch(&touchdown, &touch_pt))
    {
        if(touchdown)
            pMsg -> msgID = EGL_MSG_TOUCHED;
        else
            pMsg -> msgID = EGL_MSG_UNTOUCHED;
        pMsg -> param.point.x = touch_pt.x;
        pMsg -> param.point.y = touch_pt.y;
        return TRUE;
    }
    return FALSE;
}

int main()
{
    ...
    ...
    int type = egl_show_messagebox("messagebox test", MB_OKCANCEL, my_touchinput);
    ...
}
```

## ► MessageBox Example.

### Example

```
void btn_callback(EGL_HANDLE h, int event)
{
    if(event == BTN_CLICKED)
    {
        debugprintf("button clicked.");
    }
}

BOOL my_touchinput(EGL_MSG* pMsg)
{
    BOOL  touchdown;
    EGL_POINT  touch_pt;
    if(process_touch(&touchdown, &touch_pt))
    {
        if(touchdown)
            pMsg -> msgID = EGL_MSG_TOUCHED;
        else
            pMsg -> msgID = EGL_MSG_UNTOUCHED;
        pMsg -> param.point.x = touch_pt.x;
        pMsg -> param.point.y = touch_pt.y;
        return TRUE;
    }
    return FALSE;
}

extern BOOL process_touch(BOOL* touchdown, EGL_POINT* pPoint);

int main()
{
    ...
    EGL_POINT touch_pt;
    BOOL touchdown = FALSE;
    EGL_HANDLE hWin;
    EGL_HANDLE btn;
    egl_init();
    btn = egl_create_button(100, 100, 100, 50, "Button Ex");
    egl_button_callback(btn, btn_callback);
}
```

```
egl_window_add_object(hWin, btn);
egl_window_show(hWin);
egl_draw();

int type = egl_show_messagebox("messagebox test", MB_OKCANCEL, my_touchinput);

while(1)
{
    if( process_touch( &touchdown, &touch_pt ) )
        egl_user_touch_input( touchdown, &touch_pt );

    egl_draw();
}
}
```

# EGL Font

Function	Description
<a href="#">egl_get_font</a>	Object가 사용하는 font 정보를 반환한다.
<a href="#">egl_set_font</a>	Object가 사용 할 font를 설정한다.
<a href="#">egl_font_set_bkmode</a>	Font의 배경색 사용 여부를 결정한다.
<a href="#">egl_font_get_bk_color</a>	Font의 배경색 정보를 가져온다.
<a href="#">egl_font_set_bk_color</a>	Font의 배경색을 설정한다.
<a href="#">egl_font_get_color</a>	Font의 색 정보를 가져온다.
<a href="#">egl_font_set_color</a>	Font의 색을 설정한다.
<a href="#">create_bitfont</a>	Font사용을 위해 Bit방식의 font를 생성한다.
<a href="#">release_bitfont</a>	Create한 bit방식의 font를 release한다.
<a href="#">create_bmpfont</a>	Font사용을 위해 이미지 font를 생성한다.
<a href="#">bmpfont_release</a>	Create한 이미지 font를 release한다.
<a href="#">draw_text</a>	문자를 출력한다.
<a href="#">draw_text_pivot</a>	문자를 좌/우 90도 회전하여 출력한다.
<a href="#">draw_text_len</a>	문자를 지정한 길이만큼 출력한다.
<a href="#">draw_text_in_box</a>	지정한 영역안에 문자를 출력한다.
<a href="#">text_width</a>	문자열의 길이를 확인한다.

## ▶ egl\_get\_font function

```
EGL_FONT* egl_get_font(  
    EGL_HANDLE h  
);
```

### Overview

Object가 사용하는 font 정보를 반환한다.

### Parameter

EGL\_HANDLE h                      Font 정보를 가져 올 object handle.

### Return Value

Object가 사용하고 있는 font 정보. (EGL\_FONT Struct)

### Example

```
EGL_HANDLE object;  
...  
...  
EGL_FONT* cur_font = egl_get_font(object);
```

## ▶ egl\_set\_font function

```
EGL_FONT* egl_set_font(  
    EGL_HANDLE h,  
    EGL_FONT* font  
);
```

### Overview

Object가 사용 할 font를 설정한다.

### Parameter

EGL_HANDLE h	Font를 설정할 object handle.
EGL_FONT* font	설정 할 font.

### Return Value

이전에 설정되어 있던 font 정보.

### Example

```
EGL_HANDLE object;  
EGL_FONT* font;  
...  
font = create_bitfont();  
egl_set_font(object , font);
```

## ▶ egl\_font\_set\_bkmode function

```
void egl_font_set_bkmode(  
    EGL_FONT* font,  
    int mode  
);
```

### Overview

Font의 배경색 사용 여부를 결정한다.

### Parameter

EGL_FONT* font	설정을 변경 할 font.
int mode	배경 색 사용 여부. TRUE or FALSE.

### Return Value

없음.

### Example

```
EGL_FONT* font;  
...  
font = create_bitfont();  
egl_set_font(object , font);  
egl_font_set_bkmode(font, TRUE);
```

## ▶ egl\_font\_get\_bk\_color function

```
EGL_COLOR egl_font_get_bk_color(  
    EGL_FONT* font  
);
```

### Overview

Font의 배경색 정보를 가져온다.

### Parameter

EGL\_FONT\* font                      배경색 정보를 가져 올 font..

### Return Value

현재 설정 되어 있는 font의 배경색.

### Example

```
EGL_FONT* font;  
EGL_COLOR font_bk_color;  
...  
font = create_bitfont();  
font_bk_color = egl_font_get_bk_color(font);
```

## ▶ egl\_font\_set\_bk\_color function

```
EGL_COLOR egl_font_set_bk_color(  
    EGL_FONT* font,  
    EGL_COLOR clr  
);
```

### Overview

Font의 배경색을 설정한다.

### Parameter

EGL_FONT* font	배경색을 설정 할 font.
EGL_COLOR clr	배경색. ( MAKE_COLORREF( ) 매크로를 사용하여 설정)

### Return Value

이전에 설정 되어 있던 font의 배경색.

### Example

```
EGL_FONT* font;  
EGL_COLOR font_bk_color;  
...  
font = create_bitfont();  
font_bk_color = egl_font_set_bk_color(font, MAKE_COLORREF(0, 255, 0));
```

## ▶ egl\_font\_get\_color function

```
EGL_COLOR egl_font_get_color(  
    EGL_FONT* font  
);
```

### Overview

Font의 색 정보를 가져온다.

### Parameter

EGL\_FONT\* font                      색 정보를 가져 올 font.

### Return Value

현재 설정 되어 있는 font 색.

### Example

```
EGL_FONT* font;  
EGL_COLOR font_color;  
...  
font = create_bitfont();  
font_color = egl_font_get_color(font);
```

## ▶ egl\_font\_set\_color function

```
EGL_COLOR egl_font_set_color(  
    EGL_FONT* font,  
    EGL_COLOR clr  
);
```

### Overview

Font의 색을 설정한다.

### Parameter

EGL_FONT* font	색을 설정 할 font.
EGL_COLOR clr	Font 색. ( MAKE_COLORREF() 매크로를 사용하여 설정)

### Return Value

이전에 설정 되어 있던 font 색.

### Example

```
EGL_FONT* font;  
EGL_COLOR font_color;  
...  
font = create_bitfont();  
font_color = egl_font_set_color(font, MAKE_COLORREF(0, 0, 255));
```

## ▶ create\_bitfont function

```
EGL_FONT* create_bitfont( void );
```

### Overview

Font사용을 위해 Bit방식의 font를 생성한다.

### Parameter

없음.

### Return Value

생성된 bit font 구조체 포인터.

### Example

```
EGL_FONT* bit_font;  
EGL_HANDLE object;  
...  
bit_font = create_bitfont();  
egl_set_font(object, bit_font );  
...  
draw_text(bit_font, 100, 100, "font test");
```

## ▶ **release\_bitfont** function

```
void release_bitfont(  
    EGL_FONT* pFont  
);
```

### Overview

Create한 bit방식의 font를 release한다.

### Parameter

EGL\_FONT\* pFont                      Release 대상 bit font.

### Return Value

없음.

### Example

```
EGL_FONT* bit_font;  
...  
bit_font = create_bitfont();  
...  
release_bitfont(bit_font);
```

## ▶ create\_bmpfont function

```
EGL_FONT* create_bmpfont(  
    const char *fname  
);
```

### Overview

Font사용을 위해 이미지 font를 생성한다.

### Parameter

const char \*fname                      이미지 font file name.

### Return Value

생성된 이미지 font 구조체 포인터.

### Example

```
EGL_FONT* bm_font;  
EGL_HANDLE object;  
...  
bm_font = create_bmpfont("font/batang_24.fnt");  
egl_set_font(object, bm_font);  
...  
draw_text(bm_font, 100, 100, "image font test");
```

## ▶ **bmfont\_release** function

```
void bmfont_release(  
    EGL_FONT* pFont  
);
```

### Overview

Create한 이미지 font를 release한다.

### Parameter

EGL\_FONT\* pFont                      Release 대상 이미지 font.

### Return Value

없음.

### Example

```
EGL_FONT* bm_font;  
...  
bm_font = create_bmpfont("font/batang_24.fnt");  
...  
bmfont_release(bm_font);
```

## ▶ draw\_text function

```
int draw_text(  
    EGL_FONT* pFont,  
    int x,  
    int y,  
    const char* text  
);
```

### Overview

문자를 출력한다.

### Parameter

EGL_FONT* pFont	문자 출력에 사용할 font.
int x	출력할 x좌표.
int y	출력할 y좌표.
const char* text	출력할 문자열.

### Return Value

출력한 문자열 갯수.

### Example

```
EGL_FONT* bit_font;  
...  
bit_font = create_bitfont();  
...  
draw_text(bit_font, 100, 100, "font test");
```

## ▶ draw\_text\_pivot function

```
int draw_text_pivot(  
    EGL_FONT* pFont,  
    int x,  
    int y,  
    const char* text,  
    int pivot  
);
```

### Overview

문자를 좌/우 90도 회전하여 출력한다.

### Parameter

EGL_FONT* pFont	문자 출력에 사용할 font.
int x	출력할 x좌표.
int y	출력할 y좌표.
const char* text	출력할 문자열.
int pivot	좌/우 90도 회전 여부. PIVOT_RIGHT / PIVOT_LEFT

### Return Value

출력한 문자열 갯수.

### Example

```
EGL_FONT* bit_font;  
...  
bit_font = create_bitfont();  
...  
draw_text_pivot(bit_font, 100, 100, "font test",PIVOT_RIGHT);
```

## ▶ draw\_text\_len function

```
void draw_text_len(  
    EGL_FONT* pFont,  
    int x,  
    int y,  
    const char* text,  
    int len  
);
```

### Overview

문자를 지정한 길이만큼 출력한다.

### Parameter

EGL_FONT* pFont	문자 출력에 사용할 font.
int x	출력할 x좌표.
int y	출력할 y좌표.
const char* text	출력할 문자열.
int len	출력할 문자 개수

### Return Value

없음.

### Example

```
EGL_FONT* bit_font;  
...  
bit_font = create_bitfont();  
...  
draw_text_len(bit_font, 100, 100, "font test",4);
```

## ▶ draw\_text\_in\_box function

```
void draw_text_in_box(  
    EGL_FONT* pFont,  
    EGL_RECT* pRect,  
    const char* text,  
    int align  
);
```

### Overview

지정한 영역 안에 문자를 출력한다.

### Parameter

EGL\_FONT\* pFont

문자 출력에 사용할 font.

EGL\_RECT\* pRect

출력할 영역.

```
typedef struct _tag_RECT{  
    int x;  
    int y;  
    int w;  
    int h;  
} EGL_RECT;
```

const char\* text

출력할 문자열.

int align

영역 안에서의 정렬.

```
EGL_ALIGN_LEFT,  
EGL_ALIGN_RIGHT,  
EGL_ALIGN_TOP,  
EGL_ALIGN_BOTTOM,  
EGL_ALIGN_CENTER,  
EGL_ALIGN_MULTILINE
```

### Return Value

없음.

## Example

```
EGL_FONT* bit_font;
EGL_RECT rect;
...
bit_font = egl_create_default_font();
rect.x = 100;
rect.y = 100;
rect.w = 200;
rect.h = 40;
draw_text_in_box(bit_font, &rect, "font test", EGL_ALIGN_CENTER);
```

## ▶ `text_width` function

```
int text_width(  
    EGL_FONT* font,  
    const char* str  
);
```

### Overview

문자열의 길이(pixel 단위)를 확인한다.

### Parameter

<code>EGL_FONT* font</code>	문자열에 사용 할 font. (font마다 한글자의 width가 다르기 때문에 지정해 주어야 된다.)
<code>const char* str</code>	Width값을 확인 할 문자열.

### Return Value

문자열 width.

### Example

```
EGL_FONT* bit_font;  
int font_width;  
...  
bit_font = egl_create_default_font();  
...  
font_width = text_width(bit_font, "font test");
```

# EGL Primitives

Function	Description
<a href="#">draw_line</a>	line을 그린다.
<a href="#">draw_hline</a>	가로 line을 그린다.
<a href="#">draw_vline</a>	세로 line을 그린다.
<a href="#">draw_thickline</a>	Thickline line을 그린다.
<a href="#">draw_rect</a>	사각형을 그린다.
<a href="#">draw_rectfill</a>	안이 채워진 사각형을 그린다.
<a href="#">draw_rectfill_gradient</a>	gradient효과가 들어간 사각형을 그린다.
<a href="#">draw_rectfill_h_gradient</a>	가로로 gradient효과가 들어간 사각형을 그린다.
<a href="#">draw_rectfill_v_gradient</a>	세로로 gradient효과가 들어간 사각형을 그린다.
<a href="#">draw_roundrect</a>	모서리가 둥근 사각형을 그린다.
<a href="#">draw_roundrectfill</a>	모서리가 둥근 안이 채워진 사각형을 그린다.
<a href="#">draw_arc</a>	호, 반원, 원을 그린다.
<a href="#">draw_pie</a>	파이 모양의 도형을 그린다.
<a href="#">draw_piefill</a>	안이 채워진 파이 모양의 도형을 그린다.
<a href="#">draw_ellipse</a>	타원을 그린다.
<a href="#">draw_ellipsefill</a>	안이 채워진 타원을 그린다.
<a href="#">draw_circle</a>	원을 그린다.
<a href="#">draw_circlefill</a>	안이 채워진 원을 그린다.

Function	Description
<a href="#">draw_bezier</a>	베이어 곡선을 그린다.
<a href="#">draw_polyline</a>	지정한 좌표를 잇는 선을 그린다.
<a href="#">draw_polygon</a>	지정한 좌표를 잇는 다각형을 그린다.
<a href="#">draw_polygonfill</a>	지정한 좌표를 잇는 안이 채워진 다각형을 그린다.

## ▶ draw\_line function

```
void draw_line(  
    int x,  
    int y,  
    int x2,  
    int y2,  
    EGL_COLOR c  
);
```

### Overview

(x, y)에서 (x2,y2)까지 line을 그린다.

### Parameter

int x	Line의 시작 x좌표.
int y	Line의 시작 y좌표.
int x2	Line의 끝 x좌표.
int y2	Line의 끝 y좌표.
EGL_COLOR c	Line의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_line(100, 100, 200, 200, MAKE_COLORREF(0, 0, 0));  
draw_line(100, 100, 100, 200, MAKE_COLORREF(255, 0, 0));
```

## ▶ draw\_hline function

```
void draw_hline(  
    int x,  
    int y,  
    int x2,  
    EGL_COLOR c  
);
```

### Overview

(x, y)에서 (x2, y)까지 가로 line을 그린다.

### Parameter

int x	Line의 시작 x좌표.
int y	Line의 시작과 끝 y좌표.
int x2	Line의 끝 x좌표.
EGL_COLOR c	Line의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_hline(100, 100, 200, MAKE_COLORREF(0, 0, 0));  
draw_hline(100, 200, 200, MAKE_COLORREF(0, 0, 255));
```

## ▶ draw\_vline function

```
void draw_vline(  
    int x,  
    int y,  
    int y2,  
    EGL_COLOR c  
);
```

### Overview

(x, y)에서 (x, y2)까지 세로 line을 그린다.

### Parameter

int x	Line의 시작과 끝 x좌표.
int y	Line의 시작 y좌표.
int y2	Line의 끝 y좌표.
EGL_COLOR c	Line의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_vline(100, 100, 200, MAKE_COLORREF(0, 0, 0));  
draw_vline(200, 100, 200, MAKE_COLORREF(0, 255, 0));
```

## ▶ draw\_thickline function

```
void draw_thickline(  
    int x1,  
    int y1,  
    int x2,  
    int y2,  
    U8 width,  
    EGL_COLOR color  
);
```

### Overview

(x1, y1)에서 (x2,y2)까지 설정한 두께의 line을 그린다.

### Parameter

int x1	Thick line의 시작 x좌표.
int y1	Thick line의 시작 y좌표.
int x2	Thick line의 끝 x좌표.
int y2	Thick line의 끝 y좌표.
U8 width	Thick line의 두께.
EGL_COLOR color	Thick line의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_thickline(100, 130, 200, 130, 2, MAKE_COLORREF(0, 0, 0));  
draw_thickline(100, 200, 100, 300, 3, MAKE_COLORREF(255, 0, 0));
```

## ▶ draw\_rect function

```
void draw_rect(  
    int x,  
    int y,  
    int w,  
    int h,  
    EGL_COLOR c  
);
```

### Overview

(x, y)부터 가로크기가 w, 세로크기가 h인 사각형을 그린다.

### Parameter

int x	사각형의 시작 x좌표.
int y	사각형의 시작 y좌표.
int w	사각형의 가로 크기.
int h	사각형의 세로 크기.
EGL_COLOR c	사각형 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_rect(100, 100, 200, 200, MAKE_COLORREF(0, 0, 0));  
draw_rect(150, 150, 100, 100, MAKE_COLORREF(0, 0, 255));
```

## ▶ draw\_rectfill function

```
void draw_rectfill(  
    int x,  
    int y,  
    int w,  
    int h,  
    EGL_COLOR c  
);
```

### Overview

(x, y)부터 가로크기가 w, 세로크기가 h인 안이 채워진 사각형을 그린다.

### Parameter

int x	사각형의 시작 x좌표.
int y	사각형의 시작 y좌표.
int w	사각형의 가로 크기.
int h	사각형의 세로 크기.
EGL_COLOR c	사각형 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_rectfill(100, 100, 50, 50, MAKE_COLORREF(0, 255, 0));  
draw_rectfill(200, 200, 50, 50, MAKE_COLORREF(0, 0, 255));
```

## ▶ draw\_rectfill\_gradient function

```
void draw_rectfill_gradient(  
    int x,  
    int y,  
    int w,  
    int h,  
    EGL_COLOR startcolor,  
    EGL_COLOR endcolor,  
    BOOL bVertical  
);
```

### Overview

(x, y)부터 가로크기가 w, 세로크기가 h인 gradient 사각형을 그린다.

### Parameter

int x	사각형의 시작 x좌표.
int y	사각형의 시작 y좌표.
int w	사각형의 가로 크기.
int h	사각형의 세로 크기.
EGL_COLOR startcolor	Gradient 시작 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)
EGL_COLOR endcolor	Gradient 끝 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)
BOOL bVertical	Gradient 방향. TRUE == vertical, FALSE == horizontal.

### Return Value

없음.

### Example

```
draw_rectfill_gradient(100,100,200,50,MAKE_COLORREF(0,255,0),MAKE_COLORREF(255,255,255),FALSE);  
draw_rectfill_gradient(100,200,50,200,MAKE_COLORREF(255,255,255),MAKE_COLORREF(0,0,255),TRUE);
```

## ▶ draw\_rectfill\_h\_gradient function

```
void draw_rectfill_h_gradient(  
    int x,  
    int y,  
    int w,  
    int h,  
    EGL_COLOR startcolor,  
    EGL_COLOR endcolor  
);
```

### Overview

(x, y)부터 가로크기가 w, 세로크기가 h인 가로방향 gradient 사각형을 그린다.

### Parameter

int x	사각형의 시작 x좌표.
int y	사각형의 시작 y좌표.
int w	사각형의 가로 크기.
int h	사각형의 세로 크기.
EGL_COLOR startcolor	gradient 시작 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)
EGL_COLOR endcolor	gradient 끝 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_rectfill_h_gradient(50,50,200,50,MAKE_COLORREF(255,255,255),MAKE_COLORREF(255,0,0));
```

## ▶ draw\_rectfill\_v\_gradient function

```
void draw_rectfill_v_gradient(  
    int x,  
    int y,  
    int w,  
    int h,  
    EGL_COLOR startcolor,  
    EGL_COLOR endcolor  
);
```

### Overview

(x, y)부터 가로크기가 w, 세로크기가 h인 세로방향 gradient 사각형을 그린다.

### Parameter

int x	사각형의 시작 x좌표.
int y	사각형의 시작 y좌표.
int w	사각형의 가로 크기.
int h	사각형의 세로 크기.
EGL_COLOR startcolor	gradient 시작 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)
EGL_COLOR endcolor	gradient 끝 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_rectfill_v_gradient(50,50,200,50,MAKE_COLORREF(255,255,255),MAKE_COLORREF(255,0,0));
```

## ▶ draw\_roundrect function

```
void draw_roundrect(  
    int x0,  
    int y0,  
    int w,  
    int h,  
    int corner,  
    EGL_COLOR c  
);
```

### Overview

(x0, y0)부터 가로크기가 w, 세로크기가 h인 둥근 사각형을 그린다.

### Parameter

int x0	둥근 사각형의 시작 x좌표.
int y0	둥근 사각형의 시작 y좌표.
int w	둥근 사각형의 가로 크기.
int h	둥근 사각형의 세로 크기.
int corner	모서리 둥근 정도.
EGL_COLOR c	둥근 사각형 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_roundrect(100, 100, 100, 100, 10, MAKE_COLORREF(0, 0, 0));  
draw_roundrect(400, 200, 200, 200, 20, MAKE_COLORREF(0, 255, 0));
```

## ▶ draw\_roundrectfill function

```
void draw_roundrectfill(  
    int x0,  
    int y0,  
    int w,  
    int h  
    int corner,  
    EGL_COLOR c  
);
```

### Overview

(x0, y0)부터 가로크기가 w, 세로크기가 h인 안이 채워진 둥근 사각형을 그린다.

### Parameter

int x0	둥근 사각형의 시작 x좌표.
int y0	둥근 사각형의 시작 y좌표.
int w	둥근 사각형의 가로 크기.
int h	둥근 사각형의 세로 크기.
int corner	모서리 둥근 정도.
EGL_COLOR c	둥근 사각형 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_roundrectfill(100, 100, 100, 100, 10, MAKE_COLORREF(255, 0, 0));  
draw_roundrectfill(300, 200, 200, 200, 15, MAKE_COLOEREF(0, 255, 0));
```

## ▶ draw\_arc function

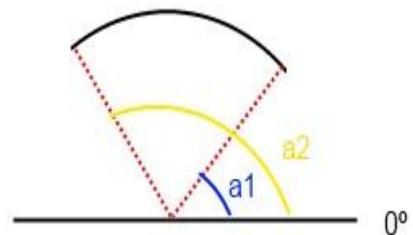
```
void draw_arc(  
    int x,  
    int y,  
    int rx,  
    int ry,  
    int a1,  
    int a2,  
    EGL_COLOR c  
);
```

### Overview

(x, y)를 원점으로 하고, 가로 반지름이 rx, 세로 반지름이 ry인 곡면을 그린다.  
a1을 시작 각도로 하여 각 a2까지 곡면을 그린다.

### Parameter

int x	원점의 x좌표.
int y	원점의 y좌표.
int rx	곡면의 가로 반지름.
int ry	곡면의 세로 반지름.
int a1	곡면의 시작 각도.
int a2	곡면의 끝 각도.
EGL_COLOR c	곡면의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)



### Return Value

없음.

### Example

```
draw_arc(100, 100, 30, 30, 0, 90, MAKE_COLORREF(255, 0, 0));  
draw_arc(200, 200, 40, 30, 90, 180, MAKE_COLORREF(0, 0, 255));
```

## ▶ draw\_pie function

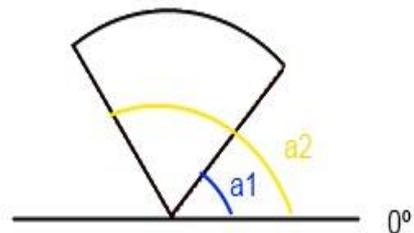
```
draw_pie(  
    int x,  
    int y,  
    int rx,  
    int ry,  
    int a1,  
    int a2,  
    EGL_COLOR c  
);
```

### Overview

(x, y)를 원점으로 하고, 가로 반지름이 rx, 세로 반지름이 ry인 파이모양을 그린다.  
a1을 시작 각도로 하여 각 a2까지 그린다.

### Parameter

int x	원점의 x좌표.
int y	원점의 y좌표.
int rx	파이의 가로 반지름.
int ry	파이의 세로 반지름.
int a1	파이의 시작 각도.
int a2	파이의 끝 각도.
EGL_COLOR c	파이의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)



### Return Value

없음.

### Example

```
draw_pie(100, 100, 50, 50, 45, 90, MAKE_COLORREF(0, 0, 255));  
draw_pie(200, 100, 60, 60, 90, 180, MAKE_COLORREF(255, 0, 0));
```

## ▶ draw\_piefill function

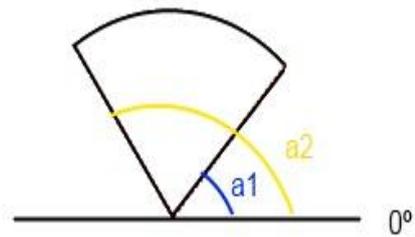
```
void draw_piefill(  
    int x,  
    int y,  
    int rx,  
    int ry,  
    int a1,  
    int a2,  
    EGL_COLOR c  
);
```

### Overview

(x, y)를 원점으로 하고, 가로 반지름이 rx, 세로 반지름이 ry인 채워진 파이모양을 그린다.  
a1을 시작 각도로 하여 각 a2까지 그린다.

### Parameter

int x	원점의 x좌표.
int y	원점의 y좌표.
int rx	파이의 가로 반지름.
int ry	파이의 세로 반지름.
int a1	파이의 시작 각도.
int a2	파이의 끝 각도.
EGL_COLOR c	파이의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)



### Return Value

없음.

### Example

```
draw_pie(100, 100, 50, 50, 45, 90, MAKE_COLORREF(0, 0, 255));  
draw_pie(200, 100, 60, 60, 90, 180, MAKE_COLORREF(255, 0, 0));
```

## ▶ draw\_ellipse function

```
void draw_ellipse(  
    int x,  
    int y,  
    int rx,  
    int ry,  
    EGL_COLOR c  
);
```

### Overview

(x, y)를 원점으로 하고, 가로 반지름이 rx, 세로 반지름이 ry인 타원을 그린다.

### Parameter

int x	원점의 x좌표.
int y	원점의 y좌표.
int rx,	가로 반지름.
int ry	세로 반지름.
EGL_COLOR c	타원의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_ellipse(100, 100, 50, 100, MAKE_COLORREF(0, 0, 255));  
draw_ellipse(300, 200, 100, 50, MAKE_COLORREF(255, 0, 0));
```

## ▶ draw\_ellipsefill function

```
void draw_ellipsefill(  
    int x,  
    int y,  
    int rx,  
    int ry,  
    EGL_COLOR c  
);
```

### Overview

(x, y)를 원점으로 하고, 가로 반지름이 rx, 세로 반지름이 ry인 채워진 타원을 그린다.

### Parameter

int x	원점의 x좌표.
int y	원점의 y좌표.
int rx,	가로 반지름.
int ry	세로 반지름.
EGL_COLOR c	타원의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_ellipsefill(100, 100, 50, 100, MAKE_COLORREF(0, 0, 255));  
draw_ellipsefill(300, 200, 100, 50, MAKE_COLORREF(255, 0, 255));
```

## ▶ draw\_circle function

```
void draw_circle(  
    int x,  
    int y,  
    int r,  
    EGL_COLOR c  
);
```

### Overview

원점이 (x, y)고 반지름이 r인 원을 그린다.

### Parameter

int x	원점의 x좌표.
int y	원점의 y좌표.
int r	반지름.
EGL_COLOR c	원의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_circle(100, 100, 50, MAKE_COLORREF(255, 0, 255));  
draw_circle(300, 200, 100, MAKE_COLORREF(0, 0, 255));
```

## ▶ draw\_circlefill function

```
void draw_circlefill(  
    int x,  
    int y,  
    int r,  
    EGL_COLOR c  
);
```

### Overview

원점이 (x, y)고 반지름이 r인 채워진 원을 그린다.

### Parameter

int x	원점의 x좌표.
int y	원점의 y좌표.
int r	반지름.
EGL_COLOR c	원의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
draw_circle(100, 100, 70, MAKE_COLORREF(0, 0, 255));  
draw_circle(300, 200, 50, MAKE_COLORREF(0, 255, 0));
```

## ▶ draw\_bezier function

```
void draw_bezier(  
    EGL_POINT* pts,  
    int n,  
    int s,  
    EGL_COLOR c  
);
```

### Overview

Bezier curve를 그린다.

### Parameter

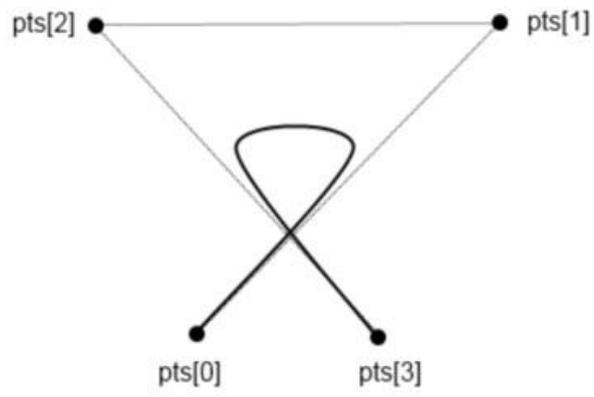
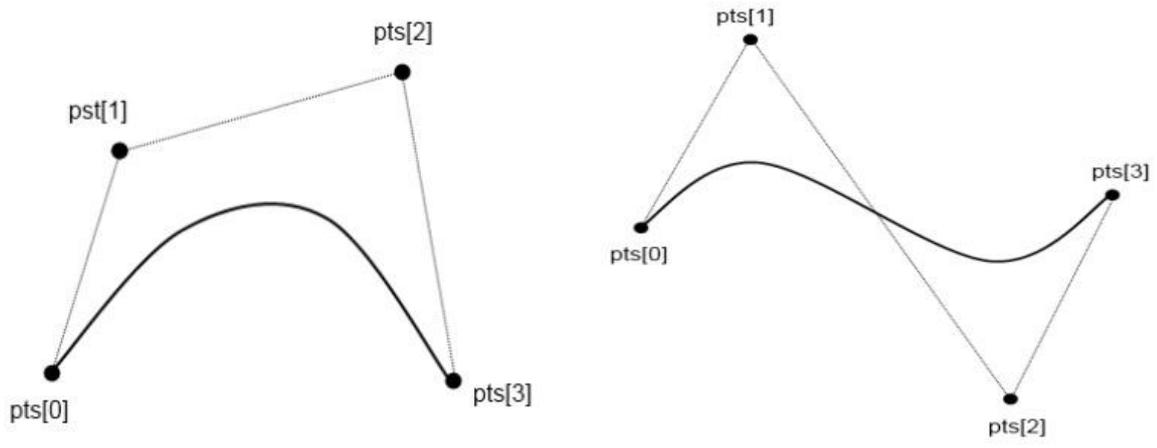
EGL_POINT* pts	좌표 배열.
int n	좌표 개수.
int s	point 개수. 좌표와 좌표사이에 몇 개의 point를 draw할지 결정.
EGL_COLOR c	curve의 color.

### Return Value

없음.

### Example

```
EGL_POINT* pts[4];  
pts[0].x = 100; pts[0].y = 200;  
pts[1].x = 200; pts[1].y = 50;  
pts[2].x = 300; pts[2].y = 50;  
pts[3].x = 400; pts[3].y = 200;  
  
draw_bezier(pts, 4, 20, MAKE_COLORREF(255, 0, 255));
```



< Bezier curve image >

## ▶ draw\_polyline function

```
void draw_polyline(  
    EGL_POINT* p,  
    int n,  
    EGL_COLOR c  
);
```

### Overview

지정한 좌표를 잇는 선을 그린다.

### Parameter

EGL_POINT* p	좌표 배열.
int n	좌표 개수.
EGL_COLOR c	선의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
EGL_POINT p[3];  
p[0].x = 100; p[0].y = 100;  
p[1].x = 130; p[1].y = 100;  
p[2].x = 150; p[2].y = 150;  
draw_polyline(p, 3, MAKE_COLORREF(0, 0, 0));
```

## ▶ draw\_polygon function

```
void draw_polygon(  
    EGL_POINT* ptable,  
    int cnt,  
    EGL_COLOR c  
);
```

### Overview

지정한 좌표를 잇는 다각형을 그린다.

### Parameter

EGL_POINT* p	좌표 배열.
int n	좌표 개수.
EGL_COLOR c	다각형의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
EGL_POINT ptable[5];  
ptable[0].x = 100; ptable[0].y = 100;  
ptable[1].x = 150; ptable[1].y = 50;  
ptable[2].x = 200; ptable[2].y = 100;  
ptable[3].x = 135; ptable[3].y = 150;  
ptable[4].x = 115; ptable[4].y = 150;  
draw_polygon(ptable, 5, MAKE_COLORREF(0, 0, 255));
```

## ▶ draw\_polygonfill function

```
void draw_polygonfill(  
    EGL_POINT* ptable,  
    int cnt,  
    EGL_COLOR c  
);
```

### Overview

지정한 좌표를 잇는 채워진 다각형을 그린다.

### Parameter

EGL_POINT* p	좌표 배열.
int n	좌표 개수.
EGL_COLOR c	다각형의 color. (MAKE_COLORREF(r,g,b) 매크로를 사용하여 설정)

### Return Value

없음.

### Example

```
EGL_POINT ptable[3];  
ptable[0].x = 100; ptable[0].y = 100;  
ptable[1].x = 150; ptable[1].y = 150;  
ptable[2].x = 100; ptable[2].y = 150;  
draw_polygon(ptable, 3, MAKE_COLORREF(0, 255, 0));
```

# EGL etc.

Function	Description
<a href="#">egl_init</a>	egl 초기화 함수.
<a href="#">egl_show_object</a>	Object를 draw 여부를 결정한다.
<a href="#">egl_object_set_redraw</a>	Object redraw할지를 결정한다.

Define	
EGL_MSG_ID	<pre>typedef enum enumMSG {     EGL_MSG_DRAW = 0,     EGL_MSG_DELETE,     EGL_MSG_FOCUS,     EGL_MSG_UNFOCUS,     EGL_MSG_KEY_UP,     EGL_MSG_KEY_DOWN,     EGL_MSG_TOUCHED,     EGL_MSG_UNTOUCHED,     EGL_MSG_MOVE,     EGL_MSG_TIMETICK } EGL_MSG_ID;</pre>
EGL_MSG	<pre>typedef struct _tagMessage {     EGL_MSGID msgID;     EGL_HANDLE hObj;     EGL_HANDLE hWin;     Union     {         EGL_POINT point;         U32 key;     } param; } EGL_MSG;</pre>

## ▶ egl\_init function

```
BOOL egl_init( void );
```

### Overview

egl 초기화를 수행한다. embedded graphic library를 사용하기 위해서 반드시 호출해주어야 한다.

### Parameter

없음.

### Return Value

TRUE or FALSE

## ▶ egl\_show\_object function

```
void egl_show_object(  
    EGL_HANDLE h,  
    BOOL bShow  
);
```

### Overview

Object를 draw 여부를 결정한다. bShow 값을 FALSE설정 하면 egl\_draw( ) 호출 시 해당 object는 draw되지 않는다.

### Parameter

EGL_HANDLE h	Draw 여부를 결정할 object handle.
BOOL bShow	Draw 여부. TRUE or FALSE.

### Return Value

없음.

## ▶ egl\_object\_set\_redraw function

```
void egl_object_set_redraw(  
    EGL_HANDLE handle  
);
```

### Overview

Object redraw할지를 결정한다. 참고로 library에서 object의 변화가 있을 경우 이 함수를 호출하지 않아도 redraw가 자동으로 설정된다.

### Parameter

EGL\_HANDLE handle                      Object를 redraw 설정 할 object handle.

### Return Value

없음.