

# SDL2 references for eForth Windows

version 1.0 - lundi 28 octobre 2024



## Autor

- Marc PETREMANN

## Contents

Autor.....	1
<b>SDL2.....</b>	<b>3</b>
CreateRenderer  window index flag -- render.....	3
CreateWindow  zstr x y w h fl -- win.....	3
DestroyRenderer  render -- fl.....	4
DestroyWindow  win -- fl.....	4
GetError  -- n.....	4
GetWindowFlags  window -- win-flag.....	4
GetWindowSize  windows *w *h -- fl.....	4
GetWindowSizeInPixels  windows *w *h -- fl.....	4
Init  n -- n.....	5
Quit  --.....	6
RenderClear  render -- 0 err.....	6
RenderPresent  render --.....	6
SDL2.dll  -- <name>.....	6
SDL_INIT_VIDEO  -- n.....	6
SetRenderDrawColor  renderer r g b a -- fl.....	6
SetWindowSize  window w h --.....	7

# SDL2

## CreateRenderer **window index flag -- render**

Create a 2D rendering context for a window.

Parameters:

- **window** the window where rendering is displayed
- **index** the index of the rendering driver to initialize, or -1 to initialize the first one supporting the requested flags
- **flags** 0, or one or more SDL\_RendererFlags OR'd together.

```
variable WIN0

z" My first window with SDL2"
    X0_SCREEN_POSITION Y0_SCREEN_POSITION SCREEN_WIDTH SCREEN_HEIGHT
    SDL_WINDOW_SHOWN    SDL.CreateWindow  WIN0 !

variable REN0

WIN0 @ -1 0 CreateRenderer  REN0 !
```

## CreateWindow **zstr x y w h fl -- win**

Create a window with the specified position, dimensions, and flags.

Parameters:

- **title** the title of the window, in UTF-8 encoding
- **x** the x position of the window, SDL\_WINDOWPOS\_CENTERED, or SDL\_WINDOWPOS\_UNDEFINED
- **y** the y position of the window, SDL\_WINDOWPOS\_CENTERED, or SDL\_WINDOWPOS\_UNDEFINED
- **w** the width of the window, in screen coordinates
- **h** the height of the window, in screen coordinates
- **flags** 0, or one or more SDL\_WindowFlags OR'd together

Returns **win** that was created or 0 on failure.

```
\ define size and position for SDL window
800 constant SCREEN_WIDTH
400 constant SCREEN_HEIGHT
```

```

200 constant X0_SCREEN_POSITION
50 constant Y0_SCREEN_POSITION

z" My first window with SDL2"
    X0_SCREEN_POSITION Y0_SCREEN_POSITION
    SCREEN_WIDTH SCREEN_HEIGHT
    SDL_WINDOW_SHOWN CreateWindow
    value WIN0

```

## DestroyRenderer **render -- fl**

Destroy the rendering context for a window and free associated textures.

```

\ free ressources, end renderer and window
: freeRessources ( -- )
    REN0 DestroyRenderer drop
    WIN0 DestroyWindow drop
    Quit
;

```

## DestroyWindow **win -- fl**

Destroy a window.

```

\ WIN0 must be declared by value and set by CreateWindow
WIN0 DestroyWindow

```

## GetError **-- n**

Retrieve a message about the last error that occurred on the current thread.

## GetWindowFlags **window -- win-flag**

Get the window flags.

## GetWindowSize **windows \*w \*h -- fl**

Get the size of a window's client area.

Parameters:

- **window** the window to query the width and height from.
- **w** a pointer filled in with the width of the window, in screen coordinates
- **h** a pointer filled in with the height of the window, in screen coordinates

## GetWindowSizeInPixels **windows \*w \*h -- fl**

Get the size of a window in pixels.

Parameters:

- **window** the window from which the drawable size should be queried
- **w** a pointer to variable for storing the width in pixels
- **h** a pointer to variable for storing the height in pixels

```
variable WIN.width
variable WIN.height
: draw ( -- )
    REN0 255 255 255 255 SetRenderDrawColor drop
    REN0 RenderClear drop
    REN0 RenderPresent drop
    WIN0 WIN.width WIN.height GetWindowSizeInPixels drop
;
draw
WIN.width @ . \ display: 800
WIN.height @ . \ display: 400
```

## Init **n -- n**

Initialize the SDL library.

n must be one of

SDL\_INIT\_TIMER \ timer subsystem

SDL\_INIT\_AUDIO \ audio subsystem

SDL\_INIT\_VIDEO \ video subsystem; automatically initializes the events subsystem

SDL\_INIT\_JOYSTICK \ joystick subsystem; automatically initializes the events subsystem

SDL\_INIT\_HAPTIC \ haptic (force feedback) subsystem

SDL\_INIT\_GAMECONTROLLER \ controller subsystem; automatically initializes the joystick subsystem

SDL\_INIT\_EVENTS \ events subsystem

SDL\_INIT\_SENSOR

Returns 0 on success or a negative error code on failure. Call **GetError** for more information.

```
\ Initialize SDL with error management
: SDL.init ( n -- )
    Init
    if
        ." SDL could not intialize! SDL_Error: " getError .
    then
;
;
```

```
SDL_INIT_VIDEO SDL.init
```

## Quit --

Clean up all initialized subsystems.

## RenderClear **renderer -- 0 |err**

Clear the current rendering target with the drawing color.

```
\ RENO is a value previously initialized with CreateRenderer
RENO RenderClear drop
```

## RenderPresent **renderer --**

Update the screen with any rendering performed since the previous call.

## SDL2.dll -- <name>

Entry point to SDL2.dll library

```
\ Destroy a window.
z" SDL_DestroyWindow"          1 SDL2.dll DestroyWindow ( window -- )
```

## SDL\_INIT\_VIDEO -- n

Constant. Tells the SDL that you want to initialize the video subsystem.

```
SDL_INIT_VIDEO SDL.Init
```

## SetRenderDrawColor **renderer r g b a -- fl**

Set the color used for drawing operations (Rect, Line and Clear)

Parameters

- **renderer** the rendering context
- **r** the red value used to draw on the rendering target
- **g** the green value used to draw on the rendering target
- **b** the blue value used to draw on the rendering target
- **a** the alpha value used to draw on the rendering target; usually **SDL\_ALPHA\_OPAQUE** (255). Use **SetRenderDrawBlendMode** to specify how the alpha channel is used

## SetWindowSize `window w h --`

Set the size of a window's client area.

Parameters:

- **window** the window to change
- **w** the width of the window in pixels, in screen coordinates, must be > 0
- **h** the height of the window in pixels, in screen coordinates, must be > 0

The window size in screen coordinates may differ from the size in pixels. Use **GetRendererOutputSize** to get the real client area size in pixels.

```
WIN0 400 200 SetWindowSize drop
```