

Lazy Foo' Productions

[SDL Forums](#) [SDL Tutorials](#) [Articles](#) [OpenGL Forums](#)



[News](#) [FAQs](#) [Games](#) [Contact](#) [Donations](#)

Hello SDL: Your First Graphics Window



Last Updated 2/8/15

This tutorial covers the first major stepping stone: getting a window to pop up.

Now that you have SDL set up, it's time to make a bare bones SDL graphics application that renders a quad on the screen.

```
//Using SDL and standard IO
#include <SDL.h>
#include <stdio.h>

//Screen dimension constants
const int SCREEN_WIDTH = 640;
const int SCREEN_HEIGHT = 480;
```

At the top of our source file we include SDL since we're going to need the SDL functions and datatypes to make any SDL code. We're also going to include C standard IO to print errors to the console. You're probably more used to using iostream, but I use printf in my applications because it's more thread safe. For these early applications, use what ever you are most comfortable with.

After including our headers, we declare the width and height of the window we're going to render to.

```
int main( int argc, char* args[] )
{
    //The window we'll be rendering to
    SDL_Window* window = NULL;

    //The surface contained by the window
    SDL_Surface* screenSurface = NULL;

    //Initialize SDL
    if( SDL_Init( SDL_INIT_VIDEO ) < 0 )
    {
        printf( "SDL could not initialize! SDL_Error: %s\n", SDL_GetError() );
    }
}
```

Here's the top of our main function. It's important that we have the arguments of the function be an integer followed by a char* array and the return type be an integer. Any other type of main function will cause an undefined reference to main. SDL requires this type of main so it is compatible with multiple platforms.

We then declare our SDL window which we will be creating in a little bit. Following that we have a screen SDL surface. A SDL surface is just a 2D image. A 2D image can be loaded from a file or it can be the image inside of a window. In this case it will be the image we see inside of the window on the screen.

After declaring our window and screen surface, we initialize SDL. You can't call any SDL functions without initializing SDL first. Since all we care about is using SDL's video subsystem, we will only be passing it the SDL_INIT_VIDEO flag.

When there's an error, SDL_Init returns -1. When there's an error we want to print to the console what happened, other wise the application will just flash for a second and then go away.

If you have never used printf before, it stands for print format. It prints the string in the first argument with the variables in the following arguments. When there's an error here, "SDL could not initialize! SDL_Error: " will be written to the console followed by the string returned by SDL_GetError. That %s is special formatting. %s means output a string from our variable list. Since SDL_GetError is the only argument, the string it returns will be added. SDL_GetError returns the latest error produced by an SDL function.

SDL_GetError is a very useful function. Whenever something goes wrong you need to know why. SDL_GetError will let you know if any errors happened inside of any SDL function.

```
else
{
    //Create window
    window = SDL_CreateWindow( "SDL Tutorial", SDL_WINDOWPOS_UNDEFINED, SDL_WINDOWPOS_UNDEFINED, SCREEN_WIDTH, SCREEN_HEIGHT, SDL_WINDOW_
    if( window == NULL )
    {
        printf( "Window could not be created! SDL_Error: %s\n", SDL_GetError() );
    }
}
```

If SDL initialized successfully, we'll want to create a window using SDL_CreateWindow. The first argument sets the window's caption or this part of the window:



The next two arguments define the x and y position the window is created in. Since we don't care where it is created, we just put SDL_WINDOWPOS_UNDEFINED for the x and y position.

The next two arguments define the window's width and height. The last argument are the creation flags. SDL_WINDOW_SHOWN makes sure the window is shown when it is created.

If there is an error, SDL_CreateWindow returns NULL. If there's no window, we want to print out the error to the console.

```
else
{
    //Get window surface
    screenSurface = SDL_GetWindowSurface( window );

    //Fill the surface white
    SDL_FillRect( screenSurface, NULL, SDL_MapRGB( screenSurface->format, 0xFF, 0xFF, 0xFF ) );

    //Update the surface
    SDL_UpdateWindowSurface( window );

    //Wait two seconds
    SDL_Delay( 2000 );
}
}
```

If the window was created successfully, we want to get the window's surface so we can draw to it. SDL_GetWindowSurface does just this.

To keep this tutorial simple, all we're going to do here is fill the window's surface white using SDL_FillRect. Don't worry too much about this function here. This tutorial is only concerned about getting a window to pop up.

An important thing to know about rendering is that just because you've drawn something to the screen surface doesn't mean you'll see it. After you've done all your drawing you need to update the window so it shows everything you drew. A call to SDL_UpdateWindowSurface will do this.

If all we do is create the window, fill it, and update it, all we're going to see is a window flash for a second and then close. To keep it from disappearing, we'll call SDL_Delay. SDL_Delay will wait for a given amount of milliseconds. A millisecond is 1/1000 of a second. This means the code above will make the window wait for 2000 1/1000 of a second or 2 seconds.

An important thing to note is that when SDL is delaying, it cannot accept input from the keyboard or mouse. Don't panic when you run this program and it doesn't respond. We haven't given it the code to handle the mouse and keyboard.

```
//Destroy window
SDL_DestroyWindow( window );

//Quit SDL subsystems
SDL_Quit();

return 0;
}
```

After the window has delayed there for 2 seconds, we'll destroy the window to free its memory. This will also take care of the SDL_Surface we got from it. After everything is deallocated, we quit SDL and return 0 to terminate the program.

Download the media and source code for this tutorial [here](#).

[Back to SDL Tutorials](#)

G+ 分享 14 Tweet

[SDL Forums](#) [SDL Tutorials](#) [Articles](#) [OpenGL Forums](#)



Fastest VPN for China

De-Free Truly Anonymous & Secure

[News](#)
[FAQs](#)
[Games](#)
[Contact](#)
[Donations](#)

