
Subject: Re: How to configure for SDL2 project?
Posted by [flim](#) on Wed, 19 May 2021 04:01:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

I test the nightly build today, I create a new package with SDL in MyApp, the package build successfully.

But when I build this code, the linker failed. Any idea?

```
() : Linking has failed
(): lld-link: error: undefined symbol: SDL_main
(): >>> referenced by c:\projects\sdl\src\main\windows\sdl_windows_main.c:175
(): >>>          SDL2main.lib(SDL_windows_main.obj):(main_getcmdline)
(): clang-11: error: linker command failed with exit code 1 (use -v to see invocation)
```

```
#ifdef _WIN32
#include <SDL.h>
#else
#include <SDL2/SDL.h>
#endif
```

```
#include <stdbool.h>
#include <stdio.h>
```

```
bool is_running = false;
```

```
SDL_Window* window = NULL;
SDL_Renderer* renderer = NULL;
```

```
bool initialize_window(void) {
    if (SDL_Init(SDL_INIT_EVERYTHING) != 0 ) {
        fprintf(stderr, "Error initializing SDL.\n");
        return false;
    }

    // Create a SDL Window
    window = SDL_CreateWindow(
        NULL,
        SDL_WINDOWPOS_CENTERED,
        SDL_WINDOWPOS_CENTERED,
        800,
        600,
        SDL_WINDOW_BORDERLESS
    );

    if (!window) {
        fprintf(stderr, "Error creating SDL window.\n");
    }
}
```

```

        return false;
    }

    // Create a SDL renderer
    renderer = SDL_CreateRenderer(window, -1, 0);
    if (!renderer) {
        fprintf(stderr, "Error creating SDL renderer.\n");
        return false;
    }

    return true;
}

void setup(void) {

}

void process_input(void) {
    SDL_Event event;
    SDL_PollEvent(&event);

    switch (event.type) {
    case SDL_QUIT:
        is_running = false;
        break;
    case SDL_KEYDOWN:
        if (event.key.keysym.sym == SDLK_ESCAPE)
            is_running = false;
        break;
    }
}

void update(void) {

}

void render(void) {
    SDL_SetRenderDrawColor(renderer, 255, 0, 0, 255);
    SDL_RenderClear(renderer);

    SDL_RenderPresent(renderer);
}

int main(void) {

    is_running = initialize_window();

    setup();
}

```

```
while (is_running) {  
    process_input();  
    update();  
    render();  
}  
  
return 0;  
}
```
