Source Code:

SDL2-2.0.3.zip - GPG signed
SDL2-2.0.3.tar.gz - GPG signed

Runtime Binaries:

Windows:
SDL2-2.0.3-win32-x86.zip (32-bit Windows)
SDL2-2.0.3-win32-x64.zip (64-bit Windows)

Mac OS X:
SDL2-2.0.3.dmg (Intel 10.5+)

Linux:
Please contact your distribution maintainer for updates.

Development Libraries:

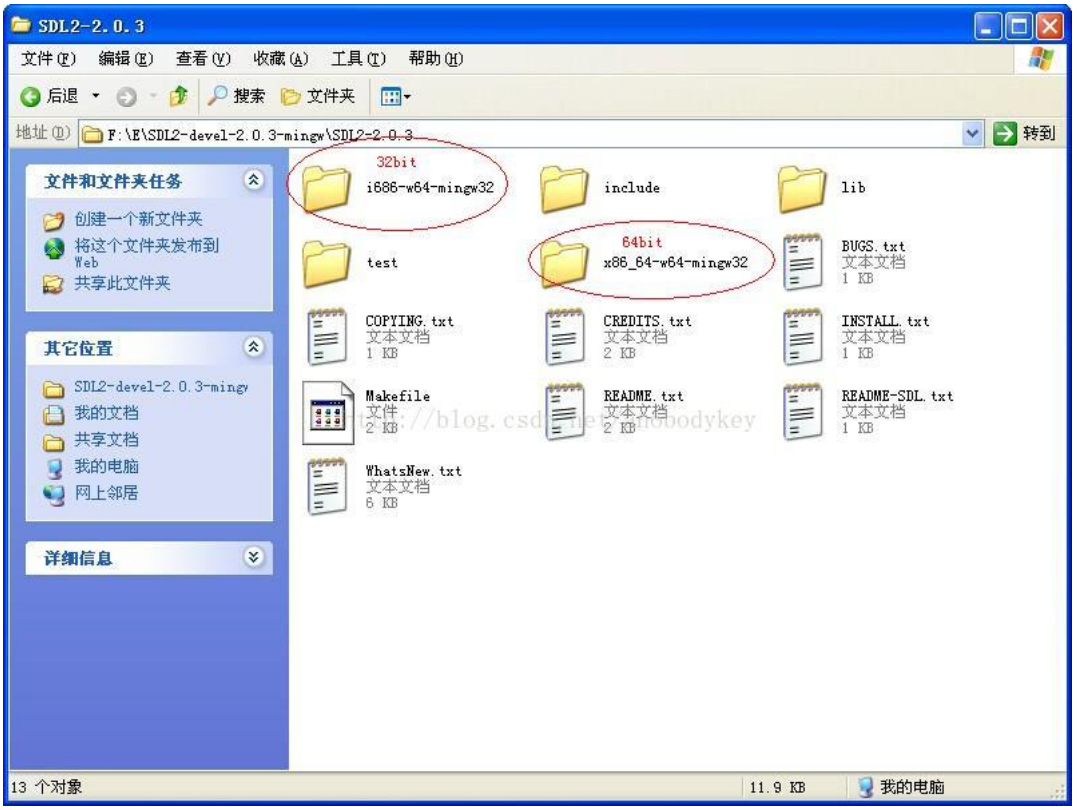
Windows:
SDL2-devel-2.0.3-VC.zip (Visual C++ 32/64-bit)
SDL2-devel-2.0.3-mingw.tar.gz (MinGW 32/64-bit)

Mac OS X:
SDL2-2.0.3.dmg (Intel 10.5+)

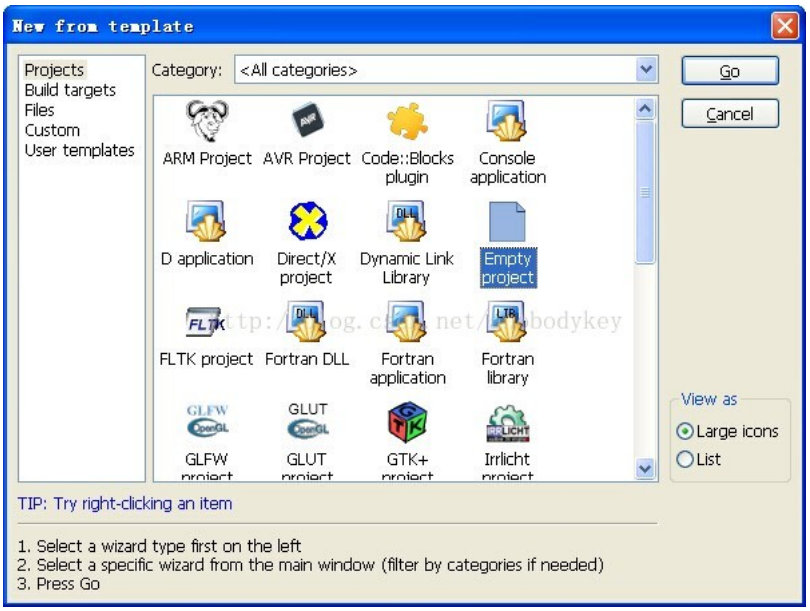
Linux:
Please contact your distribution maintainer for updates.

iOS & Android:
Projects for these platforms are included with the source.

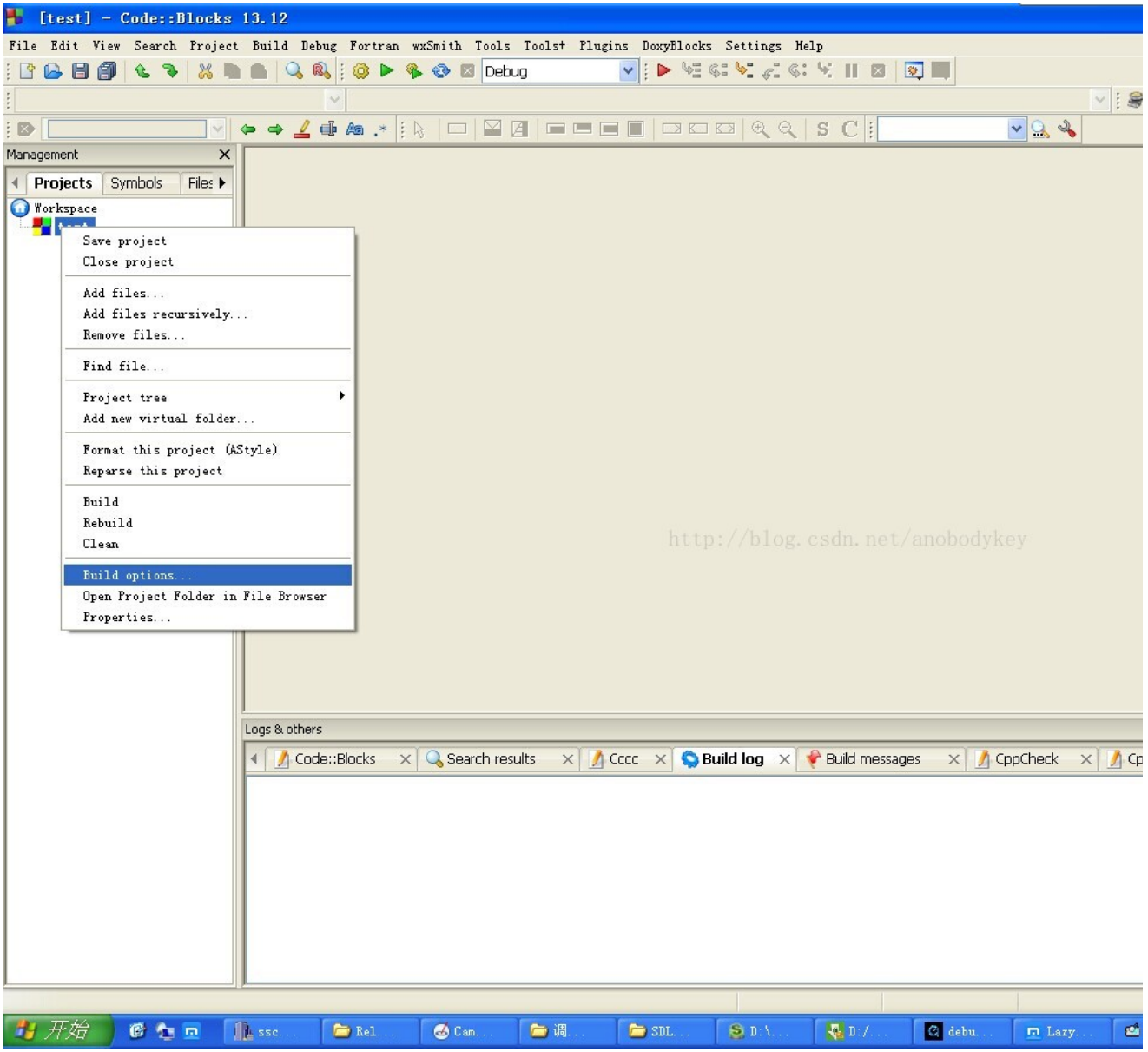
只需开发库就可以了，由于我的开发环境是CodeBlocks所以下载的是SDL2-devel-2.0.3-mingw.tar.gz，解压到本地，如下



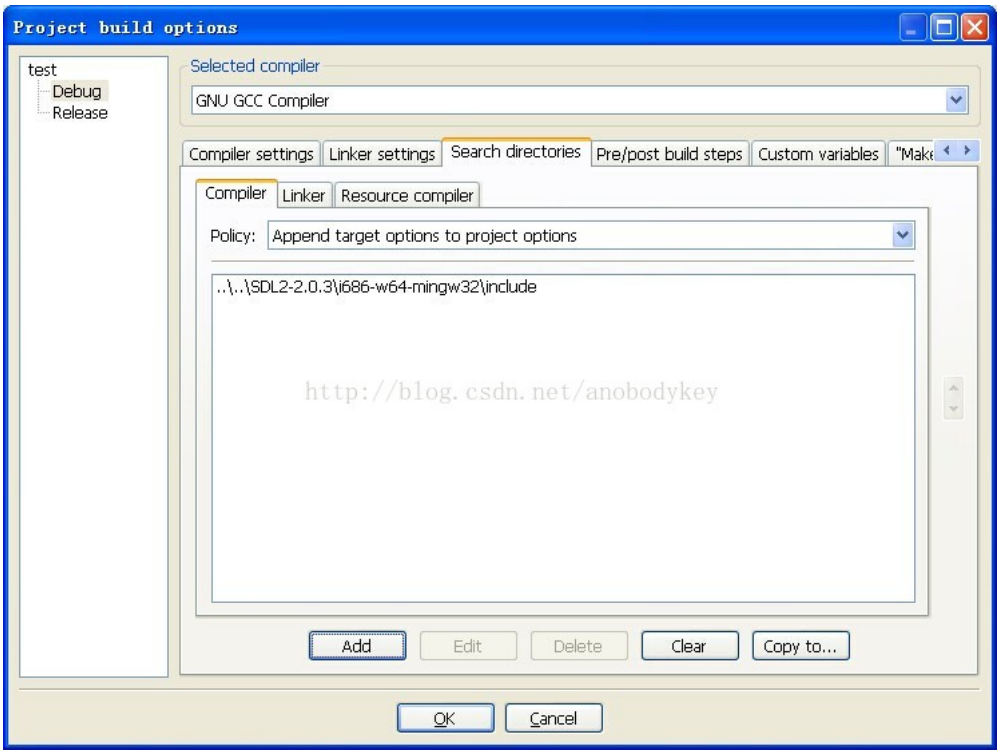
i686-w64-mingw32是32bit的，x86_64-w64-mingw32是64bit的，因此只需要前者就可以了。
打开CodeBlocks新建一个空白工程，



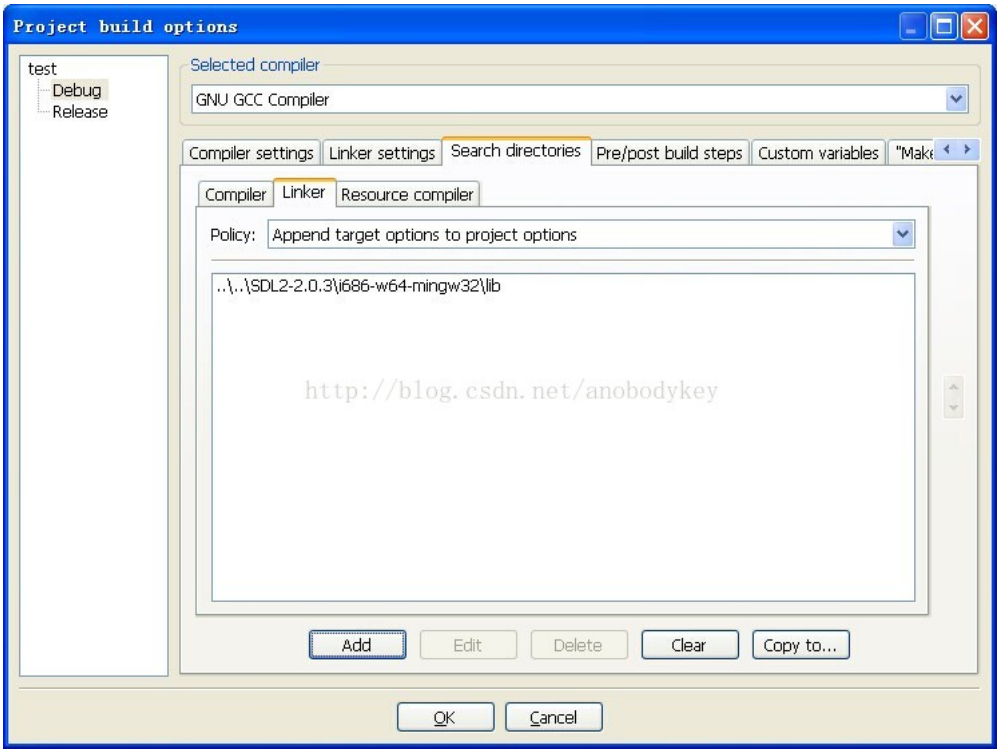
之后右击工程-》编译属性



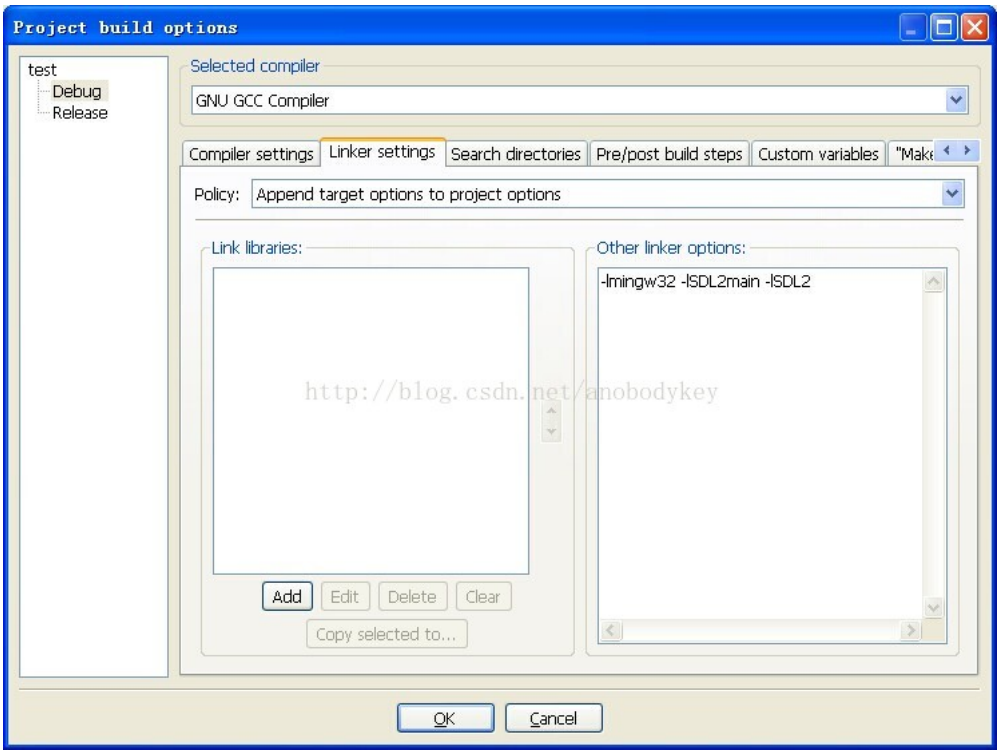
在选项中设置头文件和库文件的路径，在Search directories中，点击Add按钮，如下



以及

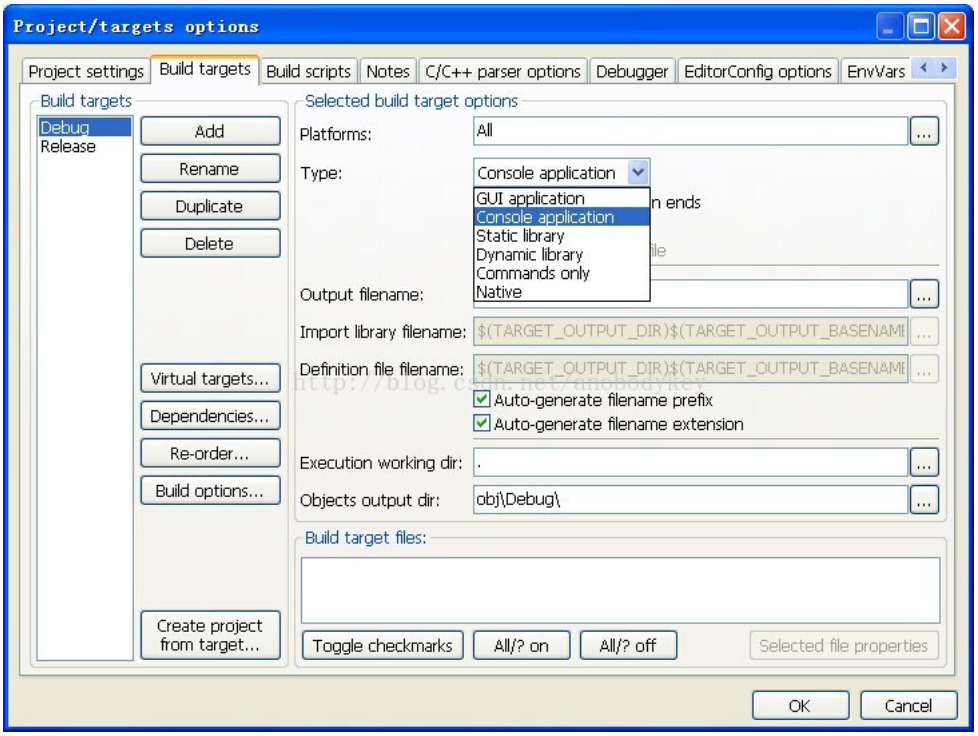


在连接器设置即Linker Settings中增加链接选项，如下



最后点击OK键确认。

右击工程-》属性-》Build Targets中设置目标类型为控制台应用，如下：



64位的库 会报link 失败使用 32位的库就好了

最后将i686-w64-mingw32\bin目录下的SDL2.dll文件拷贝到C:\WINDOWS\system32目录下，至此环境设置完毕，开写代码

[cpp] view plain copy

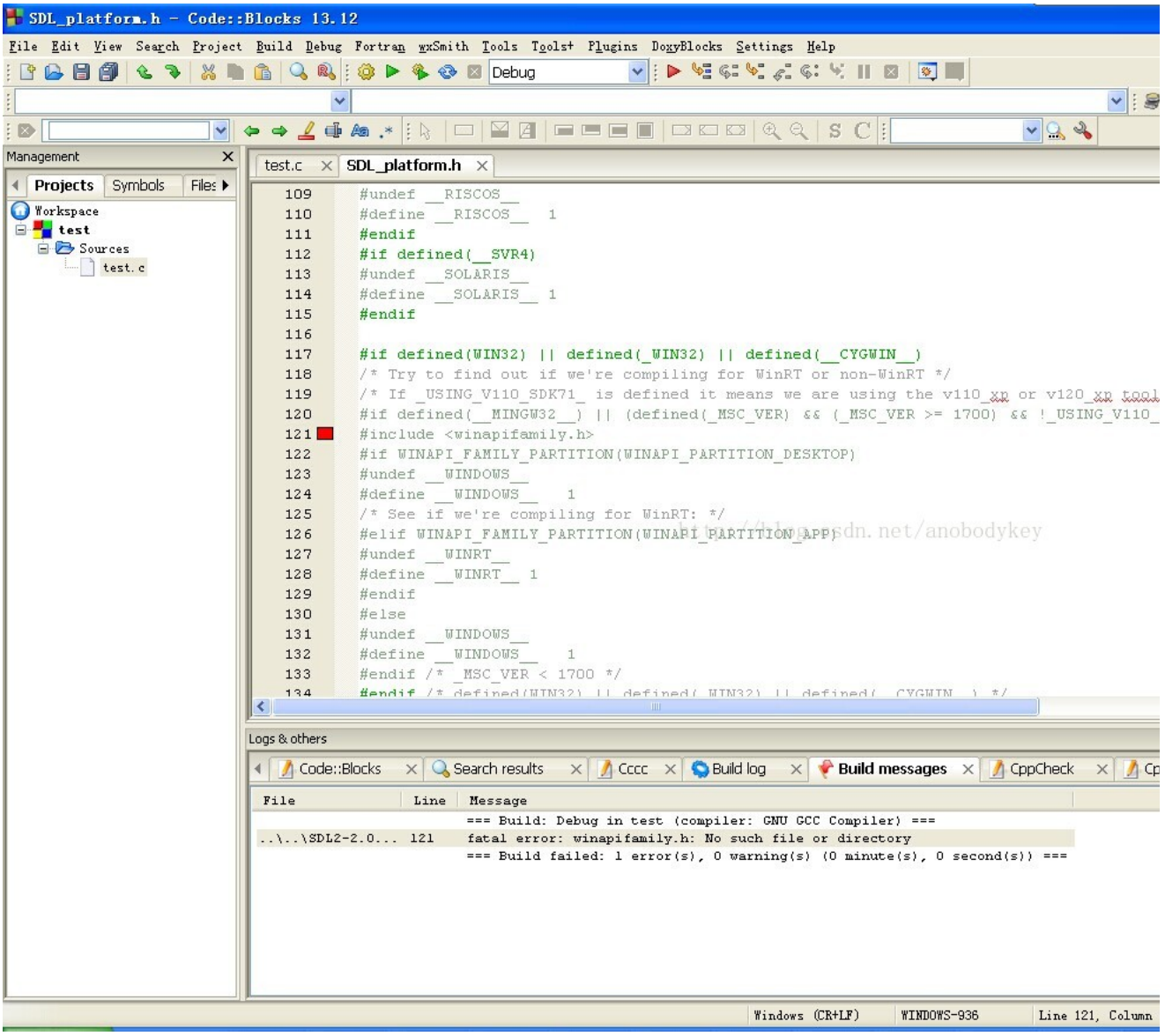
在CODE上查看代码片

派生到我的代码片

```
1. #include <SDL2/SDL.h> /*All SDL App's need this*/
2. #include <stdio.h>
3. int main()
4. {
```

```
5.     printf("Initializing SDL."); /* Initialize defaults, Video and Audio */
6.     if((SDL_Init(SDL_INIT_VIDEO|SDL_INIT_AUDIO)==-1))
7.     {
8.         printf("Could not initialize SDL: %s.", SDL_GetError());
9.         exit(-1);
10.    }
11.    printf("SDL initialized.");
12.    printf("Quiting SDL."); /* Shutdown all subsystems */
13.    SDL_Quit();
14.    printf("Quiting....");
15.    exit(0);
16. }
```

编译代码，出现fatal error: winapifamily.h: No such file or directory错误，找不到winapifamily.h文件，如图



网上查看了一哈说winapifamily.h是在win8中才有的头文件，而我的主机环境是windows XP，显然没得这个头文件，在SDL_platform.h中有如下解释

[cpp] view plain copy

在CODE上查看代码片

派生到我的代码片

```
1. /* Try to find out if we're compiling for WinRT or non-WinRT */
```

```
2. /* If _USING_V110_SDK71_ is defined it means we are using the v110_xp or v120_xp toolset. */
```

尝试查明我们编译是在WinRT还是non-WinRT，度娘了一下，WinRT只会在win8中使用，我们肯定是后者，如果_USING_V110_SDK71_被定义，意味着我们正在使用v110_xp或者v120_xp工具链。
修改代码在主程序中输出__MINGW32__、_MSC_VER、_USING_V110_SDK71这三个宏定义的值，

[cpp] view plain copy
在CODE上查看代码片
派生到我的代码片

```
1. printf("__MINGW32__:%d\n",__MINGW32__);
2. printf("_MSC_VER:%d\n",_MSC_VER);
3. printf("_USING_V110_SDK71_:%d\n",_USING_V110_SDK71_);
```

编译的结果提示后两者没有定义，如下

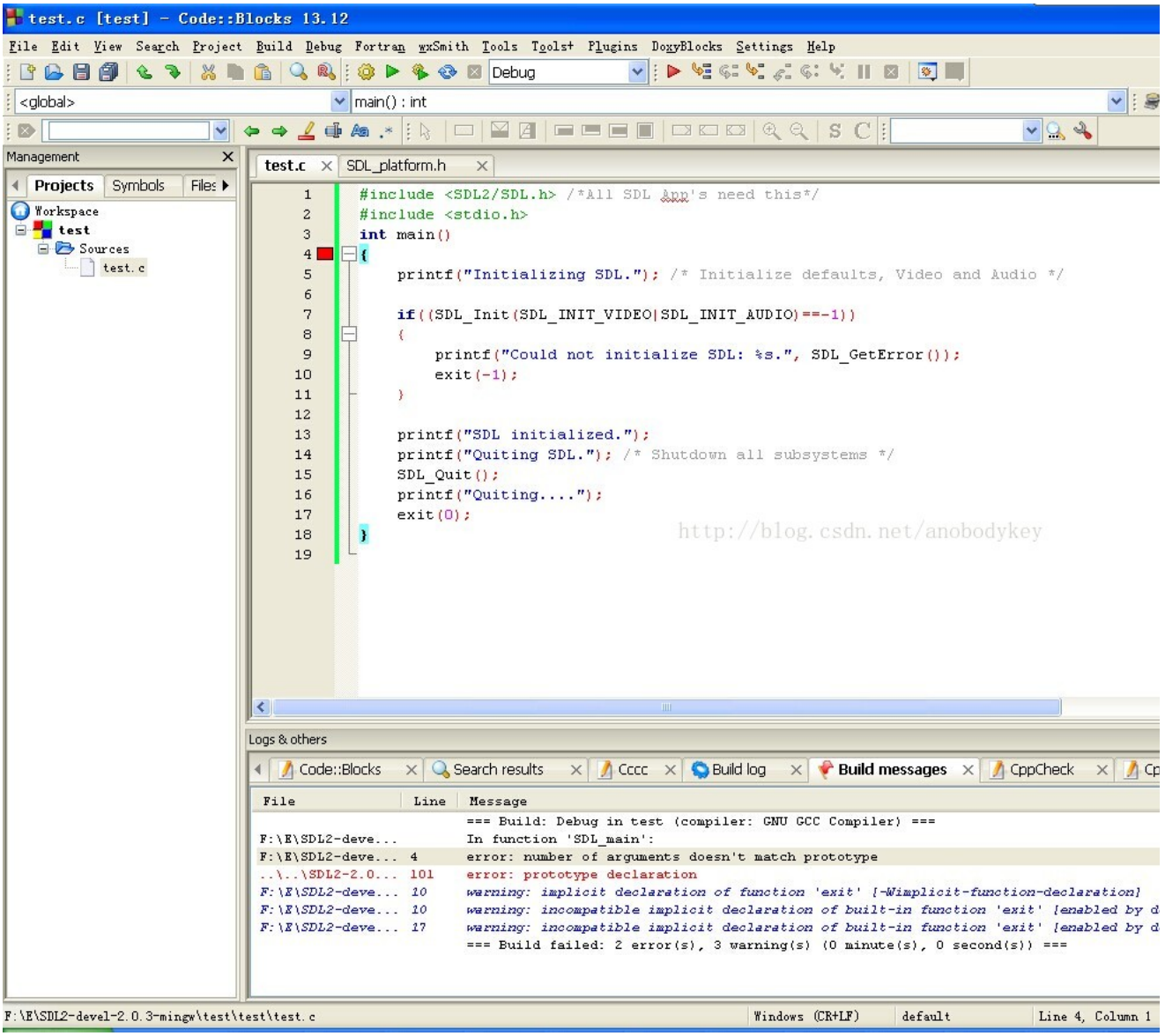
[html] view plain copy
在CODE上查看代码片
派生到我的代码片

```
1. ||=== Build: Debug in test (compiler: GNU GCC Compiler) ===|
2. F:\E\SDL2-devel-2.0.3-mingw\test\test\test.c||In function 'main':|
3. F:\E\SDL2-devel-2.0.3-mingw\test\test\test.c|13|warning: implicit declaration of function 'print' [-Wimplicit-function-declaration]|
4. F:\E\SDL2-devel-2.0.3-mingw\test\test\test.c|14|error: '_MSC_VER' undeclared (first use in this function)|
5. F:\E\SDL2-devel-2.0.3-mingw\test\test\test.c|14|note: each undeclared identifier is reported only once for each function it appears in|
6. F:\E\SDL2-devel-2.0.3-mingw\test\test\test.c|15|error: '_USING_V110_SDK71_' undeclared (first use in this function)|
7. F:\E\SDL2-devel-2.0.3-mingw\test\test\test.c|20|warning: implicit declaration of function 'exit' [-Wimplicit-function-declaration]|
8. F:\E\SDL2-devel-2.0.3-mingw\test\test\test.c|20|warning: incompatible implicit declaration of built-in function 'exit' [enabled by default]|
9. ||=== Build failed: 2 error(s), 3 warning(s) (0 minute(s), 0 second(s)) ===|
```

由于我们使用的编译器是mingw32,因此__MINGW32__是有定义的，并且值为1，但是我们并不需要winapifamily.h头文件，因此偷个懒，在SDL_platform.h中取消__MINGW32__的宏定义，如下

```
112  #if defined(__SVR4)
113  #undef __SOLARIS__
114  #define __SOLARIS__ 1
115  #endif
116  #undef __MINGW32__
117  #if defined(WIN32) || defined(_WIN32) || defined(__CYGWIN__)
118  /* Try to find out if we're compiling for WinRT or non-WinRT */
119  /* If _USING_V110_SDK71_ is defined it means we are using the v110_xp or v120_xp
120  #if defined(__MINGW32__) || (defined(_MSC_VER) && (_MSC_VER >= 1700) && !_USING_V
121  #include <winapifamily.h>
122  #if WINAPI_FAMILY_PARTITION(WINAPI_PARTITION_DESKTOP)
123  #undef __WINDOWS__
124  #define __WINDOWS__ 1
125  /* See if we're compiling for WinRT: */
126  #elif WINAPI_FAMILY_PARTITION(WINAPI_PARTITION_APP)
127  #undef __WINRT__
128  #define __WINRT__ 1
129  #endif
130  #else
131  #undef __WINDOWS__
132  #define __WINDOWS__    1
133  #endif /* _MSC_VER < 1700 */
134  #endif /* defined(WIN32) || defined(_WIN32) || defined(__CYGWIN__) */
135
```


再次编译代码，发现没有之前的错误， 转而提示error: number of arguments doesn't match prototype



这是因为在SDL中重定义了main函数，在SDL_main.h中，

[cpp] view plain copy

在CODE上查看代码片

派生到我的代码片

```
#if defined(SDL_MAIN_NEEDED) || defined(SDL_MAIN_AVAILABLE)
#define main    SDL_main
#endif

/**
 *  The prototype for the application's main() function
 */
extern C_LINKAGE int SDL_main(int argc, char *argv[]);
```

因此修改我们的主函数入口加上参数即可，如下

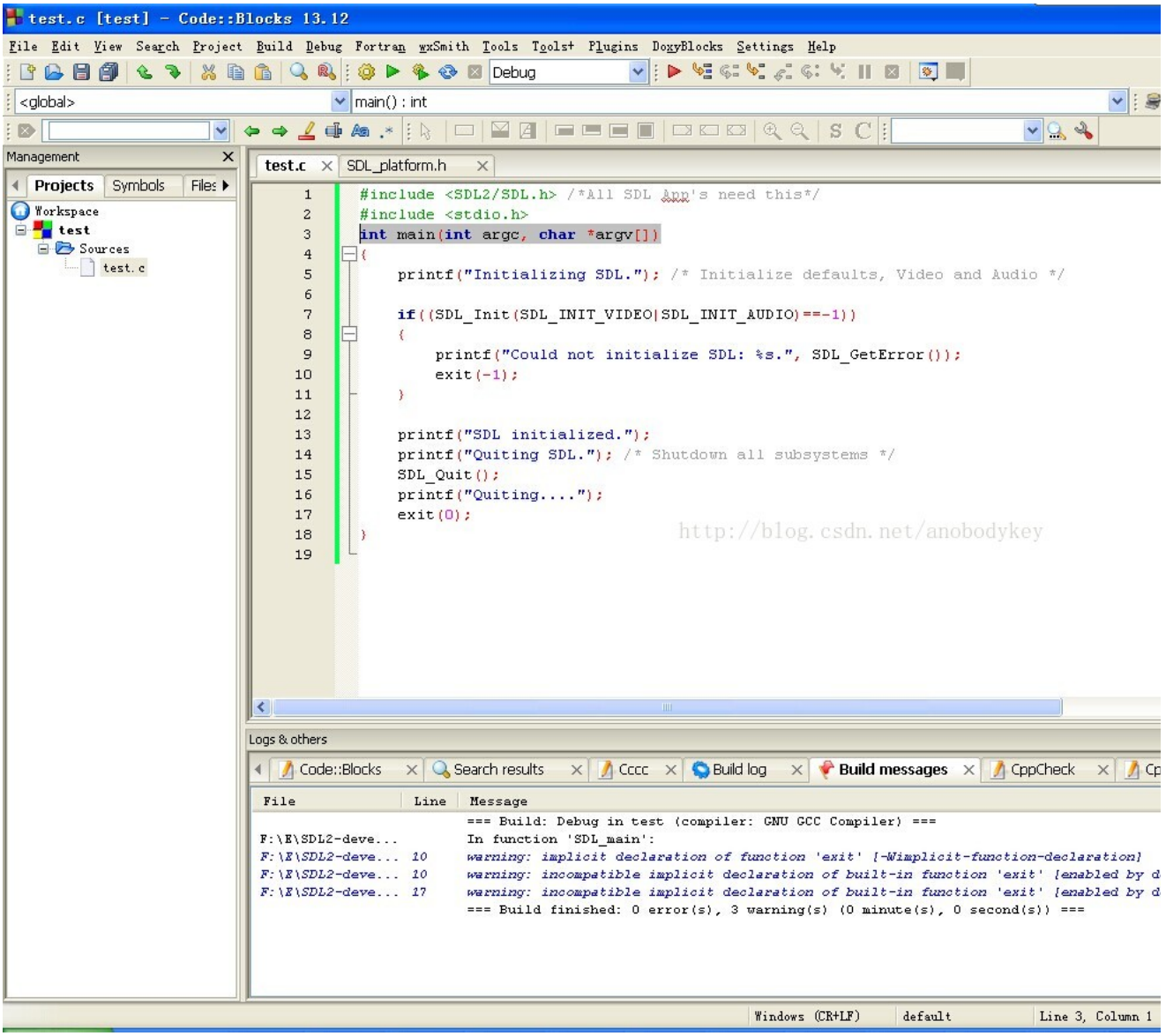
[cpp] view plain copy

在CODE上查看代码片

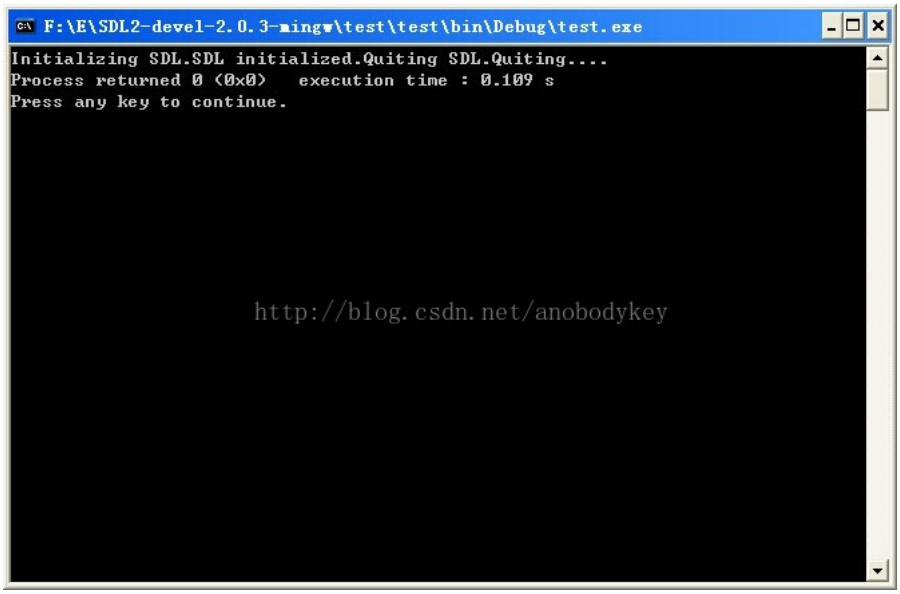
派生到我的代码片

```
int main(int argc, char *argv[])
```

再次编译，成功。



运行的结果如下图所示：



之后我们来一个有图形的代码，

[cpp] view plain copy

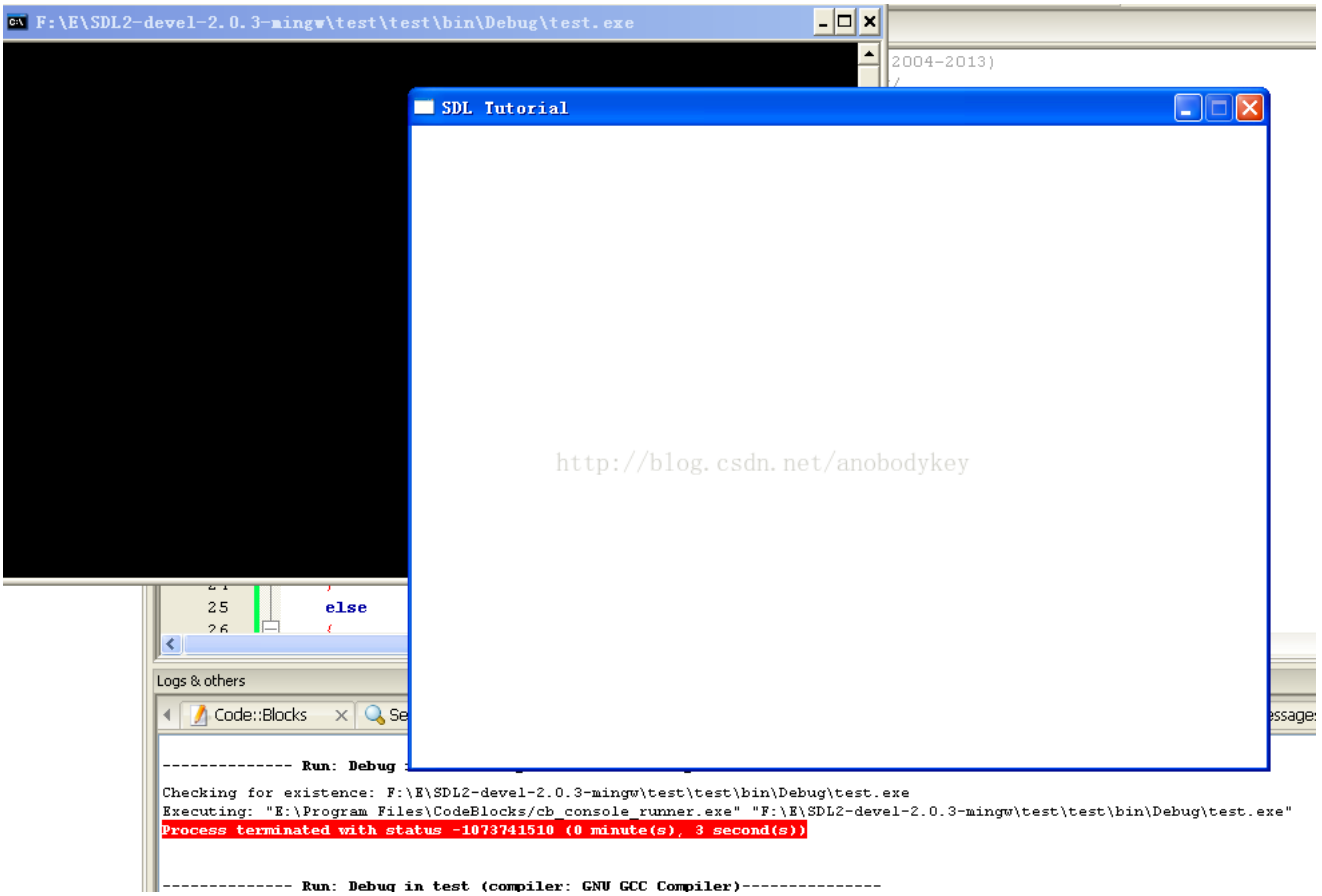
在CODE上查看代码片

派生到我的代码片

```cpp
/*This source code copyrighted by Lazy Foo' Productions (2004-2013)
and may not be redistributed without written permission.*/

//Using SDL and standard IO
#include <SDL2/SDL.h>
#include <stdio.h>

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

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() );
    }
    else
    {
        //Create window

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

    //Destroy window
    SDL_DestroyWindow( window );

    //Quit SDL subsystems
    SDL_Quit();

}
```

```
55.     return 0;
56. }
```

编译运行，如下



修改RGB颜色代码

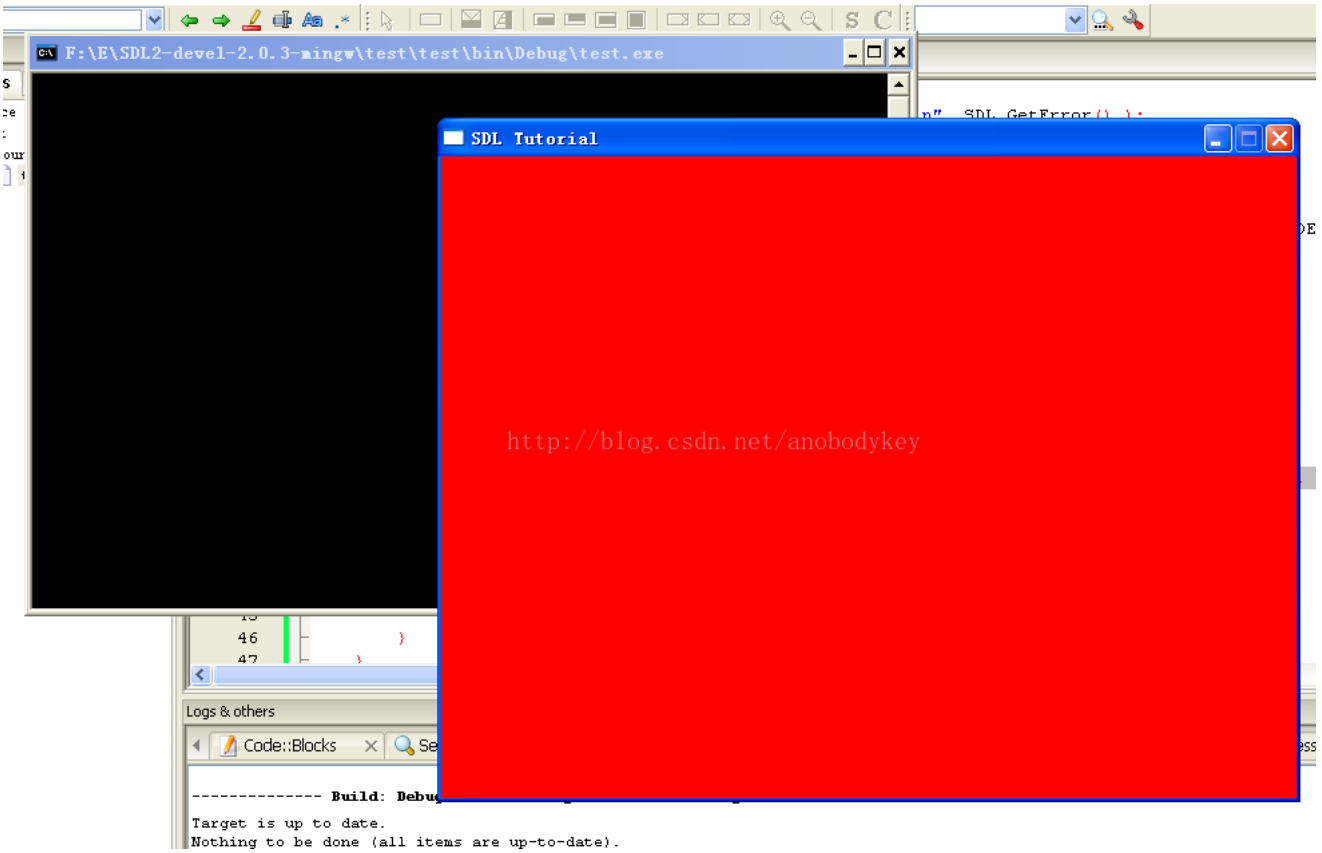
[cpp] view plain copy

在CODE上查看代码片

派生到我的代码片

```
1. SDL_FillRect( screenSurface, NULL, SDL_MapRGB( screenSurface->format, 0xFF, 0x00, 0x00 ) );
```

此时窗口应该是红色的，如下



至此，SDL2安装就告一段落了。