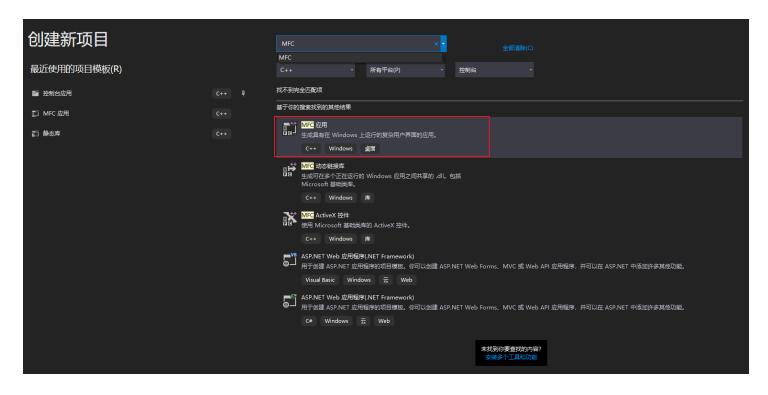
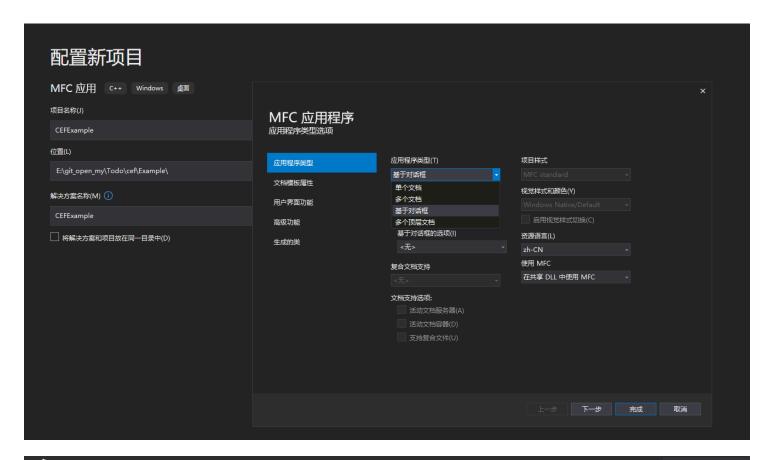
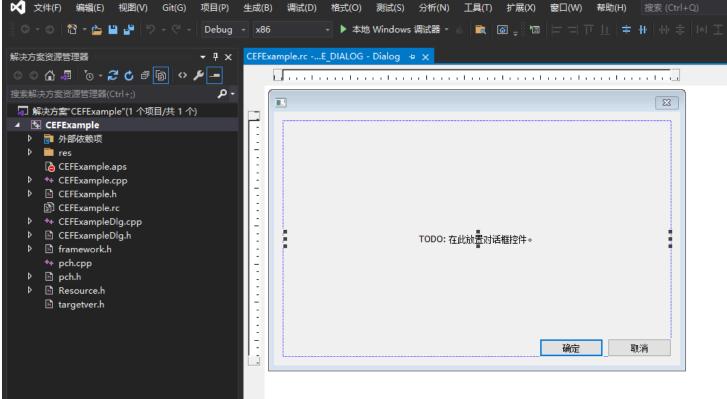
CEF 在 MFC 中的使用

工程配置

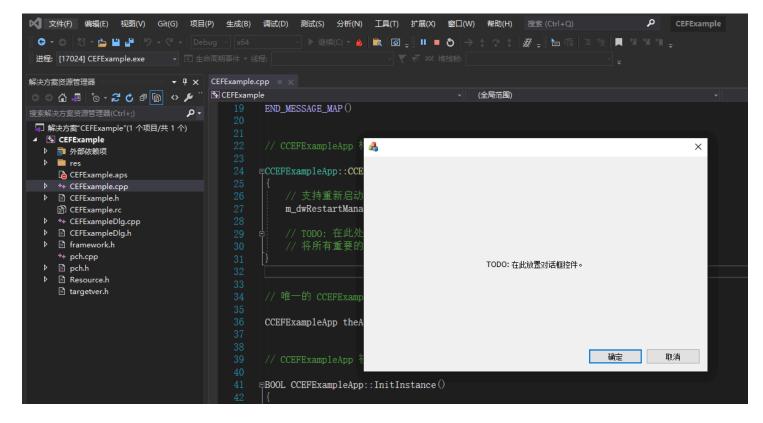
1、首先创建一个MFC对话框工程





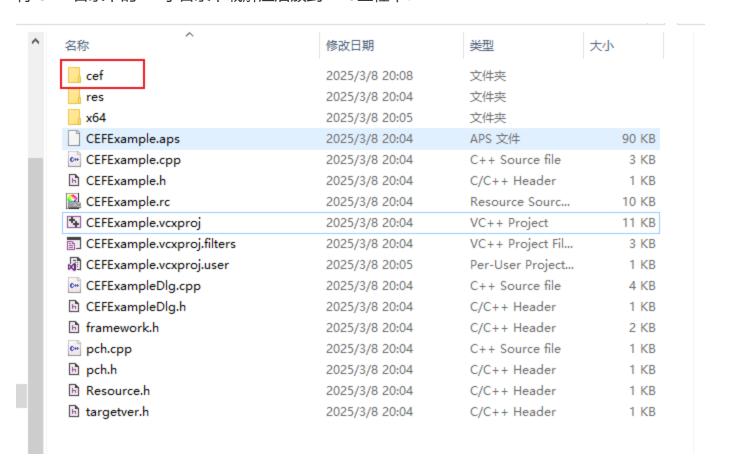


创建完运行测试效果如下

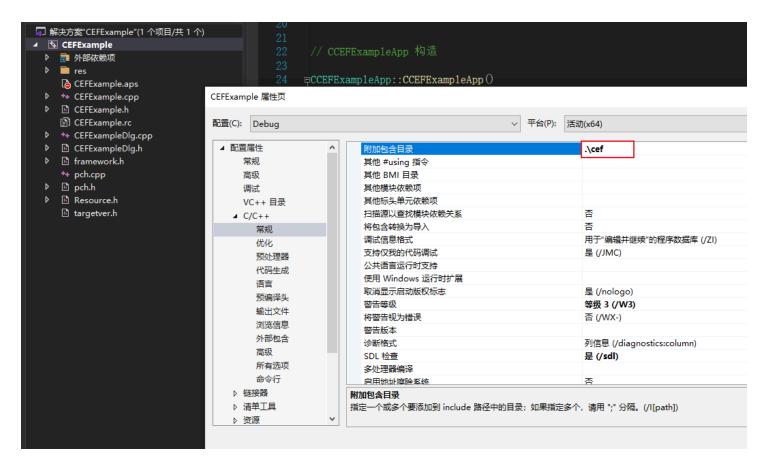


2、MFC工程引入CEF库

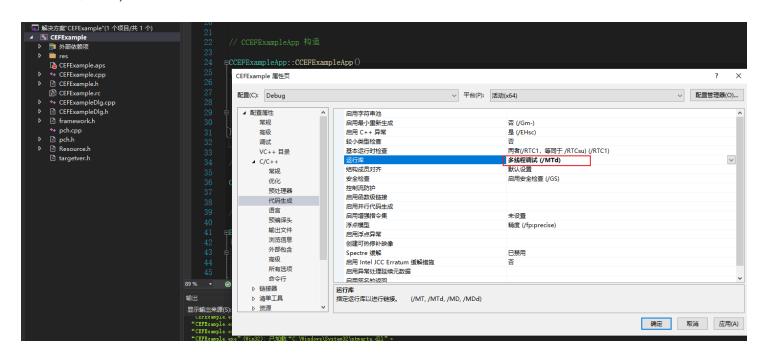
将 CEF 目录下的 cef子目录下载解压后放到MFC工程中:



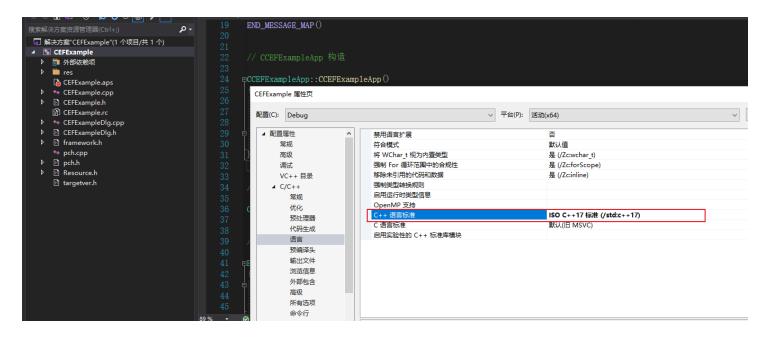
然后在VS中对工程右键 -> 属性 -> C/C++ -> 常规 -> 附加包含目录,添加".\cef"



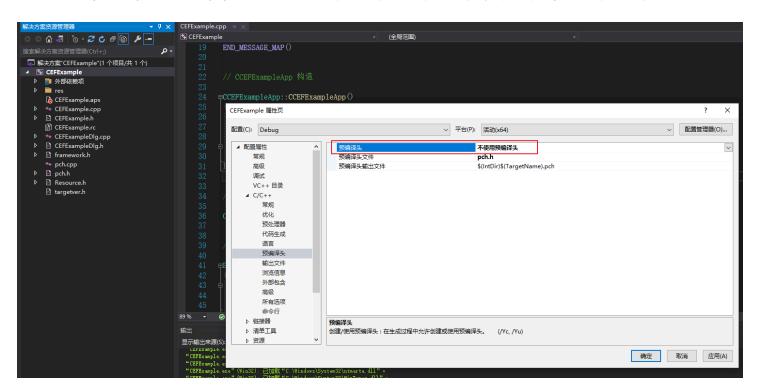
然后在VS中对工程右键 -> 属性 -> C/C++ -> 代码生成 -> 运行库,调整为 MT (debug调整为MTd, Release为MT)



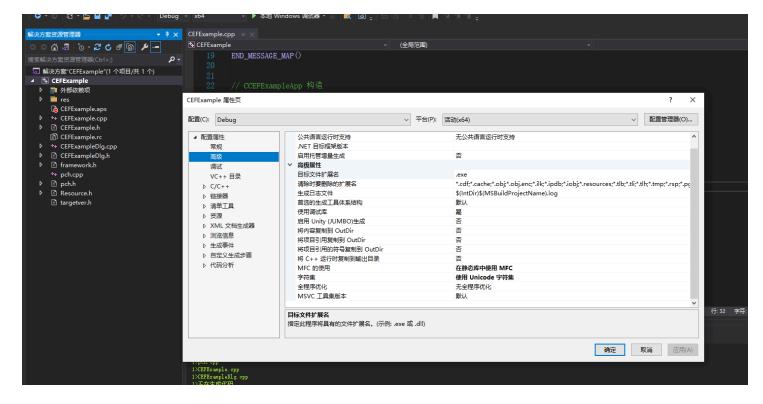
然后在VS中对工程右键 -> 属性 -> C/C++ -> 语言 -> c++语言标准,调整为 MT (debug调整为MTd, Release为MT)



然后在VS中对工程右键 -> 属性 -> C/C++ -> 预编译头 -> 预编译头, 调整为不适用预编译头

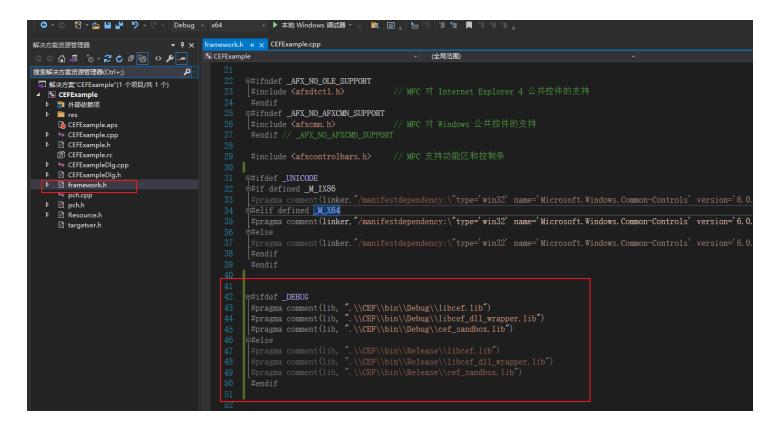


然后在VS中对工程右键 -> 属性 -> 高级 -> MFC的使用, 改为"在静态库中使用MFC"



然后在工程中找到 framework.h, 在最下面添加:

```
#ifdef _DEBUG
#pragma comment(lib, ".\\CEF\\bin\\Debug\\libcef.lib")
#pragma comment(lib, ".\\CEF\\bin\\Debug\\libcef_dll_wrapper.lib")
#pragma comment(lib, ".\\CEF\\bin\\Debug\\cef_sandbox.lib")
#else
#pragma comment(lib, ".\\CEF\\bin\\Release\\libcef.lib")
#pragma comment(lib, ".\\CEF\\bin\\Release\\libcef_dll_wrapper.lib")
#pragma comment(lib, ".\\CEF\\bin\\Release\\libcef_sandbox.lib")
#endif
```



然后对工程右键生成,生成信息中会显示工程exe生成位置,一般是代码的解决方案文件所在目录同级的 x64\Debug\中,定位到exe所在位置,在CEF目录的子目录中下载 Run64.zip, 然后解压后将对应子项目下的文件全部拷贝到exe所在目录:

Herry.	ISBAHAA	~=	A3	
locales	2025/3/8 20:27	文件夹		
CEFExample.exe	2025/3/8 20:23	应用程序	13,203 KB	
CEFExample.pdb	2025/3/8 20:23	Program Debug	32,564 KB	
chrome_100_percent.pak	2025/3/8 15:24	PAK 文件	656 KB	
chrome_200_percent.pak	2025/3/8 15:24	PAK 文件	1,034 KB	
chrome_elf.dll	2025/3/8 15:24	应用程序扩展	1,806 KB	
icudtl.dat	2025/3/8 15:24	DAT 文件	10,222 KB	
libcef.dll	2025/3/8 15:24	应用程序扩展	338,567 KB	
₫ libEGL.dll	2025/3/8 15:24	应用程序扩展	515 KB	
ibGLESv2.dll	2025/3/8 15:24	应用程序扩展	13,428 KB	
resources.pak	2025/3/8 15:24	PAK 文件	8,879 KB	
v8_context_snapshot.bin	2025/3/8 15:24	BIN 文件	1,451 KB	
vk_swiftshader.dll	2025/3/8 15:24	应用程序扩展	5,416 KB	
vk_swiftshader_icd.json	2025/3/8 15:24	JSON File	1 KB	
vulkan-1.dll	2025/3/8 15:24	应用程序扩展	909 KB	
cef cef_binary_13	3. Example	Run64	.gitattribi	utes
4.5+gdb2810 +chromium- 3.0.6943.142	06 13	TO T	igitation	
.gitignore cef.zip	BZ2 cef_binary_133. 4.5+gdb28106	readme.md	ZI Run64.z	
	+chromium-13 3.0.6943.142		CEF下子	目录

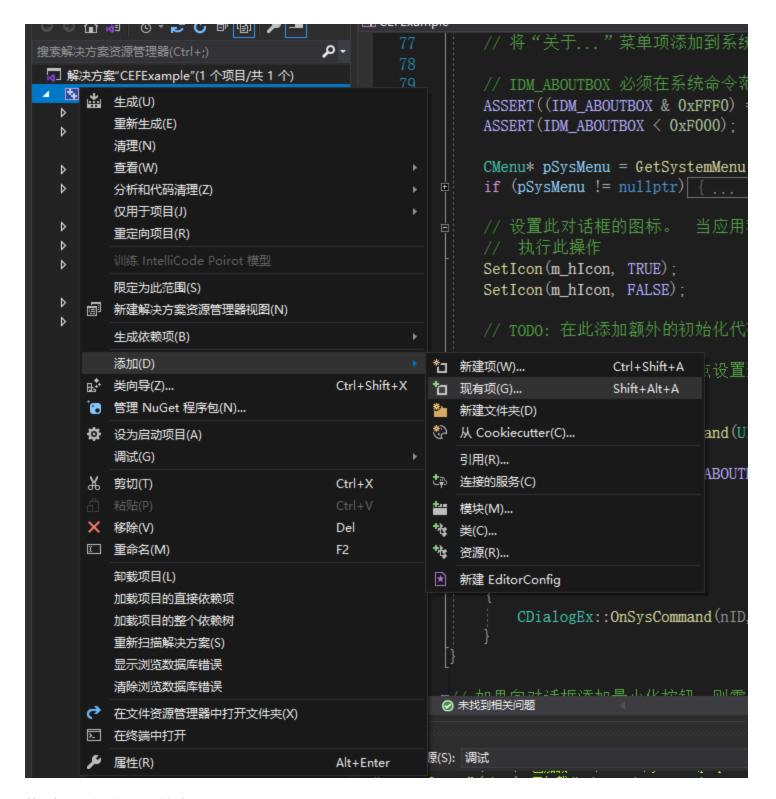
至此, CEF库导入完成

测试代码

首先去将cef目录下的子目录 test 中将测试类拷贝到工程下,这几个类其实来自于CEF源码中自带的 cefsimple工程,只是把simple_app.cc 中的默认路径改成了百度,其他无任何修改:

cef	2025/3/8 20:08	文件夹	
res	2025/3/8 20:04	文件夹	
x64	2025/3/8 20:05	文件夹	
CEFExample.aps	2025/3/8 20:04	APS 文件	90 KB
CEFExample.cpp	2025/3/8 20:04	C++ Source file	3 KB
□ CEFExample.h	2025/3/8 20:04	C/C++ Header	1 KB
CEFExample.rc	2025/3/8 20:04	Resource Sourc	10 KB
CEFExample.vcxproj	2025/3/8 20:21	VC++ Project	11 KB
CEFExample.vcxproj.filters	2025/3/8 20:04	VC++ Project Fil	3 KB
CEFExample.vcxproj.user	2025/3/8 20:05	Per-User Project	1 KB
CEFExampleDlg.cpp	2025/3/8 20:29	C++ Source file	4 KB
□ CEFExampleDlg.h	2025/3/8 20:04	C/C++ Header	1 KB
framework.h	2025/3/8 20:20	C/C++ Header	2 KB
pch.cpp	2025/3/8 20:04	C++ Source file	1 KB
	2025/3/8 20:04	C/C++ Header	1 KB
Resource.h	2025/3/8 20:04	C/C++ Header	1 KB
simple_app.cc	2025/3/8 14:19	C++ Source file	7 KB
simple_app.h	2025/3/1 7:56	C/C++ Header	1 KB
simple_handler.cc	2025/3/8 14:30	C++ Source file	6 KB
	2025/3/1 7:56	C/C++ Header	3 KB
🗈 targetver.h	2025/3/8 20:04	C/C++ Header	1 KB

在工程中导入这几个文件,工程右键 -> 添加 -> 现有项,选择刚才考过来的文件



找到工程名称+Dlg 的类,

```
#include "include/cef_command_line.h"
#include "include/cef_sandbox_win.h"
#include "simple_app.h"
#include "simple_handler.h"
```

```
CEFExampleDlg.cpp → X CEFExample.cpp
解决方案资源管理器
                                         № CEFExample
O O A 제 To - 2 C 리 📵 💠 🔑 💻
                                                                                                (全局范围)
                                    - م
                                                   ₽// CEFExampleDlg.cpp: 实现文件

√ 解决方案"CEFExample"(1 个项目/共 1 个)

▲ CEFExample

  ▶ 💼 外部依赖项
                                                   #include "pch. h"
#include "framework. h"
#include "CEFExample. h"
""
  ▶ 🗎 res
     CEFExample.aps
  ▶ ++ CEFExample.cpp
                                                    #include "CEFExampleD1g.h"
  CEFExample.h
                                                   #include "afxdialogex.h"
     CEFExample.rc
  ▶ ++ CEFExampleDlg.cpp
  CEFExampleDlg.h
                                                   □#ifdef _DEBUG
  ▶ 🖪 framework.h
                                                    #define new DEBUG_NEW
      ++ pch.cpp
  ▶ 🖹 pch.h
  ▶ 🖪 Resource.h
                                                   □#include "include/cef_command_line.h"
  ▶ ++ simple_app.cc
                                                    #include "include/cef_sandbox_win.h"
  simple_app.h
                                                  #include "simple_app.h"
#include "simple_handler.h"
  > ++ simple_handler.cc
  simple_handler.h
      🖹 targetver.h
                                                   // 用于应用程序"关于"菜单项的 CAboutD1g 对话框
                                                         CAboutD1g();
                                                     // 对话框数据
```

在当前类定位到 OnInitDialog 函数

```
解决方案资源管理器
                                   CEFExampleDlg.cpp + X CEFExample.cpp
                                   1 CEFExample
                                                                               - → CCEFExampleDlg
O O 🖒 🚜 To - 2 C d 🕫 📵 💠 🔑 💻
                                                ON_WM_SYSCOMMAND()
                                                ON_WM_PAINT()

√ 解决方案"CEFExample"(1 个项目/共 1 个)

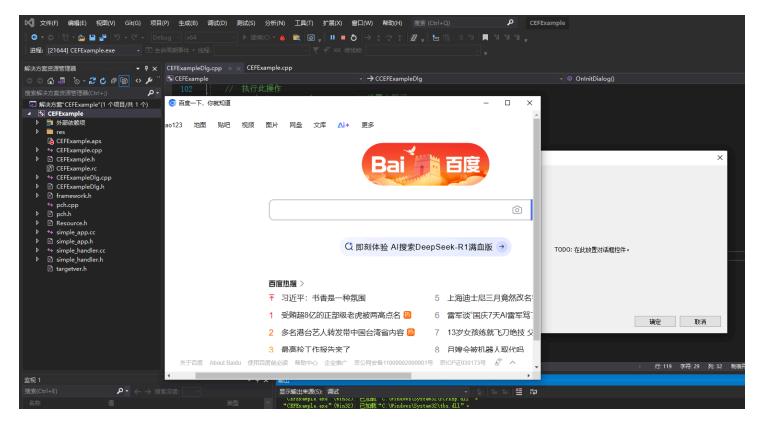
                                                ON_WM_QUERYDRAGICON()
END_MESSAGE_MAP()
  ▶ 💼 外部依赖项
  ▶ res
     CEFExample.aps
                                             // CCEFExampleDlg 消息处理程序
    ++ CEFExample.cpp
   CEFExample.h
                                            □BOOL CCEFExampleD1g::OnInitDialog()
    CEFExample.rc
    ++ CEFExampleDlg.cpp
     CDialogEx::OnInitDialog();
    framework.h
     ++ pch.cpp
  ▶ 🖪 pch.h
  Resource.h
                                                // IDM_ABOUTBOX 必须在系统命令范围内。
     🖹 targetver.h
                                                ASSERT ((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
                                                ASSERT (IDM_ABOUTBOX < 0xF000);
                                                CMenu* pSysMenu = GetSystemMenu(FALSE);
                                                if (pSysMenu != nullptr) { ... }
                                                // 设置此对话框的图标。 当应用程序主窗口不是对话框时,框架将自动
                                                                              // 设置大图标
                                                SetIcon(m_hIcon, TRUE);
                                                                              // 设置小图标
                                                SetIcon(m_hIcon, FALSE);
```

在TODO 后添加:

```
HINSTANCE hInstance = AfxGetInstanceHandle();
int exit code;
void* sandbox_info = nullptr;
CefMainArgs main args(hInstance);
exit_code = CefExecuteProcess(main_args, nullptr, sandbox_info);
if (exit_code >= 0) {
    return exit_code;
}
CefRefPtr<CefCommandLine> command line = CefCommandLine::CreateCommandLine();
command_line->InitFromString(::GetCommandLineW());
CefSettings settings;
settings.no_sandbox = true;
CefRefPtr<SimpleApp> app(new SimpleApp);
CefInitialize(main_args, settings, app.get(), sandbox_info);
CefRunMessageLoop();
CefShutdown();
```

```
CEFExampleDlg.cpp → X CEFExample.cpp
中方案资源管理器
                                   T CEFExample
                                                                                 → CCEFExampleDlg
O 🔐 🕫 To - 운 C 라 📵 💠 🔑 🗕
                                                     执行此操作
                                                                                 // 设置大图标
                                                 SetIcon(m_hIcon, TRUE);
解决方案"CEFExample"(1 个项目/共 1 个)
                                                 SetIcon(m_hIcon, FALSE);
№ CEFExample
  ■ 外部依赖项
  res
   CEFExample.aps
                                                 HINSTANCE hInstance = AfxGetInstanceHandle();
   ++ CEFExample.cpp
                                                 int exit_code;
  void* sandbox_info = nullptr;
  CEFExample.rc
   ** CEFExampleDlg.cpp
                                                 CefMainArgs main_args(hInstance);
   exit_code = CefExecuteProcess(main_args, nullptr, sandbox_info);
  framework.h
                                                 if (exit_code >= 0) {
   ** pch.cpp
                                                     return exit_code;
  Resource.h
                                                 CefRefPtr<CefCommandLine > command_line = CefCommandLine::CreateCommandLine();
  ++ simple_app.cc
                                                 command_line->InitFromString(::GetCommandLineW());
  simple_app.h
                                                 CefSettings settings;
   ++ simple_handler.cc
                                                 settings.no_sandbox = true
  imple_handler.h
                                                 CefRefPtr<SimpleApp> app (new SimpleApp);
   🖹 targetver.h
                                                 CefInitialize(main_args, settings, app. get(), sandbox_info);
                                                 CefRunMessageLoop();
                                                 CefShutdown();
                                                 return TRUE: // 除非将焦点设置到控件, 否则返回 TRUE
                                            □void CCEFExamp1eD1g::OnSysCommand(UINT nID, LPARAM 1Param)
```

然后运行



会看到一个有浏览器并且打开了百度的页面,

以及两个空的对话框。

这份测试代码是有bug的,不过只是为了测试CEF库是否可以正常使用,相关bug暂不定位处理;