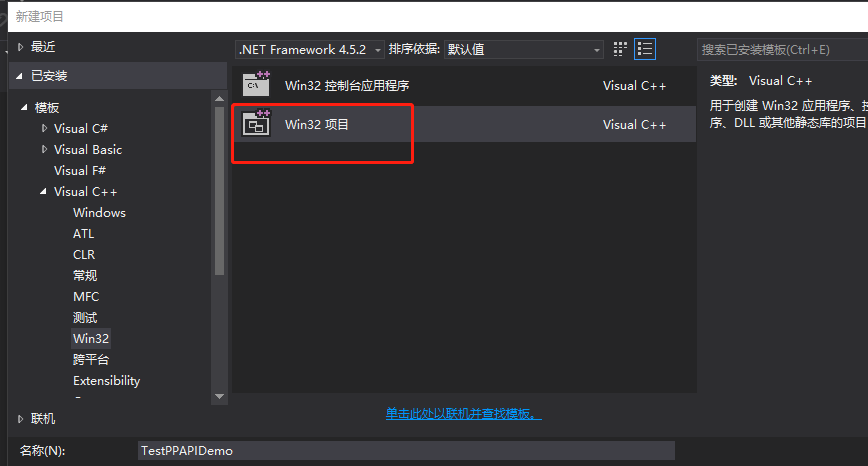
1. 创建一个win32 dll工程

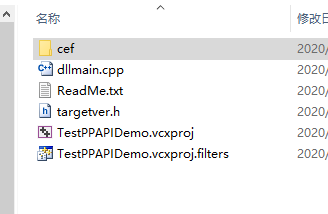


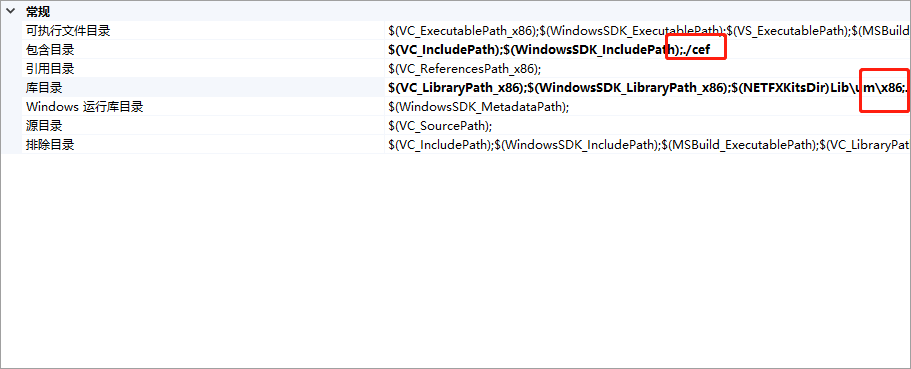
选择dll

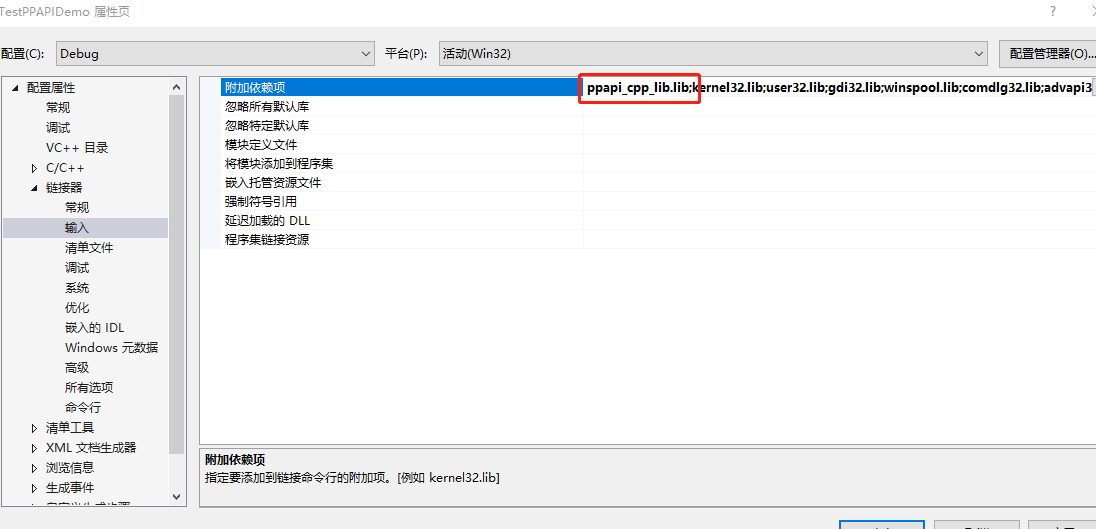


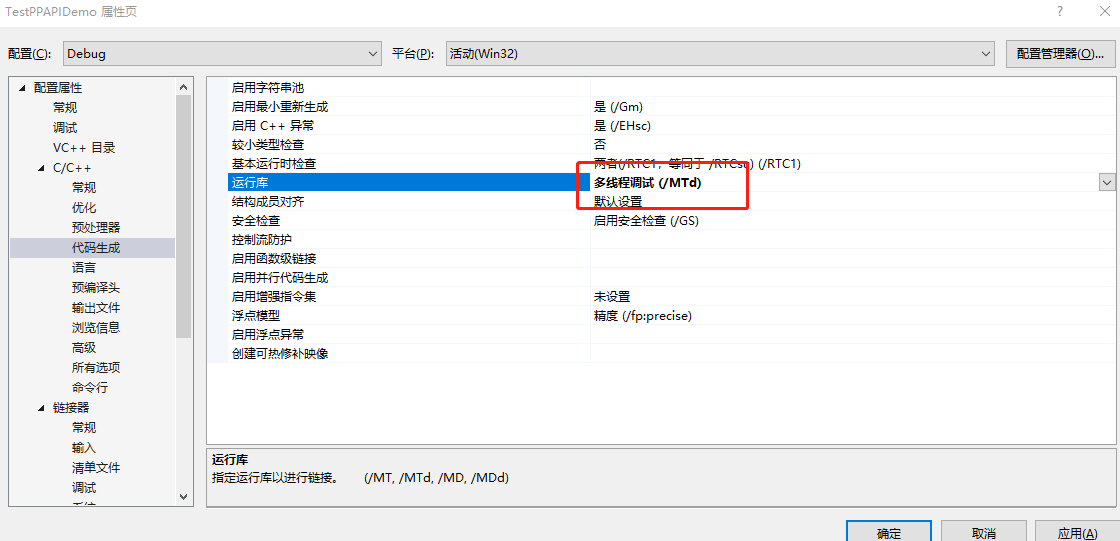
1. 修改工程

拷贝thirdpart的cef目录拷贝到当前工程目录，并设置引用路径和库









工程删除无用文件stdafx.h、stdafx.cpp、TestPPAPIDemo.cpp

dllmain.cpp代码修改为如下代码：

//dllmain.cpp

#include "ppapi/cpp/instance.h"

#include "ppapi/cpp/module.h"

#include "ppapi/cpp/var.h"

class HelloPPAPIInstance : public pp::*Instance* {

public:

explicit HelloPPAPIInstance(*PP\_Instance* instance) : pp::*Instance*(instance) {

pp::*Var* var\_reply = pp::*Var*("hello from Pepper Plugin");

*PostMessage*(var\_reply);

}

virtual ~HelloPPAPIInstance() {}

virtual void HandleMessage(const pp::*Var*& var\_message) {

if (var\_message.*is\_string*()) {

*printf*("HandleMessage: %s\n", var\_message.*AsString*().*c\_str*());

}

}

};

class HelloPPAPIModule : public pp::*Module* {

public:

HelloPPAPIModule() : pp::*Module*() {}

virtual ~HelloPPAPIModule() {}

virtual pp::*Instance*\* CreateInstance(*PP\_Instance* instance) {

return new HelloPPAPIInstance(instance);

}

};

namespace pp {

*Module*\* CreateModule() {

return new HelloPPAPIModule();

}

} // namespace pp

编译生成TestPPAPIDemo.dll

1. 安装electron

命令行进入到html目录下



npm install --save-dev electron@^6.0.2 --arch=ia32

**创建main.js文件：**

<!DOCTYPE html>

<html>

<head>

<title>hello ppapi</title>

<meta charset="UTF-8">

</head>

<body>

<embed id="plugin" type="application/x-ppapi-HelloPPAPI" width="1" height="1"></embed>

<input type="text" id="inputBox" value="" />

<button onclick="SendString()" >发送消息</button>

<script type="text/javascript">

function HandleMessage(message\_event) {

if (message\_event.data) {

alert("The string was a palindrome.");

} else {

alert("The string was not a palindrome.");

}

}

window.addEventListener('load', res=>{

var plugin = document.getElementById("plugin");

plugin.addEventListener("message", HandleMessage, false);

});

function SendString() {

var plugin = document.getElementById("plugin");

var inputBox = document.getElementById("inputBox");

// Send the string to the plugin using postMessage. This results in a call

// to Instance::HandleMessage in C++ (or PPP\_Messaging::HandleMessage in C).

plugin.postMessage(inputBox.value);

}

</script>

</body>

</html>

创建main.js：

const { app, BrowserWindow } = require('electron')

var cwd = process.cwd();

app.commandLine.appendSwitch('register-pepper-plugins', `${cwd}\\..\\Debug\\TestPPAPIDemo.dll;application/x-ppapi-HelloPPAPI`);

function createWindow () {

// 创建浏览器窗口

let win = new BrowserWindow({

width: 800,

height: 600,

webPreferences: {

nodeIntegration: true,

'plugins': true

}

})

// 加载index.html文件

win.loadFile('index.html')

}

app.whenReady().then(createWindow)

创建package.json：

{

"name": "your-app",

"version": "0.1.0",

"main": "main.js",

"scripts": {

"start": "electron ."

}

}

这样就完成了ppapi的基本demo，消息传递是通postMessage发送，