<https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3>

# CefSharp中文帮助文档

[**Edit**](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3/_edit)[**New Page**](https://github.com/cefsharp/CefSharp/wiki/_new)

Roby edited this page on 17 Jul 2018 · [20 revisions](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3/_history)

# 目录

## [1、基础知识](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a1)

[1.1 cefsharp设置默认语言](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a1_1)  
[1.2 cefSharp 服务器运行要求](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a1_2)  
[1.3 cefsharp设置网页接受语言AcceptLanguage](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a1_3)  
[1.4 cef设置userAgent](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a1_4)

## [2、配置管理](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a2)

[2.1、CommandLine配置](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a2_1)

## [3、请求流程处理](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a3)

[3.1 读取网页源代码](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a3_1)  
[3.2 获取页面中的指定文件内容(.jpg,.js等)](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a3_2)  
[3.3 过滤某些页面内容，例如图片或某些文字](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a3_3)  
[3.4 文件进一步获取，获取完整内容](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a3_4)  
[3.5 文件进一步获取，获取完整内容（优化，Content-Length不一致）](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a3_5)

## [4、 Cookie处理](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a4)

[4.1 设置cookie](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a4_1)  
[4.2 读取cookie](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a4_2)

## [5、 Js处理](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a5)

[5.1 基本的同步js操作](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a5_1)  
[5.2 其他Frame操作](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a5_2)

## [6、 资源清理](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a6)

[6.1 浏览器本身处理](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a6_1)  
[6.2 需要关闭浏览器负载程序时操作](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a6_2)

## [7、 扩展功能](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a7)

## [8、 常见错误](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a8)

[8.1 下载CefSharp master zip code文件编译报错](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a8_1)  
[8.2 无法创建新的堆栈防护页面](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a8_2)  
[8.3 结束浏览器后，CefSharp.BrowserSubprocess.exe进行无法结束](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a8_3)

## [9、 GitHub---Wiki(部分内容使用了Google进行翻译)](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a9)

[9.1 如何使用.NET 调用 JavaScript方法？](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a9_1)  
[9.2 如何获取 Javascript 方法返回的结果？](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a9_2)  
[9.3 如何暴露.NET类，提供给Javascript？](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a9_3)  
[9.4 为什么我得到一个错误有关“无法加载文件或程序集 CefSharp.Core.dll ”或它的一个依赖。指定的模块找不到。“当试图运行基于我CefSharp的应用程序？它编译成功，但不运行？它运行我的开发机器上，虽然抛出一个异常，当我把它复制到另一台计算机？](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a9_4)  
[9.5 为什么当我将 ChromiumWebBrowser 添加到我的应用程序时，Visual Studio WPF设计器不起作用？](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a9_5)  
[9.6 如何在目标应用程序中包含Visual Studio C ++ 2012/2013可再发行组件？](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a9_6)  
[9.7 如何重写 Javascript 的 window.alert 事件和相似的？](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a9_7)  
[9.8 CefSharp3二进制程序在哪里？](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a9_8)  
[9.9 Windows XP/2003支持？](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a9_9)  
[9.10 当我重新发布使用CefSharp的应用程序时，需要包括什么文件？](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a9_10)  
[9.11 构建过程不能复制CEF文件](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a9_11)  
[9.12 为什么 IndexedDB 返回 QuotaExceededError？](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a9_12)  
[9.13 在C＃中如何处理Javascript事件？](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a9_13)  
[9.14 如何实现 WinForms 拖放？](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a9_14)

## [10、 参考网址](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a10)

## [11、 其他](https://github.com/cefsharp/CefSharp/wiki/CefSharp%E4%B8%AD%E6%96%87%E5%B8%AE%E5%8A%A9%E6%96%87%E6%A1%A3#a11)

## 1、基础知识

### 1.1 cefsharp设置默认语言

cefsharp是不错的浏览器内核封装版本之一，默认语言是en-US，这个一直困扰着项目，项目好多处需修改，后来经多次尝试，才发现，原来设置默认语言这么简单。 Loacal 属性就是对CefSharp运行语言环境进行设置  
var setting = new CefSharp.CefSettings();

// 设置语言  
setting.Locale = "zh-CN";  
CefSharp.Cef.Initialize(setting, true, false); 以上这段代码一定要在new ChromiumWebBrowser之前调用

### 1.2 cefSharp 服务器运行要求

.net framework 环境和 vc++ 2013 runtime (x86/x64)

### 1.3 cefsharp设置网页接受语言AcceptLanguage

什么是 cefsharp设置网页接受语言AcceptLanguage  
1.设置浏览器的请求控制器  
 webView.RequestHandler = new RequestHandler();   
2.新建RequestHandler类继承IRequestHandler接口，实现方法OnBeforeResourceLoad,新版本如果又变更但是整体思路不变，内部处理是一致的请见谅。

public bool OnBeforeResourceLoad(IWebBrowser browser, IRequestResponse requestResponse)

{

IDictionary<string, string> headers = requestResponse.Request.GetHeaders();

headers.Add("Accept-Language", "zh,zh-cn,zh-tw");

requestResponse.Request.SetHeaders(headers);

return false;

}

### 1.4 cef设置userAgent

var setting = new CefSharp.CefSettings();

setting.UserAgent = "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.102 Safari/537.36";

CefSharp.Cef.Initialize(setting, true, false);

## 2、配置管理

### 2.1、CommandLine配置

所有配置：<http://peter.sh/experiments/chromium-command-line-switches/>  
样例参考：

var settings = new CefSettings();

settings.CefCommandLineArgs.Add("no-proxy-server", "1");

settings.CefCommandLineArgs.Add("proxy-server", "ProxyAddress");

Cef.Initialize(settings, true, true);

## 3、请求流程处理

### 3.1 读取网页源代码

在页面加载完成后处理， 依赖最低环境 4.5.2

async void browser\_FrameLoadEnd(object sender, FrameLoadEndEventArgs e)

{

Log.WriteLog("browser\_FrameLoadEnd:" + e.Url);

var result = await browser.GetSourceAsync();

}

如果想在4.0下环境操作需要使用。

 var task = browser.GetSourceAsync();

task.Wait();

string content = task.Result;

### 3.2 获取页面中的指定文件内容(.jpg,.js等)

A.首先需要对ChromiumWebBrowser 的 IRequestHandler RequestHandler进行实现。  
B.需要对 IRequestHandler 的IResponseFilter IRequestHandler.GetResourceResponseFilter 方法进行重写。  
C.需要写一个类实现 IResponseFilter 接口。  
D.然后就可用在IResponseFilter 的  
FilterStatus Filter(Stream dataIn, out long dataInRead, Stream dataOut, out long dataOutWritten) 方法中，读取指定内容的流了。

### 3.3 过滤某些页面内容，例如图片或某些文字

具体实现，参考 3.2 ，在最后的Filter方法中，对返回的dataOut不进行赋值或者，取到值，然后replace处理，返回其他数据，即可。

### 3.4 文件进一步获取，获取完整内容

说明：由于很多文件无法获取到完整内容，再者具体文件内容在Filter里面进行了控制，而Fileter的内容依赖于IRequestHandler所以，外部只能操作Handler得到数据。 所以需要在，Filter和Hanlder类中，使用事件来传递具体的内容。代码如下。 Filter类如下：

public class TestImageFilter : IResponseFilter

{

public event Action<byte[]> NotifyData;

private int contentLength = 0;

private List<byte> dataAll = new List<byte>();

public void SetContentLength(int contentLength)

{

this.contentLength = contentLength;

}

public FilterStatus Filter(System.IO.Stream dataIn, out long dataInRead, System.IO.Stream dataOut, out long dataOutWritten)

{

try

{

if (dataIn == null)

{

dataInRead = 0;

dataOutWritten = 0;

return FilterStatus.Done;

}

dataInRead = dataIn.Length;

dataOutWritten = Math.Min(dataInRead, dataOut.Length);

dataIn.CopyTo(dataOut);

dataIn.Seek(0, SeekOrigin.Begin);

byte[] bs = new byte[dataIn.Length];

dataIn.Read(bs, 0, bs.Length);

dataAll.AddRange(bs);

if (dataAll.Count == this.contentLength)

{

// 通过这里进行通知

NotifyData(dataAll.ToArray());

return FilterStatus.Done;

}

else if (dataAll.Count < this.contentLength)

{

dataInRead = dataIn.Length;

dataOutWritten = dataIn.Length;

return FilterStatus.NeedMoreData;

}

else

{

return FilterStatus.Error;

}

}

catch (Exception ex)

{

dataInRead = dataIn.Length;

dataOutWritten = dataIn.Length;

return FilterStatus.Done;

}

}

public bool InitFilter()

{

return true;

}

}

Filter类有了，那我们如何知道数据流的具体长度呢？这就需要在Handler的实现的其他方法里面寻找了。

bool IRequestHandler.OnResourceResponse(IWebBrowser browserControl, IBrowser browser, IFrame frame, IRequest request, IResponse response)

{

//NOTE: You cannot modify the response, only the request

// You can now access the headers

//var headers = response.ResponseHeaders;

try

{

var content\_length = int.Parse(response.ResponseHeaders["Content-Length"]);

if (this.filter != null)

{

this.filter.SetContentLength(content\_length);

}

}

catch { }

return false;

}

private TestImageFilter filter = null;

public event Action<byte[]> NotifyData;

IResponseFilter IRequestHandler.GetResourceResponseFilter(IWebBrowser browserControl, IBrowser browser, IFrame frame, IRequest request, IResponse response)

{

var url = new Uri(request.Url);

if (url.AbsoluteUri.Contains("http://test.test.com/somehead?"))

{

this.filter = new TestImageFilter();

filter.NotifyData += filter\_NotifyData;

return filter;

}

return null;

}

void filter\_NotifyData(byte[] data)

{

if (NotifyData != null)

{

NotifyData(data);

}

}

此方法位IRequestHandler的一部分实现，通过实现函数：IRequestHandler.GetResourceResponseFilter得到资源文件的长度，然后长度传入Filter,在Filter中控制从而得到整个数据的真正长度。

### 3.5 文件进一步获取，获取完整内容（优化，Content-Length不一致）

由于，部分站点，返回数据是分片了的，即：不能通过，Content-Length的长度来判断，程序的流是否完成。  
所以需要其他方式处理，即：单个http请求完成的时候，会调用Complete方法，所以可以在这里处理。  
下面是测试代码： Filter管理类

public class FilterManager

{

private static Dictionary<string, IResponseFilter> dataList = new Dictionary<string, IResponseFilter>();

public static IResponseFilter CreateFilter(string guid)

{

lock (dataList)

{

var filter = new TestImageFilter();

dataList.Add(guid, filter);

return filter;

}

}

public static IResponseFilter GetFileter(string guid)

{

lock (dataList)

{

return dataList[guid];

}

}

}

TestFilter类，对流进行合并

public class TestImageFilter : IResponseFilter

{

public List<byte> dataAll = new List<byte>();

public FilterStatus Filter(System.IO.Stream dataIn, out long dataInRead, System.IO.Stream dataOut, out long dataOutWritten)

{

try

{

if (dataIn == null || dataIn.Length == 0)

{

dataInRead = 0;

dataOutWritten = 0;

return FilterStatus.Done;

}

dataInRead = dataIn.Length;

dataOutWritten = Math.Min(dataInRead, dataOut.Length);

dataIn.CopyTo(dataOut);

dataIn.Seek(0, SeekOrigin.Begin);

byte[] bs = new byte[dataIn.Length];

dataIn.Read(bs, 0, bs.Length);

dataAll.AddRange(bs);

dataInRead = dataIn.Length;

dataOutWritten = dataIn.Length;

return FilterStatus.NeedMoreData;

}

catch (Exception ex)

{

dataInRead = dataIn.Length;

dataOutWritten = dataIn.Length;

return FilterStatus.Done;

}

}

public bool InitFilter()

{

return true;

}

}

最后是部分的。IRequestHandler实现代码

public class RequestHandler : IRequestHandler

{

// 略去代码 ...

public event Action<byte[]> NotifyMsg;

IResponseFilter IRequestHandler.GetResourceResponseFilter(IWebBrowser browserControl, IBrowser browser, IFrame frame, IRequest request, IResponse response)

{

var url = new Uri(request.Url);

if (url.AbsoluteUri.Contains("https://res.wx.qq.com/zh\_CN/htmledition/v2/css/base/base2e4e03.css"))

{

var filter = FilterManager.CreateFilter(request.Identifier.ToString());

return filter;

}

return null;

}

void filter\_NotifyData(byte[] data)

{

if (NotifyMsg != null)

{

NotifyMsg(data);

}

}

void IRequestHandler.OnResourceLoadComplete(IWebBrowser browserControl, IBrowser browser, IFrame frame, IRequest request, IResponse response, UrlRequestStatus status, long receivedContentLength)

{

if (request.Url.Contains("https://res.wx.qq.com/zh\_CN/htmledition/v2/css/base/base2e4e03.css"))

{

var filter = FilterManager.GetFileter(request.Identifier.ToString()) as TestImageFilter;

filter\_NotifyData(filter.dataAll.ToArray());

}

}

}

## 4、Cookie处理

### 4.1 设置cookie

var cookieManager = CefSharp.Cef.GetGlobalCookieManager();

await cookieManager.SetCookieAsync("http://" + domain, new CefSharp.Cookie()

{

Domain = domain,

Name = name,

Value = value,

Expires = DateTime.MinValue

});

### 4.2 读取cookie

建立Cookie读取对象，继承接口 ICookieVisitor

public class CookieVisitor : CefSharp.ICookieVisitor

{

public event Action<CefSharp.Cookie> SendCookie;

public bool Visit(CefSharp.Cookie cookie, int count, int total, ref bool deleteCookie)

{

deleteCookie = false;

if (SendCookie != null)

{

SendCookie(cookie);

}

return true;

}

}

在browser事件中进行处理

private void browser\_FrameLoadEnd(object sender, CefSharp.FrameLoadEndEventArgs e)

{

var cookieManager = CefSharp.Cef.GetGlobalCookieManager();

CookieVisitor visitor = new CookieVisitor();

visitor.SendCookie += visitor\_SendCookie;

cookieManager.VisitAllCookies(visitor);

}

/// 回调事件

private void visitor\_SendCookie(CefSharp.Cookie obj)

{

cookies += obj.Domain.TrimStart('.') + "^" + obj.Name + "^" + obj.Value + "$";

}

## 5、Js处理

### 5.1 基本的同(异)步js操作

browser.GetBrowser().MainFrame.ExecuteJavaScriptAsync("document.getElementById('testid').click();");

browser.GetBrowser().MainFrame.ExecuteJavaScriptAsync("document.getElementById('testid2').value='123'");

### 5.2 其他Frame操作

string script = "if(document.getElementById('img\_out\_10000')){ document.getElementById('img\_out\_10000').click(); }";

var list = browser.GetBrowser().GetFrameNames();

if (list.Count > 1)

{

browser.GetBrowser().GetFrame(list[1]).ExecuteJavaScriptAsync(script);

}

### 5.3 js回调，C#方法

参见本文档：9.3

## 6、资源清理

### 6.1 浏览器本身处理

static ChromiumWebBrowser()

{

if (CefSharpSettings.ShutdownOnExit)

{

Application.ApplicationExit += OnApplicationExit;

}

}

private static void OnApplicationExit(object sender, EventArgs e)

{

Cef.Shutdown();

}

### 6.2 需要关闭浏览器负载程序时操作

try

{

browser.CloseDevTools();

browser.GetBrowser().CloseBrowser(true);

}

catch { }

try

{

if (browser != null)

{

browser.Dispose();

Cef.Shutdown();

}

}

catch { }

## 7、扩展功能

## 8、常见错误

### 8.1 下载CefSharp master zip code文件编译报错

下载地址：<https://codeload.github.com/cefsharp/CefSharp/zip/master>  
错误提示：

由于编译路径存在中文导致。

### 8.2 无法创建新的堆栈防护页面

可能是由于 进程“ CefSharp.BrowserSubprocess.exe ”没有正常结束掉，一直占用内存增加，直到...

### 8.3 结束浏览器后，CefSharp.BrowserSubprocess.exe进行无法结束

需要确保，创建浏览器的线程和调用Cef.ShutDown();的线程确保在同一个线程中进行操作。 如果是主线程创建。关闭程序后BrowserSubprocess进程可以直接退出。 如果是非主线程，需要在创建浏览器，即：把浏览器绑定到控件的线程，调用shutdown方法。

## 9、GitHub---Wiki(部分内容使用了Google进行翻译)

本文档的大部分将被替换为新的 [一般使用指南](https://github.com/cefsharp/CefSharp/wiki/General-Usage) - 必须阅读！

### 快速FAQ TOC [英文版本](https://github.com/cefsharp/CefSharp/wiki/Frequently-asked-questions)

本FAQ〜10项。尝试列出一些很酷的CefSharp特性和一些最常见的问题的的。

有关更多提示，请查看不断增加的问题列表[faq-able!](https://github.com/cefsharp/CefSharp/issues?q=label%3Afaq-able!)

1. 如何使用.NET 调用 JavaScript方法？
2. 如何获取 Javascript 方法返回的结果呢？
3. 如何暴露.NET类，提供给Javascript？
4. “无法加载文件或程序集 CefSharp.Core.dll ”或它的一个依赖。
5. 为什么当我将 ChromiumWebBrowser 添加到我的应用程序时，Visual Studio WPF设计器不起作用？
6. 如何在目标应用程序中包含Visual Studio C ++ 2012/2013可再发行组件？
7. 如何重写 Javascript 的 window.alert 事件和相似的？
8. CefSharp3二进制程序在哪里？
9. Windows XP/2003支持？
10. 当我重新发布使用CefSharp的应用程序时，需要包括什么文件？
11. 构建过程不能复制CEF文件
12. 为什么 IndexedDB 返回 QuotaExceededError？
13. 在C＃中如何处理Javascript事件？
14. 如何实现 WinForms 拖放？

### 1. 如何使用.NET 调用 JavaScript方法？

简单代码可能看起来像这样：

var script = string.Format("document.body.style.background = '{0}'", colors[color\_index++]);if (color\_index >= colors.Length)

{

color\_index = 0;

}

browser.GetMainFrame().ExecuteJavaScriptAsync(script);

##### 什么时候可以开始执行 Javascript ？

不幸的是加载DOM之前OnFrameLoadStart被调用，所以你需要使用如下之一：FrameLoadEnd / LoadingStateChanged / IRenderProcessMessageHandler.OnContextCreated。下面是几个例子

不幸的是，在加载DOM之前调用 OnFrameLoadStart ，所以你需要使用下面的一个方法： FrameLoadEnd/LoadingStateChanged/IRenderProcessMessageHandler.OnContextCreated 。这里有几个例子

browser.RenderProcessMessageHandler = new RenderProcessMessageHandler();

public class RenderProcessMessageHandler : IRenderProcessMessageHandler

{

// Wait for the underlying `Javascript Context` to be created, this is only called for the main frame.

// If the page has no javascript, no context will be created.

void IRenderProcessMessageHandler.OnContextCreated(IWebBrowser browserControl, IBrowser browser, IFrame frame)

{

const string script = "document.addEventListener('DOMContentLoaded', function(){ alert('DomLoaded'); });";

frame.ExecuteJavaScriptAsync(script);

}

}

//Wait for the page to finish loading (all resources will have been loaded, rendering is likely still happening)

browser.LoadingStateChanged += (sender, args) =>

{

//Wait for the Page to finish loading

if (args.IsLoading == false)

{

browser.ExecuteJavaScriptAsync("alert('All Resources Have Loaded');");

}

}

//Wait for the MainFrame to finish loading

browser.FrameLoadEnd += (sender, args) =>

{

//Wait for the MainFrame to finish loading

if(args.Frame.IsMain)

{

args.Frame.ExecuteJavaScriptAsync("alert('MainFrame finished loading');");

}

};

注意：脚本是在 Frame 级别执行，永远页面至少有一个Frame（ MainFrame ）。 IWebBrowser.ExecuteScriptAsync 扩展方法是为了向后兼容，可以将其用作在主框架上执行 js 的快捷方式。

### 2. 如何获取 Javascript 方法返回的结果？

如果需要评估返回值的代码，使用 Task<JavascriptResponse> EvaluateScriptAsync(string script, TimeSpan? timeout) 方法。 JavaScript代码是异步执行的，因此使用.NET Task 类返回一个响应，其中包含错误消息，结果和一个成功（bool）标志。

// Get Document Height

var task = frame.EvaluateScriptAsync("(function() { var body = document.body, html = document.documentElement; return Math.max( body.scrollHeight, body.offsetHeight, html.clientHeight, html.scrollHeight, html.offsetHeight ); })();", null);

task.ContinueWith(t =>

{

if (!t.IsFaulted)

{

var response = t.Result;

EvaluateJavaScriptResult = response.Success ? (response.Result ?? "null") : response.Message;

}

}, TaskScheduler.FromCurrentSynchronizationContext());

有关更详细的例子 [Gist](https://gist.github.com/amaitland/9d354376960b0cd9305a#file-oneplusone-cs)

**Notes**

* 脚本是在 Frame 级别执行，永远页面至少有一个Frame（ MainFrame ）
* 只能返回一般的值（如int，bool，string等） - 不是您自己定义的复杂（用户定义）类型。这是因为没有（容易的）方式将随机的JavaScript对象暴露给.NET World，至少不是今天。但是，一种可能的技术是，将要返回到.NET代码的Javascript对象转换为JSON对象，使用 JSON.toStringify() 方法转化为JSON字符串，并将该字符串返回到您的.NET代码。然后，您可以将该字符串解码为一个类似JSON.net的.NET对象。有关更多信息，请参阅此MSDN链接。 (<https://msdn.microsoft.com/library/cc836459(v=vs.94).aspx>)

### 3. 如何暴露.NET类，提供给Javascript？

参考 [般使用指南 - 如何公开.Net类](https://github.com/cefsharp/CefSharp/wiki/General-Usage#3-how-do-you-expose-a-net-class-to-javascript)

### 4. 为什么我得到一个错误有关“无法加载文件或程序集 CefSharp.Core.dll”或它的一个依赖。指定的模块找不到。“当试图运行基于我CefSharp的应用程序？它编译成功，但不运行？它运行我的开发机器上，虽然抛出一个异常，当我把它复制到另一台计算机？

这是一个常见的错误，通常是以下之一

1. VC++ 2012/2013 可重新分发包\*是为了在非开发者机器上运行CefSharp所必需的。有关详细信息，请参阅下面的FAQ＃6。 您可以将所需的dll作为应用程序的一部分。
2. 执行文件夹中不存在所有依赖关系。 CefSharp 包括 非托管 的dll和资源，这些文件通过安装 Nuget 包时包含在项目中的两个 .props 文件复制到执行文件夹。 请参阅下面所需的文件列表，确保所需的文件存在。
3. 你通过安装程序打包应用程序进行发布，并且不会在目标计算机上运行。 默认情况下，安装程序不包括 非托管 资源，您需要手动添加它们。 对于 ClickOnce ，请参阅＃1314其他用户提出的一些指针和解决方案。

可以在这里找到所需文件的列表：[Output files description (Redistribution)](https://github.com/cefsharp/CefSharp/wiki/Output-files-description-table-(Redistribution))

**注**: 如果在 XAML 中初始化 WPF 控件时遇到 FileNotFoundException ，这也适用。

**注2**：如果从源代码编译（不推荐使用 Nuget 包），并注意到您不能再以debug模式构建，但是release构建工作正常，您可能需要修复Visual Studio版本。 这种情况发生在少数情况下，您将获得如上所示，丢失的非托管.dll文件相同的提示信息。

### 5. 为什么当我将 ChromiumWebBrowser 添加到我的应用程序时，Visual Studio WPF设计器不起作用？

**注** 版本 57.0.0 增加了最小的设计器支持，有关详细信息，请参见 [https://github.com/cefsharp/CefSharp/pull/1989。](https://github.com/cefsharp/CefSharp/pull/1989%E3%80%82)

旧版本不支持 WPF 设计器，这是不幸的，CefSharp是用C++代码编写的，后者又引用了CEF DLLs 导致的结果。这种情况下，Visual Studio 设计器不能够很好的支持。[在论坛的主题](https://groups.google.com/forum/#!topic/cefsharp/rGGXh6wAkCU)将进一步解释，并提供可能的解决方法。 我认为（不幸的是）最简单的解决方法是接受一个事实，即支持并不容易，但是还有其他方法，如果你真的觉得你需要旧版本。

### 6. 如何在目标应用程序中包含[Visual Studio C ++ 2012/2013 再发布](https://support.microsoft.com/zh-cn/help/2977003/the-latest-supported-visual-c-downloads)？

CefSharp 需要 VC ++运行时 。你必须安装/包括这与你的应用程序的几个选项：

* 你可以在你希望运行 CefSharp 基于应用程序的每台计算机上安装 VC++ 。一旦安装更新就可以通过 Windows Update 进行管理。
* 您可以设置在 Visual Studio C++ 再分发作为安装程序的先决条件（即的ClickOnce或WiX的工具集）
* 通过复制到您的项目在此文件夹（只有当你安装了Visual Studio中的匹配版本存在）的内容：

＃对于VC ++ 2012（86）

C:\Program Files (x86)\Microsoft Visual Studio 11.0\VC\redist\x86\Microsoft.VC110.CRT

＃对于VC ++ 2012（64）

C:\Program Files (x86)\Microsoft Visual Studio 11.0\VC\redist\x64\Microsoft.VC110.CRT

＃对于VC ++ 2013（86）

C:\Program Files (x86)\Microsoft Visual Studio 12.0\VC\redist\x86\Microsoft.VC120.CRT

＃对于VC ++ 2013（64）

C:\Program Files (x86)\Microsoft Visual Studio 12.0\VC\redist\x64\Microsoft.VC120.CRT

随着第三方法，你将不再需要为前提的 Visual C++ 2012/2013 运行时文件 安装到客户端。就目前的计划是留在VC2013官方CefSharp二进制版本。可行时，我们可能会切换到VC2015。如果你从源代码构建自己的你部署的过程中必须与您构建环境。对于官方的 Nuget 版本的详细信息请参见[发布分支表](https://github.com/cefsharp/CefSharp#release-branches)。

欲了解更多信息，请参阅[重新分发 Visual C++ 文件](https://msdn.microsoft.com/zh-cn/library/ms235299.aspx)。要下载，请访问[Visual Studio C++ 2012/2013 再发行](https://support.microsoft.com/zh-cn/help/2977003/the-latest-supported-visual-c-downloads)

一个新的选项是 JetBrains 在 Nuget.org 上有一个[Microsoft.VC120.CRT.JetBrains](https://www.nuget.org/packages/Microsoft.VC120.CRT.JetBrains/12.0.21005.2) ，似乎包含相关文件，您仍然需要连接一些后期构建任务将其复制到您的bin文件夹。

**注意** 当从源代码构建时，确保在部署到没有安装 Visual Studio 的机器时以 Release 模式进行编译。 Visual C++ 使用一套不同的调试和发布版本的运行时库。 Debug 运行时库仅安装有 Visual Studio 中。如果你使用他们已经建在释放模式官方的 NuGet软件包 ，您可以随后建立在调试模式下您的应用程序，因为只有在 Visual C++ 项目需要在 Release 模式下进行编译。

### 7. 如何重写 Javascript 的 window.alert 事件和相似的事件？

ChromiumWebBrowser 控件提供了一个名为 JsDialogHandler 的属性。 该属性可以设置为一个类，可以用来覆盖这些对话框的行为。

### 8. CefSharp3二进制程序在哪里？

CefSharp3被发布为 Nuget 包。 请参阅<https://github.com/cefsharp/CefSharp/blob/master/README.md#nuget-packages> 以获取最新的稳定版本和预发行版本。

对于使用NuGet软件包的一个非常简单的示例项目，请参阅[CefSharp.MinimalExample](https://github.com/cefsharp/CefSharp.MinimalExample) 存储库。 克隆它/下载源码，如果你想要一个真正的小而简单的例子，如何使用CefSharp3。

***请注意：*** 平台目标

使用这些包时，**必须** 选择 x86 或 x64 。 如果您选择 AnyCPU ，NuGet magic将无法正常工作。\_请参阅[steps to configure your solution here](https://github.com/cefsharp/CefSharp/issues/576#issuecomment-61926661)。

### 9. Windows XP/2003支持？

我们不再支持 Windows XP/2003 。官方的 Nuget 软件包没有在 Windows XP/2003 上进行测试。 您可能会喜欢使用目标为 VC++ 2013 的较新版本（ VC++ 2012 packages 有一个已知问题）。（记住 CefSharp.Example.\* 项目是使用 .Net 4.5 编译的，所以如果你想使用提供的例子，你必须重新定位它们并相应地更新代码）。

请不要创建与 Windows XP 相关的 **新的** 问题。 以下 <https://github.com/cefsharp/CefSharp/wiki/Windows-XP-No-Longer-Supported> Wiki 页面可能很有用，可以由任何具有 GitHub 帐户的人进行编辑，因此请随时提供，您认为可能证明有用的任何信息。

### 10. 当我重新发布使用CefSharp的应用程序时，需要包括什么文件？

请参阅: [Output files description (Redistribution)](https://github.com/cefsharp/CefSharp/wiki/Output-files-description-table-(Redistribution))

### 11. 构建过程不能复制CEF文件

有时，编译过程不能复制CEF文件，并且会在最终失败之前重试多次。 即使应用程序关闭，CefSharp.BrowserSubprocess.exe 仍然运行时也会发生这种情况。 解决方案,使用任务管理器，手动杀死 CefSharp.BrowserSubprocess.exe 。

### 12. 为什么 IndexedDB 返回 QuotaExceededError？

使用IndexedDB时，必须将CefSettings CachePath设置为当前用户具有写入权限的目录。 在大多数情况下，您可以使用这样的代码创建缓存目录，并在Initialize() 期间将设置传递给Cef：

// On Win7 this will create a directory in AppData.

var cache = System.IO.Path.Combine(Environment.GetFolderPath (Environment.SpecialFolder.ApplicationData), System.IO.Path.Combine("MyApplication","cache"));

if (!System.IO.Directory.Exists (cache))

System.IO.Directory.CreateDirectory (cache);

// Set the CachePath during initialization.

var settings = new CefSettings(){CachePath = cache};

Cef.Initialize (settings);

### 13. 在C＃中如何处理Javascript事件？

public class BoundObject

{

public void OnFrameLoadEnd (object sender, FrameLoadEndEventArgs e)

{

if(e.Frame.IsMain)

{

browser.ExecuteScriptAsync(@"

document.body.onmouseup = function()

{

bound.onSelected(window.getSelection().toString());

}

");

}

}

public void OnSelected(string selected)

{

MessageBox.Show("The user selected some text [" + selected + "]");

}

}

// ...

// After your ChromiumWebBrowser has been instantiated (for WPF directly after `InitializeComponent();` in the control constructor).

var obj = new BoundObject();

browser.RegisterJsObject("bound", obj);

browser.FrameLoadEnd += obj.OnFrameLoadEnd ;

***请注意：***

* 在创建 ChromiumWebBrowser 实例之后，应该直接调用 RegisterJsObject
* "bound" 将作为可在网页上运行的任何JS脚本访问的顶级对象创建的对象的名称。 您可以使用任何您喜欢的，但确保它不与现有的JS对象冲突，并且名称与C＃和JS匹配。
* 此示例使用 FrameLoadEnd 事件即时将JS代码注入到网页中。 如果您控制了网页，可以将JS代码添加到网页中。
* 我们使用单个字符串参数调用 OnSelected C＃方法。 您可以使用任何数量的参数，也可以不使用任何参数，但都必须是原始类型。
* 确保将复杂的JS对象转换为字符串，或将其转换为JSON。
* 由于CefSharp将通过其名称搜索 OnSelected 方法，请确保您的混淆器不会将该方法重命名为别的东西，否则将不会被执行。

### 14. 如何实现 WinForms 拖放？

一个可能的解决方案概述 <https://github.com/cefsharp/CefSharp/issues/1593#issuecomment-304451518>

## 10、参考网址

<http://www.cnblogs.com/TianFang/p/4658151.html>