

官方: [参考地址](#)

编译分三大步:

- 一、配置必须的环境
- 二、下载
- 三、编译

一、准备工作

- 1.Win10系统, 必须64位, 至少8GB的RAM, 我采用win10 64位, 16GRAM
- 2.比较新的VS,最近免费的社区版 (编译不同版本要求不一样, 具体看CEF官网帮助, 我用的是VS2019), 需要安装“C++桌面组件”和“MFC和ATL支持”,最好安装在默认路径
- 3.Win10 SDK (VS2019中有)
- 4.至少200G剩余空间, NTFS文件系统, 部分文件超过4G, 部分资料显示最少60G, 编译结束后发现远超60G
- 5.设置系统区域为英语 (美国)。(控制面板-区域-管理-更改系统区域设置-英语 (美国)), 设置完需要重启
- 6.创建工作目录, 路径不能包含空格和特殊字符。例如f:\cef
- 7.下载 [编译工具包](#), 解压缩至工作目录。例如f:\cef\depot_tools

建议从git仓库获取最新的:

```
1 git clone https://chromium.googlesource.com/chromium/tools/depot_tools.git
4
```

8.下载 [编译脚本](#) 至工作目录。例如f:\cef

9.在工作目录下创建源码目录。例如f:\cef\source

10.安装ninja,并加到系统环境变量下载地址:[ninja](#)

11.一个稳定的 VPN 网络可以让你快速下载 CEF 和 Chromium 代码

二、配置环境变量

1.配置depot tools



2.配置ninja环境变量

三、配置下载脚本并下载

新建一个 bat脚本并命名为 git_4430.bat

在脚本中写入如下代码

```
1 set CEF_USE_GN=1
2 set GN_DEFINES=is_official_build=true proprietary_codecs=true ffmpeg_brandin
3 set GYP_DEFINES=buildtype=Official
4 set GYP_MSVS_VERSION=2019
5 set CEF_ARCHIVE_FORMAT=tar.bz2
6 set GN_ARGUMENTS=--ide-vs2019 --sln-cef --filters=//cef/*
7 python automate-git.py --download-dir=f:\cef\source --depot-tools-dir=f:\cef
8 pause
```

代理运行(需要管理员):

```
1 echo proxy setup ...
2
3 netsh winhttp set proxy 127.0.0.1:10809
4 set https_proxy=http://127.0.0.1:10809
5 set http_proxy=http://127.0.0.1:10809
6 echo env setup ...
7
8 set CEF_USE_GN=1
9 set GN_DEFINES=is_official_build=true proprietary_codecs=true ffmpeg_brandin
10 set GYP_DEFINES=buildtype=Official
11 set GYP_MSVS_VERSION=2019
12 set CEF_ARCHIVE_FORMAT=tar.bz2
13 set GN_ARGUMENTS=--ide-vs2019 --sln-cef --filters=//cef/*
14 python automate-git.py --download-dir=f:\cef\source --depot-tools-dir=f:\cef
15 pause
```

10809端口要改成自己实际的

两个脚本根据实际情况选择其一

GN_DEFINES 参数介绍:

ffmpeg_branding 和 proprietary_codecs 表示开启多媒体编解码支持, 但默认仅支持一小部分, 后面介绍如何支持更多。

is_official_build 决定了是否是编译正式版本, 指定该参数为 true 基本上都是为了产品发布使用, 同时也会在创建解决方案的时候生成带有 sandbox 的解决方案 (如 Release_GN_x86_sandbox), 而不指定这个参数是没有的。如果你不是为了贡献代码, 那默认这个都加上吧。

use_jumbo_build 官方资料默认指定, 表示是否启用试验性的 jumbo 编译, 编译过程会加快很多 (至少快 1 小时), 但是占用 CPU 和内存 (尤其是内存) 会翻倍。

is_component_build 官方资料默认指定, 但我们没有开启, 这个参数表示是否启用组件化编译, 设置为 true 以后, base、ffmpeg 等等都会被编译为动态库, 使用时也是动态链接, 编译出来的 cef_sandbox.lib 只有几兆大小, 并且你需要复制很多 dll 文件到项目目录下才能运行。

automate-git.py 参数介绍

--branch 表示你要下载哪个版本的代码, CEF 每个版本都有固定的分支, 你去 CEF 项目页面看分支名称指定即可, 这里我们编译 2019年9月份目前最新的版本 3809。

--no-build 表示只下载代码而不编译, 这里只为下载代码, 我们还要修改支持多媒体的参数, 所以不进行编译

--no-distrib 不执行打包项目, 这里只为下载代码, 我们还要修改支持多媒体的参数, 所以不进行打包

--force-clean 如果你曾经执行过这个脚本, 可能会出错, 则加上这个参数, 它执行清理残留文件 (你也可以手动在 chromium 源码目录执行git clean -xdf来清理目录中的多余内容)。

automate-git.py 的其他参数可以手动执行 python automate-git.py --help 来查看

注:

如果 VPN 不太稳定, 可能会经常中断, 造成脚本运行错误, 重新运行脚本即可, 源码主要有三部分, depot_tools源码包和chromium源码, 如果发现相应部分的源码已经同步完成, 可以分别使用--no-depot-tools-update, --no-cef-update, --no-chromium-update 来不进行相应部分的代码更新。

自动构建脚本在下载源码时, 并没有进度提示, 比如下下载chromium/src目录时会非常长, 会一直提示still working on src..., 请耐心等待。

3.cmd (管理员权限运行) git_4430.bat脚本

```
1 完整目录结构:
2 f:/
3   cef/
4   automate-git.py
5   depot_tools/
6   source/
```

四、修改文件使得CEF支持MP4能力

修改: source\chromium\src\third_party\ffmpeg\chromium\scripts\build_ffmpeg.py 文件

```
1 configure_flags['Chrome'].extend([
2     '--enable-decoder=aac,h264,mp3',
3     '--enable-demuxer=aac,mp3,mov',
4     '--enable-parser=aac,h264,mpeg4audio',
5 ])
```

改为

```
1 configure_flags['Chrome'].extend([
2     '--enable-decoder=aac,h264,mp3,mpeg4,amrnb,amrwb,flv',
3     '--enable-demuxer=aac,mp3,mov,avi,amr,flv',
4     '--enable-parser=aac,h264,mpeg4audio,mpeg4video,h263',
5 ])
```

分别打开:

f:\cef\source\chromium\src\third_party\ffmpeg\chromium\config\Chrome\win\ia32\config.h

f:\cef\source\chromium\src\third_party\ffmpeg\chromium\config\Chrome\win\x64\config.h

在原有配置宏FFMPEG_CONFIGURATION里增加以下:

```
1 --enable-decoder='rv10,rv20,rv30,rv40,cook,h263,mpeg4,msmpeg4v1,msmpeg4v2,
2 --enable-demuxer='rm,mpegvideo,avi,avisynth,h263,aac,amr,ac3,flv,mpgts,mpe
3 --enable-parser='mpegvideo,rv30,rv40,h263,mpeg4video,ac3'
```

注: 但在90.6.7中看到FFMPEG_CONFIGURATION节点是被禁用的

因在90.6.7中看到FFMPEG_CONFIGURATION是被禁用因此我们使用下面这种方式

分别修改:

f:\cef\source\chromium\src\third_party\ffmpeg\chromium\config\Chrome\win\ia32\config.h

f:\cef\source\chromium\src\third_party\ffmpeg\chromium\config\Chrome\win\x64\config.h

把部分#define 由0改为1

```
1 #define CONFIG_FLV_DECODER 1
2 #define CONFIG_H263_DECODER 1
3 #define CONFIG_H263I_DECODER 1
4 #define CONFIG_MPEG4_DECODER 1
5 #define CONFIG_MSMPEG4VIDEO_DECODER 1
6 #define CONFIG_MSMPEG4V1_DECODER 1
7 #define CONFIG_MSMPEG4V2_DECODER 1
8 #define CONFIG_MSMPEG4V3_DECODER 1
9 #define CONFIG_RV10_DECODER 1
10 #define CONFIG_RV20_DECODER 1
11 #define CONFIG_RV30_DECODER 1
12 #define CONFIG_RV40_DECODER 1
13 #define CONFIG_AC3_DECODER 1
14 #define CONFIG_AMRNB_DECODER 1
15 #define CONFIG_AMRWB_DECODER 1
16 #define CONFIG_COOK_DECODER 1
17 #define CONFIG_STPR_DECODER 1
18 #define CONFIG_FLV_ENCODER 1
19 #define CONFIG_H263_ENCODER 1
20 #define CONFIG_MPEG4_ENCODER 1
21 #define CONFIG_MSMPEG4V2_ENCODER 1
22 #define CONFIG_MSMPEG4V3_ENCODER 1
23 #define CONFIG_RV10_ENCODER 1
24 #define CONFIG_RV20_ENCODER 1
25 #define CONFIG_AAC_ENCODER 1
26 #define CONFIG_AC3_PARSER 1
27 #define CONFIG_COOK_PARSER 1
28 #define CONFIG_H263_PARSER 1
29 #define CONFIG_MPEG4VIDEO_PARSER 1
30 #define CONFIG_MPEGVIDEO_PARSER 1
31 #define CONFIG_RV30_PARSER 1
32 #define CONFIG_RV40_PARSER 1
33 #define CONFIG_STPR_PARSER 1
34 #define CONFIG_AAC_DEMUXER 1
35 #define CONFIG_AMR_DEMUXER 1
36 #define CONFIG_AMRNB_DEMUXER 1
37 #define CONFIG_AMRWB_DEMUXER 1
38 #define CONFIG_AVI_DEMUXER 1
39 #define CONFIG_AVISYNTH_DEMUXER 1
40 #define CONFIG_FLV_DEMUXER 1
41 #define CONFIG_H263_DEMUXER 1
42 #define CONFIG_MPEGTS_DEMUXER 1
43 #define CONFIG_MPEGVIDEO_DEMUXER 1
44 #define CONFIG_RM_DEMUXER 1
45 #define CONFIG_AC3_MUXER 1
46 #define CONFIG_AMR_MUXER 1
47 #define CONFIG_AVI_MUXER 1
48 #define CONFIG_FLV_MUXER 1
49 #define CONFIG_H263_MUXER 1
50 #define CONFIG_MPEGTS_MUXER 1
51 #define CONFIG_RM_MUXER 1
```

因为改的比较多, 这里我们通过python的脚本帮我们修改

1.新建一个mp4_support.txt把上面的内容复制进去

```
1 # 第一参数为上面要修改的define 的文件路径, 第二参数为 F:\cef\source\chromium\
2 import sys
3 import shutil
4 import re
5 import os
6
7 def replace(change,content):
8     str_array = re.findall("#define%s\w+.*",change)
9     str_replace=str_array[0]
10    str_replace+="0"
11    str_dest=str_array[0]
12    str_dest+="1"
13    return content.replace(str_replace,str_dest)
14
15 if len(sys.argv) > 2 :
16     src_file_name =sys.argv[1]
17     dest_file_name=sys.argv[2]
18 else:
19     src_file_name =raw_input("Please input src file path name:").replace("\n",
20     dest_file_name =raw_input("Please input dest file path name:").replace("\n",
21
22 file_src_handle = open(src_file_name,"r")
23 file_src_lines = file_src_handle.readlines()
24 file_src_handle.close()
25 file_dest_handle = open(dest_file_name,"r")
26 dest_file_content = file_dest_handle.read()
27 file_dest_handle.close()
28
29 for line in file_src_lines:
30     dest_file_content = Replace(line,dest_file_content)
31
32 write_file_path = os.getcwd()+"\\%s.path.basename(dest_file_name)
33 ready_copy = open(write_file_path,"w")
34 ready_copy.write(dest_file_content)
35 ready_copy.close()
36
37 shutil.copy(write_file_path,dest_file_name)
38 os.remove(write_file_path)
39 print("Support mp4 Success!!!")
```

2.新建一个update_mp4.py的脚本, 内容如下:

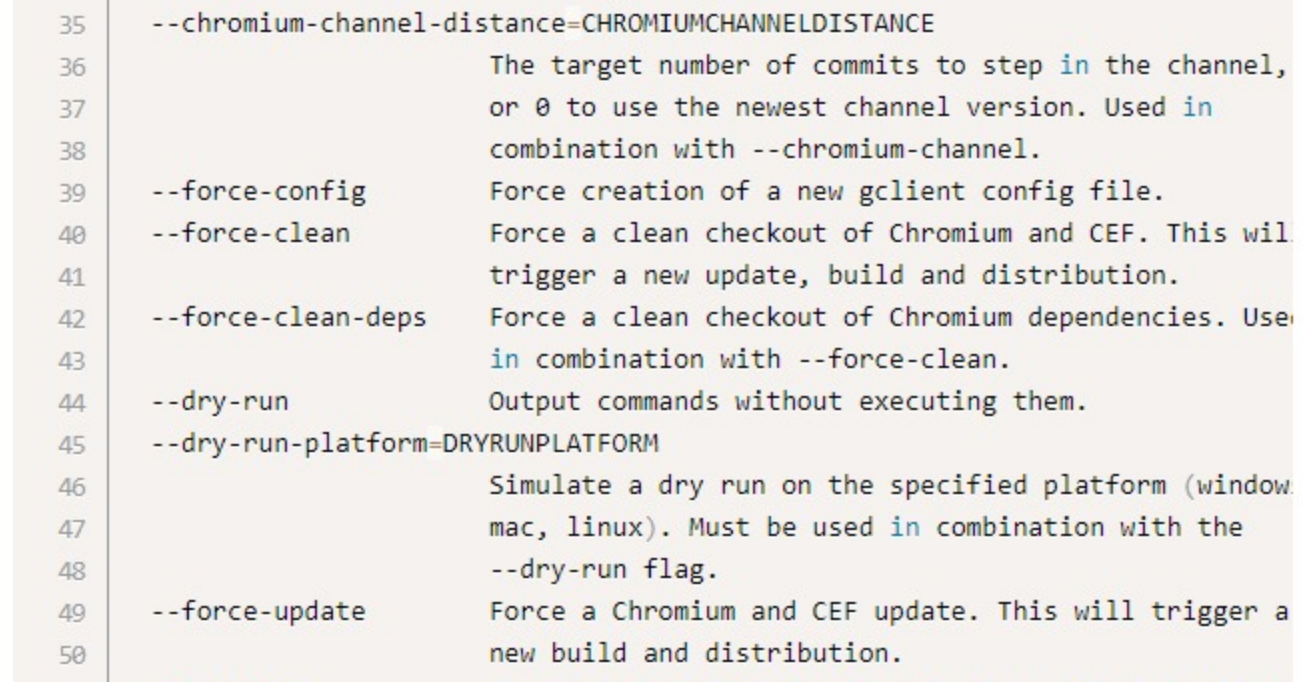
```
1 # 第一参数为上面要修改的define 的文件路径, 第二参数为 F:\cef\source\chromium\
2 import sys
3 import shutil
4 import re
5 import os
6
7 def replace(change,content):
8     str_array = re.findall("#define%s\w+.*",change)
9     str_replace=str_array[0]
10    str_replace+="0"
11    str_dest=str_array[0]
12    str_dest+="1"
13    return content.replace(str_replace,str_dest)
14
15 if len(sys.argv) > 2 :
16     src_file_name =sys.argv[1]
17     dest_file_name=sys.argv[2]
18 else:
19     src_file_name =raw_input("Please input src file path name:").replace("\n",
20     dest_file_name =raw_input("Please input dest file path name:").replace("\n",
21
22 file_src_handle = open(src_file_name,"r")
23 file_src_lines = file_src_handle.readlines()
24 file_src_handle.close()
25 file_dest_handle = open(dest_file_name,"r")
26 dest_file_content = file_dest_handle.read()
27 file_dest_handle.close()
28
29 for line in file_src_lines:
30     dest_file_content = Replace(line,dest_file_content)
31
32 write_file_path = os.getcwd()+"\\%s.path.basename(dest_file_name)
33 ready_copy = open(write_file_path,"w")
34 ready_copy.write(dest_file_content)
35 ready_copy.close()
36
37 shutil.copy(write_file_path,dest_file_name)
38 os.remove(write_file_path)
39 print("Support mp4 Success!!!")
```

3.新建一个bat 脚本用于启动我们的update_mp4.py脚本(也可以直接通过cmd进行启动) bat脚本内容如下:

```
1 python update_mp4.py F:\cef\script\mp4_support.txt F:\cef\source\chromium\
2 pause
```

4.分别执行两个脚本

目录如下:



五、创建编译脚本

新建两个脚本分别对应x86、x64

build_x64_4430.bat文件内容如下:

```
1 set CEF_USE_GN=1
2 set GN_DEFINES=is_official_build=true proprietary_codecs=true ffmpeg_brandin
3 set GYP_DEFINES=buildtype=Official
4 set GYP_MSVS_VERSION=2019
5 set CEF_ARCHIVE_FORMAT=tar.bz2
6 set DEPOT_TOOLS_WIN_TOOLCHAIN=0
7 set GYP_GENERATORS=ninja,msvs-ninja
8 set GN_ARGUMENTS=--ide-vs2019 --sln-cef --filters=//cef/*
9 set WIN_CUSTOM_TOOLCHAIN=1
10 set CEF_VCVARS=none
11 set GYP_MSVS_OVERRIDE_PATH=C:\Program Files (x86)\Microsoft Visual Studio\20
12 set SDK_ROOT=C:\Program Files (x86)\Windows Kits\10
13 set INCLUDE=C:\Program Files (x86)\Windows Kits\10\Include\10.0.19041.0\x86;C:\Pro
14 set PATH=C:\Program Files (x86)\Windows Kits\10\bin\10.0.19041.0\x86;C:\Prog
15 set LIB=C:\Program Files (x86)\Windows Kits\10\Lib\10.0.19041.0\um\x86;C:\Pr
16 set VS_CRT_ROOT=C:\Program Files (x86)\Microsoft Visual Studio\2019\Enterpri
17
18 python automate-git.py --download-dir=f:\cef\source --depot-tools-dir=f:\cef
19 pause
```

build_x86_4430.bat文件内容如下:

```
1 set CEF_USE_GN=1
2 set GN_DEFINES=is_official_build=true proprietary_codecs=true ffmpeg_brandin
3 set GYP_DEFINES=buildtype=Official
4 set GYP_MSVS_VERSION=2019
5 set CEF_ARCHIVE_FORMAT=tar.bz2
6 set DEPOT_TOOLS_WIN_TOOLCHAIN=0
7 set GYP_GENERATORS=ninja,msvs-ninja
8 set GN_ARGUMENTS=--ide-vs2019 --sln-cef --filters=//cef/*
9 set WIN_CUSTOM_TOOLCHAIN=1
10 set CEF_VCVARS=none
11 set GYP_MSVS_OVERRIDE_PATH=C:\Program Files (x86)\Microsoft Visual Studio\20
12 set SDK_ROOT=C:\Program Files (x86)\Windows Kits\10
13 set INCLUDE=C:\Program Files (x86)\Windows Kits\10\Include\10.0.19041.0\x86;C:\Pro
14 set PATH=C:\Program Files (x86)\Windows Kits\10\bin\10.0.19041.0\x86;C:\Prog
15 set LIB=C:\Program Files (x86)\Windows Kits\10\Lib\10.0.19041.0\um\x86;C:\Pr
16 set VS_CRT_ROOT=C:\Program Files (x86)\Microsoft Visual Studio\2019\Enterpri
17
18 python automate-git.py --download-dir=f:\cef\source --depot-tools-dir=f:\cef
19 pause
```

注:根据自己的VS版本、路径、Win10 SDK版本、路径进行对应更改

备注: bat中的选项可以有:

--download-dir 源码下载目录

--depot-tools-dir 工具包目录

--branch 源码分支(默认trunk主分支)

--checkout 指定patch版本 (默认最近提交)

--no-build 下载完不自动开始编译 (我们需要修改编译配置)

--no-update 确定源码下载完仅重新编译的使用

--force-build 强制编译 (发现在有成功编译的时候再编译不会执行, 可以加上这个

--no-debug-build 只编译release版本

--no-release-build 只编译debug版本

六、执行对应环境的脚本进入漫长的等待

七.编译完成

输出目录为source\chromium\src\out\Release.exe可测试, x86 和 source\chromium\src\cef\binary_distrib, Release_GN_x86下cefclient.exe可以测试, binary_distrib下有cef_binary_90.6.7-g19ba721+chromium-90.0.4430.212_windows32

通过cmd 运行cefclient.exe -url = <http://html5test.com> -single-process 结果如下:


```
1 automate-git.py 参数
2
3 Usage: automate-git.py [options]
4
5 This utility implements automation for the download, update, build and
6 distribution of CEF.
7
8 Options:
9 -h, --help show this help message and exit
10 --download-dir=DIR Download directory with no spaces [required].
11 --depot-tools-dir=DIR Download directory for depot_tools.
12 --depot-tools-archive=DEPOTTOOLSOURCE Zip archive file that contains a single top-level depot_tools directory.
13 --branch=BRANCH Branch of CEF to build (master, 3987, ...). This will be used to name the CEF download directory and to identify the correct URL if --url is not specified. The default value is master.
14 --url=URL CEF download URL. If not specified the default URL will be used.
15 --chromium-url=CHROMIUMURL Chromium download URL. If not specified the default URL will be used.
16 --checkout=CHECKOUT Version of CEF to checkout. If not specified the most recent remote version of the branch will be used.
17 --chromium-checkout=CHROMIUMCHECKOUT Version of Chromium to checkout (Git branch/hash/tag This overrides the value specified by CEF in CHROMIUM_BUILD_COMPATIBILITY.txt.
18 --chromium-channel=CHROMIUMCHANNEL Chromium channel to check out (canary, dev, beta or stable). This overrides the value specified by CEF in CHROMIUM_BUILD_COMPATIBILITY.txt.
19 --chromium-channel-distance=CHROMIUMCHANNELDISTANCE The target number of commits to step in the channel, or 0 to use the newest channel version. Used in combination with --chromium-channel.
20 --force-config Force creation of a new gclient config file.
21 --force-clean Force a clean checkout of Chromium and CEF. This will trigger a new update, build and distribution.
22 --force-clean-deps Force a clean checkout of Chromium dependencies. Use in combination with --force-clean.
23 --dry-run Output commands without executing them.
24 --dry-run-platform=DRYRUNPLATFORM Simulate a dry run on the specified platform (windows, mac, linux). Must be used in combination with the --dry-run flag.
25 --force-update Force a Chromium and CEF update. This will trigger a new build and distribution.
26 --no-update Do not update Chromium or CEF. Pass --force-build or --force-distrib if you desire a new build or distribution.
27 --no-cef-update Do not update CEF. Pass --force-build or --force-distrib if you desire a new build or distribution.
28 --force-cef-update Force a CEF update. This will cause local changes in the CEF checkout to be discarded and patch files to be reapplied.
29 --no-chromium-update Do not update Chromium.
30 --no-depot-tools-update Do not update depot_tools.
31 --fast-update Update existing Chromium/CEF checkouts for fast incremental builds by attempting to minimize the number of modified files. The update will fail if there are unstaged CEF changes or if Chromium change are not included in a patch file.
32 --force-patch-update Force update of patch files.
33 --resave Resave patch files.
34 --log-chromium-changes Create a log of the Chromium changes.
35 --force-build Force CEF debug and release builds. This builds [build-target] on all platforms and chrome_sandbox on Linux.
36 --no-build Do not build CEF.
37 --build-target=BUILDTARGET Target name(s) to build (defaults to "cefclient").
38 --build-tests Also build the test target specified via --test-target.
39 --no-debug-build Don't perform the CEF debug build.
40 --no-release-build Don't perform the CEF release build.
41 --verbose-build Show all command lines while building.
42 --build-failure-limit=BUILDFAILURELIMIT Keep going until N jobs fail.
43 --build-log-file Write build logs to file. The file will be named "build-[branch]-[debug|release].log" in the download directory.
44 --x64-build Create a 64-bit build.
45 --arm-build Create an ARM build.
46 --arm64-build Create an ARM64 build.
47 --run-tests Run the ceftests target.
48 --no-debug-tests Don't run debug build tests.
49 --no-release-tests Don't run release build tests.
50 --test-target=TESTTARGET Test target name to build (defaults to "ceftests").
51 --test-prefix=TESTPREFIX Prefix for running the test executable (e.g. "xvfb" on Linux).
52 --test-args=TESTARGS Arguments that will be passed to the test executable
53 --force-distrib Force creation of a CEF binary distribution.
54 --no-distrib Don't create a CEF binary distribution.
55 --minimal-distrib Create a minimal CEF binary distribution.
56 --minimal-distrib-only Create a minimal CEF binary distribution only.
57 --client-distrib Create a client CEF binary distribution.
58 --client-distrib-only Create a client CEF binary distribution only.
59 --sandbox-distrib Create a cef_sandbox static library distribution.
60 --sandbox-distrib-only Create a cef_sandbox static library distribution only.
61 --no-distrib-docs Don't create CEF documentation.
62 --no-distrib-archive Don't create archives for output directories.
63 --clean-artifacts Clean the artifacts output directory.
64 CEF distrib cef name, child of chromium/src/cef/binary_distrib
65 --distrib-subdir=PREFIX=PREFIX CEF distrib dir name suffix, child of chromium/src/cef/binary_distrib
```

参考地址: [参考地址](#) [参考地址](#)