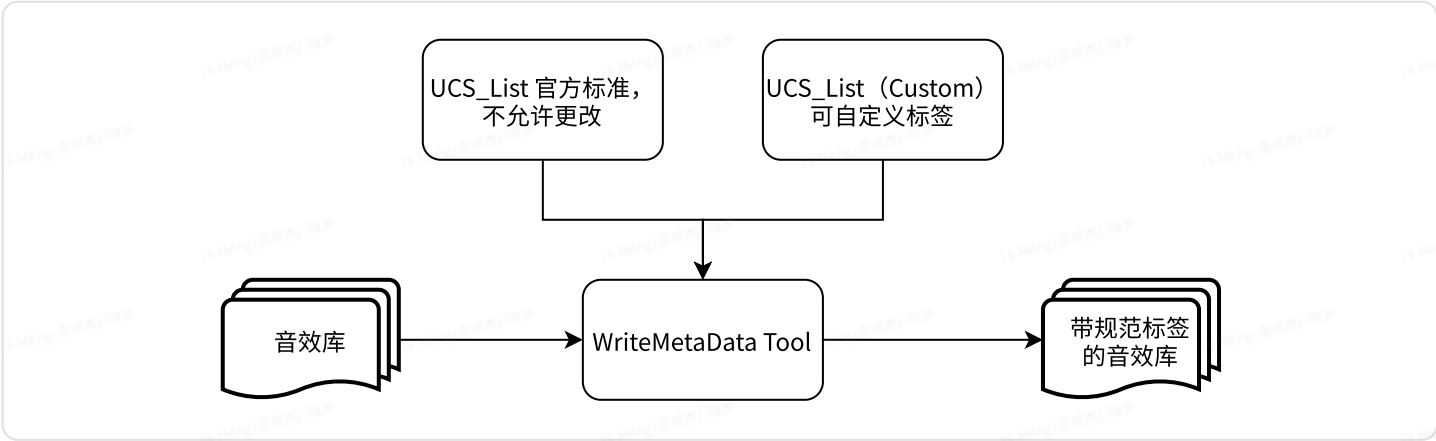


Writemetadata Readme



关于UCS



通用类别系统（UCS）是由[Tim Nielsen](#)，[Justin Drury](#)，[Kai Paquin](#)等人发起的公共领域倡议，并得到来自世界各地的声音图书馆员，供应商和用户的支持。

我们的目标是提供并鼓励使用一组类别列表来分类音效。我们希望通过这样做，我们可以提供一个框架来一致地分类声音效果，提供文件名结构的统一性，并减轻维护声音效果库的痛苦。

我们还希望提供工具，使维护自己的个人或专业库的每个人都更容易命名和分类音效。

Category	SubCategory	CatID	CatShort	Synonyms - Comma Separated
AIR	BLOW	AIRBlow	AIR	compressed air, depressurise, release,
AIR	BURST	AIRBrst	AIR	air blast, air poof, chuff, spurt, bla
AIR	HISS	AIRHiss	AIR	air release, exhaust, expel
AIR	MISC	AIRMisc	AIR	miscellaneous
AIR	SUCTION	AIRSuck	AIR	vacuum, suck, suction
AIRCRAFT	HELICOPTER	AEROHeli	AERO	gyrocopter, whirlybird, chopper, ghet
AIRCRAFT	INTERIOR	AEROInt	AERO	jet, airplane, aircraft, passenger je
AIRCRAFT	JET	AEROJet	AERO	airliner, jetliner, jumbo jet, learje
AIRCRAFT	MECHANISM	AEROMech	AERO	landing gear, cockpit lever, rudder, :
AIRCRAFT	MILITARY	AEROMil	AERO	fighter jet, bomber, stealth bomber, b
AIRCRAFT	MISC	AEROMisc	AERO	blimp, glider, hot air balloon, ultra
AIRCRAFT	PROP	AEROProp	AERO	biplane, triplane, turboprop, stunt p
AIRCRAFT	RADIO CONTROLLED	AERORadio	AERO	quadcopters, uav, unmanned aerial veh
AIRCRAFT	ROCKET	AERORckt	AERO	rocket, missile, jetpack, space shuttl
ALARMS	BELL	ALRMBell	ALRM	fire bell, school bell, railroad cross
ALARMS	BUZZER	ALRMBuzr	ALRM	door buzzer, game show buzzer
ALARMS	CLOCK	ALRMClok	ALRM	digital alarm clock, alarm clock
ALARMS	ELECTRONIC	ALRMElec	ALRM	fire detector
ALARMS	MISC	ALRMMisc	ALRM	miscellaneous
ALARMS	SIREN	ALRMSirn	ALRM	klaxon, ahooga horn, air raid siren
AMBIENCE	AIR	AMBAir	AMB	desert, winter, quiet air, night air,
AMBIENCE	ALPINE	AMBAlpn	AMB	alps, mountains, rockies, tetons, mow

FILENAME	DESCRIPTION
Armor Chainmail Impact 01.wav	ARMOUR CHAIN MAIL IMPACT Hitting fine chain mail
Armor Chainmail Impact 02.wav	ARMOUR CHAIN MAIL IMPACT Hitting chain mail with bigger rings
Armor Full Plate Impact 01.wav	ARMOUR FULL PLATE IMPACT Hitting loose full plate
Armor Full Plate Impact 02.wav	ARMOUR FULL PLATE IMPACT Hitting large full plate
Armor Half Plate Impact 01.wav	ARMOUR HALF PLATE IMPACT Hitting armour with chain, leather and plate elements
Armor Half Plate Impact 02.wav	ARMOUR HALF PLATE IMPACT Hitting tighter half plate armour
Armor Leather Impact 01.wav	ARMOUR LEATHER IMPACT Hitting leather body, short whoosh in
Armor Leather Impact 02.wav	ARMOUR LEATHER IMPACT Hitting smaller leather armour, short whoosh in
Axe Medium Hit.wav	AXE MEDIUM HIT Medium bone breaks, short scrapes
Axe Small Hit.wav	AXE SMALL HIT Short impact, blood splatter and short scrapes
Axe Throwing.wav	AXE THROWING Multi rotating whoosh in, metallic impact
Axe Two-Handed Hit.wav	AXE TWO HANDED HIT Large axe impact, medium whoosh in
Bow Arrow Flyby Air.wav	BOW ARROW FLY BY AIR Neutral arrow or bolt flyby
Bow Arrow Flyby Fire Hard.wav	BOW ARROW FLY BY FIRE HARD Crackling, fast burning arrow flyby
Bow Arrow Flyby Fire Soft.wav	BOW ARROW FLY BY FIRE SOFT Slower and less crackling burning arrow flyby
Bow Arrow Flyby Whistle.wav	BOW ARROW FLY BY WHISTLE Whistling arrow or bolt flyby
Bow Arrow Impact Flesh.wav	BOW ARROW IMPACT FLESH Whistling fly in, blood drops
Bow Arrow Impact Metal.wav	BOW ARROW IMPACT METAL Whistling fly in, shield or plate armour hit
Bow Arrow Impact Wood.wav	BOW ARROW IMPACT WOOD Whistling fly in, vibrating arrow hit
Bow Fire Arrow Shot.wav	BOW FIRE ARROW SHOT Shooting burning arrows
Bow Flatbow Shot.wav	BOW FLATBOW SHOT Light bow shots, whistling arrows
Bow Loading.wav	BOW LOADING Charging a bow, leather and rope creaks with wood elements
Bow Longbow Shot.wav	BOW LONGBOW SHOT Heavy bow shots, airy neutral arrows
Bow Recurve Shot.wav	BOW RECURVE SHOT Medium bow shots, short metallic rattling
Crossbow Bolt Flyby.wav	CROSSBOW BOLT FLY BY Fast whistling bolt or arrow flybys
Crossbow Bolt Impact Flesh.wav	CROSSBOW BOLT IMPACT FLESH Neutral fly in, hit with small metal element

示例: *Medieval_Weapons_DS_Metadata.pdf*

关于Writemetadata Tool

对于符合UCS命名规则的资源，使用 UCS Metadata 元数据渲染脚本，可以快速地把元数据写到资源中，支持在REAPER媒体浏览器以及Soundminer中读取。

核心第三方库

<https://www.npmjs.com/package/ffmetadata>

```
ffmetadata.read(file, [options], callback)
```

The `callback` function is called (`callback(err, data)`) with an error or an object containing metadata from `file`.

If `options.dryRun` is truthy, the `ffmpeg` process is not actually invoked and instead an array of the arguments that *would* have been used to invoke `ffmpeg` is returned synchronously. The `callback` argument is not used in this case.

`options.coverPath`: Option to provide a path, where `ffmpeg` will save cover image. Unfortunately, `ffmpeg` will always convert resulted image, based on extension in provided `coverPath`.

```
ffmetadata.write(file, data, [options], callback)
```

Write metadata to `file` and optionally append additional attachments (e.g., artwork image).

The `data` object should contain metadata fields supported by FFmpeg. See the [metadata section](#) above for more information.

The `options` object may be provided with any of the following fields:

- `options.attachments` : An array of files that should be appended as additional streams to the output. This can be used to e.g., attach artwork images; see the [artwork section](#) above for more information.
- `options["id3v1"]` : If this property is truthy, id3 v1 will also be included. This is useful if compatibility with *Windows Explorer* is desired.
- `options["id3v2.3"]` : If this property is truthy, id3 v2.3 will be used (instead of the default of v2.4). This is useful if compatibility with *Windows Explorer* is desired (see [#8](#)).
- `options.dryRun` : If this property is truthy, the `ffmpeg` process is not actually invoked and instead an array of the arguments that *would* have been used to invoke `ffmpeg` is returned synchronously. The `callback` argument is not used in this case.

`callback(err)` is called when finished, with an error if one occurred.

界面概览：



安装及配置环境：

1、~~（不用装了）~~安装node.js

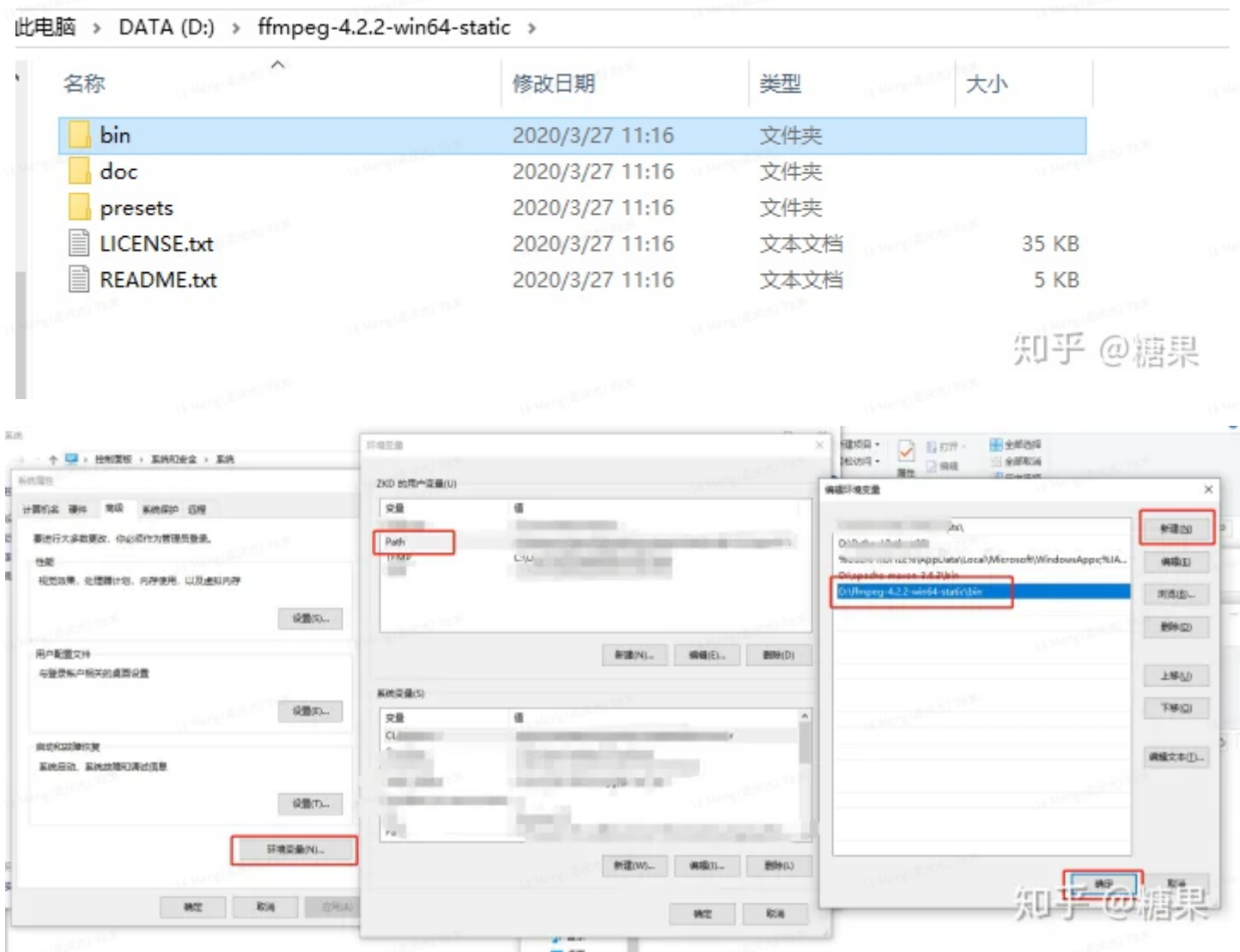
windows:  [node-v16.18.0-x86.msi](#)

mac:  [node-v18.12.1.pkg](#)

2、安装ffmpeg并配置环境 lil Meng(孟庆杰)

windows:  [ffmpeg-5.1.2-full_build.7z](#)

1. 下载完成后直接进行解压，然后配置环境变量点击系统属性窗口中的环境变量按钮。在“用户变量”区域选择Path变量，点击编辑按钮，在其原始内容后输入ffmpeg的bin路径，例如我的就是“D:\ffmpeg-4.2.2-win64-static\bin”。下面有图示



2. 打开命令提示符窗口。输入命令“`ffmpeg -version`”。如果命令提示窗口返回FFmpeg的版本信息，那么就说明安装成功了。


```
管理员: 命令提示符
Microsoft Windows [版本 10.0.18362.720]
(c) 2019 Microsoft Corporation. 保留所有权利。

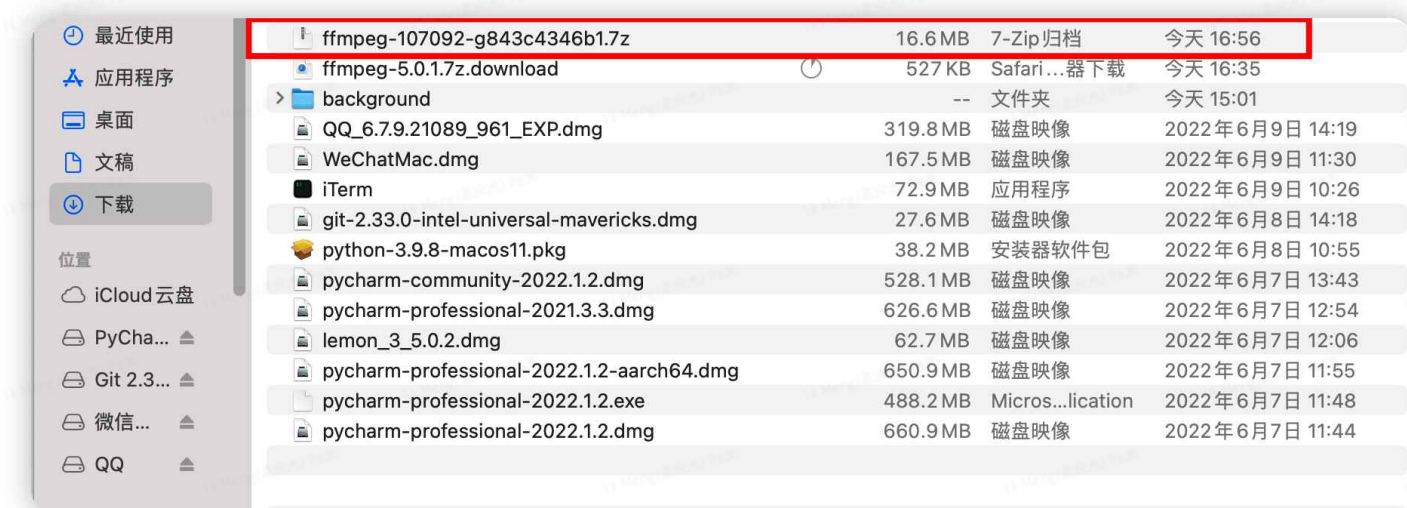
C:\Windows\system32>ffmpeg -version
ffmpeg version 4.2.2 Copyright (c) 2000-2019 the FFmpeg developers
built with gcc 9.2.1 (GCC) 20200122
configuration: --enable-gpl --enable-version3 --enable-sdl2 --enable-fontconfig --enable-gnutls --enable-iconv --enable-
libass --enable-libdav1d --enable-libbluray --enable-libfreetype --enable-libmp3lame --enable-libopencl --enable-
libopenjpeg --enable-libopenm2 --enable-libopus --enable-libshine --enable-lsnp --enable-libsoxr --enable-
libtheora --enable-libtwolame --enable-libvpx --enable-libwavpack --enable-libwebp --enable-libx264 --enable-libx265 --e
nable-libxml2 --enable-libzimg --enable-lzma --enable-zlib --enable-gmp --enable-libvidstab --enable-libvorbis --enable-
libvo-amrwbenc --enable-libmysofa --enable-libspeex --enable-libxvid --enable-libaom --enable-libbrotli --enable-amf --enab
le-ffnvcodec --enable-cuda-llvm --enable-d3d11va --enable-nvenc --enable-nvdec --enable-dxva2 --enable-avisynth --enable-lib
openmpt
libavutil      56. 31.100 / 56. 31.100
libavcodec     58. 54.100 / 58. 54.100
libavformat    58. 29.100 / 58. 29.100
libavdevice    58.  8.100 / 58.  8.100
libavfilter     7. 57.100 /  7. 57.100
libswscale     5.  5.100 /  5.  5.100
libswresample  3.  5.100 /  3.  5.100
libpostproc   55.  5.100 / 55.  5.100

C:\Windows\system32>
```

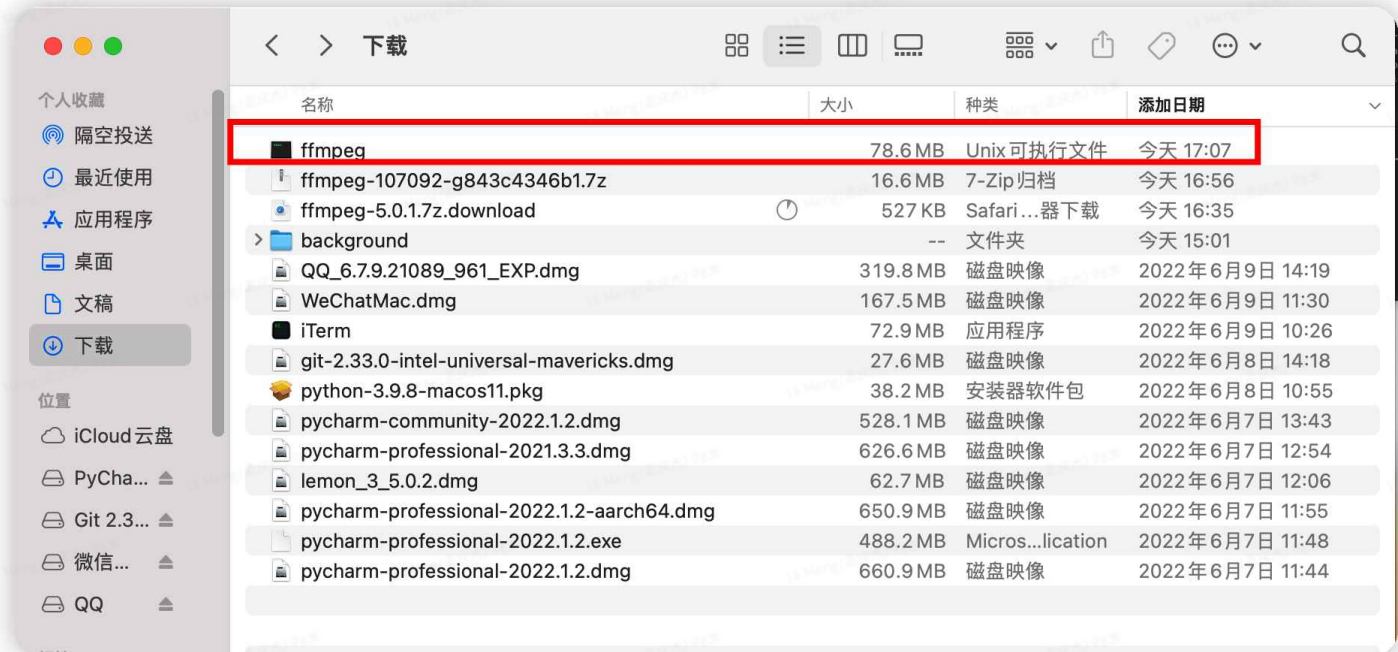
知乎 @糖果

mac: [ffmpeg-109086-geb1e359a14.7z](#)

1. 下载完成之后，打开电脑的访达，找到我们刚刚下载好的文件



2. 解压缩之后，会出现“ffmpeg”



3. 打开上图中的红框里的“ffmpeg”，就会弹出终端页面，等待进程完成，这时ffmpeg应在Downloads里。

```
aminer — ffmpeg — 80x24

Last login: Wed Jun 15 16:46:47 on ttys001
/Users/aminer/Downloads/ffmpeg ; exit;
aminer@aminderMacBook-Pro ~ % /Users/aminer/Downloads/ffmpeg ; exit;
ffmpeg version N-107092-g843c4346b1-tessus Copyright (c) 2000-2022 the FFmpeg de
velopers
built with Apple clang version 11.0.0 (clang-1100.0.33.17)
configuration: --cc=/usr/bin/clang --prefix=/opt/ffmpeg --extra-version=tessus
--enable-avisynth --enable-fontconfig --enable-gpl --enable-libaom --enable-lib
ass --enable-libbluray --enable-libdav1d --enable-libfreetype --enable-libgsm --
enable-libmodplug --enable-libmp3lame --enable-libmysofa --enable-libopencore-am
rnb --enable-libopencore-amrwb --enable-libopenh264 --enable-libopenjpeg --enabl
e-libopus --enable-librubberband --enable-libshine --enable-libsnappp --enable-l
ibsoxr --enable-libspeex --enable-libtheora --enable-libtwolame --enable-libvids
tab --enable-libvmaf --enable-libvo-amrwbenc --enable-libvorbis --enable-libvpx
--enable-libwebp --enable-libx264 --enable-libx265 --enable-libxavs --enable-lib
xvid --enable-libzimg --enable-libzmq --enable-libzvbi --enable-version3 --pkg-c
onfig-flags=--static --disable-ffplay
libavutil      57. 26.100 / 57. 26.100
libavcodec     59. 33.100 / 59. 33.100
libavformat    59. 24.100 / 59. 24.100
libavdevice    59.  6.100 / 59.  6.100
libavfilter     8. 40.100 /  8. 40.100
libswscale     6.  6.100 /  6.  6.100
libswresample  4.  6.100 /  4.  6.100
```

4. 配置环境变量

打开终端逐行输入设置以下指令

```
1 cd ~/Downloads/
```

```
2 sudo mkdir -p /usr/local/bin/  
3 sudo cp ./ffmpeg /usr/local/bin  
4 sudo chmod ugo+x /usr/local/bin/ffmpeg
```

将ffmpeg添加至\$PATH:

```
1 open -e ~/.bash_profile
```

在文件最后面加入以下命令:

```
1 export PATH="/usr/local/bin:$PATH"
```

保存, 然后关闭终端并重启, 输入 `which ffmpeg` 应当返回 `/usr/local/bin/ffmpeg`.

3、下载writemetadata

win:

 [writemetadata-win32-x64.zip](#)

mac:根据macOS的版本下载

 [writemetadata-darwin-arm64.zip](#)

 [writemetadata-darwin-x64.zip](#)