HTTrack version 3.48-24

格式: httrack <URLs> [-参数] [+<URL\_FILTER>] [-<URL\_FILTER>] [+<mime:MIME\_FILTER>] [-<mime:MIME\_FILTER>]

下面列出的选项: (\*是默认值)

普通参数:

O path for mirror/logfiles+cache (-O path\_mirror[,path\_cache\_and\_logfiles]) (--path <param>)

行动方案:

w \*镜像网站(--mirror)

W 镜像网站, semi-automatic (asks questions) (--mirror-wizard)

g 把文件保存在当前目录(saved in the current directory) (--get-files)

i 使用缓存继续中断镜像(--continue)

Y 镜像在一级页中的所有链接 (mirror links) (--mirrorlinks)

代理选项:

P 代理的使用(-P proxy:port or -P user:pass@proxy:port) (--proxy <param>)

%f \*使用代理服务器的FTP (f0 don't use) (--httpproxy-ftp[=N])

%b 使用本地主机名/发送请求 (-%b hostname) (--bind <param>)

限制选项:

rN 设置镜像深度为n (\* r9999) (--depth[=N])

%eN 设置外部链接深度为n(\* %e0) (--ext-depth[=N])

mN 对于一个非HTML文件的最大文件长度 (--max-files[=N])

mN,N2 非HTML的最大文件长度（N）和HTML（N2）

MN 最大的整体大小，可以上传/扫描(--max-size[=N])

EN 最大镜时间 (60=1 minute, 3600=1 hour) (--max-time[=N])

AN 最大传输速率以字节/秒 (1000=1KB/s max) (--max-rate[=N])

%cN最大连接数/秒(\*%c10) (--connection-per-second[=N])

GN暂停传输，如果N个字节达到, 并等待锁文件被删除 (--max-pause[=N])

流量控制:

cN 多个连接数 (\*c8) (--sockets[=N])默认是8

TN 超时, 非响应。链接关闭后的秒数 (--timeout[=N])

RN 重试次数, 在超时或非致命错误的情况下 (\*R1) (--retries[=N])

JN 交通堵塞的控制, 最低传输速率 (bytes/seconds) 容忍一个链接 (--min-rate[=N])

HN host is abandonned if: 0=never, 1=timeout, 2=slow, 3=timeout or slow (--host-control[=N])

链接选项:

%P \*扩展分析, 试图解析所有链接, 即使在未知的标签或者JavaScript(%P0 don't use) (--extended-parsing[=N])

n 让非HTML文件的“近”的HTML文件(ex: 位于外部的图像) (--near)

t 测试所有的URL(即使是被禁止的） (--test)

%L <file> 添加位于此文本文件中的所有网址(每行一个网址) (--list <param>)

%S <file> 添加位于此文本文件中的所有扫描规则(每行一个扫描规则) (--urllist <param>)

编译选项:

NN结构类型 (0 \*原始结构, 1+: see below) (--structure[=N])

或用户定义的结构 (-N "%h%p/%n%q.%t")

%N 延迟型检查, 不要做任何链接测试，但等待文件下载开始，(experimental) (%N0 don't use, %N1 use for unknown extensions, \* %N2 always use)

%D 缓存延迟类型检查, 在更新过程中不要等待远程类型, 为了加速他们 (%D0 wait, \* %D1 don't wait) (--cached-delayed-type-check)

%M 生成一个RFC MIME封装完整档案 (.mht) (--mime-html)

LN long names (L1 \*long names / L0 8-3 conversion / L2 ISO9660 compatible) (--long-names[=N])

KN 保持原来的链接 (e.g. http://www.adr/link) (K0 \*relative link, K absolute links, K4 original links, K3 absolute URI links, K5 transparent proxy link) (--keep-links[=N])

x 通过错误页替换外部HTML链接 (--replace-external)

%x 不包括任何外部密码保护的网站的密码 (%x0 include) (--disable-passwords)

%q \*include query string for local files (useless, for information purpose only) (%q0 don't include) (--include-query-string)

o \*产生错误时输出的HTML文件 (404..) (o0 don't generate) (--generate-errors)

X \*更新后清除旧文件 (X0 keep delete) (--purge-old[=N])

%p 保存的HTML文件'as is' (identical to '-K4 -%F ""') (--preserve)

%T 链接转换为UTF-8 (--utf8-conversion)

爬行选项:

bN 在cookies文件中使用cookie(0=do not accept,\* 1=accept) (--cookies[=N])

u 如果未知的检查文档类型 (cgi,asp..) (u0 don't check, \* u1 check but /, u2 check always) (--check-type[=N])

j \*解析java类 (j0 don't parse, bitmask: |1 parse default, |2 don't parse .class |4 don't parse .js |8 don't be aggressive) (--parse-java[=N])

sN 遵循robots.txt和meta robots标签 (0=never,1=sometimes,\* 2=always, 3=always (even strict rules)) (--robots[=N])

%h force HTTP/1.0 requests (减少更新功能，只用于旧的服务器或代理服务器) (--http-10)

%k 如果可能保持速度，极大地减少小文件和测试请求等待时间 (%k0 don't use) (--keep-alive)

%B 宽容的要求 (接受一些服务器上的虚假响应，但不接受标准！) (--tolerant)

%s 更新的攻击: 各种攻击限制重新传输时更新 (相同的大小，假响应。) (--updatehack)

%u url 攻击: 各种攻击限制重复的URL (strip //, www.foo.com==foo.com..) (--urlhack)

%A 假设一个类型 (cgi,asp..) 总是带着一个MIME类型链接 (-%A php3,cgi=text/html;dat,bin=application/x-zip) (--assume <param>)

快捷方式: '--assume standard' 相当于-%A php2 php3 php4 php cgi asp jsp pl cfm nsf=text/html

也可以用来强制特定的文件类型: --assume foo.cgi=text/html

@iN 互联网协议； (0=both ipv6+ipv4, 4=ipv4 only, 6=ipv6 only) (--protocol[=N])

%w 禁用特定外部MIME模块 (-%w htsswf -%w htsjava) (--disable-module <param>)

浏览器ID:

F 用户代理字段在HTTP标头发送 (-F "user-agent name") (--user-agent <param>)

%R 默认的Referer字段在HTTP标头发送(--referer <param>)

%E 从电子邮件地址发送的HTTP头 (--from <param>)

%F 在HTML代码页脚字符串 (-%F "Mirrored [from host %s [file %s [at %s]]]" (--footer <param>)

%l 常见的语言 (-%l "fr, en, jp, \*" (--language <param>)

%a 接受的格式 (-%a "text/html,image/png;q=0.9,\*/\*;q=0.1" (--accept <param>)

%X 额外的HTTP头线 (-%X "X-Magic: 42" (--headers <param>)

Log, index, cache

C 创建/使用缓存更新和重试次数 (C0 no cache,C1 cache is prioritary,\* C2 test update before) (--cache[=N])

k 存储缓存中的所有文件 (not useful if files on disk) (--store-all-in-cache)

%n 不要重新下载本地删除的文件 (--do-not-recatch)

%v 在屏幕上显示文件名下载(in realtime) - \* %v1 short version - %v2 full animation (--display)

Q 不安静的日志模式 (--do-not-log)

q 不启用的模式 (--quiet)

z 日志-额外的信息(--extra-log)

Z 日志-调试 (--debug-log)

v 登录屏幕 (--verbose)

f \*日志文件(--file-log)

f2 一个单一的日志文件 (--single-log)

I \*做一个index (I0 don't make) (--index)

%i 为项目文件夹做顶部索引 (\* %i0 don't make) (--build-top-index)

%I 让这个镜像是一个可搜索的索引 (\* %I0 don't make) (--search-index)

高级选项:

PN优先模式: (\* p3) (--priority[=N])

p0 只扫描，不保存任何东西(for checking links)

p1 只保存HTML文件

p2 不使用HTML文件

\*p3 保存所有文件

p7 get html files before, then treat other files

S stay on the same directory (--stay-on-same-dir)

D \*只能进入子目录(--can-go-down)

U 只能上目录 (--can-go-up)

B 可以上下到目录结构(--can-go-up-and-down)

a \*停留在同一个地址 (--stay-on-same-address)

d 停留在同一个主域 (--stay-on-same-domain)

l 在同一公司 (eg: .com) (--stay-on-same-tld)

e 在网络上到处走(--go-everywhere)

%H 调试HTTP标头在日志文件 (--debug-headers)

大师的选择: (do NOT use if possible)

#X \*使用优化的配置(limited memory boundary checks) (--fast-engine)

#0 滤波器的测试 (-#0 '\*.gif' 'www.bar.com/foo.gif') (--debug-testfilters <param>)

#1 简化测试(-#1 ./foo/bar/../foobar)

#2 型式试验 (-#2 /foo/bar.php)

#C 缓存列表 (-#C '\*.com/spider\*.gif' (--debug-cache <param>)

#R 缓存修复 (damaged cache) (--repair-cache)

#d 调试分析器 (--debug-parsing)

#E 在meta.zip new.zip缓存元数据提取

#f 总是刷新日志文件(--advanced-flushlogs)

#FN 过滤器的最大数量 (--advanced-maxfilters[=N])

#h 版本信息 (--version)

#K 扫描输入(debug) (--debug-scanstdin)

#L 最大链接数(-#L1000000) (--advanced-maxlinks[=N])

#p 显示丑陋的进度信息 (--advanced-progressinfo)

#P 抓URL (--catch-url)

#R 老的FTP程序 (debug) (--repair-cache)

#T 生成转移行动。记录每分钟(--debug-xfrstats)

#u 等待时间 (--advanced-wait)

#Z 产生传输率统计每分钟(--debug-ratestats)

危险的选项: (不要使用，除非你完全知道你在做什么)

%! 绕过内置的安全限制，旨在避免带宽滥用(带宽, 同时连接) (--disable-security-limits)

重要注意事项: 危险的选择，只适用于专家

使用它与极端护理

Command-line specific options:

V 在每个文件后执行系统命令 ($0 is the filename: -V "rm \$0") (--userdef-cmd <param>)

%W 使用外部库函数作为一个包装器 (-%W myfoo.so[,myparameters]) (--callback <param>)

详细: Option N

N0 网站结构 (default)

N1 在网页HTML，图像或其他文件在网页/图像/

N2 在网络/ HTML HTML，图像/网页/图像

N3 在网页HTML，图像/网页/其他

N4 在网页HTML，图像/网页/ XXX，XXX是文件扩展名（所有的GIF将放置到网页/ GIF为例）

N5 图像/网页/ HTML / XXX和HTML网页等

N99 所有的文件在网络/，与随机名称（小工具！）

N100 Site-structure, without www.domain.xxx/

N101 Identical to N1 exept that "web" is replaced by the site's name

N102 Identical to N2 exept that "web" is replaced by the site's name

N103 Identical to N3 exept that "web" is replaced by the site's name

N104 Identical to N4 exept that "web" is replaced by the site's name

N105 Identical to N5 exept that "web" is replaced by the site's name

N199 Identical to N99 exept that "web" is replaced by the site's name

N1001 Identical to N1 exept that there is no "web" directory

N1002 Identical to N2 exept that there is no "web" directory

N1003 Identical to N3 exept that there is no "web" directory (option set for g option)

N1004 Identical to N4 exept that there is no "web" directory

N1005 Identical to N5 exept that there is no "web" directory

N1099 Identical to N99 exept that there is no "web" directory

详细: 用户自定义选项 N

'%n' 无文件类型的文件名 (ex: image)

'%N' 文件名，包括文件类型(ex: image.gif)

'%t' 文件类型(ex: gif)

'%p' Path [without ending /] (ex: /someimages)

'%h' 主机名； (ex: www.someweb.com)

'%M' URL MD5 (128 bits, 32 ascii bytes)

'%Q' 查询字符串的MD5(128 bits, 32 ascii bytes)

'%k' 完整的查询字符串

'%r' 协议名称(ex: http)

'%q' 小的查询字符串的MD5 (16 bits, 4 ascii bytes)

'%s?' 短名称版本 (ex: %sN)

'%[param]' 在查询字符串参数变量

'%[param:before:after:empty:notfound]' advanced variable extraction

Details:用户定义的选项n和高级变量提取

%[param:before:after:empty:notfound]

param : 参数名称

before : 字符串前面加上如果参数被发现

after : 如果找到参数，将追加字符串

notfound : 如果找不到参数的字符串替换

empty : 如果参数为空的字符串替换

所有字段，除了第一个（参数名称），可以是空的

详细: 参数 K

K0 foo.cgi?q=45 -> foo4B54.html?q=45 (relative URI, default)

K -> http://www.foobar.com/folder/foo.cgi?q=45 (absolute URL) (--keep-links[=N])

K3 -> /folder/foo.cgi?q=45 (absolute URI)

K4 -> foo.cgi?q=45 (original URL)

K5 -> http://www.foobar.com/folder/foo4B54.html?q=45 (transparent proxy URL)

快捷键:

--mirror <URLs> \*做一个网站的镜像（s） (default)

--get <URLs> 获取文件表面，不要寻求其他网址(-qg)

--list <text file> 添加位于此文本文件中的所有网址(-%L)

--mirrorlinks <URLs> 镜像第一级页面中的所有链接(-Y)

--testlinks <URLs> 页面中的测试链接 (-r1p0C0I0t)

--spider <URLs> 趴站(s),测试链接中的错误和警告 (-p0C0I0t)

--testsite <URLs> 一样的方式--spider

--skeleton <URLs> 做一个镜像, 但只有HTML文件(-p1)

--update 更新镜像， 不确认 (-iC2)

--continue 继续镜像, 不确认 (-iC1)

--catchurl 创建一个临时代理来捕获一个网址或一个表单后的网址

--clean 清除缓存和日志文件

--http10 force http/1.0 requests (-%h)

实例: Option %W: External callbacks prototypes

see htsdefines.h

例子: httrack www.someweb.com/bob/

means: mirror site www.someweb.com/bob/ and only this site

例子: httrack www.someweb.com/bob/ www.anothertest.com/mike/ +\*.com/\*.jpg -mime:application/\*

means: mirror the two sites together (with shared links) and accept any .jpg files on .com sites

例子: httrack www.someweb.com/bob/bobby.html +\* -r6

means get all files starting from bobby.html, with 6 link-depth, and possibility of going everywhere on the web

例子: httrack www.someweb.com/bob/bobby.html --spider -P proxy.myhost.com:8080

runs the spider on www.someweb.com/bob/bobby.html using a proxy

例子: httrack --update

updates a mirror in the current folder

例子: httrack

will bring you to the interactive mode

例子: httrack --continue

continues a mirror in the current folder

HTTrack version 3.48-24

Copyright (C) 1998-2016 Xavier Roche and other contributors