NAME

Wget − The non−interactive network downloader.

SYNOPSIS

wget [*option*]... [*URL*]...

DESCRIPTION

GNU Wget is a free utility for non-interactive download of files from the Web. It supports HTTP, HTTPS, and FTP protocols, as well as retrieval through HTTP proxies.

Wget is non−interactive, meaning that it can work in the background, while the user is not logged on. This allows you to start a retrieval and disconnect from the system, letting Wget finish the work. By contrast, most of the Web browsers require constant user's presence, which can be a great hindrance when transferring a lot of data.

Wget can follow links in HTML and XHTML pages and create local versions of remote web sites, fully recreating the directory structure of the original site. This is sometimes referred to as ‘‘recursive downloading.’’ While doing that, Wget respects the Robot Exclusion Standard (*/robots.txt*). Wget can be instructed to convert the links in downloaded HTML files to the local files for offline viewing.

Wget has been designed for robustness over slow or unstable network connections; if a download fails due to a network problem, it will keep retrying until the whole file has been retrieved. If the server supports regetting, it will instruct the server to continue the download from where it left off.

OPTIONS

Option Syntax

Since Wget uses GNU getopt to process command-line arguments, every option has a long form along with the short one. Long options are more convenient to remember, but take time to type. You may freely mix different option styles, or specify options after the command-line arguments. Thus you may write:

```
wget -r --tries=10 http://fly.srk.fer.hr/ -o log
```

The space between the option accepting an argument and the argument may be omitted. Instead **−o log** you can write **−olog**.

You may put several options that do not require arguments together, like:

```
wget -drc <URL>
```

This is a complete equivalent of:

```
wget -d -r -c <URL>
```

Since the options can be specified after the arguments, you may terminate them with `−−`. So the following will try to download URL **−x**, reporting failure to *log*:

```
wget -o log -- -x
```

The options that accept comma-separated lists all respect the convention that specifying an empty list clears its value. This can be useful to clear the *.wgetrc* settings. For instance, if your *.wgetrc* sets `exclude_directories` to */cgi−bin*, the following example will first reset it, and then set it to exclude */˜nobody* and */˜somebody*. You can also clear the lists in *.wgetrc*.

```
wget -X '' -X /˜nobody,/˜somebody
```

Most options that do not accept arguments are *boolean* options, so named because their state can be captured with a yes-or-no (‘‘boolean’’) variable. For example, `−−`**follow−ftp** tells Wget to follow FTP links from HTML files and, on the other hand, `−−`**no−glob** tells it not to perform file globbing on FTP URLs. A boolean option is either *affirmative* or *negative* (beginning with `−−`**no**). All such options share several properties.

Unless stated otherwise, it is assumed that the default behavior is the opposite of what the option accomplishes. For example, the documented existence of `−−`**follow−ftp** assumes that the default is to *not* follow FTP links from HTML pages.

Affi rmative options can be negated by prepending the **−−no−** to the option name; negative options can be negated by omitting the **−−no−** prefi x. This might seem superfluous−−−if the default for an affi rmative option is to not do something, then why provide a way to explicitly turn it off? But the startup fi le may in fact change the default. For instance, using `follow_ftp = off` in *.wgetrc* makes Wget *not* follow FTP links by default, and using **−−no−follow−ftp** is the only way to restore the factory default from the command line.

Basic Startup Options

−V
−−version
Display the version of Wget.

−h
−−help
Print a help message describing all of Wget's command-line options.

−b
−−background
Go to background immediately after startup. If no output fi le is specifi ed via the **−o**, output is redirected to *wget-log*.

−e *command*
−−execute *command*
Execute *command* as if it were a part of *.wgetrc*. A command thus invoked will be executed *after* the commands in *.wgetrc*, thus taking precedence over them. If you need to specify more than one wgetrc command, use multiple instances of **−e**.

Logging and Input File Options

−o *logfile*
−−output−file=*logfile*
Log all messages to *logfile*. The messages are normally reported to standard error.

−a *logfile*
−−append−output=*logfile*
Append to *logfile*. This is the same as **−o**, only it appends to *logfile* instead of overwriting the old log fi le. If *logfile* does not exist, a new fi le is created.

−d
−−debug
Turn on debug output, meaning various information important to the developers of Wget if it does not work properly. Your system administrator may have chosen to compile Wget without debug support, in which case **−d** will not work. Please note that compiling with debug support is always safe−−−Wget compiled with the debug support will *not* print any debug info unless requested with **−d**.

−q
−−quiet
Turn off Wget's output.

−v
−−verbose
Turn on verbose output, with all the available data. The default output is verbose.

−nv
−−no−verbose
Turn off verbose without being completely quiet (use **−q** for that), which means that error messages and basic information still get printed.

−i *fi le*
−−input−file=*fi le*

Read URLs from *fi le*. If **−** is specifi ed as *fi le*, URLs are read from the standard input. (Use **./−** to read from a fi le literally named **−**.)

If this function is used, no URLs need be present on the command line. If there are URLs both on the command line and in an input fi le, those on the command lines will be the fi rst ones to be retrieved. The *fi le* need not be an HTML document (but no harm if it is)−−−it is enough if the URLs are just listed sequentially.

However, if you specify **−−force−html**, the document will be regarded as **html**. In that case you may have problems with relative links, which you can solve either by adding `<base href="`*`url`*`">` to the documents or by specifying **−−base=***url* on the command line.

−F
−−force−html

When input is read from a fi le, force it to be treated as an HTML fi le. This enables you to retrieve relative links from existing HTML fi les on your local disk, by adding `<base href="`*`url`*`">` to HTML, or using the **−−base** command-line option.

−B *URL*
−−base=*URL*

Prepends *URL* to relative links read from the fi le specifi ed with the **−i** option.

Download Options

−−bind−address=*ADDRESS*

When making client TCP/IP connections, bind to *ADDRESS* on the local machine. *ADDRESS* may be specifi ed as a hostname or IP address. This option can be useful if your machine is bound to multiple IPs.

−t *number*
−−tries=*number*

Set number of retries to *number*. Specify 0 or **inf** for infi nite retrying. The default is to retry 20 times, with the exception of fatal errors like ‘‘connection refused’’ or ‘‘not found’’ (404), which are not retried.

−O *fi le*
−−output−document=*fi le*

The documents will not be written to the appropriate fi les, but all will be concatenated together and written to *fi le*. If **−** is used as *fi le*, documents will be printed to standard output, disabling link conversion. (Use **./−** to print to a fi le literally named **−**.)

Note that a combination with **−k** is only well-defi ned for downloading a single document.

−nc
−−no−clobber

If a fi le is downloaded more than once in the same directory, Wget's behavior depends on a few options, including **−nc**. In certain cases, the local fi le will be *clobbered*, or overwritten, upon repeated download. In other cases it will be preserved.

When running Wget without **−N**, **−nc**, or **−r**, downloading the same fi le in the same directory will result in the original copy of *fi le* being preserved and the second copy being named *fi le***.1**. If that fi le is downloaded yet again, the third copy will be named *fi le***.2**, and so on. When **−nc** is specifi ed, this behavior is suppressed, and Wget will refuse to download newer copies of *fi le*. Therefore, ‘‘`no−clobber`’’ is actually a misnomer in this mode−−−it's not clobbering that's prevented (as the numeric suffi xes were already preventing clobbering), but rather the multiple version saving that's prevented.

When running Wget with **−r**, but without **−N** or **−nc**, re-downloading a fi le will result in the new copy simply overwriting the old. Adding **−nc** will prevent this behavior, instead causing the original version to be preserved and any newer copies on the server to be ignored.

When running Wget with **−N**, with or without **−r**, the decision as to whether or not to download a newer copy of a file depends on the local and remote timestamp and size of the file. **−nc** may not be specified at the same time as **−N**.

Note that when **−nc** is specified, files with the suffixes **.html** or **.htm** will be loaded from the local disk and parsed as if they had been retrieved from the Web.

−c
−−continue

Continue getting a partially-downloaded file. This is useful when you want to finish up a download started by a previous instance of Wget, or by another program. For instance:

```
        wget -c ftp://sunsite.doc.ic.ac.uk/ls-lR.Z
```

If there is a file named *ls−lR.Z* in the current directory, Wget will assume that it is the first portion of the remote file, and will ask the server to continue the retrieval from an offset equal to the length of the local file.

Note that you don't need to specify this option if you just want the current invocation of Wget to retry downloading a file should the connection be lost midway through. This is the default behavior. **−c** only affects resumption of downloads started *prior* to this invocation of Wget, and whose local files are still sitting around.

Without **−c**, the previous example would just download the remote file to *ls−lR.Z.1*, leaving the truncated *ls−lR.Z* file alone.

Beginning with Wget 1.7, if you use **−c** on a non-empty file, and it turns out that the server does not support continued downloading, Wget will refuse to start the download from scratch, which would effectively ruin existing contents. If you really want the download to start from scratch, remove the file.

Also beginning with Wget 1.7, if you use **−c** on a file which is of equal size as the one on the server, Wget will refuse to download the file and print an explanatory message. The same happens when the file is smaller on the server than locally (presumably because it was changed on the server since your last download attempt)−−−because ''continuing'' is not meaningful, no download occurs.

On the other side of the coin, while using **−c**, any file that's bigger on the server than locally will be considered an incomplete download and only `(length(remote) - length(local))` bytes will be downloaded and tacked onto the end of the local file. This behavior can be desirable in certain cases−−−for instance, you can use **wget −c** to download just the new portion that's been appended to a data collection or log file.

However, if the file is bigger on the server because it's been *changed*, as opposed to just *appended* to, you'll end up with a garbled file. Wget has no way of verifying that the local file is really a valid prefix of the remote file. You need to be especially careful of this when using **−c** in conjunction with **−r**, since every file will be considered as an ''incomplete download'' candidate.

Another instance where you'll get a garbled file if you try to use **−c** is if you have a lame HTTP proxy that inserts a ''transfer interrupted'' string into the local file. In the future a ''rollback'' option may be added to deal with this case.

Note that **−c** only works with FTP servers and with HTTP servers that support the `Range` header.

−−progress=*type*

Select the type of the progress indicator you wish to use. Legal indicators are ''dot'' and ''bar''.

The ''bar'' indicator is used by default. It draws an ASCII progress bar graphics (a.k.a ''thermometer'' display) indicating the status of retrieval. If the output is not a TTY, the ''dot'' bar will be used by default.

Use **−−progress=dot** to switch to the ''dot'' display. It traces the retrieval by printing dots on the screen, each dot representing a fixed amount of downloaded data.

When using the dotted retrieval, you may also set the *style* by specifying the type as **dot:***style*. Different styles assign different meaning to one dot. With the `default` style each dot represents 1K, there are ten dots in a cluster and 50 dots in a line. The `binary` style has a more ''computer''−like orientation−−−8K dots, 16−dots clusters and 48 dots per line (which makes for 384K lines). The `mega` style is suitable for downloading very large files−−−each dot represents 64K retrieved, there are eight dots in a cluster, and 48 dots on each line (so each line contains 3M).

Note that you can set the default style using the `progress` command in *.wgetrc*. That setting may be overridden from the command line. The exception is that, when the output is not a TTY, the ''dot'' progress will be favored over ''bar''. To force the bar output, use **−−progress=bar:force**.

−N
−−timestamping
Turn on time−stamping.

−S
−−server−response
Print the headers sent by HTTP servers and responses sent by FTP servers.

−−spider
When invoked with this option, Wget will behave as a Web *spider*, which means that it will not download the pages, just check that they are there. For example, you can use Wget to check your bookmarks:

```
        wget --spider --force-html -i bookmarks.html
```

This feature needs much more work for Wget to get close to the functionality of real web spiders.

−T seconds
−−timeout=*seconds*
Set the network timeout to *seconds* seconds. This is equivalent to specifying **−−dns−timeout**, **−−connect−timeout**, and **−−read−timeout**, all at the same time.

When interacting with the network, Wget can check for timeout and abort the operation if it takes too long. This prevents anomalies like hanging reads and infinite connects. The only timeout enabled by default is a 900−second read timeout. Setting a timeout to 0 disables it altogether. Unless you know what you are doing, it is best not to change the default timeout settings.

All timeout-related options accept decimal values, as well as subsecond values. For example, **0.1** seconds is a legal (though unwise) choice of timeout. Subsecond timeouts are useful for checking server response times or for testing network latency.

−−dns−timeout=*seconds*
Set the DNS lookup timeout to *seconds* seconds. DNS lookups that don't complete within the specified time will fail. By default, there is no timeout on DNS lookups, other than that implemented by system libraries.

−−connect−timeout=*seconds*
Set the connect timeout to *seconds* seconds. TCP connections that take longer to establish will be aborted. By default, there is no connect timeout, other than that implemented by system libraries.

−−read−timeout=*seconds*
Set the read (and write) timeout to *seconds* seconds. The ''time'' of this timeout refers *idle time*: if, at any point in the download, no data is received for more than the specified number of seconds, reading fails and the download is restarted. This option does not directly affect the duration of the entire download.

Of course, the remote server may choose to terminate the connection sooner than this option requires. The default read timeout is 900 seconds.

−−limit−rate=*amount*

Limit the download speed to *amount* bytes per second. Amount may be expressed in bytes, kilobytes with the **k** suffix, or megabytes with the **m** suffix. For example, **−−limit−rate=20k** will limit the retrieval rate to 20KB/s. This is useful when, for whatever reason, you don't want Wget to consume the entire available bandwidth.

This option allows the use of decimal numbers, usually in conjunction with power suffixes; for example, **−−limit−rate=2.5k** is a legal value.

Note that Wget implements the limiting by sleeping the appropriate amount of time after a network read that took less time than specified by the rate. Eventually this strategy causes the TCP transfer to slow down to approximately the specified rate. However, it may take some time for this balance to be achieved, so don't be surprised if limiting the rate doesn't work well with very small files.

−w *seconds*
−−wait=*seconds*

Wait the specified number of seconds between the retrievals. Use of this option is recommended, as it lightens the server load by making the requests less frequent. Instead of in seconds, the time can be specified in minutes using the `m` suffix, in hours using `h` suffix, or in days using `d` suffix.

Specifying a large value for this option is useful if the network or the destination host is down, so that Wget can wait long enough to reasonably expect the network error to be fixed before the retry.

−−waitretry=*seconds*

If you don't want Wget to wait between *every* retrieval, but only between retries of failed downloads, you can use this option. Wget will use *linear backoff*, waiting 1 second after the first failure on a given file, then waiting 2 seconds after the second failure on that file, up to the maximum number of *seconds* you specify. Therefore, a value of 10 will actually make Wget wait up to (1 + 2 + ... + 10) = 55 seconds per file.

Note that this option is turned on by default in the global *wgetrc* file.

−−random−wait

Some web sites may perform log analysis to identify retrieval programs such as Wget by looking for statistically significant similarities in the time between requests. This option causes the time between requests to vary between 0 and 2 * *wait* seconds, where *wait* was specified using the **−−wait** option, in order to mask Wget's presence from such analysis.

A recent article in a publication devoted to development on a popular consumer platform provided code to perform this analysis on the fly. Its author suggested blocking at the class C address level to ensure automated retrieval programs were blocked despite changing DHCP-supplied addresses.

The **−−random−wait** option was inspired by this ill-advised recommendation to block many unrelated users from a web site due to the actions of one.

−−no−proxy

Don't use proxies, even if the appropriate `*_proxy` environment variable is defined.

For more information about the use of proxies with Wget,

−Q *quota*
−−quota=*quota*

Specify download quota for automatic retrievals. The value can be specified in bytes (default), kilobytes (with **k** suffix), or megabytes (with **m** suffix).

Note that quota will never affect downloading a single file. So if you specify **wget −Q10k ftp://wuarchive.wustl.edu/ls−lR.gz**, all of the *ls−lR.gz* will be downloaded. The same goes even when several URLs are specified on the command−line. However, quota is respected when retrieving either recursively, or from an input file. Thus you may safely type **wget −Q2m −i sites**−−−download will be aborted when the quota is exceeded.

Setting quota to 0 or to **inf** unlimits the download quota.

−−no−dns−cache

Turn off caching of DNS lookups. Normally, Wget remembers the IP addresses it looked up from DNS so it doesn't have to repeatedly contact the DNS server for the same (typically small) set of hosts it retrieves from. This cache exists in memory only; a new Wget run will contact DNS again.

However, it has been reported that in some situations it is not desirable to cache host names, even for the duration of a short-running application like Wget. With this option Wget issues a new DNS lookup (more precisely, a new call to `gethostbyname` or `getaddrinfo`) each time it makes a new connection. Please note that this option will *not* affect caching that might be performed by the resolving library or by an external caching layer, such as NSCD.

If you don't understand exactly what this option does, you probably won't need it.

−−restrict−file−names=*mode*

Change which characters found in remote URLs may show up in local fi le names generated from those URLs. Characters that are *restricted* by this option are escaped, i.e. replaced with **%HH**, where **HH** is the hexadecimal number that corresponds to the restricted character.

By default, Wget escapes the characters that are not valid as part of fi le names on your operating system, as well as control characters that are typically unprintable. This option is useful for changing these defaults, either because you are downloading to a non-native partition, or because you want to disable escaping of the control characters.

When mode is set to ''unix'', Wget escapes the character **/** and the control characters in the ranges 0−−31 and 128−−159. This is the default on Unix-like OS'es.

When mode is set to ''windows'', Wget escapes the characters ****, **|**, **/**, **:**, **?**, **"**, *****, **<**, **>**, and the control characters in the ranges 0−−31 and 128−−159. In addition to this, Wget in Windows mode uses **+** instead of **:** to separate host and port in local fi le names, and uses **@** instead of **?** to separate the query portion of the fi le name from the rest. Therefore, a URL that would be saved as **www.xemacs.org:4300/search.pl?input=blah** in Unix mode would be saved as **www.xemacs.org+4300/search.pl@input=blah** in Windows mode. This mode is the default on Windows.

If you append **,nocontrol** to the mode, as in **unix,nocontrol**, escaping of the control characters is also switched off. You can use **−−restrict−file−names=nocontrol** to turn off escaping of control characters without affecting the choice of the OS to use as fi le name restriction mode.

−4
−−inet4−only
−6
−−inet6−only

Force connecting to IPv4 or IPv6 addresses. With **−−inet4−only** or **−4**, Wget will only connect to IPv4 hosts, ignoring AAAA records in DNS, and refusing to connect to IPv6 addresses specifi ed in URLs. Conversely, with **−−inet6−only** or **−6**, Wget will only connect to IPv6 hosts and ignore A records and IPv4 addresses.

Neither options should be needed normally. By default, an IPv6−aware Wget will use the address family specifi ed by the host's DNS record. If the DNS responds with both IPv4 and IPv6 addresses, Wget will them in sequence until it fi nds one it can connect to. (Also see `−−prefer−family` option described below.)

These options can be used to deliberately force the use of IPv4 or IPv6 address families on dual family systems, usually to aid debugging or to deal with broken network confi guration. Only one of **−−inet6−only** and **−−inet4−only** may be specifi ed at the same time. Neither option is available in Wget compiled without IPv6 support.

−−prefer−family=IPv4/IPv6/none

When given a choice of several addresses, connect to the addresses with specifi ed address family fi rst. IPv4 addresses are preferred by default.

This avoids spurious errors and connect attempts when accessing hosts that resolve to both IPv6 and IPv4 addresses from IPv4 networks. For example, **www.kame.net** resolves to **2001:200:0:8002:203:47ff:fea5:3085** and to **203.178.141.194**. When the preferred family is `IPv4`, the IPv4 address is used first; when the preferred family is `IPv6`, the IPv6 address is used first; if the specified value is `none`, the address order returned by DNS is used without change.

Unlike **−4** and **−6**, this option doesn't inhibit access to any address family, it only changes the *order* in which the addresses are accessed. Also note that the reordering performed by this option is *stable*−−−it doesn't affect order of addresses of the same family. That is, the relative order of all IPv4 addresses and of all IPv6 addresses remains intact in all cases.

−−retry−connrefused
Consider ''connection refused'' a transient error and try again. Normally Wget gives up on a URL when it is unable to connect to the site because failure to connect is taken as a sign that the server is not running at all and that retries would not help. This option is for mirroring unreliable sites whose servers tend to disappear for short periods of time.

−−user=*user*
−−password=*password*
Specify the username *user* and password *password* for both FTP and HTTP file retrieval. These parameters can be overridden using the **−−ftp−user** and **−−ftp−password** options for FTP connections and the **−−http−user** and **−−http−password** options for HTTP connections.

Directory Options

−nd
−−no−directories
Do not create a hierarchy of directories when retrieving recursively. With this option turned on, all files will get saved to the current directory, without clobbering (if a name shows up more than once, the filenames will get extensions **.n**).

−x
−−force−directories
The opposite of **−nd**−−−create a hierarchy of directories, even if one would not have been created otherwise. E.g. **wget −x http://fly.srk.fer.hr/robots.txt** will save the downloaded file to *fly.srk.fer.hr/robots.txt*.

−nH
−−no−host−directories
Disable generation of host-prefixed directories. By default, invoking Wget with **−r http://fly.srk.fer.hr/** will create a structure of directories beginning with *fly.srk.fer.hr/*. This option disables such behavior.

−−protocol−directories
Use the protocol name as a directory component of local file names. For example, with this option, **wget −r http://***host* will save to **http/***host***/...** rather than just to *host***/...**.

−−cut−dirs=*number*
Ignore *number* directory components. This is useful for getting a fine-grained control over the directory where recursive retrieval will be saved.

Take, for example, the directory at **ftp://ftp.xemacs.org/pub/xemacs/**. If you retrieve it with **−r**, it will be saved locally under *ftp.xemacs.org/pub/xemacs/*. While the **−nH** option can remove the *ftp.xemacs.org/* part, you are still stuck with *pub/xemacs*. This is where **−−cut−dirs** comes in handy; it makes Wget not ''see'' *number* remote directory components. Here are several examples of how **−−cut−dirs** option works.

```
No options        -> ftp.xemacs.org/pub/xemacs/
-nH               -> pub/xemacs/
-nH --cut-dirs=1  -> xemacs/
-nH --cut-dirs=2  -> .

--cut-dirs=1      -> ftp.xemacs.org/xemacs/
...
```

If you just want to get rid of the directory structure, this option is similar to a combination of **−nd** and **−P**. However, unlike **−nd**, **−−cut−dirs** does not lose with subdirectories−−−for instance, with **−nH −−cut−dirs=1**, a *beta/* subdirectory will be placed to *xemacs/beta*, as one would expect.

−P *prefix*
−−directory−prefix=*prefix*
Set directory prefix to *prefix*. The *directory prefix* is the directory where all other files and subdirectories will be saved to, i.e. the top of the retrieval tree. The default is **.** (the current directory).

HTTP Options

−E
−−html−extension
If a file of type **application/xhtml+xml** or **text/html** is downloaded and the URL does not end with the regexp **\.[Hh][Tt][Mm][Ll]?**, this option will cause the suffix **.html** to be appended to the local filename. This is useful, for instance, when you're mirroring a remote site that uses **.asp** pages, but you want the mirrored pages to be viewable on your stock Apache server. Another good use for this is when you're downloading CGI-generated materials. A URL like **http://site.com/article.cgi?25** will be saved as *article.cgi?25.html*.

Note that filenames changed in this way will be re-downloaded every time you re-mirror a site, because Wget can't tell that the local *X.html* file corresponds to remote URL *X* (since it doesn't yet know that the URL produces output of type **text/html** or **application/xhtml+xml**. To prevent this re−downloading, you must use **−k** and **−K** so that the original version of the file will be saved as *X.orig*.

−−http−user=*user*
−−http−password=*password*
Specify the username *user* and password *password* on an HTTP server. According to the type of the challenge, Wget will encode them using either the `basic` (insecure) or the `digest` authentication scheme.

Another way to specify username and password is in the URL itself. Either method reveals your password to anyone who bothers to run `ps`. To prevent the passwords from being seen, store them in *.wgetrc* or *.netrc*, and make sure to protect those files from other users with `chmod`. If the passwords are really important, do not leave them lying in those files either−−−edit the files and delete them after Wget has started the download.

−−no−cache
Disable server-side cache. In this case, Wget will send the remote server an appropriate directive (**Pragma: no-cache**) to get the file from the remote service, rather than returning the cached version. This is especially useful for retrieving and flushing out-of-date documents on proxy servers.

Caching is allowed by default.

−−no−cookies
Disable the use of cookies. Cookies are a mechanism for maintaining server-side state. The server sends the client a cookie using the `Set−Cookie` header, and the client responds with the same cookie upon further requests. Since cookies allow the server owners to keep track of visitors and for sites to exchange this information, some consider them a breach of privacy. The default is to use cookies; however, *storing* cookies is not on by default.

−−load−cookies *file*

Load cookies from *file* before the first HTTP retrieval. *file* is a textual file in the format originally used by Netscape's *cookies.txt* file.

You will typically use this option when mirroring sites that require that you be logged in to access some or all of their content. The login process typically works by the web server issuing an HTTP cookie upon receiving and verifying your credentials. The cookie is then resent by the browser when accessing that part of the site, and so proves your identity.

Mirroring such a site requires Wget to send the same cookies your browser sends when communicating with the site. This is achieved by **−−load−cookies**−−−simply point Wget to the location of the *cookies.txt* file, and it will send the same cookies your browser would send in the same situation. Different browsers keep textual cookie files in different locations:

Netscape 4.x.

The cookies are in *˜/.netscape/cookies.txt*.

Mozilla and Netscape 6.x.

Mozilla's cookie file is also named *cookies.txt*, located somewhere under *˜/.mozilla*, in the directory of your profile. The full path usually ends up looking somewhat like *˜/.mozilla/default/some-weird-string/cookies.txt*.

Internet Explorer.

You can produce a cookie file Wget can use by using the File menu, Import and Export, Export Cookies. This has been tested with Internet Explorer 5; it is not guaranteed to work with earlier versions.

Other browsers.

If you are using a different browser to create your cookies, **−−load−cookies** will only work if you can locate or produce a cookie file in the Netscape format that Wget expects.

If you cannot use **−−load−cookies**, there might still be an alternative. If your browser supports a ''cookie manager'', you can use it to view the cookies used when accessing the site you're mirroring. Write down the name and value of the cookie, and manually instruct Wget to send those cookies, bypassing the ''official'' cookie support:

```
        wget --no-cookies --header "Cookie: <name>=<value>"
```

−−save−cookies *file*

Save cookies to *file* before exiting. This will not save cookies that have expired or that have no expiry time (so−called ''session cookies''), but also see **−−keep−session−cookies**.

−−keep−session−cookies

When specified, causes **−−save−cookies** to also save session cookies. Session cookies are normally not saved because they are meant to be kept in memory and forgotten when you exit the browser. Saving them is useful on sites that require you to log in or to visit the home page before you can access some pages. With this option, multiple Wget runs are considered a single browser session as far as the site is concerned.

Since the cookie file format does not normally carry session cookies, Wget marks them with an expiry timestamp of 0. Wget's **−−load−cookies** recognizes those as session cookies, but it might confuse other browsers. Also note that cookies so loaded will be treated as other session cookies, which means that if you want **−−save−cookies** to preserve them again, you must use **−−keep−session−cookies** again.

−−ignore−length

Unfortunately, some HTTP servers (CGI programs, to be more precise) send out bogus `Content−Length` headers, which makes Wget go wild, as it thinks not all the document was retrieved. You can spot this syndrome if Wget retries getting the same document again and again, each time claiming that the (otherwise normal) connection has closed on the very same byte.

With this option, Wget will ignore the `Content−Length` header−−−as if it never existed.

−−header=*header-line*

Send *header-line* along with the rest of the headers in each HTTP request. The supplied header is sent as−is, which means it must contain name and value separated by colon, and must not contain newlines.

You may define more than one additional header by specifying **−−header** more than once.

```
wget --header='Accept-Charset: iso-8859-2' \
     --header='Accept-Language: hr'        \
       http://fly.srk.fer.hr/
```

Specification of an empty string as the header value will clear all previous user-defined headers.

As of Wget 1.10, this option can be used to override headers otherwise generated automatically. This example instructs Wget to connect to localhost, but to specify **foo.bar** in the `Host` header:

```
wget --header="Host: foo.bar" http://localhost/
```

In versions of Wget prior to 1.10 such use of **−−header** caused sending of duplicate headers.

−−proxy−user=*user*
−−proxy−password=*password*

Specify the username *user* and password *password* for authentication on a proxy server. Wget will encode them using the `basic` authentication scheme.

Security considerations similar to those with **−−http−password** pertain here as well.

−−referer=*url*

Include 'Referer: *url*' header in HTTP request. Useful for retrieving documents with server-side processing that assume they are always being retrieved by interactive web browsers and only come out properly when Referer is set to one of the pages that point to them.

−−save−headers

Save the headers sent by the HTTP server to the file, preceding the actual contents, with an empty line as the separator.

−U *agent-string*
−−user−agent=*agent-string*

Identify as *agent-string* to the HTTP server.

The HTTP protocol allows the clients to identify themselves using a `User−Agent` header field. This enables distinguishing the WWW software, usually for statistical purposes or for tracing of protocol violations. Wget normally identifies as **Wget/***version*, *version* being the current version number of Wget.

However, some sites have been known to impose the policy of tailoring the output according to the `User−Agent`−supplied information. While this is not such a bad idea in theory, it has been abused by servers denying information to clients other than (historically) Netscape or, more frequently, Microsoft Internet Explorer. This option allows you to change the `User−Agent` line issued by Wget. Use of this option is discouraged, unless you really know what you are doing.

Specifying empty user agent with **−−user−agent=**`""` instructs Wget not to send the `User−Agent` header in HTTP requests.

−−post−data=*string*
−−post−file=*file*

Use POST as the method for all HTTP requests and send the specified data in the request body. `−−post−data` sends *string* as data, whereas `−−post−file` sends the contents of *file*. Other than that, they work in exactly the same way.

Please be aware that Wget needs to know the size of the POST data in advance. Therefore the argument to `−−post−file` must be a regular file; specifying a FIFO or something like */dev/stdin* won't work. It's not quite clear how to work around this limitation inherent in HTTP/1.0. Although HTTP/1.1 introduces *chunked* transfer that doesn't require knowing the request length in advance, a client can't use chunked unless it knows it's talking to an HTTP/1.1 server. And it can't know that until it receives a response, which in turn requires the request to have been completed — a chicken-and-egg problem.

Note: if Wget is redirected after the POST request is completed, it will not send the POST data to the redirected URL. This is because URLs that process POST often respond with a redirection to a regular page, which does not desire or accept POST. It is not completely clear that this behavior is optimal; if it doesn't work out, it might be changed in the future.

This example shows how to log to a server using POST and then proceed to download the desired pages, presumably only accessible to authorized users:

```
# Log in to the server.  This can be done only once.
wget --save-cookies cookies.txt \
     --post-data 'user=foo&password=bar' \
     http://server.com/auth.php

# Now grab the page or pages we care about.
wget --load-cookies cookies.txt \
     -p http://server.com/interesting/article.php
```

If the server is using session cookies to track user authentication, the above will not work because **−−save−cookies** will not save them (and neither will browsers) and the *cookies.txt* file will be empty. In that case use **−−keep−session−cookies** along with **−−save−cookies** to force saving of session cookies.

HTTPS (SSL/TLS) Options

To support encrypted HTTP (HTTPS) downloads, Wget must be compiled with an external SSL library, currently OpenSSL. If Wget is compiled without SSL support, none of these options are available.

−−secure−protocol=*protocol*

Choose the secure protocol to be used. Legal values are **auto**, **SSLv2**, **SSLv3**, and **TLSv1**. If **auto** is used, the SSL library is given the liberty of choosing the appropriate protocol automatically, which is achieved by sending an SSLv2 greeting and announcing support for SSLv3 and TLSv1. This is the default.

Specifying **SSLv2**, **SSLv3**, or **TLSv1** forces the use of the corresponding protocol. This is useful when talking to old and buggy SSL server implementations that make it hard for OpenSSL to choose the correct protocol version. Fortunately, such servers are quite rare.

−−no−check−certificate

Don't check the server certificate against the available certificate authorities. Also don't require the URL host name to match the common name presented by the certificate.

As of Wget 1.10, the default is to verify the server's certificate against the recognized certificate authorities, breaking the SSL handshake and aborting the download if the verification fails. Although this provides more secure downloads, it does break interoperability with some sites that worked with previous Wget versions, particularly those using self−signed, expired, or otherwise invalid certificates. This option forces an ''insecure'' mode of operation that turns the certificate verification errors into warnings and allows you to proceed.

If you encounter ''certificate verification'' errors or ones saying that ''common name doesn't match requested host name'', you can use this option to bypass the verification and proceed with the download. *Only use this option if you are otherwise convinced of the site's authenticity, or if you really don't care about the validity of its certificate.* It is almost always a bad idea not to check the certificates when transmitting confidential or important data.

−−certificate=*file*

Use the client certificate stored in *file*. This is needed for servers that are configured to require certificates from the clients that connect to them. Normally a certificate is not required and this switch is optional.

−−certificate−type=*type*
Specify the type of the client certificate. Legal values are **PEM** (assumed by default) and **DER**, also known as **ASN1**.

−−private−key=*file*
Read the private key from *file*. This allows you to provide the private key in a file separate from the certificate.

−−private−key−type=*type*
Specify the type of the private key. Accepted values are **PEM** (the default) and **DER**.

−−ca−certificate=*file*
Use *file* as the file with the bundle of certificate authorities (''CA'') to verify the peers. The certificates must be in PEM format.

Without this option Wget looks for CA certificates at the system-specified locations, chosen at OpenSSL installation time.

−−ca−directory=*directory*
Specifies directory containing CA certificates in PEM format. Each file contains one CA certificate, and the file name is based on a hash value derived from the certificate. This is achieved by processing a certificate directory with the `c_rehash` utility supplied with OpenSSL. Using **−−ca−directory** is more efficient than **−−ca−certificate** when many certificates are installed because it allows Wget to fetch certificates on demand.

Without this option Wget looks for CA certificates at the system-specified locations, chosen at OpenSSL installation time.

−−random−file=*file*
Use *file* as the source of random data for seeding the pseudo-random number generator on systems without */dev/random*.

On such systems the SSL library needs an external source of randomness to initialize. Randomness may be provided by EGD (see **−−egd−file** below) or read from an external source specified by the user. If this option is not specified, Wget looks for random data in `$RANDFILE` or, if that is unset, in *$HOME/.rnd*. If none of those are available, it is likely that SSL encryption will not be usable.

If you're getting the ''Could not seed OpenSSL PRNG; disabling SSL.'' error, you should provide random data using some of the methods described above.

−−egd−file=*file*
Use *file* as the EGD socket. EGD stands for *Entropy Gathering Daemon*, a user-space program that collects data from various unpredictable system sources and makes it available to other programs that might need it. Encryption software, such as the SSL library, needs sources of non-repeating randomness to seed the random number generator used to produce cryptographically strong keys.

OpenSSL allows the user to specify his own source of entropy using the `RAND_FILE` environment variable. If this variable is unset, or if the specified file does not produce enough randomness, OpenSSL will read random data from EGD socket specified using this option.

If this option is not specified (and the equivalent startup command is not used), EGD is never contacted. EGD is not needed on modern Unix systems that support */dev/random*.

FTP Options

−−ftp−user=*user*
−−ftp−password=*password*
Specify the username *user* and password *password* on an FTP server. Without this, or the corresponding startup option, the password defaults to **−wget@**, normally used for anonymous FTP.

Another way to specify username and password is in the URL itself. Either method reveals your password to anyone who bothers to run `ps`. To prevent the passwords from being seen, store them in

.wgetrc or *.netrc*, and make sure to protect those files from other users with `chmod`. If the passwords are really important, do not leave them lying in those files either−−−edit the files and delete them after Wget has started the download.

−−no−remove−listing
Don't remove the temporary *.listing* files generated by FTP retrievals. Normally, these files contain the raw directory listings received from FTP servers. Not removing them can be useful for debugging purposes, or when you want to be able to easily check on the contents of remote server directories (e.g. to verify that a mirror you're running is complete).

Note that even though Wget writes to a known filename for this file, this is not a security hole in the scenario of a user making *.listing* a symbolic link to */etc/passwd* or something and asking `root` to run Wget in his or her directory. Depending on the options used, either Wget will refuse to write to *.listing*, making the globbing/recursion/time−stamping operation fail, or the symbolic link will be deleted and replaced with the actual *.listing* file, or the listing will be written to a *.listing.number* file.

Even though this situation isn't a problem, though, `root` should never run Wget in a non-trusted user's directory. A user could do something as simple as linking *index.html* to */etc/passwd* and asking `root` to run Wget with **−N** or **−r** so the file will be overwritten.

−−no−glob
Turn off FTP globbing. Globbing refers to the use of shell-like special characters (*wildcards*), like *****, **?**, **[** and **]** to retrieve more than one file from the same directory at once, like:

```
        wget ftp://gnjilux.srk.fer.hr/*.msg
```

By default, globbing will be turned on if the URL contains a globbing character. This option may be used to turn globbing on or off permanently.

You may have to quote the URL to protect it from being expanded by your shell. Globbing makes Wget look for a directory listing, which is system−specific. This is why it currently works only with Unix FTP servers (and the ones emulating Unix `ls` output).

−−no−passive−ftp
Disable the use of the *passive* FTP transfer mode. Passive FTP mandates that the client connect to the server to establish the data connection rather than the other way around.

If the machine is connected to the Internet directly, both passive and active FTP should work equally well. Behind most firewall and NAT configurations passive FTP has a better chance of working. However, in some rare firewall configurations, active FTP actually works when passive FTP doesn't. If you suspect this to be the case, use this option, or set `passive_ftp=off` in your init file.

−−retr−symlinks
Usually, when retrieving FTP directories recursively and a symbolic link is encountered, the linked-to file is not downloaded. Instead, a matching symbolic link is created on the local filesystem. The pointed-to file will not be downloaded unless this recursive retrieval would have encountered it separately and downloaded it anyway.

When **−−retr−symlinks** is specified, however, symbolic links are traversed and the pointed-to files are retrieved. At this time, this option does not cause Wget to traverse symlinks to directories and recurse through them, but in the future it should be enhanced to do this.

Note that when retrieving a file (not a directory) because it was specified on the command−line, rather than because it was recursed to, this option has no effect. Symbolic links are always traversed in this case.

−−no−http−keep−alive
Turn off the ‘‘keep−alive’’ feature for HTTP downloads. Normally, Wget asks the server to keep the connection open so that, when you download more than one document from the same server, they get transferred over the same TCP connection. This saves time and at the same time reduces the load on the server.

This option is useful when, for some reason, persistent (keep−alive) connections don't work for you, for example due to a server bug or due to the inability of server-side scripts to cope with the connections.

Recursive Retrieval Options

−r
−−recursive
Turn on recursive retrieving.

−l *depth*
−−level=*depth*
Specify recursion maximum depth level *depth*. The default maximum depth is 5.

−−delete−after
This option tells Wget to delete every single file it downloads, *after* having done so. It is useful for pre-fetching popular pages through a proxy, e.g.:

```
        wget -r -nd --delete-after http://whatever.com/~popular/page/
```

The **−r** option is to retrieve recursively, and **−nd** to not create directories.

Note that **−−delete−after** deletes files on the local machine. It does not issue the **DELE** command to remote FTP sites, for instance. Also note that when **−−delete−after** is specified, **−−convert−links** is ignored, so **.orig** files are simply not created in the first place.

−k
−−convert−links
After the download is complete, convert the links in the document to make them suitable for local viewing. This affects not only the visible hyperlinks, but any part of the document that links to external content, such as embedded images, links to style sheets, hyperlinks to non-HTML content, etc.

Each link will be changed in one of the two ways:

- The links to files that have been downloaded by Wget will be changed to refer to the file they point to as a relative link.

  Example: if the downloaded file */foo/doc.html* links to */bar/img.gif*, also downloaded, then the link in *doc.html* will be modified to point to **../bar/img.gif**. This kind of transformation works reliably for arbitrary combinations of directories.

- The links to files that have not been downloaded by Wget will be changed to include host name and absolute path of the location they point to.

  Example: if the downloaded file */foo/doc.html* links to */bar/img.gif* (or to *../bar/img.gif*), then the link in *doc.html* will be modified to point to *http://hostname/bar/img.gif*.

Because of this, local browsing works reliably: if a linked file was downloaded, the link will refer to its local name; if it was not downloaded, the link will refer to its full Internet address rather than presenting a broken link. The fact that the former links are converted to relative links ensures that you can move the downloaded hierarchy to another directory.

Note that only at the end of the download can Wget know which links have been downloaded. Because of that, the work done by **−k** will be performed at the end of all the downloads.

−K
−−backup−converted
When converting a file, back up the original version with a **.orig** suffix. Affects the behavior of **−N**.

−m
−−mirror
Turn on options suitable for mirroring. This option turns on recursion and time−stamping, sets infinite recursion depth and keeps FTP directory listings. It is currently equivalent to **−r −N −l inf**

−−no−remove−listing.

−p
−−page−requisites

This option causes Wget to download all the files that are necessary to properly display a given HTML page. This includes such things as inlined images, sounds, and referenced stylesheets.

Ordinarily, when downloading a single HTML page, any requisite documents that may be needed to display it properly are not downloaded. Using **−r** together with **−l** can help, but since Wget does not ordinarily distinguish between external and inlined documents, one is generally left with ''leaf documents'' that are missing their requisites.

For instance, say document *1.html* contains an `<IMG>` tag referencing *1.gif* and an `<A>` tag pointing to external document *2.html*. Say that *2.html* is similar but that its image is *2.gif* and it links to *3.html*. Say this continues up to some arbitrarily high number.

If one executes the command:

```
wget -r -l 2 http://<site>/1.html
```

then *1.html*, *1.gif*, *2.html*, *2.gif*, and *3.html* will be downloaded. As you can see, *3.html* is without its requisite *3.gif* because Wget is simply counting the number of hops (up to 2) away from *1.html* in order to determine where to stop the recursion. However, with this command:

```
wget -r -l 2 -p http://<site>/1.html
```

all the above files *and 3.html*'s requisite *3.gif* will be downloaded. Similarly,

```
wget -r -l 1 -p http://<site>/1.html
```

will cause *1.html*, *1.gif*, *2.html*, and *2.gif* to be downloaded. One might think that:

```
wget -r -l 0 -p http://<site>/1.html
```

would download just *1.html* and *1.gif*, but unfortunately this is not the case, because **−l 0** is equivalent to **−l inf**−−−that is, infinite recursion. To download a single HTML page (or a handful of them, all specified on the command-line or in a **−i** URL input file) and its (or their) requisites, simply leave off **−r** and **−l**:

```
wget -p http://<site>/1.html
```

Note that Wget will behave as if **−r** had been specified, but only that single page and its requisites will be downloaded. Links from that page to external documents will not be followed. Actually, to download a single page and all its requisites (even if they exist on separate websites), and make sure the lot displays properly locally, this author likes to use a few options in addition to **−p**:

```
wget -E -H -k -K -p http://<site>/<document>
```

To finish off this topic, it's worth knowing that Wget's idea of an external document link is any URL specified in an `<A>` tag, an `<AREA>` tag, or a `<LINK>` tag other than `<LINK REL="stylesheet">`.

−−strict−comments

Turn on strict parsing of HTML comments. The default is to terminate comments at the first occurrence of **−−>**.

According to specifications, HTML comments are expressed as SGML *declarations*. Declaration is special markup that begins with **<!** and ends with **>**, such as **<!DOCTYPE ...>**, that may contain comments between a pair of **−−** delimiters. HTML comments are ''empty declarations'', SGML declarations without any non-comment text. Therefore, **<!−−foo−−>** is a valid comment, and so is **<!−−one−− −−two−−>**, but **<!−−1−−2−−>** is not.

On the other hand, most HTML writers don't perceive comments as anything other than text delimited with **<!−−** and **−−>**, which is not quite the same. For example, something like **<!−−−−−−−−−−−−>**

works as a valid comment as long as the number of dashes is a multiple of four (!). If not, the comment technically lasts until the next `--`, which may be at the other end of the document. Because of this, many popular browsers completely ignore the specification and implement what users have come to expect: comments delimited with `<!--` and `-->`.

Until version 1.9, Wget interpreted comments strictly, which resulted in missing links in many web pages that displayed fine in browsers, but had the misfortune of containing non-compliant comments. Beginning with version 1.9, Wget has joined the ranks of clients that implements ''naive'' comments, terminating each comment at the first occurrence of `-->`.

If, for whatever reason, you want strict comment parsing, use this option to turn it on.

Recursive Accept/Reject Options

−A *acclist* **−−accept** *acclist*
−R *rejlist* **−−reject** *rejlist*

Specify comma-separated lists of file name suffixes or patterns to accept or reject (@pxref{Types of Files} for more details).

−D *domain-list*
−−domains=*domain-list*

Set domains to be followed. *domain-list* is a comma-separated list of domains. Note that it does *not* turn on **−H**.

−−exclude−domains *domain-list*

Specify the domains that are *not* to be followed..

−−follow−ftp

Follow FTP links from HTML documents. Without this option, Wget will ignore all the FTP links.

−−follow−tags=*list*

Wget has an internal table of HTML tag / attribute pairs that it considers when looking for linked documents during a recursive retrieval. If a user wants only a subset of those tags to be considered, however, he or she should be specify such tags in a comma-separated *list* with this option.

−−ignore−tags=*list*

This is the opposite of the **−−follow−tags** option. To skip certain HTML tags when recursively looking for documents to download, specify them in a comma-separated *list*.

In the past, this option was the best bet for downloading a single page and its requisites, using a command-line like:

```
        wget --ignore-tags=a,area -H -k -K -r http://<site>/<document>
```

However, the author of this option came across a page with tags like `<LINK REL="home" HREF="/">` and came to the realization that specifying tags to ignore was not enough. One can't just tell Wget to ignore `<LINK>`, because then stylesheets will not be downloaded. Now the best bet for downloading a single page and its requisites is the dedicated **−−page−requisites** option.

−H
−−span−hosts

Enable spanning across hosts when doing recursive retrieving.

−L
−−relative

Follow relative links only. Useful for retrieving a specific home page without any distractions, not even those from the same hosts.

−I *list*
−−include−directories=*list*

Specify a comma-separated list of directories you wish to follow when downloading (@pxref{Directory−Based Limits} for more details.) Elements of *list* may contain wildcards.

−X *list*
−−exclude−directories=*list*

Specify a comma-separated list of directories you wish to exclude from download (@pxref{Directory−Based Limits} for more details.) Elements of *list* may contain wildcards.

−np
−−no−parent

Do not ever ascend to the parent directory when retrieving recursively. This is a useful option, since it guarantees that only the fi les *below* a certain hierarchy will be downloaded.

EXAMPLES

The examples are divided into three sections loosely based on their complexity.

Simple Usage

- Say you want to download a URL. Just type:

```
        wget http://fly.srk.fer.hr/
```

- But what will happen if the connection is slow, and the fi le is lengthy? The connection will probably fail before the whole fi le is retrieved, more than once. In this case, Wget will try getting the fi le until it either gets the whole of it, or exceeds the default number of retries (this being 20). It is easy to change the number of tries to 45, to insure that the whole fi le will arrive safely:

```
        wget --tries=45 http://fly.srk.fer.hr/jpg/flyweb.jpg
```

- Now let’s leave Wget to work in the background, and write its progress to log fi le *log*. It is tiring to type **−−tries**, so we shall use **−t**.

```
        wget -t 45 -o log http://fly.srk.fer.hr/jpg/flyweb.jpg &
```

  The ampersand at the end of the line makes sure that Wget works in the background. To unlimit the number of retries, use **−t inf**.

- The usage of FTP is as simple. Wget will take care of login and password.

```
        wget ftp://gnjilux.srk.fer.hr/welcome.msg
```

- If you specify a directory, Wget will retrieve the directory listing, parse it and convert it to HTML. Try:

```
        wget ftp://ftp.gnu.org/pub/gnu/
        links index.html
```

Advanced Usage

- You have a fi le that contains the URLs you want to download? Use the **−i** switch:

```
        wget -i <file>
```

  If you specify **−** as fi le name, the URLs will be read from standard input.

- Create a fi ve levels deep mirror image of the GNU web site, with the same directory structure the original has, with only one try per document, saving the log of the activities to *gnulog*:

```
        wget -r http://www.gnu.org/ -o gnulog
```

- The same as the above, but convert the links in the HTML fi les to point to local fi les, so you can view the documents off−line:

```
        wget --convert-links -r http://www.gnu.org/ -o gnulog
```

- Retrieve only one HTML page, but make sure that all the elements needed for the page to be displayed, such as inline images and external style sheets, are also downloaded. Also make sure the downloaded page references the downloaded links.

```
wget -p --convert-links http://www.server.com/dir/page.html
```

The HTML page will be saved to *www.server.com/dir/page.html*, and the images, stylesheets, etc., somewhere under *www.server.com/*, depending on where they were on the remote server.

- The same as the above, but without the *www.server.com/* directory. In fact, I don't want to have all those random server directories anyway−−−just save *all* those files under a *download/* subdirectory of the current directory.

  ```
  wget -p --convert-links -nH -nd -Pdownload \
       http://www.server.com/dir/page.html
  ```

- Retrieve the index.html of **www.lycos.com**, showing the original server headers:

  ```
  wget -S http://www.lycos.com/
  ```

- Save the server headers with the file, perhaps for post−processing.

  ```
  wget --save-headers http://www.lycos.com/
  more index.html
  ```

- Retrieve the first two levels of **wuarchive.wustl.edu**, saving them to */tmp*.

  ```
  wget -r -l2 -P/tmp ftp://wuarchive.wustl.edu/
  ```

- You want to download all the GIFs from a directory on an HTTP server. You tried **wget http://www.server.com/dir/*.gif**, but that didn't work because HTTP retrieval does not support globbing. In that case, use:

  ```
  wget -r -l1 --no-parent -A.gif http://www.server.com/dir/
  ```

  More verbose, but the effect is the same. **−r −l1** means to retrieve recursively, with maximum depth of 1. **−−no−parent** means that references to the parent directory are ignored, and **−A.gif** means to download only the GIF files. **−A "*.gif"** would have worked too.

- Suppose you were in the middle of downloading, when Wget was interrupted. Now you do not want to clobber the files already present. It would be:

  ```
  wget -nc -r http://www.gnu.org/
  ```

- If you want to encode your own username and password to HTTP or FTP, use the appropriate URL syntax.

  ```
  wget ftp://hniksic:mypassword@unix.server.com/.emacs
  ```

  Note, however, that this usage is not advisable on multi-user systems because it reveals your password to anyone who looks at the output of `ps`.

- You would like the output documents to go to standard output instead of to files?

  ```
  wget -O - http://jagor.srce.hr/ http://www.srce.hr/
  ```

  You can also combine the two options and make pipelines to retrieve the documents from remote hotlists:

  ```
  wget -O - http://cool.list.com/ | wget --force-html -i -
  ```

Very Advanced Usage

- If you wish Wget to keep a mirror of a page (or FTP subdirectories), use **−−mirror** (**−m**), which is the shorthand for **−r −l inf −N**. You can put Wget in the crontab file asking it to recheck a site each Sunday:

  ```
  crontab
  0 0 * * 0 wget --mirror http://www.gnu.org/ -o /home/me/weeklog
  ```

- In addition to the above, you want the links to be converted for local viewing. But, after having read this manual, you know that link conversion doesn't play well with timestamping, so you also want Wget to back up the original HTML files before the conversion. Wget invocation would look like this:

```
wget --mirror --convert-links --backup-converted  \
     http://www.gnu.org/ -o /home/me/weeklog
```

- But you've also noticed that local viewing doesn't work all that well when HTML files are saved under extensions other than **.html**, perhaps because they were served as *index.cgi*. So you'd like Wget to rename all the files served with content-type **text/html** or **application/xhtml+xml** to *name.html*.

```
wget --mirror --convert-links --backup-converted \
     --html-extension -o /home/me/weeklog        \
     http://www.gnu.org/
```

  Or, with less typing:

```
wget -m -k -K -E http://www.gnu.org/ -o /home/me/weeklog
```

FILES

/usr/local/etc/wgetrc
Default location of the *global* startup file.

.wgetrc
User startup file.

BUGS

You are welcome to send bug reports about GNU Wget to <**bug−wget@gnu.org**>.

Before actually submitting a bug report, please try to follow a few simple guidelines.

1. Please try to ascertain that the behavior you see really is a bug. If Wget crashes, it's a bug. If Wget does not behave as documented, it's a bug. If things work strange, but you are not sure about the way they are supposed to work, it might well be a bug.

2. Try to repeat the bug in as simple circumstances as possible. E.g. if Wget crashes while downloading **wget −rl0 −kKE −t5 −Y0 http://yoyodyne.com −o /tmp/log**, you should try to see if the crash is repeatable, and if will occur with a simpler set of options. You might even try to start the download at the page where the crash occurred to see if that page somehow triggered the crash.

   Also, while I will probably be interested to know the contents of your *.wgetrc* file, just dumping it into the debug message is probably a bad idea. Instead, you should first try to see if the bug repeats with *.wgetrc* moved out of the way. Only if it turns out that *.wgetrc* settings affect the bug, mail me the relevant parts of the file.

3. Please start Wget with **−d** option and send us the resulting output (or relevant parts thereof). If Wget was compiled without debug support, recompile it−−−it is *much* easier to trace bugs with debug support on.

   Note: please make sure to remove any potentially sensitive information from the debug log before sending it to the bug address. The `-d` won't go out of its way to collect sensitive information, but the log *will* contain a fairly complete transcript of Wget's communication with the server, which may include passwords and pieces of downloaded data. Since the bug address is publically archived, you may assume that all bug reports are visible to the public.

4. If Wget has crashed, try to run it in a debugger, e.g. `gdb 'which wget' core` and type `where` to get the backtrace. This may not work if the system administrator has disabled core files, but it is safe to try.

SEE ALSO

GNU Info entry for *wget*.

AUTHOR

Originally written by Hrvoje Niksic <hniksic@xemacs.org>.

COPYRIGHT