

Wget Cheatsheet

By Dejan Panovski • Updated on Feb 14, 2026 • [Download PDF](#)

Quick reference for downloading files and mirroring websites with wget

Wget is a command-line utility for downloading files over HTTP, HTTPS, and FTP. It supports recursive downloads, resume, bandwidth limiting, and website mirroring. This cheatsheet covers the most useful commands for everyday use.

Basic Downloads

Download files to the current directory.

wget URL	Download a file
<code>wget -q URL</code>	Download silently (no output)
<code>wget -nv URL</code>	Less verbose (errors and basic info only)
<code>wget -v URL</code>	Verbose output (default)
<code>wget -d URL</code>	Debug output

Save Options

Control where and how files are saved.

<code>wget -O name.txt URL</code>	Save with a specific filename
<code>wget -O - URL</code>	Output to stdout
<code>wget -P /path/ URL</code>	Save to a specific directory
<code>wget --content-disposition URL</code>	Use server-suggested filename
<code>wget --no-clobber URL</code>	Skip if file already exists
<code>wget -nc URL</code>	Short form of <code>--no-clobber</code>

Resume and Speed

Resume interrupted downloads and control bandwidth.

<code>wget -c URL</code>	Resume a partial download
<code>wget --limit-rate=2m URL</code>	Limit speed to 2 MB/s
<code>wget --limit-rate=500k URL</code>	Limit speed to 500 KB/s
<code>wget -b URL</code>	Download in background
<code>wget -b -o log.txt URL</code>	Background with custom log

Multiple Files

Download several files at once.

<code>wget URL1 URL2 URL3</code>	Download multiple URLs
<code>wget -i urls.txt</code>	Download URLs from a file
<code>wget -i - < urls.txt</code>	Read URLs from stdin

Recursive Download

Download directory trees and linked pages.

<code>wget -r URL</code>	Recursive download
<code>wget -r -l 2 URL</code>	Limit depth to 2 levels
<code>wget -r --no-parent URL</code>	Do not ascend to parent directory
<code>wget -r --accept=jpg,png URL</code>	Accept only these file types
<code>wget -r --reject=mp4,avi URL</code>	Reject these file types
<code>wget -r -A "*.pdf" URL</code>	Download only PDFs
<code>wget -r --follow-tags=a URL</code>	Follow only <code><a></code> tags

Website Mirroring

Create offline copies of websites.

<code>wget -m URL</code>	Mirror a website
<code>wget -m -k -p URL</code>	Mirror with local link conversion
<code>wget -m -k -p -E URL</code>	Mirror and add <code>.html</code> extensions
<code>wget -m --wait=2 URL</code>	Mirror with 2 second delay between requests
<code>wget -m --random-wait URL</code>	Mirror with random delay (0.5x-1.5x of <code>--wait</code>)

Authentication

Access protected resources.

<code>wget --user=USER --password=PASS URL</code>	HTTP basic auth
<code>wget --ask-password URL</code>	Prompt for password
<code>wget --header="Authorization: Bearer TOKEN" URL</code>	Token auth
<code>wget --ftp-user=USER --ftp-password=PASS URL</code>	FTP auth

Headers and User Agent

Send custom headers and change identity.

<code>wget --header="Key: Value" URL</code>	Add custom header
<code>wget --header="Accept: application/json" URL</code>	Request JSON
<code>wget -U "CustomAgent/1.0" URL</code>	Change user agent
<code>wget --referer=URL URL</code>	Set referer header

SSL/TLS

Handle HTTPS connections and certificates.

<code>wget --no-check-certificate URL</code>	Skip certificate verification
<code>wget --ca-certificate=ca.crt URL</code>	Use custom CA certificate
<code>wget --certificate=client.crt URL</code>	Client certificate
<code>wget --private-key=client.key URL</code>	Client private key
<code>wget --https-only URL</code>	Only follow HTTPS links

FTP

Work with FTP servers.

<code>wget ftp://server/file</code>	Download file
<code>wget -r ftp://server/dir/</code>	Download directory recursively
<code>wget --ftp-user=USER --ftp-password=PASS ftp://server/</code>	Authenticated FTP
<code>wget --no-passive-ftp ftp://server/file</code>	Use active FTP mode
<code>wget --no-remove-listing ftp://server/dir/</code>	Keep <code>.listing</code> files

Retries and Timeouts

Control retry behavior and connection timing.

<code>wget --tries=5 URL</code>	Retry up to 5 times
<code>wget --retry-connrefused URL</code>	Retry on connection refused
<code>wget --waitretry=10 URL</code>	Wait 10 seconds between retries
<code>wget --timeout=30 URL</code>	Set all timeouts to 30 seconds
<code>wget --connect-timeout=10 URL</code>	Connection timeout only
<code>wget --read-timeout=30 URL</code>	Read timeout only
<code>wget --dns-timeout=5 URL</code>	DNS timeout only

Output and Logging

Control progress display and log output.

<code>wget -q URL</code>	Suppress all output
<code>wget -nv URL</code>	Print errors and basic info only
<code>wget -o log.txt URL</code>	Log output to file
<code>wget -a log.txt URL</code>	Append to log file
<code>wget --show-progress -q URL</code>	Quiet but show progress bar
<code>wget --progress=dot URL</code>	Dot-style progress indicator

Timestamping and Caching

Download only new or updated files.

<code>wget -N URL</code>	Download only if remote file is newer
<code>wget --no-cache URL</code>	Disable server-side caching
<code>wget --spider URL</code>	Check if URL exists (do not download)
<code>wget --spider -r URL</code>	Check all links recursively

Common Patterns

Frequently used command combinations.

<code>wget -q -O - URL tar -xzf - -C /path</code>	Download and extract in one step
<code>wget -c --limit-rate=1m -P /tmp URL</code>	Resume to directory with speed limit
<code>wget -r -np -nH --cut-dirs=2 URL</code>	Recursive without host and path prefix
<code>wget -m -k -p --wait=1 -e robots=off URL</code>	Full mirror ignoring robots.txt

Related Guides

Use these articles for detailed wget workflows.

[Wget Command in Linux with Examples](#)

Full wget tutorial with practical examples

[Curl Command in Linux with Examples](#)

Alternative HTTP client for API interactions

[How to Use SCP Command to Securely Transfer Files](#)

Transfer files between hosts over SSH

[How to Use Rsync for Local and Remote Data Transfer](#)

Incremental file sync and transfer