# The Ultimate Wget Download Guide With 15 Awesome Examples

**wget** utility is the best option to download files from internet. wget can pretty much handle all complex download situations including large file downloads, recursive downloads, non-interactive downloads, multiple file downloads etc.,

In this article let us review how to use **wget** for various download scenarios using 15 awesome wget examples.

## 1. Download Single File with wget

The following example downloads a single file from internet and stores in the current directory.

```
$ wget http://www.openss7.org/repos/tarballs/strx25-0.9.2.1.tar.bz2
```

While downloading it will show a progress bar with the following information:

- %age of download completion (for e.g. 31% as shown below)
- Total amount of bytes downloaded so far (for e.g. 1,213,592 bytes as shown below)
- Current download speed (for e.g. 68.2K/s as shown below)
- Remaining time to download (for e.g. eta 34 seconds as shown below)

Download in progress:

```
$ wget http://www.openss7.org/repos/tarballs/strx25-0.9.2.1.tar.bz2
Saving to: `strx25-0.9.2.1.tar.bz2.1'

31% [=================> 1,213,592   68.2K/s  eta 34s
```

Download completed:

```
$ wget http://www.openss7.org/repos/tarballs/strx25-0.9.2.1.tar.bz2
Saving to: `strx25-0.9.2.1.tar.bz2'

100%[======================>] 3,852,374   76.8K/s   in 55s

2009-09-25 11:15:30 (68.7 KB/s) - `strx25-0.9.2.1.tar.bz2' saved [3852374/3852374]
```

## 2. Download and Store With a Different File name Using wget -O

By default wget will pick the filename from the last word after last forward slash, which may not be appropriate always.

**Wrong:** Following example will download and store the file with name: download_script.php?src_id=7701

```
$ wget http://www.vim.org/scripts/download_script.php?src_id=7701
```

Even though the downloaded file is in zip format, it will get stored in the file as shown below.

```
$ ls
download_script.php?src_id=7701
```

**Correct:** To correct this issue, we can specify the output file name using the -O option as:

```
$ wget -O taglist.zip http://www.vim.org/scripts/download_script.php?src_id=7701
```

## 3. Specify Download Speed / Download Rate Using wget –limit-rate

While executing the wget, by default it will try to occupy full possible bandwidth. This might not be acceptable when you are downloading huge files on production servers. So, to avoid that we can limit the download speed using the –limit-rate as shown below.

In the following example, the download speed is limited to 200k

```
$ wget --limit-rate=200k http://www.openss7.org/repos/tarballs/strx25-0.9.2.1.tar.bz2
```

## 4. Continue the Incomplete Download Using wget -c

Restart a download which got stopped in the middle using wget -c option as shown below.

```
$ wget -c http://www.openss7.org/repos/tarballs/strx25-0.9.2.1.tar.bz2
```

This is very helpful when you have initiated a very big file download which got interrupted in the middle. Instead of starting the whole download again, you can start the download from where it got interrupted using option -c

**Note:** If a download is stopped in middle, when you restart the download again without the option -c, wget will append .1 to the filename automatically as a file with the previous name already exist. If a file with .1 already exist, it will download the file with .2 at the end.

## 5. Download in the Background Using wget -b

For a huge download, put the download in background using wget option -b as shown below.

```
$ wget -b http://www.openss7.org/repos/tarballs/strx25-0.9.2.1.tar.bz2
Continuing in background, pid 1984.
Output will be written to `wget-log'.
```

It will initiate the download and gives back the shell prompt to you. You can always check the status of the download using tail -f as shown below.

```
$ tail -f wget-log
Saving to: `strx25-0.9.2.1.tar.bz2.4'

     0K .......... .......... .......... .......... ..........  1% 65.5K 57s
    50K .......... .......... .......... .......... ..........  2% 85.9K 49s
   100K .......... .......... .......... .......... ..........  3% 83.3K 47s
   150K .......... .......... .......... .......... ..........  5% 86.6K 45s
   200K .......... .......... .......... .......... ..........  6% 33.9K 56s
   250K .......... .......... .......... .......... ..........  7%  182M 46s
   300K .......... .......... .......... .......... ..........  9% 57.9K 47s
```

Also, make sure to review our previous multitail article on how to use tail command effectively to view multiple files.

## 6. Mask User Agent and Display wget like Browser Using wget –user-agent

Some websites can disallow you to download its page by identifying that the user agent is not a browser. So you can mask the user agent by using –user-agent options and show wget like a browser as shown below.

```
$ wget --user-agent="Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.3) Gecko/2008092416 Firefox/3.0
```

## 7. Test Download URL Using wget –spider

When you are going to do scheduled download, you should check whether download will happen fine or not at scheduled time. To do so, copy the line exactly from the schedule, and then add –spider option to check.

```
$ wget --spider DOWNLOAD-URL
```

If the URL given is correct, it will say

```
$ wget --spider download-url
Spider mode enabled. Check if remote file exists.
```

```
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Remote file exists and could contain further links,
but recursion is disabled -- not retrieving.
```

This ensures that the downloading will get success at the scheduled time. But when you had give a wrong URL, you will get the following error.

```
$ wget --spider download-url
Spider mode enabled. Check if remote file exists.
HTTP request sent, awaiting response... 404 Not Found
Remote file does not exist -- broken link!!!
```

You can use the spider option under following scenarios:

- Check before scheduling a download.
- Monitoring whether a website is available or not at certain intervals.
- Check a list of pages from your bookmark, and find out which pages are still exists.

## 8. Increase Total Number of Retry Attempts Using wget –tries

If the internet connection has problem, and if the download file is large there is a chance of failures in the download. By default wget retries 20 times to make the download successful.

If needed, you can increase retry attempts using –tries option as shown below.

```
$ wget --tries=75 DOWNLOAD-URL
```

## 9. Download Multiple Files / URLs Using Wget -i

First, store all the download files or URLs in a text file as:

```
$ cat > download-file-list.txt
URL1
URL2
URL3
URL4
```

Next, give the download-file-list.txt as argument to wget using -i option as shown below.

```
$ wget -i download-file-list.txt
```

## 10. Download a Full Website Using wget –mirror

Following is the command line which you want to execute when you want to download a full website and made available for local viewing.

```
$ wget --mirror -p --convert-links -P ./LOCAL-DIR WEBSITE-URL
```

- –mirror : turn on options suitable for mirroring.
- -p : download all files that are necessary to properly display a given HTML page.
- –convert-links : after the download, convert the links in document for local viewing.
- -P ./LOCAL-DIR : save all the files and directories to the specified directory.

## 11. Reject Certain File Types while Downloading Using wget –reject

You have found a website which is useful, but don’t want to download the images you can specify the following.

```
$ wget --reject=gif WEBSITE-TO-BE-DOWNLOADED
```

## 12. Log messages to a log file instead of stderr Using wget -o

When you wanted the log to be redirected to a log file instead of the terminal.

```
$ wget -o download.log DOWNLOAD-URL
```

## 13. Quit Downloading When it Exceeds Certain Size Using wget -Q

When you want to stop download when it crosses 5 MB you can use the following wget command line.

```
$ wget -Q5m -i FILE-WHICH-HAS-URLS
```

**Note:** This quota will not get effect when you do a download a single URL. That is irrespective of the quota size everything will get downloaded when you specify a single file. This quota is applicable only for recursive downloads.

## 14. Download Only Certain File Types Using wget -r -A

You can use this under following situations:

- Download all images from a website
- Download all videos from a website
- Download all PDF files from a website

```
$ wget -r -A.pdf http://url-to-webpage-with-pdfs/
```

## 15. FTP Download With wget

You can use wget to perform FTP download as shown below.

Anonymous FTP download using Wget

```
$ wget ftp-url
```

FTP download using wget with username and password authentication.

```
$ wget --ftp-user=USERNAME --ftp-password=PASSWORD DOWNLOAD-URL
```