

wget vs curl: How to Download Files Using wget and curl

Both wget and curl are command line utilities that do the same thing.

- They both can be used to download files using FTP and HTTP(s).
- You can also send HTTP POST request using curl and wget
- However curl provides APIs that can be used by programmers inside their own code. curl uses libcurl which is a cross-platform library.
- wget is just a command-line tool without any APIs.
- Curl also supports lot more protocols that wget doesn't support. For example: SCP, SFTP, TFTP, TELNET, LDAP(S), FILE, POP3, IMAP, SMTP, RTMP and RTSP.
- There is a major advantage of using wget. wget supports recursive download, while curl doesn't.

Wget Examples

The following example downloads the file and stores in the same name as the remote server.

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

The following example download the file and stores in a different name than the remote server. This is helpful when the remote URL doesn't contain the file name in the url as shown in the example below.

```
wget -O taglist.zip http://www.vim.org/scripts/download_script.php?src_id=770
```

More wget examples: [The Ultimate Wget Download Guide With 15 Awesome Examples](#)

Curl Examples

```
$ curl -O http://www.openss7.org/repos/tarballs/strx25-0.9.2.1.tar.bz2
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total   Spent    Left   Speed
 28 3762k    28 1085k    0     0  72771      0  0:00:52  0:00:15  0:00:37 54267
```

Option -O (upper-case O) is important. Without this, curl will start dumping the downloaded file on the stdout. Using -O, it downloads the files in the same name as the remote server. In the above example, we are downloading strx25-0.9.2.1.tar.bz2, so the downloaded file will also be the same name.

Instead of -O, you can also specify, "--remote-name" as shown below. Both are the same.

```
$ curl --remote-name http://www.openss7.org/repos/tarballs/strx25-0.9.2.1.tar 2
```

While curl is downloading it gives the following useful information:

- % – The total % of the download that was completed as of now. When it gets to 100% the download is completed. In the above example, it has downloaded only 28% of the file.
- Total – The total size of the file
- Received – The total size of the file that has been downloaded so far. In the above example, it has downloaded 1085k so far (out of 3762k total)
- Xferd – This will be used when you upload some files to the remote server. During upload, this will indicate the total size of the file that has been uploaded so far. Since we are downloading a file, in this example, this is 0.
- Average Speed Dload – This indicates the average download speed.
- Average Speed Upload – While uploading a file, this will indicate the average upload speed
- Time Total – This indicates the total time it will take to download (or upload) the whole file based on the current download (or upload) speed. In this example, it will take approximately a total of 52 seconds to download this file.
- Time Spent – The time curl has spent so far downloading (or uploading) the file. In this example, it has spent 15 seconds so far.
- Time Left – This is calculated based on “Time Total” – “Time Spent”.
- Current Speed – This indicates the current download/upload speed. Compare this with Average Speed Dload/Upload to see how fast or slow your system is downloading currently.

If you want to download the file and store it in a different name than the name of the file in the remote server, use -o (lower-case o) as shown below. This is helpful when the remote URL doesn't contain the file name in the url as shown in the example below.

```
$ curl -o taglist.zip http://www.vim.org/scripts/download_script.php?src_id= 1
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total   Dload  Upload   Total   Spent    Left     Speed
100 50243  100 50243    0     0  170k      0  --:--:-- --:--:-- --:--:--  400k
```

In the above example, there is no file name in the remote URL, it just calls a php script that passes some parameter to it. However, the file will be downloaded and saved as taglist.zip on your local system. Instead of -o, you can also specify, “-output”.

More curl examples: [15 Practical Linux cURL Command Examples](#)