

15 Practical Linux cURL Command Examples (cURL Download Examples)

 thegeekstuff.com /2012/04/curl-examples/

cURL is a software package which consists of command line tool and a library for transferring data using URL syntax.

cURL supports various protocols like, DICT, FILE, FTP, FTPS, Gopher, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMTP, SMTPS, Telnet and TFTP.

This article provides 15 practical cURL usage examples.

1. Download a Single File

The following command will get the content of the URL and display it in the STDOUT (i.e on your terminal).

```
$ curl
http://www.centos.org
```

To store the output in a file, you can redirect it as shown below. This will also display some additional download statistics.

```
$ curl http://www.centos.org > centos-org.html
% Total    % Received % Xferd  Average Speed   Time    Time     Time
Current
                                 Dload  Upload  Total  Spent  Left
Speed
100 27329    0 27329    0    0   104k    0 --:--:-- --:--:-- --:--:--
167k
```

2. Save the cURL Output to a file

We can save the result of the curl command to a file by using -o/-O options.

- -o (lowercase o) the result will be saved in the filename provided in the command line
- -O (uppercase O) the filename in the URL will be taken and it will be used as the filename to store the result

```
$ curl -o mygettext.html
http://www.gnu.org/software/gettext/manual/gettext.html
```

Now the page gettext.html will be saved in the file named 'mygettext.html'. You can also note that when running curl with -o option, it displays the progress meter for the download as follows.



15 cURL Examples

% Total Current	% Received	% Xferd	Average Speed		Time	Time	Time
			Dload	Upload	Total	Spent	Left
Speed							
66 1215k	66 805k	0 0	33060	0	0:00:37	0:00:24	0:00:13
45900							
100 1215k	100 1215k	0 0	39474	0	0:00:31	0:00:31	--:--:--
68987							

When you use curl -O (uppercase O), it will save the content in the file named 'gettext.html' itself in the local machine.

```
$ curl -O
http://www.gnu.org/software/gettext/manual/gettext.html
```

Note: When curl has to write the data to the terminal, it disables the Progress Meter, to avoid confusion in printing. We can use '>|'-o'|'-O' options to move the result to a file.

Similar to cURL, you can also use wget to download files. Refer to [wget examples](#) to understand how to use wget effectively.

3. Fetch Multiple Files at a time

We can download multiple files in a single shot by specifying the URLs on the command line.

Syntax:

```
$ curl -O URL1 -O
URL2
```

The below command will download both index.html and gettext.html and save it in the same name under the current directory.

```
$ curl -O http://www.gnu.org/software/gettext/manual/html_node/index.html -O
http://www.gnu.org/software/gettext/manual/gettext.html
```

Please note that when we download multiple files from a same sever as shown above, curl will try to re-use the connection.

4. Follow HTTP Location Headers with -L option

By default CURL doesn't follow the HTTP Location headers. It is also termed as Redirects. When a requested web page is moved to another place, then an HTTP Location header will be sent as a Response and it will have where the actual web page is located.

For example, when someone types google.com in the browser from India, it will be automatically redirected to 'google.co.in'. This is done based on the HTTP Location header as shown below.

```
$ curl http://www.google.com

<TITLE>302 Moved</TITLE>
<H1>302 Moved</H1>
The document has moved
<A HREF="http://www.google.co.in/">here</A>
```

The above output says that the requested document is moved to 'http://www.google.co.in/'.

We can insist curl to follow the redirection using -L option, as shown below. Now it will download the google.co.in's html source code.

```
$ curl -L
http://www.google.com
```

5. Continue/Resume a Previous Download

Using curl -C option, you can continue a download which was stopped already for some reason. This will be helpful when you download large files, and the download got interrupted.

If we say '-C -', then curl will find from where to start resuming the download. We can also give an offset '-C <offset>'. The given offset bytes will be skipped from the beginning for the source file.

Start a big download using curl, and press Ctrl-C to stop it in between the download.

```
$ curl -O
http://www.gnu.org/software/gettext/manual/gettext.html
#####                20.1%
```

Note: -# is used to display a progress bar instead of a progress meter.

Now the above download was stopped at 20.1%. Using "curl -C -", we can continue the download from where it left off earlier. Now the download continues from 20.1%.

```
curl -C - -O
http://www.gnu.org/software/gettext/manual/gettext.html
#####                21.1%
```

6. Limit the Rate of Data Transfer

You can limit the amount at which the data gets transferred using --limit-rate option. You can specify the maximum transfer rate as argument.

```
$ curl --limit-rate 1000B -O
http://www.gnu.org/software/gettext/manual/gettext.html
```

The above command is limiting the data transfer to 1000 Bytes/second. curl may use higher transfer rate for short span of time. But on an average, it will come around to 1000B/second.

The following was the progress meter for the above command. You can see that the current speed is near to the 1000 Bytes.

% Total Current	% Received	% Xferd	Average Speed		Time Total	Time Spent	Time Left
			Dload	Upload			
Speed							
1 1215k	1 13601	0 0	957	0	0:21:40	0:00:14	0:21:26
999							
1 1215k	1 14601	0 0	960	0	0:21:36	0:00:15	0:21:21
999							
1 1215k	1 15601	0 0	962	0	0:21:34	0:00:16	0:21:18
999							

7. Download a file only if it is modified before/after the given time

We can get the files that are modified after a particular time using `-z` option in curl. This will work for both FTP & HTTP.

```
$ curl -z 21-Dec-11
http://www.example.com/yy.html
```

The above command will download the `yy.html` only if it is modified later than the given date and time

```
$ curl -z -21-Dec-11
http://www.example.com/yy.html
```

The above command will download the `yy.html`, if it is modified before than the given date and time.

Please refer 'man curl_getdate' for the various syntax supported for the date expression

8. Pass HTTP Authentication in cURL

Sometime, websites will require a username and password to view the content (can be done with `.htaccess` file). With the help of `-u` option, we can pass those credentials from cURL to the web server as shown below.

```
$ curl -u username:password
URL
```

Note: By default curl uses Basic HTTP Authentication. We can specify other authentication method using `-ntlm` | `-digest`.

9. Download Files from FTP server

cURL can also be used to download files from FTP servers. If the given FTP path is a directory, by default it will list the files under the specific directory.

```
$ curl -u ftpuser:ftppass -O
ftp://ftp_server/public_html/xss.php
```

The above command will download the `xss.php` file from the ftp server and save it in the local directory.

```
$ curl -u ftpuser:ftppass -O
ftp://ftp_server/public_html/
```

Here, the given URL refers to a directory. So cURL will list all the files and directories under the given URL

If you are new to FTP/SFTP, refer [ftp sftp tutorial for beginners](#).

10. List/Download using Ranges

cURL supports ranges to be given in the URL. When a range is given, files matching within the range will be downloaded. It will be helpful to download packages from the FTP mirror sites.

```
$ curl ftp://ftp.uk.debian.org/debian/pool/main/[a-z]/
```

The above command will list out all the packages from a-z ranges in the terminal.

11. Upload Files to FTP Server

Curl can also be used to upload files to the FTP server with -T option.

```
$ curl -u ftpuser:ftppass -T myfile.txt  
ftp://ftp.testserver.com
```

The above command will upload the file named myfile.txt to the FTP server. You can also upload multiple files at a same time using the range operations.

```
$ curl -u ftpuser:ftppass -T "{file1,file2}"  
ftp://ftp.testserver.com
```

Optionally we can use "." to get the input from STDIN and transfer to the remote.

```
$ curl -u ftpuser:ftppass -T -  
ftp://ftp.testserver.com/myfile_1.txt
```

The above command will get the input from the user from Standard Input and save the contents in the ftp server under the name 'myfile_1.txt'.

You can provide one '-T' for each URL and the pair specifies what to upload where.

12. More Information using Verbose and Trace Option

You can get to know what is happening using the -v option. -v option enable the verbose mode and it will print the details

```
curl -v  
http://google.co.in
```

The about command will output the following

```

* About to connect() to www.google.co.in port 80 (#0)
*   Trying 74.125.236.56... connected
* Connected to www.google.co.in (74.125.236.56) port 80 (#0)
> GET / HTTP/1.1
> User-Agent: curl/7.21.0 (i486-pc-linux-gnu) libcurl/7.21.0 OpenSSL/0.9.8o
zlib/1.2.3.4 libidn/1.15 libssh2/1.2.6
> Host: www.google.co.in
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Date: Tue, 10 Apr 2012 11:18:39 GMT
< Expires: -1
< Cache-Control: private, max-age=0
< Content-Type: text/html; charset=ISO-8859-1
< Set-Cookie:
PREF=ID=7c497a6b15cc092d:FF=0:TM=1334056719:LM=1334056719:S=UORpBwxFmTRkbXLj;
expires=Thu, 10-Apr-2014 11:18:39 GMT; path=/; domain=.google.co.in
.
.

```

If you need more detailed information then you can use the `--trace` option. The trace option will enable a full trace dump of all incoming/outgoing data to the given file

```

=> Send header, 169 bytes (0xa9)
0000: 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31 0d 0a GET /
HTTP/1.1..
0010: 55 73 65 72 2d 41 67 65 6e 74 3a 20 63 75 72 6c User-Agent:
curl
..
0060: 2e 32 2e 33 2e 34 20 6c 69 62 69 64 6e 2f 31 2e .2.3.4
libidn/1.

0070: 31 35 20 6c 69 62 73 73 68 32 2f 31 2e 32 2e 36 15
libssh2/1.2.6
0080: 0d 0a 48 6f 73 74 3a 20 77 77 77 2e 67 6f 6f 67 ..Host:
www.goog
0090: 6c 65 2e 63 6f 2e 69 6e 0d 0a 41 63 63 65 70 74
le.co.in..Accept
00a0: 3a 20 2a 2f 2a 0d 0a 0d 0a : */*.....
== Info: HTTP 1.0, assume close after body
<= Recv header, 17 bytes (0x11)
0000: 48 54 54 50 2f 31 2e 30 20 32 30 30 20 4f 4b 0d HTTP/1.0 200
OK.
0010: 0a

```

This verbose and trace option will come in handy when curl fails due to some reason and we don't know why.

13. Get Definition of a Word using DICT Protocol

You can use cURL to get the definition for a word with the help of DICT protocol. We need to pass a Dictionary Server URL to it.

```

$ curl
dict://dict.org/d:bash

```

The above command will list the meaning for bash as follows

```
151 "Bash" gcide "The Collaborative International Dictionary of English v.0.48"
```

```
Bash \Bash\, v. t. [imp. & p. p. {Bashed}; p. pr. & vb. n. {Bashing}.] [Perh. of imitative origin; or cf. Dan. baske to strike, bask a blow, Sw. basa to beat, bas a beating.] To strike heavily; to beat; to crush. [Prov. Eng. & Scot.] --Hall Caine. [1913 Webster]
```

```
        Bash her open with a rock.                --Kipling. [Webster 1913 Suppl.]
```

```
.  
151 "Bash" gcide "The Collaborative International Dictionary of English v.0.48"
```

```
Bash \Bash\, n.  
    1. a forceful blow, especially one that does damage to its target. [PJC]
```

Now you can see that it uses “The Collaborative International Dictionary of English”. There are many dictionaries are available. We can list all the dictionaries using

```
$ curl dict://dict.org/show:db
```

```
jargon "The Jargon File (version 4.4.7, 29 Dec 2003)"  
foldoc "The Free On-line Dictionary of Computing (26 July 2010)"  
easton "Easton's 1897 Bible Dictionary"  
hitchcock "Hitchcock's Bible Names Dictionary (late 1800's)"  
bouvier "Bouvier's Law Dictionary, Revised 6th Ed (1856)"
```

Now in-order to find the actual meaning of Bash in computer we can search for bash in “foldoc” dictionary as follows

```
$ curl  
dict://dict.org/d:bash:foldoc
```

The result will be,

```
bash
```

```
    Bourne Again SHell.  {GNU}'s {command interpreter} for {Unix}.  
    Bash is a {Posix}-compatible {shell} with full {Bourne shell} syntax, and some {C shell} commands built in.  The Bourne Again Shell supports {Emacs}-style command-line editing, job control, functions, and on-line help.  Written by Brian Fox of {UCSB}.
```

For more details with regard to DICT please read [RFC2229](#)

14. Use Proxy to Download a File

We can specify cURL to use proxy to do the specific operation using -x option. We need to specify the host and port of the proxy.

```
$ curl -x proxyserver.test.com:3128  
http://google.co.in
```

15. Send Mail using SMTP Protocol

cURL can also be used to send mail using the SMTP protocol. You should specify the from-address, to-address, and the mailserver ip-address as shown below.

```
$ curl --mail-from blah@test.com --mail-rcpt foo@test.com  
smtp://mailserver.com
```

Once the above command is entered, it will wait for the user to provide the data to mail. Once you've composed your message, type . (period) as the last line, which will send the email immediately.

```
Subject: Testing  
This is a test  
mail  
.
```