

NAME

CURLOPT_COOKIEFILE – filename to read cookies from

SYNOPSIS

```
#include <curl/curl.h>
```

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_COOKIEFILE, char *filename);
```

DESCRIPTION

Pass a pointer to a null-terminated string as parameter. It should point to the filename of your file holding cookie data to read. The cookie data can be in either the old Netscape / Mozilla cookie data format or regular HTTP headers (Set-Cookie style) dumped to a file.

It also enables the cookie engine, making libcurl parse and send cookies on subsequent requests with this handle.

By passing the empty string ("") to this option, you enable the cookie engine without reading any initial cookies. If you tell libcurl the filename is "-" (a single minus sign), libcurl instead reads from stdin.

This option only **reads** cookies. To make libcurl write cookies to file, see *CURLOPT_COOKIEJAR(3)*.

If you read cookies from a plain HTTP headers file and it does not specify a domain in the Set-Cookie line, then the cookie is not sent since the cookie domain cannot match the target URL's. To address this, set a domain in Set-Cookie line (doing that includes subdomains) or preferably: use the Netscape format.

The application does not have to keep the string around after setting this option.

If you use this option multiple times, you add more files to read cookies from. Setting this option to NULL disables the cookie engine and clears the list of files to read cookies from.

The cookies are loaded from the specified file(s) when the transfer starts, not when this option is set.

libcurl ignores filenames which do not exist or point to a directory.

SECURITY CONCERNS

This document previously mentioned how specifying a non-existing file can also enable the cookie engine. While true, we strongly advise against using that method as it is too hard to be sure that files that stay that way in the long run.

DEFAULT

NULL

PROTOCOLS

This functionality affects http only

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com/foo.bin");

        /* get cookies from an existing file */
        curl_easy_setopt(curl, CURLOPT_COOKIEFILE, "/tmp/cookies.txt");

        result = curl_easy_perform(curl);
```

```
    curl_easy_cleanup(curl);  
  }  
}
```

Cookie file format

The cookie file format and general cookie concepts in curl are described online here:
<https://curl.se/docs/http-cookies.html>

AVAILABILITY

Added in curl 7.1

RETURN VALUE

curl_easy_setopt(3) returns a CURLcode indicating success or error.

CURLE_OK (0) means everything was OK, non-zero means an error occurred, see *libcurl-errors(3)*.

SEE ALSO

CURLOPT_COOKIE(3), **CURLOPT_COOKIEJAR(3)**, **CURLOPT_COOKIESESSION(3)**