

**NAME**

CURLOPT\_COOKIEJAR – filename to store cookies to

**SYNOPSIS**

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

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

**DESCRIPTION**

Pass a *filename* as a char \*, null-terminated. This makes libcurl write all internally known cookies to the specified file when *curl\_easy\_cleanup(3)* is called. If no cookies are kept in memory at that time, no file is created. Specify "-" as filename to instead have the cookies written to stdout. Using this option also enables cookies for this session, so if you for example follow a redirect it makes matching cookies get sent accordingly.

Note that libcurl does not read any cookies from the cookie jar specified with this option. To read cookies from a file, use *CURLOPT\_COOKIEFILE(3)*.

If the cookie jar file cannot be created or written to (when the *curl\_easy\_cleanup(3)* is called), libcurl does not and cannot report an error for this. Using *CURLOPT\_VERBOSE(3)* or *CURLOPT\_DEBUGFUNCTION(3)* displays a warning, but that is the only visible feedback you get about this possibly lethal situation.

Cookies are imported in the Set-Cookie format without a domain name are not exported by this option.

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

Using this option multiple times makes the last set string override the previous ones. Set it to NULL to disable its use again.

**SECURITY CONCERNS**

libcurl cannot fully protect against attacks where an attacker has write access to the same directory where it is directed to save files. This is particularly sensitive if you save files using elevated privileges.

libcurl creates the file to store cookies using default file permissions, meaning that on \*nix systems you may need to restrict your umask to prevent other users on the same system to access the file.

**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");

        /* export cookies to this file when closing the handle */
        curl_easy_setopt(curl, CURLOPT_COOKIEJAR, "/tmp/cookies.txt");

        result = curl_easy_perform(curl);

        /* close the handle, write the cookies */
        curl_easy_cleanup(curl);
    }
}
```

```
}  
}
```

**AVAILABILITY**

Added in curl 7.9

**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\_COOKIEFILE(3)**, **CURLOPT\_COOKIELIST(3)**