

**NAME**

CURLOPT\_COOKIEJAR – file name to store cookies to

**SYNOPSIS**

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

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

**DESCRIPTION**

Pass a *filename* as char \*, zero terminated. This will make libcurl write all internally known cookies to the specified file when *curl\_easy\_cleanup(3)* is called. If no cookies are known, no file will be 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 location it will make matching cookies get sent accordingly.

Note that libcurl doesn't read any cookies from the cookie jar. If you want to read cookies from a file, use *CURLOPT\_COOKIEFILE(3)*.

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

**DEFAULT**

NULL

**PROTOCOLS**

HTTP

**EXAMPLE**

TODO

**AVAILABILITY**

Along with HTTP

**RETURN VALUE**

Returns CURLE\_OK if HTTP is supported, CURLE\_UNKNOWN\_OPTION if not, or CURLE\_OUT\_OF\_MEMORY if there was insufficient heap space.

**SEE ALSO**

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