

NAME

CURLOPT_SHARE – share handle to use

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_SHARE, CURLSH *share);
```

DESCRIPTION

Pass a *share* handle as a parameter. The share handle must have been created by a previous call to *curl_share_init(3)*. Setting this option, makes this curl handle use the data from the shared handle instead of keeping the data to itself. This enables several curl handles to share data. If the curl handles are used simultaneously in multiple threads, you **must** use the locking methods in the share handle. See *curl_share_setopt(3)* for details.

If you add a share that is set to share cookies, your easy handle uses that cookie cache and get the cookie engine enabled. If you stop sharing an object that was using cookies (or change to another object that does not share cookies), the easy handle gets its cookie engine disabled.

Data that the share object is not set to share is dealt with the usual way, as if no share was used.

Set this option to NULL again to stop using that share object.

DEFAULT

NULL

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    CURL *curl2 = curl_easy_init(); /* a second handle */
    if(curl) {
        CURLcode result;
        CURLSH *shobject = curl_share_init();
        curl_share_setopt(shobject, CURLSHOPT_SHARE, CURL_LOCK_DATA_COOKIE);

        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com/");
        curl_easy_setopt(curl, CURLOPT_COOKIEFILE, "");
        curl_easy_setopt(curl, CURLOPT_SHARE, shobject);
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);

        /* the second handle shares cookies from the first */
        curl_easy_setopt(curl2, CURLOPT_URL, "https://example.com/second");
        curl_easy_setopt(curl2, CURLOPT_COOKIEFILE, "");
        curl_easy_setopt(curl2, CURLOPT_SHARE, shobject);
        result = curl_easy_perform(curl2);
        curl_easy_cleanup(curl2);

        curl_share_cleanup(shobject);
    }
}
```

AVAILABILITY

Added in curl 7.10

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), **CURLSHOPT_SHARE(3)**