

**NAME**

curl\_share\_setopt - Set options for a shared object

**SYNOPSIS**

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

```
CURLSHcode curl_share_setopt(CURLSH *share, CURLSHoption option, parameter);
```

**DESCRIPTION**

Set the *option* to *parameter* for the given *share*.

**OPTIONS****CURLSHOPT\_LOCKFUNC**

The *parameter* must be a pointer to a function matching the following prototype:

```
void lock_function(CURL *handle, curl_lock_data data, curl_lock_access access, void *userptr);
```

*data* defines what data libcurl wants to lock, and you must make sure that only one lock is given at any time for each kind of data.

*access* defines what access type libcurl wants, shared or single.

*userptr* is the pointer you set with *CURLSHOPT\_USERDATA*.

**CURLSHOPT\_UNLOCKFUNC**

The *parameter* must be a pointer to a function matching the following prototype:

```
void unlock_function(CURL *handle, curl_lock_data data, void *userptr);
```

*data* defines what data libcurl wants to unlock, and you must make sure that only one lock is given at any time for each kind of data.

*userptr* is the pointer you set with *CURLSHOPT\_USERDATA*.

**CURLSHOPT\_SHARE**

The *parameter* specifies a type of data that should be shared. This may be set to one of the values described below.

**CURL\_LOCK\_DATA\_COOKIE**

Cookie data will be shared across the easy handles using this shared object.

**CURL\_LOCK\_DATA\_DNS**

Cached DNS hosts will be shared across the easy handles using this shared object. Note that when you use the multi interface, all easy handles added to the same multi handle will share DNS cache by default without this having to be used!

**CURL\_LOCK\_DATA\_SSL\_SESSION**

SSL session IDs will be shared across the easy handles using this shared object. This will reduce the time spent in the SSL handshake when reconnecting to the same server. Note SSL session IDs are reused within the same easy handle by default. Note this symbol was added in 7.10.3 but was not implemented until 7.23.0.

**CURLSHOPT\_UNSHARE**

This option does the opposite of *CURLSHOPT\_SHARE*. It specifies that the specified *parameter* will no longer be shared. Valid values are the same as those for *CURLSHOPT\_SHARE*.

**CURLSHOPT\_USERDATA**

The *parameter* allows you to specify a pointer to data that will be passed to the *lock\_function* and *unlock\_function* each time it is called.

**RETURN VALUE**

CURLSHE\_OK (zero) means that the option was set properly, non-zero means an error occurred as `<curl/curl.h>` defines. See the *libcurl-errors.3* man page for the full list with descriptions.

**SEE ALSO**

`curl_share_cleanup(3)`, `curl_share_init(3)`