

NAME

CURLSHOPT_SHARE – add data to share

SYNOPSIS

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

```
CURLSHcode curl_share_setopt(CURLSH *share, CURLSHOPT_SHARE, int type);
```

DESCRIPTION

The *type* parameter specifies what specific data that should be shared and kept in the share object that was created with *curl_share_init(3)*. The given *type* must be one of the values described below. You can set *CURLSHOPT_SHARE(3)* multiple times with different data arguments to have the share object share multiple types of data. Unset a type again by setting *CURLSHOPT_UNSHARE(3)*.

If any of the data is to be shared in multiple threads then mutex callbacks must be set as well. See *CURLSHOPT_LOCKFUNC(3)* and *CURLSHOPT_UNLOCKFUNC(3)*.

CURL_LOCK_DATA_COOKIE

Cookie data is shared across the easy handles using this shared object. Note that this does not activate an easy handle's cookie handling. You can do that separately by using *CURLOPT_COOKIEFILE(3)* for example.

It is not supported to share cookies between multiple concurrent threads.

CURL_LOCK_DATA_DNS

Cached DNS hosts are 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 share the DNS cache by default without using this option.

CURL_LOCK_DATA_SSL_SESSION

SSL sessions are shared across the easy handles using this shared object. This reduces the time spent in the SSL handshake when reconnecting to the same server.

Note that when you use the multi interface, all easy handles added to the same multi handle share the SSL session cache by default without using this option.

CURL_LOCK_DATA_CONNECT

Put the connection cache in the share object and make all easy handles using this share object share the connection cache.

It is not supported to share connections between multiple concurrent threads.

Connections that are used for HTTP/2 or HTTP/3 multiplexing only get additional transfers added to them if the existing connection is held by the same multi or easy handle. libcurl does not support doing multiplexed streams in different threads using a shared connection.

Note that when you use the multi interface, all easy handles added to the same multi handle share the connection cache by default without using this option.

CURL_LOCK_DATA_PSL

The Public Suffix List stored in the share object is made available to all easy handle bound to the later. Since the Public Suffix List is periodically refreshed, this avoids updates in too many different contexts.

Added in 7.61.0.

Note that when you use the multi interface, all easy handles added to the same multi handle share the PSL cache by default without using this option.

CURL_LOCK_DATA_HSTS

The in-memory HSTS cache.

It is not supported to share the HSTS between multiple concurrent threads.

Added in 7.88.0

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
    CURLSHcode sh;
    CURLSH *share = curl_share_init();
    sh = curl_share_setopt(share, CURLSHOPT_SHARE, CURL_LOCK_DATA_COOKIE);
    if(sh)
        printf("Error: %s\n", curl_share_strerror(sh));
}
```

AVAILABILITY

Added in curl 7.10.3

RETURN VALUE

CURLSHE_OK (zero) means that the option was set properly, non-zero means an error occurred. See *libcurl-errors(3)* for the full list with descriptions.

SEE ALSO

CURLSHOPT_UNSHARE(3), curl_share_cleanup(3), curl_share_init(3), curl_share_setopt(3)