

NAME

CURLSHOPT_UNLOCKFUNC – mutex unlock callback

SYNOPSIS

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

```
void unlockcb(CURL *handle, curl_lock_data data, void *clientp);
```

```
CURLSHcode curl_share_setopt(CURLSH *share, CURLSHOPT_UNLOCKFUNC, unlockcb);
```

DESCRIPTION

Set a mutex unlock callback for the share object. There is a corresponding *CURLSHOPT_LOCKFUNC(3)* callback called when the mutex is first locked.

The *unlockcb* argument must be a pointer to a function matching the prototype shown above. The arguments to the callback are:

handle is the currently active easy handle in use when the share object is released.

The *data* argument tells what kind of data libcurl wants to unlock. Make sure that the callback uses a different lock for each kind of data.

clientp is the private pointer you set with *CURLSHOPT_USERDATA(3)*. This pointer is not used by libcurl itself.

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
extern void mutex_unlock(CURL *, curl_lock_data, void *);

int main(void)
{
    CURLSHcode sh;
    CURLSH *share = curl_share_init();
    sh = curl_share_setopt(share, CURLSHOPT_UNLOCKFUNC, mutex_unlock);
    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_LOCKFUNC(3), *curl_share_cleanup(3)*, *curl_share_init(3)*, *curl_share_setopt(3)*