

**NAME**

CURLOPT\_SOCKOPTFUNCTION – set callback for setting socket options

**SYNOPSIS**

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

```
typedef enum {
    CURLSOCKTYPE_IPCXN, /* socket created for a specific IP connection */
    CURLSOCKTYPE_ACCEPT, /* socket created by accept() call */
    CURLSOCKTYPE_LAST /* never use */
} curlsocktype;
```

```
#define CURL_SOCKOPT_OK 0
#define CURL_SOCKOPT_ERROR 1 /* causes libcurl to abort and return
    CURLE_ABORTED_BY_CALLBACK */
#define CURL_SOCKOPT_ALREADY_CONNECTED 2
```

```
int sockopt_callback(void *clientp,
                    curl_socket_t curlfd,
                    curlsocktype purpose);
```

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_SOCKOPTFUNCTION, sockopt_callback);
```

**DESCRIPTION**

Pass a pointer to your callback function, which should match the prototype shown above.

When set, this callback function gets called by libcurl when the socket has been created, but before the connect call to allow applications to change specific socket options. The callback's *purpose* argument identifies the exact purpose for this particular socket:

*CURLSOCKTYPE\_IPCXN* for actively created connections or since 7.28.0 *CURLSOCKTYPE\_ACCEPT* for FTP when the connection was setup with PORT/EPSV (in earlier versions these sockets weren't passed to this callback).

Future versions of libcurl may support more purposes. libcurl passes the newly created socket descriptor to the callback in the *curlfd* parameter so additional setsockopt() calls can be done at the user's discretion.

The *clientp* pointer contains whatever user-defined value set using the *CURLOPT\_SOCKOPTDATA(3)* function.

Return *CURL\_SOCKOPT\_OK* from the callback on success. Return *CURL\_SOCKOPT\_ERROR* from the callback function to signal an unrecoverable error to the library and it will close the socket and return *CURLE\_COULDNT\_CONNECT*. Alternatively, the callback function can return *CURL\_SOCKOPT\_ALREADY\_CONNECTED*, to tell libcurl that the socket is already connected and then libcurl will not attempt to connect it. This allows an application to pass in an already connected socket with *CURLOPT\_OPENSOCKETFUNCTION(3)* and then have this function make libcurl not attempt to connect (again).

**DEFAULT**

By default, this callback is NULL and unused.

**PROTOCOLS**

All

**EXAMPLE**

TODO

CURLOPT\_SOCKOPTFUNCTION(3)      curl\_easy\_setopt options      CURLOPT\_SOCKOPTFUNCTION(3)

**AVAILABILITY**

Added in 7.16.0. The *CURL\_SOCKOPT\_ALREADY\_CONNECTED* return code was added in 7.21.5.

**RETURN VALUE**

Returns CURLE\_OK if the option is supported, and CURLE\_UNKNOWN\_OPTION if not.

**SEE ALSO**

**CURLOPT\_SOCKOPTDATA(3), CURLOPT\_OPENSOCKETFUNCTION(3),**