

NAME

CURLOPT_ABSTRACT_UNIX_SOCKET – abstract Unix domain socket

SYNOPSIS

#include <curl/curl.h>

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_ABSTRACT_UNIX_SOCKET,
                           char *path);
```

DESCRIPTION

Enables the use of an abstract Unix domain socket instead of establishing a TCP connection to a host. The parameter should be a char * to a null-terminated string holding the path of the socket. The path is set to *path* prefixed by a NULL byte. This is the convention for abstract sockets, however it should be stressed that the path passed to this function should not contain a leading NULL byte.

On non-supporting platforms, the abstract address is interpreted as an empty string and fails gracefully, generating a runtime error.

This option shares the same semantics as *CURLOPT_UNIX_SOCKET_PATH(3)* in which documentation more details can be found. Internally, these two options share the same storage and therefore only one of them can be set per handle.

DEFAULT

NULL

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_ABSTRACT_UNIX_SOCKET, "/tmp/foo.sock");
        curl_easy_setopt(curl, CURLOPT_URL, "http://localhost");

        /* Perform the request */
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

AVAILABILITY

Added in curl 7.53.0

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_UNIX_SOCKET_PATH(3), **unix(7)**