NAME

CURLOPT_UNIX_SOCKET_PATH − Unix domain socket

SYNOPSIS

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

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

DESCRIPTION

Enables the use of Unix domain sockets as connection endpoint and sets the path to *path*. If *path* is NULL, then Unix domain sockets are disabled.

When enabled, curl connects to the Unix domain socket instead of establishing a TCP connection to the host. Since no network connection is created, curl does not resolve the DNS hostname in the URL.

The maximum path length on Cygwin, Linux and Solaris is 107. On other platforms it might be even less.

Proxy and TCP options such as *CURLOPT_TCP_NODELAY(3)* are not supported. Proxy options such as *CURLOPT_PROXY(3)* have no effect either as these are TCP−oriented, and asking a proxy server to connect to a certain Unix domain socket is not possible.

The application does not have to keep the string around after setting this option.

Using this option multiple times makes the last set string override the previous ones. Set it to NULL to disable its use again.

DEFAULT

NULL − no Unix domain sockets are used.

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_UNIX_SOCKET_PATH, "/tmp/httpd.sock");
  curl_easy_setopt(curl, CURLOPT_URL, "http://localhost/");

  result = curl_easy_perform(curl);
  curl_easy_cleanup(curl);
 }
}
```

If you are on Linux and somehow have a need for paths larger than 107 bytes, you can use the *proc* file system to bypass the limitation:

```
 int dirfd = open(long_directory_path_to_socket, O_DIRECTORY | O_RDONLY);
 char path[108];
 snprintf(path, sizeof(path), "/proc/self/fd/%d/httpd.sock", dirfd);
 curl_easy_setopt(curl_handle, CURLOPT_UNIX_SOCKET_PATH, path);
 /* Be sure to keep dirfd valid until you discard the handle */
```

AVAILABILITY

Added in curl 7.40.0

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_ABSTRACT_UNIX_SOCKET(3), **CURLOPT_OPENSOCKETFUNCTION**(3), **unix**(7)