

NAME

CURLOPT_TCP_NODELAY – the TCP_NODELAY option

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_TCP_NODELAY, long nodelay);
```

DESCRIPTION

Pass a long specifying whether the *TCP_NODELAY* option is to be set or cleared (1L = set, 0 = clear). The option is set by default. This has no effect after the connection has been established.

Setting this option to 1L disables the Nagle algorithm on connections created using this handle. The purpose of this algorithm is to minimize the number of small packets on the network (where "small packets" means TCP segments less than the Maximum Segment Size for the network).

Maximizing the amount of data sent per TCP segment is good because it amortizes the overhead of the send. In some cases small segments may need to be sent without delay. This is less efficient than sending larger amounts of data at a time, and can contribute to congestion on the network if overdone.

DEFAULT

1

PROTOCOLS

This functionality affects tcp only

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");
        /* leave Nagle enabled */
        curl_easy_setopt(curl, CURLOPT_TCP_NODELAY, 0L);
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

HISTORY

The default was changed to 1 from 0 in 7.50.2.

AVAILABILITY

Added in curl 7.11.2

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_BUFFERSIZE(3),
LOPT_TCP_KEEPALIVE(3)

CURLOPT_SOCKOPTFUNCTION(3),

CUR-