

NAME

CURLOPT_PIPEWAIT – wait for multiplexing

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_PIPEWAIT, long wait);
```

DESCRIPTION

Set *wait* to 1L to tell libcurl to prefer to wait for a connection to confirm or deny that it can do multiplexing before continuing.

When about to perform a new transfer that allows multiplexing, libcurl checks for existing connections to use. If no such connection exists it immediately continues and creates a fresh new connection to use.

By setting this option to 1 – and having *CURLOPT_PIPELINING(3)* enabled for the multi handle this transfer is associated with – libcurl instead waits for the connection to reveal if it is possible to multiplex on before it continues. This enables libcurl to much better keep the number of connections to a minimum when using multiplexing protocols.

With this option set, libcurl prefers to wait and reuse an existing connection for multiplexing rather than the opposite: prefer to open a new connection rather than waiting.

The waiting time is as long as it takes for the connection to get up and for libcurl to get the necessary response back that informs it about its protocol and support level.

DEFAULT

0 (off)

PROTOCOLS

This functionality affects http only

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com/");
        curl_easy_setopt(curl, CURLOPT_PIPEWAIT, 1L);

        /* now add this easy handle to the multi handle */
    }
}
```

AVAILABILITY

Added in curl 7.43.0

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_MAX_HOST_CONNECTIONS(3), CURLOPT_PIPELINING(3), CURLOPT_FORBID_REUSE(3), CURLOPT_FRESH_CONNECT(3)