

NAME

CURLOPT_PIPELINING – enable HTTP pipelining and multiplexing

SYNOPSIS

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

```
CURLMcode curl_multi_setopt(CURLM *handle, CURLOPT_PIPELINING, long bitmask);
```

DESCRIPTION

Pass in the **bitmask** parameter to instruct libcurl to enable HTTP pipelining and/or HTTP/2 multiplexing for this multi handle.

When enabled, libcurl will attempt to use those protocol features when doing parallel requests to the same hosts.

For pipelining, this means that if you add a second request that can use an already existing connection, the second request will be "piped" on the same connection rather than being executed in parallel.

For multiplexing, this means that follow-up requests can re-use an existing connection and send the new request multiplexed over that at the same time as other transfers are already using that single connection.

There are several other related options that are interesting to tweak and adjust to alter how libcurl spreads out requests on different connections or not etc.

Before 7.43.0, this option was set to 1 and 0 to enable and disable HTTP/1.1 pipelining.

Starting in 7.43.0, **bitmask**'s second bit also has a meaning, and you can ask for pipelining and multiplexing independently of each other by toggling the correct bits.

CURLPIPE_NOHING

Default, which means doing no attempts at pipelining or multiplexing.

CURLPIPE_HTTP1

If this bit is set, libcurl will try to pipeline HTTP/1.1 requests on connections that are already established and in use to hosts.

CURLPIPE_MULTIPLEX

If this bit is set, libcurl will try to multiplex the new transfer over an existing connection if possible. This requires HTTP/2.

DEFAULT

0 (both pipeline and multiplexing are off)

PROTOCOLS

HTTP(S)

EXAMPLE

TODO

AVAILABILITY

Added in 7.16.0. Multiplex support bit added in 7.43.0.

RETURN VALUE

Returns CURLM_OK if the option is supported, and CURLM_UNKNOWN_OPTION if not.

SEE ALSO

CURLOPT_MAX_PIPELINE_LENGTH(3), **CURLOPT_PIPELINING_SITE_BL(3)**, **CURLMOPT_CONTENT_LENGTH_PENALTY_SIZE(3)**, **CURLMOPT_CHUNK_LENGTH_PENALTY_SIZE(3)**, **CURLOPT_MAX_HOST_CONNECTIONS(3)**, **CURLMOPT_MAXCONNECTS(3)**, **CURLMOPT_MAX_HOST_CONNECTIONS(3)**,