

NAME

CURLMOPT_NETWORK_CHANGED – signal network changed

SYNOPSIS

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

```
CURLMcode curl_multi_setopt(CURLM *handle, CURLMOPT_NETWORK_CHANGED,
                             long value);
```

DESCRIPTION

Pass a long with a bitmask to tell libcurl how the multi handle should react. The following values in the mask are defined. All bits not mentioned are reserved for future extensions.

This option can be set at any time and repeatedly. Each call only affects the *currently* cached connections and DNS information. Any connection created or DNS information added afterwards is cached the usual way again. Phrasing it another way: the option is not persisted but setting it serves as a "trigger" to clear the caches.

The call affects only the connection and DNS cache of the multi handle itself and not the ones owned by SHARE handles.

CURLMNWC_CLEAR_ALL

Clear everything. (Added in 8.20.0)

CURLMNWC_CLEAR_CONNS

No longer reuse any existing connection in the multi handle's connection cache. This closes all connections that are not in use. Ongoing transfers continue on the connections they operate on.

CURLMNWC_CLEAR_DNS

Clear the multi handle's DNS cache. Ongoing transfers keep using their already resolved addresses, but future name resolutions are performed again.

DEFAULT

0, which has no effect.

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
    CURLM *m = curl_multi_init();
    /* do transfers on the multi handle */
    /* do not reuse existing connections */
    curl_multi_setopt(m, CURLMOPT_NETWORK_CHANGED, CURLMNWC_CLEAR_CONNS);
}
```

AVAILABILITY

Added in curl 8.16.0

RETURN VALUE*curl_multi_setopt(3)* returns a CURLMcode indicating success or error.CURLM_OK (0) means everything was OK, non-zero means an error occurred, see *libcurl-errors(3)*.**SEE ALSO****CURLOPT_FORBID_REUSE(3)**, **CURLOPT_FRESH_CONNECT(3)**