

NAME

CURLMOPT_RESOLVE_THREADS_MAX – max threads for threaded DNS resolver

SYNOPSIS

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

```
CURLMcode curl_multi_setopt(CURLM *handle, CURLMOPT_RESOLVE_THREADS_MAX,
                             long amount);
```

DESCRIPTION

Pass a long for the **amount**. The set number is used as the maximum number of threads to be used for the threaded DNS resolver. It has to be a positive number in the range of 32 bits.

When libcurl is built with a threaded resolver, which is the default on many systems, it uses a thread pool to lookup addresses and other properties of hostnames so other transfers are not blocked by this.

Threads are started on demand to perform the resolving and shut down again after a period of inactivity. When the maximum number of threads is reached, outstanding resolves are held in a queue and served when a thread becomes available.

The default maximum is expected to work fine for many situations. Application may override it using this option for the multi handle.

Changing this value while there are resolves in progress is possible. Increasing the value takes effect right away. Lowering the value does not close down any resolves, but ends threads above the new maximum once the resolving is done.

DEFAULT

20.

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
    CURLM *m = curl_multi_init();
    /* never use more than 5 threads for resolving */
    curl_multi_setopt(m, CURLMOPT_RESOLVE_THREADS_MAX, 5L);
}
```

AVAILABILITY

Added in curl 8.20.0

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_IPRESOLVE(3), **CURLOPT_RESOLVE(3)**