

**NAME**

CURLOPT\_FOLLOWLOCATION – follow HTTP 3xx redirects

**SYNOPSIS**

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_FOLLOWLOCATION, long enable);
```

**DESCRIPTION**

A long parameter set to 1 tells the library to follow any Location: header that the server sends as part of a HTTP header in a 3xx response. The Location: header can specify a relative or an absolute URL to follow.

libcurl will issue another request for the new URL and follow new Location: headers all the way until no more such headers are returned. *CURLOPT\_MAXREDIRS(3)* can be used to limit the number of redirects libcurl will follow.

libcurl limits what protocols it automatically follows to. The accepted protocols are set with *CURLOPT\_REDIR\_PROTOCOLS(3)*. By default libcurl will allow all protocols on redirect except those disabled for security reasons: Since 7.19.4 FILE and SCP are disabled, and since 7.40.0 SMB and SMBS are also disabled.

When following a Location:, the 3xx response code that redirected it also dictates which request method it will use in the subsequent request: For 301, 302 and 303 responses libcurl will switch method to GET unless *CURLOPT\_POSTREDIR(3)* instructs libcurl otherwise. All other 3xx codes will make libcurl send the same method again.

For users who think the existing location following is too naive, too simple or just lacks features, it is very easy to instead implement your own redirect follow logic with the use of *curl\_easy\_getinfo(3)*'s *CURLINFO\_REDIRECT\_URL(3)* option instead of using *CURLOPT\_FOLLOWLOCATION(3)*.

**DEFAULT**

0, disabled

**PROTOCOLS**

HTTP(S)

**EXAMPLE**

```
CURL *curl = curl_easy_init();
if(curl) {
    curl_easy_setopt(curl, CURLOPT_URL, "http://example.com");

    /* example.com is redirected, so we tell libcurl to follow redirection */
    curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1L);

    curl_easy_perform(curl);
}
```

**AVAILABILITY**

Along with HTTP

**RETURN VALUE**

Returns CURLE\_OK if HTTP is supported, and CURLE\_UNKNOWN\_OPTION if not.

**SEE ALSO**

*CURLOPT\_REDIR\_PROTOCOLS(3)*, *CURLOPT\_PROTOCOLS(3)*, *CURLOPT\_POSTREDIR(3)*, *CURLINFO\_REDIRECT\_URL(3)*, *CURLINFO\_REDIRECT\_COUNT(3)*,