

**NAME**

CURLOPT\_FOLLOWLOCATION – follow HTTP 3xx redirects

**SYNOPSIS**

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

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

**DESCRIPTION**

A 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.

This means that libcurl will re-send the same request on the new location 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 can limit to what protocols it will automatically follow. The accepted protocols are set with *CURLOPT\_REDIR\_PROTOCOLS(3)* and it excludes the FILE protocol by default.

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* 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)*,