NAME

CURLOPT_ERRORBUFFER − error buffer for error messages

SYNOPSIS

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

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_ERRORBUFFER, char *buf);
```

DESCRIPTION

Pass a char pointer to a buffer that libcurl may use to store human readable error messages on failures or problems. This may be more helpful than the single return code from *curl_easy_perform(3)* and related functions. The buffer must be at least **CURL_ERROR_SIZE** bytes big.

You must keep the associated buffer available until libcurl no longer needs it. Failing to do so might cause odd behavior or even crashes. libcurl might need it until you call *curl_easy_cleanup(3)* or you set the same option again to use a different pointer.

libcurl initializes the contents of the error buffer to an empty string before performing a transfer.

Do not attempt to set the contents of the buffer yourself, including in any callbacks you write that may be called by libcurl. The library may overwrite the buffer after your callback returns.

Consider *CURLOPT_VERBOSE(3)* and *CURLOPT_DEBUGFUNCTION(3)* to better debug and trace why errors happen.

Using this option multiple times makes the last set pointer override the previous ones. Set it to NULL to disable its use again.

DEFAULT

NULL

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
#include <string.h> /* for strlen() */
int main(void)
{
  CURL *curl = curl_easy_init();
  if(curl) {
    CURLcode result;
    char errbuf[CURL_ERROR_SIZE];

    curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");

    /* provide a buffer to store errors in */
    curl_easy_setopt(curl, CURLOPT_ERRORBUFFER, errbuf);

    /* set the error buffer as empty before performing a request */
    errbuf[0] = 0;

    /* perform the request */
    result = curl_easy_perform(curl);

    /* if the request did not complete correctly, show the error
    information. if no detailed error information was written to errbuf
    show the more generic information from curl_easy_strerror instead.
```

```
   */
   if(result != CURLE_OK) {
     size_t len = strlen(errbuf);
     fprintf(stderr, "\nlibcurl: (%d) ", result);
     if(len)
       fprintf(stderr, "%s%s", errbuf,
               ((errbuf[len - 1] != '\n') ? "\n" : ""));
     else
       fprintf(stderr, "%s\n", curl_easy_strerror(result));
   }
  }
}
```

HISTORY

Before curl 7.60.0, if an error code was returned but there was no error detail the buffer was untouched: not initialized.

AVAILABILITY

Added in curl 7.1

RETURN VALUE

curl_easy_setopt(3) returns a CURLcode indicating success or error.

CURLE_OK (0) means everything was OK, non−zero means an error occurred, see *libcurl−errors(3)*.

SEE ALSO

CURLOPT_DEBUGFUNCTION(3), **CURLOPT_VERBOSE**(3), **curl_easy_strerror**(3), **curl_multi_strerror**(3), **curl_share_strerror**(3), **curl_url_strerror**(3)