

**NAME**

CURLOPT\_ERRORBUFFER – set error buffer for error messages

**SYNOPSIS**

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

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

**DESCRIPTION**

Pass a char \* to a buffer that the libcurl may store human readable error messages in on failures or problems. This may be more helpful than just the 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 will cause very odd behavior or even crashes. libcurl will need it until you call *curl\_easy\_cleanup(3)* or you set the same option again to use a different pointer.

Consider *CURLOPT\_VERBOSE(3)* and *CURLOPT\_DEBUGFUNCTION(3)* to better debug and trace why errors happen.

If the library does not return an error, the buffer may not have been touched. Do not rely on the contents in those cases.

**DEFAULT**

NULL

**PROTOCOLS**

All

**EXAMPLE**

```
curl = curl_easy_init();
if(curl) {
    CURLcode res;
    char errbuf[CURL_ERROR_SIZE];

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

```
        fprintf(stderr, "%s\n", curl_easy_strerror(res));  
    }  
}
```

**AVAILABILITY**

Always

**RETURN VALUE**

Returns CURLE\_OK

**SEE ALSO**

**CURLOPT\_DEBUGFUNCTION(3), CURLOPT\_VERBOSE(3), curl\_easy\_strerror(3),  
curl\_multi\_strerror(3), curl\_share\_strerror(3)**