

NAME

CURLOPT_DEBUGFUNCTION – debug callback

SYNOPSIS

#include <curl/curl.h>

```
typedef enum {
    CURLINFO_TEXT = 0,
    CURLINFO_HEADER_IN, /* 1 */
    CURLINFO_HEADER_OUT, /* 2 */
    CURLINFO_DATA_IN, /* 3 */
    CURLINFO_DATA_OUT, /* 4 */
    CURLINFO_SSL_DATA_IN, /* 5 */
    CURLINFO_SSL_DATA_OUT, /* 6 */
    CURLINFO_END
} curl_infotype;
```

```
int debug_callback(CURL *handle,
                  curl_infotype type,
                  char *data,
                  size_t size,
                  void *userptr);
```

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_DEBUGFUNCTION,
                          debug_callback);
```

DESCRIPTION

Pass a pointer to your callback function, which should match the prototype shown above.

CURLOPT_DEBUGFUNCTION(3) replaces the standard debug function used when *CURLOPT_VERBOSE(3)* is in effect. This callback receives debug information, as specified in the *type* argument. This function must return 0. The *data* pointed to by the *char ** passed to this function WILL NOT be zero terminated, but will be exactly of the *size* as told by the *size* argument.

The *userptr* argument is the pointer set with *CURLOPT_DEBUGDATA(3)*.

Available curl_infotype values:

CURLINFO_TEXT

The data is informational text.

CURLINFO_HEADER_IN

The data is header (or header-like) data received from the peer.

CURLINFO_HEADER_OUT

The data is header (or header-like) data sent to the peer.

CURLINFO_DATA_IN

The data is protocol data received from the peer.

CURLINFO_DATA_OUT

The data is protocol data sent to the peer.

CURLINFO_SSL_DATA_OUT

The data is SSL/TLS (binary) data sent to the peer.

CURLINFO_SSL_DATA_IN

The data is SSL/TLS (binary) data received from the peer.

CURLOPT_DEBUGFUNCTION(3)

curl_easy_setopt options

CURLOPT_DEBUGFUNCTION(3)

DEFAULT

NULL

PROTOCOLS

All

EXAMPLE

<http://curl.haxx.se/libcurl/c/debug.html>

AVAILABILITY

Always

RETURN VALUE

Returns CURLE_OK

SEE ALSO

CURLOPT_VERBOSE(3), CURLOPT_DEBUGDATA(3),