NAME

curl_easy_send − sends raw data over an "easy" connection

SYNOPSIS

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

CURLcode curl_easy_send(CURL *curl, const void *buffer,
                        size_t buflen, size_t *n);
```

DESCRIPTION

This function sends arbitrary data over the established connection. You may use it together with *curl_easy_recv(3)* to implement custom protocols using libcurl. This functionality can be particularly useful if you use proxies and/or SSL encryption: libcurl takes care of proxy negotiation and connection setup.

buffer is a pointer to the data of length **buflen** that you want sent. The variable **n** points to receives the number of sent bytes.

To establish the connection, set *CURLOPT_CONNECT_ONLY(3)* option before calling *curl_easy_perform(3)* or *curl_multi_perform(3)*. Note that *curl_easy_send(3)* does not work on connections that were created without this option.

The call returns **CURLE_AGAIN** if it is not possible to send data right now - the socket is used in non−blocking mode internally. When **CURLE_AGAIN** is returned, use your operating system facilities like *select(2)* to wait until the socket is writable. The socket may be obtained using *curl_easy_getinfo(3)* with *CURLINFO_ACTIVESOCKET(3)*.

Furthermore if you wait on the socket and it tells you it is writable, *curl_easy_send(3)* may return **CURLE_AGAIN** if the only data that was sent was for internal SSL processing, and no other data could be sent.

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
  CURL *curl = curl_easy_init();
  if(curl) {
    CURLcode result;
    curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");
    /* Do not do the transfer - only connect to host */
    curl_easy_setopt(curl, CURLOPT_CONNECT_ONLY, 1L);
    result = curl_easy_perform(curl);

    if(result == CURLE_OK) {
      curl_socket_t sockfd;
      size_t sent;
      /* Extract the socket from the curl handle - we need it for waiting. */
      result = curl_easy_getinfo(curl, CURLINFO_ACTIVESOCKET, &sockfd);

      /* send data */
      result = curl_easy_send(curl, "hello", 5, &sent);
    }
  }
}
```

AVAILABILITY

Added in curl 7.18.2

RETURN VALUE

On success, returns **CURLE_OK** and stores the number of bytes actually sent into ***n**. Note that this may be less than the amount you wanted to send.

On failure, returns the appropriate error code.

This function may return **CURLE_AGAIN**. In this case, use your operating system facilities to wait until the socket is writable, and retry.

If there is no socket available to use from the previous transfer, this function returns **CURLE_UNSUPPORTED_PROTOCOL**.

SEE ALSO

curl_easy_getinfo(3), **curl_easy_perform**(3), **curl_easy_recv**(3), **curl_easy_setopt**(3)