

## NAME

`curl_easy_send` - sends raw data over an "easy" connection

## SYNOPSIS

```
#include <curl/easy.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 will take 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 will receive the number of sent bytes.

To establish the connection, set **CURLOPT\_CONNECT\_ONLY** option before calling `curl_easy_perform(3)`. Note that `curl_easy_send(3)` will not work on connections that were created without this option.

You must ensure that the socket is writable before calling `curl_easy_send(3)`, otherwise the call will return **CURLE\_AGAIN** - the socket is used in non-blocking mode internally. Use `curl_easy_getinfo(3)` with **CURLINFO\_LASTSOCKET** to obtain the socket; use your operating system facilities like `select(2)` to check if it can be written to.

## AVAILABILITY

Added in 7.18.2.

## RETURN VALUE

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

On failure, returns the appropriate error code.

## EXAMPLE

See `sendrecv.c` in `docs/examples` directory for usage example.

## SEE ALSO

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