

NAME

curl_multi_socket_all – reads/writes available data for all easy handles

SYNOPSIS

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

```
CURLMcode curl_multi_socket_all(CURLM *multi_handle,  
                                int *running_handles);
```

DESCRIPTION

This function is deprecated for performance reasons but there are no plans to remove it from the API. Use *curl_multi_socket_action(3)* instead.

At return, the integer **running_handles** points to contains the number of still running easy handles within the multi handle. When this number reaches zero, all transfers are complete/done.

Force libcurl to (re-)check all its internal sockets and transfers instead of a single one by calling *curl_multi_socket_all(3)*. Note that there should not be any reason to use this function.

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
    int running;
    int rc;
    CURLM *multi = curl_multi_init();

    rc = curl_multi_socket_all(multi, &running);
}
```

AVAILABILITY

Added in curl 7.15.4

RETURN VALUE

This function returns a CURLMcode indicating success or error.

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

The return code is for the whole multi stack. Problems still might have occurred on individual transfers even when one of these functions return OK.

SEE ALSO

curl_multi_cleanup(3), **curl_multi_fdset(3)**, **curl_multi_info_read(3)**, **curl_multi_init(3)**, **thehiper-fifo.cexample**