NAME

curl_multi_fdset - extracts file descriptor information from a multi handle

SYNOPSIS

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

CURLMcode curl_multi_fdset(CURLM *multi_handle,
                         fd_set *read_fd_set,
                         fd_set *write_fd_set,
                         fd_set *exc_fd_set,
                         int *max_fd);
```

DESCRIPTION

This function extracts file descriptor information from a given multi_handle. libcurl returns its fd_set sets. The application can use these to select() on, but be sure to FD_ZERO them before calling this function as *curl_multi_fdset(3)* only adds its own descriptors, it doesn't zero or otherwise remove any others. The *curl_multi_perform(3)* function should be called as soon as one of them is ready to be read from or written to.

To be sure to have up-to-date results, you should call *curl_multi_perform* until it does not return CURLM_CALL_MULTI_PERFORM prior to calling *curl_multi_fdset*. This will make sure that libcurl has updated the handles' socket states.

If no file descriptors are set by libcurl, *max_fd* will contain -1 when this function returns. Otherwise it will contain the higher descriptor number libcurl set.

When doing select(), you should use **curl_multi_timeout** to figure out how long to wait for action. Call *curl_multi_perform* even if no activity has been seen on the fd_sets after the timeout expires as otherwise internal retries and timeouts may not work as you'd think and want.

RETURN VALUE

CURLMcode type, general libcurl multi interface error code. See *libcurl-errors(3)*

SEE ALSO

curl_multi_cleanup(3), **curl_multi_init**(3), **curl_multi_timeout**(3), **curl_multi_perform**(3)