

**NAME**

curl\_multi\_wait - polls on all easy handles in a multi handle

**SYNOPSIS**

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

```
CURLMcode curl_multi_wait(CURLM *multi_handle,
                          struct curl_waitfd extra_fds[],
                          unsigned int extra_nfds,
                          int timeout_ms,
                          int *numfds);
```

**DESCRIPTION**

*curl\_multi\_wait(3)* polls all file descriptors used by the curl easy handles contained in the given multi handle set. It will block until activity is detected on at least one of the handles or *timeout\_ms* has passed. Alternatively, if the multi handle has a pending internal timeout that has a shorter expiry time than *timeout\_ms*, that shorter time will be used instead to make sure timeout accuracy is reasonably kept.

The calling application may pass additional curl\_waitfd structures which are similar to *poll(2)*'s pollfd structure to be waited on in the same call.

On completion, if *numfds* is non-NULL, it will be populated with the total number of file descriptors on which interesting events occurred. This number can include both libcurl internal descriptors as well as descriptors provided in *extra\_fds*.

If no extra file descriptors are provided and libcurl has no file descriptor to offer to wait for, this function will return immediately.

This function is encouraged to be used instead of *select(3)* when using the multi interface to allow applications to easier circumvent the common problem with 1024 maximum file descriptors.

**curl\_waitfd**

```
struct curl_waitfd {
    curl_socket_t fd;
    short events;
    short revents;
};
```

**CURL\_WAIT\_POLLIN**

Bit flag to curl\_waitfd.events indicating the socket should poll on read events such as new data received.

**CURL\_WAIT\_POLLPRI**

Bit flag to curl\_waitfd.events indicating the socket should poll on high priority read events such as out of band data.

**CURL\_WAIT\_POLLOUT**

Bit flag to curl\_waitfd.events indicating the socket should poll on write events such as the socket being clear to write without blocking.

**RETURN VALUE**

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

**AVAILABILITY**

This function was added in libcurl 7.28.0.

**SEE ALSO**

**curl\_multi\_fdset(3)**, **curl\_multi\_perform(3)**