

NAME

CURLOPT_IOCTLFUNCTION – callback for I/O operations

SYNOPSIS

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

```
typedef enum {
    CURLIOE_OK,          /* I/O operation successful */
    CURLIOE_UNKNOWNCMD, /* command was unknown to callback */
    CURLIOE_FAILRESTART, /* failed to restart the read */
    CURLIOE_LAST         /* never use */
} curlioerr;
```

```
typedef enum {
    CURLIOCMD_NOP,      /* no operation */
    CURLIOCMD_RESTARTREAD, /* restart the read stream from start */
    CURLIOCMD_LAST      /* never use */
} curliocmd;
```

```
curlioerr ioctl_callback(CURL *handle, int cmd, void *clientp);
```

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_IOCTLFUNCTION, ioctl_callback);
```

DESCRIPTION

Pass a pointer to your callback function, which should match the prototype shown above.

This callback function gets called by libcurl when something special I/O-related needs to be done that the library cannot do by itself. For now, rewinding the read data stream is the only action it can request. The rewinding of the read data stream may be necessary when doing an HTTP PUT or POST with a multi-pass authentication method.

The callback **must** return `CURLIOE_UNKNOWNCMD` if the input `cmd` is not `CURLIOCMD_RESTARTREAD`.

The `clientp` argument to the callback is set with the `CURLOPT_IOCTLDATA(3)` option.

This option is deprecated. Do not use it. Use `CURLOPT_SEEKFUNCTION(3)` instead to provide seeking. If `CURLOPT_SEEKFUNCTION(3)` is set, this parameter is ignored when seeking.

DEFAULT

NULL

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
#include <unistd.h> /* for lseek */

struct data {
    int fd; /* our file descriptor */
};

static curlioerr ioctl_callback(CURL *handle, int cmd, void *clientp)
{
    struct data *io = (struct data *)clientp;
    if(cmd == CURLIOCMD_RESTARTREAD) {
        lseek(io->fd, 0, SEEK_SET);
    }
}
```

```
    return CURLIOE_OK;
}
return CURLIOE_UNKNOWNCMD;
}
int main(void)
{
    struct data ioctl_data;
    CURL *curl = curl_easy_init();
    if(curl) {
        curl_easy_setopt(curl, CURLOPT_IOCTLFUNCTION, ioctl_callback);
        curl_easy_setopt(curl, CURLOPT_IOCTLDATA, &ioctl_data);
    }
}
```

DEPRECATED

Deprecated since 7.18.0.

AVAILABILITY

Added in curl 7.12.3

RETURN VALUE

curl_easy_setopt(3) returns a CURLcode indicating success or error.

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

SEE ALSO

CURLOPT_IOCTLDATA(3), CURLOPT_SEEKFUNCTION(3)