

**NAME**

CURLOPT\_READDATA – pointer passed to the read callback

**SYNOPSIS**

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_READDATA, void *pointer);
```

**DESCRIPTION**

Data *pointer* to pass to the file read function. If you use the *CURLOPT\_READFUNCTION(3)* option, this is the pointer you get as input in the fourth argument to the callback.

If you do not specify a read callback but instead rely on the default internal read function, this data must be a valid readable FILE \* (cast to 'void \*').

If you are using libcurl as a DLL on Windows, you must use the *CURLOPT\_READFUNCTION(3)* callback if you set this option, otherwise you might experience crashes.

**DEFAULT**

stdin

**PROTOCOLS**

This functionality affects all supported protocols

**EXAMPLE**

```
struct MyData {
    void *custom;
};

int main(void)
{
    CURL *curl = curl_easy_init();
    struct MyData this;
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");

        /* pass pointer that gets passed in to the
           CURLOPT_READFUNCTION callback */
        curl_easy_setopt(curl, CURLOPT_READDATA, &this);

        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

**HISTORY**

This option was once known by the older name CURLOPT\_INFILE, the name *CURLOPT\_READDATA(3)* was introduced in 7.9.7.

**AVAILABILITY**

Added in curl 7.9.7

**RETURN VALUE**

This returns CURLE\_OK.

**SEE ALSO**

**CURLOPT\_HEADERDATA(3), CURLOPT\_READFUNCTION(3), CURLOPT\_WRITEDATA(3),  
CURLOPT\_WRITEFUNCTION(3)**