

NAME

CURLOPT_PROGRESSDATA – pointer passed to the progress callback

SYNOPSIS

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

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

DESCRIPTION

Pass a *pointer* that is untouched by libcurl and passed as the first argument in the progress callback set with *CURLOPT_PROGRESSFUNCTION(3)*.

DEFAULT

NULL

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
struct progress {
    char *private;
    size_t size;
};

static int progress_callback(void *clientp,
                             double dltotal,
                             double dlnow,
                             double ultotal,
                             double ulnow)
{
    struct progress *memory = clientp;
    printf("private: %p\n", memory->private);

    /* use the values */

    return 0; /* all is good */
}

int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        struct progress data;

        /* pass struct to callback */
        curl_easy_setopt(curl, CURLOPT_PROGRESSDATA, &data);
        curl_easy_setopt(curl, CURLOPT_PROGRESSFUNCTION, progress_callback);

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

AVAILABILITY

Added in curl 7.1

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_PROGRESSFUNCTION(3), CURLOPT_XFERINFOFUNCTION(3)