NAME
> CURLOPT_WRITEDATA – custom pointer passed to the write callback

SYNOPSIS
> #include <curl/curl.h>
>
> CURLcode curl_easy_setopt(CURL *handle, CURLOPT_WRITEDATA, void *pointer);

DESCRIPTION
> A data *pointer* to pass to the write callback. If you use the *CURLOPT_WRITEFUNCTION(3)* option, this is the pointer you'll get in that callback's 4th argument. If you don't use a write callback, you must make *pointer* a 'FILE *' (cast to 'void *') as libcurl will pass this to *fwrite(3)* when writing data.
>
> The internal *CURLOPT_WRITEFUNCTION(3)* will write the data to the FILE * given with this option, or to stdout if this option hasn't been set.
>
> If you're using libcurl as a win32 DLL, you **MUST** use the *CURLOPT_WRITEFUNCTION(3)* if you set this option or you will experience crashes.

DEFAULT
> By default, this is a FILE * to stdout.

PROTOCOLS
> Used for all protocols.

EXAMPLE
> A common technique is to use the write callback to store the incoming data into a dynamically growing allocated buffer, and then this CURLOPT_WRITEDATA is used to point to a struct or the buffer to store data in. Like in the getinmemory example: http://curl.haxx.se/libcurl/c/getinmemory.html

AVAILABILITY
> Available in all libcurl versions. This option was formerly known as *CURLOPT_FILE*, the name *CURLOPT_WRITEDATA* was introduced in 7.9.7.

RETURN VALUE
> This will return CURLE_OK.

SEE ALSO
> **CURLOPT_WRITEFUNCTION**(3), **CURLOPT_READDATA**(3),