

NAME

curl_global_trace – log configuration

SYNOPSIS

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

```
CURLcode curl_global_trace(const char *config);
```

DESCRIPTION

This function configures the logging behavior to make some parts of curl more verbose or silent than others.

This function may be called during the initialization phase of a program. It does not have to be. It can be called several times even, possibly overwriting settings of previous calls.

Calling this function after transfers have been started is undefined. On some platforms/architectures it might take effect, on others not.

This function is thread-safe since libcurl 8.3.0 if *curl_version_info(3)* has the `CURL_VERSION_THREADSAFE` feature bit set (most platforms).

If this is not thread-safe, you must not call this function when any other thread in the program (i.e. a thread sharing the same memory) is running. This does not only mean no other thread that is using libcurl. Because *curl_global_init(3)* may call functions of other libraries that are similarly thread-unsafe, it could conflict with any other thread that uses these other libraries.

If you are initializing libcurl from a Windows DLL you should not initialize it from *DllMain* or a static initializer because Windows holds the loader lock during that time and it could cause a deadlock.

The *config* string is a list of comma-separated component names. Names are case-insensitive and unknown names are ignored. The special name "all" applies to all components. Names may be prefixed with '+' or '-' to enable or disable detailed logging for a component.

The list of component names is not part of curl's public API. Names may be added or disappear in future versions of libcurl. Since unknown names are silently ignored, outdated log configurations does not cause errors when upgrading libcurl. Given that, some names can be expected to be fairly stable and are listed below for easy reference.

Note that log configuration applies only to transfers where debug logging is enabled. See *CURLOPT_VERBOSE(3)* or *CURLOPT_DEBUGFUNCTION(3)* on how to control that.

TRACE COMPONENTS

- tcp Tracing of TCP socket handling: connect, sends, receives.
- ssl Tracing of SSL/TLS operations, whichever SSL backend is used in your build.
- ftp Tracing of FTP operations when this protocol is enabled in your build.
- http/2 Details about HTTP/2 handling: frames, events, I/O, etc.
- http/3 Details about HTTP/3 handling: connect, frames, events, I/O etc.
- http-proxy
Involved when transfers are tunneled through an HTTP proxy. "h1-proxy" or "h2-proxy" are also involved, depending on the HTTP version negotiated with the proxy.

In order to find out all components involved in a transfer, run it with "all" configured. You can then see all names involved in your libcurl version in the trace.

dns	Tracing of DNS operations to resolve hostnames and HTTPS records.
lib-ids	Adds transfer and connection identifiers as prefix to every call to <i>CURLOPT_DEBUGFUNCTION</i> (3). The format is <i>[n-m]</i> where <i>n</i> is the identifier of the transfer and <i>m</i> is the identifier of the connection. A literal <i>x</i> is used for internal transfers or when no connection is assigned. For example, <i>[5-x]</i> is the prefix for transfer 5 that has no connection. The command line tool <i>curl</i> uses the same format for its <i>-trace-ids</i> option. <i>lib-ids</i> is intended for libcurl applications that handle multiple transfers but have no own way to identify in trace output which transfer a trace event is connected to.
doh	Former name for DNS-over-HTTP operations. Now an alias for <i>dns</i> .
multi	Traces multi operations managing transfers' state changes and sockets poll states.
read	Traces reading of upload data from the application in order to send it to the server.
ssh	Tracing of SSH related protocols SCP and SFTP.
ssls	Tracing of SSL Session handling, e.g. caching/import/export.
smtp	Tracing of SMTP operations when this protocol is enabled in your build.
threads	Tracing of thread queue and pools, used in threaded DNS resolving.
timer	Tracing of timers set for transfers.
write	Traces writing of download data, received from the server, to the application.
ws	Tracing of WebSocket operations when this protocol is enabled in your build.

TRACE GROUPS

Besides the specific component names there are the following group names defined:

all

network

All components involved in bare network I/O, including the SSL layer.

All components that your libcurl is built with.

protocol

All components involved in transfer protocols, such as 'ftp' and 'http/2'.

proxy All components involved in use of proxies.

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
    /* log details of HTTP/2 and SSL handling */
    curl_global_trace("http/2,ssl");

    /* log all details, except SSL handling */
    curl_global_trace("all,-ssl");
}
```

Below is a trace sample where "http/2" was configured. The trace output of an enabled component appears at the beginning in brackets.

```
* [HTTP/2] [h2sid=1] cf_send(len=96) submit https://example.com/
...
* [HTTP/2] [h2sid=1] FRAME[HEADERS]
```

* [HTTP/2] [h2sid=1] 249 header bytes

...

AVAILABILITY

Added in curl 8.3.0

RETURN VALUE

If this function returns non-zero, something went wrong and the configuration may not have any effects or may only been applied partially.

SEE ALSO

curl_global_init(3), **libcurl(3)**