

NAME

libcurl-env - environment variables libcurl understands

DESCRIPTION

libcurl reads and understands a set of environment variables that if set controls and changes behaviors. This is the full list of variables to set and description of what they do. Also note that curl, the command line tool, supports a set of additional environment variables independently of this.

[scheme]_proxy

When libcurl is given a URL to use in a transfer, it first extracts the scheme part from the URL and checks if there is a given proxy set for that in its corresponding environment variable. A URL like `https://example.com` makes libcurl use the **http_proxy** variable, while a URL like `ftp://example.com` uses the **ftp_proxy** variable.

These proxy variables are also checked for in their uppercase versions, except the **http_proxy** one which is only used lowercase. Note also that some systems actually have a case insensitive handling of environment variables and then of course **HTTP_PROXY** still works.

An exception exists for the WebSocket **ws** and **wss** URL schemes, where libcurl first checks **ws_proxy** or **wss_proxy** but if they are not set, it will fall back and try the `http` and `https` versions instead if set.

ALL_PROXY

This is a setting to set proxy for all URLs, independently of what scheme is being used. Note that the scheme specific variables overrides this one if set.

CURL_SSL_BACKEND

When libcurl is built to support multiple SSL backends, it selects a specific backend at first use. If no selection is done by the program using libcurl, this variable's selection is used. Setting a name that is not a built-in alternative makes libcurl stay with the default.

SSL backend names (case-insensitive): BearSSL, GnuTLS, mbedTLS, nss, OpenSSL, rustls, Schannel, Secure-Transport, wolfSSL

HOME

When the `netrc` feature is used (`CURLOPT_NETRC(3)`), this variable is checked as the primary way to find the "current" home directory in which the `.netrc` file is likely to exist.

USERPROFILE

When the `netrc` feature is used (`CURLOPT_NETRC(3)`), this variable is checked as the

secondary way to find the "current" home directory (on Windows only) in which the `.netrc` file is likely to exist.

LOGNAME

User name to use when invoking the `ntlm-wb` tool, if `NTLMUSER` was not set.

NO_PROXY

This has the same functionality as the `CURLOPT_NOPROXY(3)` option: it gives libcurl a comma-separated list of host name patterns for which libcurl should not use a proxy.

NTLMUSER

User name to use when invoking the `ntlm-wb` tool.

SSLKEYLOGFILE

When set and libcurl runs with a SSL backend that supports this feature, libcurl saves SSL secrets into the given file name. Using those SSL secrets, other tools (such as Wireshark) can decrypt the SSL communication and analyze/view the traffic.

These secrets and this file might be sensitive. Users are advised to take precautions so that they are not stolen or otherwise inadvertently revealed.

USER

User name to use when invoking the `ntlm-wb` tool, if `NTLMUSER` and `LOGNAME` were not set.

Debug Variables

Debug variables are intended for internal use and are documented in `libcurl-env-dbg(3)`.

SEE ALSO

libcurl-env-dbg(3)