

# libcurl library

The libcurl library is a free, client-side URL transfer library, supporting a range of common Internet protocols. It is designed to be portable, efficient, and fully-featured, making it an excellent tool for transferring data with URLs from within applications. libcurl is highly versatile, being usable in a wide variety of programming languages through its bindings and interfaces.

Here's a closer look at its capabilities and features:

## Supported Protocols

libcurl supports a broad array of protocols, making it suitable for many different network operations. Some of the supported protocols include:

HTTP

HTTPS

FTP

FTPS

SCP

SFTP

TFTP

LDAP

LDAPS

DICT

FILE

Telnet

IMAP

SMTP

POP3

And more

## Key Features

**Multi-Platform:** libcurl works on a wide range of platforms, including Windows, macOS, Linux, and many others. This cross-platform support ensures that applications using libcurl can be deployed on virtually any system.

**Thread Safe:** It is designed to be safe to use in multithreaded applications, allowing for parallel transfers in the same or different threads.

**Asynchronous Operations:** libcurl can perform non-blocking/asynchronous operations, making it suitable for applications that require high performance without blocking the main thread.

**SSL/TLS Support:** It offers robust support for secure data transfers using SSL and TLS, including the ability to work with various certificate types and encryption libraries like OpenSSL, LibreSSL, BoringSSL, and others.

**Authentication and Cookies:** libcurl supports a wide range of authentication mechanisms (Basic, Digest, NTLM, and more) and can handle cookie management, which is essential for interacting with web services that require login or session management.

**Custom Headers and HTTP Methods:** Developers can customize HTTP headers and use various HTTP methods (GET, POST, PUT, DELETE, etc.), providing flexibility in how they interact with web services.

## Use Cases

**Downloading and Uploading Files:** libcurl is widely used for downloading files from the Internet or uploading them to remote servers.

**Interacting with Web APIs:** It's a popular choice for making HTTP requests to RESTful APIs, handling JSON or XML data returned by these services.

**Web Scraping:** Although primarily a transfer tool, libcurl can be used as part of a web scraping setup to retrieve web page content.

**Data Transfer in Desktop and Server Applications:** It's used in both desktop and server-side applications for any tasks that involve network data transfer.

## Conclusion

libcurl is a cornerstone library for network programming, providing a comprehensive solution for data transfer over the internet. Its wide protocol support, portability, and feature set make it an invaluable tool for developers needing to add network communication capabilities to their applications.

## Use libcurl library from Harbour

Use Cases:

**Downloading Files:** This function can download files from the internet, such as documents, images, or any other content accessible via a URL.

**API Interaction:** It can be used to interact with web APIs that require simple GET requests.

**Data Collection:** Useful for collecting data from various online sources for processing or analysis in a Harbour application.

**Web Scraping:** Although not shown directly in this code, with slight modifications, it could be adapted for web scraping purposes.

```
#include "hbcurl.ch"
```

```
function Main()
```

```
    ? urlLoad( "https://raw.githubusercontent.com/FiveTechSoft/harbourPackages/main/python/init.prg" )
```

```

return nil

function urlLoad( cUrl )
    local hCurl, nRet

    if ! Empty( hCurl := curl_easy_init() )
        curl_easy_setopt( hCurl, HB_CURLOPT_DOWNLOAD )
        curl_easy_setopt( hCurl, HB_CURLOPT_URL, cUrl )
        curl_easy_setopt( hCurl, HB_CURLOPT_SSL_VERIFYPEER, .F. )
        curl_easy_setopt( hCurl, HB_CURLOPT_SSL_VERIFYHOST, .F. )
        curl_easy_setopt( hCurl, HB_CURLOPT_FOLLOWLOCATION )
        curl_easy_setopt( hCurl, HB_CURLOPT_FILETIME, 1 )
        curl_easy_setopt( hCurl, HB_CURLOPT_DL_FILE_SETUP, SubStr( cUrl, RAT( "/", cUrl ) + 1 ) )
        curl_easy_setopt( hCurl, HB_CURLOPT_NOPROGRESS, .F. )
        curl_easy_setopt( hCurl, HB_CURLOPT_VERBOSE, .F. )
        curl_easy_setopt( hCurl, HB_CURLOPT_FAILONERROR, .T. )

        nRet = curl_easy_perform( hCurl )
        curl_easy_cleanup( hCurl )
    endif

return nRet

```

Based on the provided code snippet, `urlLoad(cUrl)` is a function that utilizes the libcurl library to download data from the internet. The code is written in a language that appears to be Harbour, a dialect of the xBase programming language. Harbour is commonly used for database applications but can also interact with web resources using extensions like libcurl. Here's a breakdown of what the `urlLoad` function does:

**Initialization:** The function starts by attempting to initialize a curl session using `curl_easy_init()`. If successful, `hCurl` holds the handle to the curl session.

**Setting Options:** Several options are set for the curl session:

`HB_CURLOPT_DOWNLOAD` indicates that the session is intended for downloading data.

`HB_CURLOPT_URL` sets the URL to download from, which is passed to the function via the `cUrl` parameter.

`HB_CURLOPT_SSL_VERIFYPEER` and `HB_CURLOPT_SSL_VERIFYHOST` are set to `.F.` (false), which disables SSL certificate verification. This is useful for testing or in controlled environments but poses a security risk in production.

`HB_CURLOPT_FOLLOWLOCATION` allows curl to follow redirects, ensuring the actual resource is reached even if the URL has moved.

`HB_CURLOPT_FILETIME` is enabled to attempt to get the remote document's timestamp.

`HB_CURLOPT_DL_FILE_SETUP` specifies the local file name for the downloaded content, derived from the last segment of the URL.

`HB_CURLOPT_NOPROGRESS` and `HB_CURLOPT_VERBOSE` are set to control the verbosity and progress reporting of the operation.

`HB_CURLOPT_FAILONERROR` is set to `.T.` (true), making the function call fail if the HTTP response code is 400 or above.

**Performing the Operation:** `curl_easy_perform(hCurl)` executes the configured download operation. The success or failure of this operation is stored in `nRet`.

**Cleanup:** The curl session is cleaned up using `curl_easy_cleanup(hCurl)` to free resources.

**Return Value:** The function returns the result of the curl operation, typically a status code indicating success or failure.

**Security Note:** The disabling of SSL certificate verification ( `SSL_VERIFYPEER` and `SSL_VERIFYHOST` ) is a significant security risk in a production environment as it makes the application vulnerable to man-in-the-middle attacks. This should only be done in a controlled environment or for testing purposes with known, safe URLs.

---

## WebPageContents

Fivewin function for urlLoad

```
#include "fivewin.ch"
function Main()
    cUrl := "https://raw.githubusercontent.com/FiveTechSoft/harbourPackages/main/python/init.prg"
    ? WebPageContents( cUrl ) // --> "// dummy file"
return nil
```