

NAME

curl_url_get - extract a part from a URL

SYNOPSIS

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

```
CURLUcode curl_url_get(const CURLU *url,  
                        CURLUPart part,  
                        char **content,  
                        unsigned int flags);
```

DESCRIPTION

Given a *url* handle of a URL object, this function extracts an individual piece or the full URL from it.

The *part* argument specifies which part to extract (see list below) and *content* points to a 'char *' to get updated to point to a newly allocated string with the contents.

The *flags* argument is a bitmask with individual features.

The returned content pointer must be freed with *curl_free(3)* after use.

FLAGS

The flags argument is zero, one or more bits set in a bitmask.

CURLU_DEFAULT_PORT

If the handle has no port stored, this option makes *curl_url_get(3)* return the default port for the used scheme.

CURLU_DEFAULT_SCHEME

If the handle has no scheme stored, this option makes *curl_url_get(3)* return the default scheme instead of error.

CURLU_NO_DEFAULT_PORT

Instructs *curl_url_get(3)* to not return a port number if it matches the default port for the scheme.

CURLU_URLDECODE

Asks *curl_url_get(3)* to URL decode the contents before returning it. It does not decode the scheme, the port number or the full URL.

The query component also gets plus-to-space conversion as a bonus when this bit is set.

Note that this URL decoding is charset unaware and you get a zero terminated string back with data that could be intended for a particular encoding.

If there are byte values lower than 32 in the decoded string, the get operation returns an error instead.

CURLU_URLENCODE

If set, *curl_url_get(3)* URL encodes the hostname part when a full URL is retrieved. If not set (default), libcurl returns the URL with the hostname raw to support IDN names to appear as-is. IDN hostnames are typically using non-ASCII bytes that otherwise gets percent-encoded.

Note that even when not asking for URL encoding, the '%' (byte 37) is URL encoded to make sure the hostname remains valid.

CURLU_PUNYCODE

If set and *CURLU_URLENCODE* is not set, and asked to retrieve the **CURLUPART_HOST** or **CURLUPART_URL** parts, libcurl returns the host name in its punycode version if it contains any non-ASCII octets (and is an IDN name).

If libcurl is built without IDN capabilities, using this bit makes *curl_url_get(3)* return *CURLUE_LACKS_IDN* if the hostname contains anything outside the ASCII range.

(Added in curl 7.88.0)

CURLU_PUNY2IDN

If set and asked to retrieve the **CURLUPART_HOST** or **CURLUPART_URL** parts, libcurl returns the hostname in its IDN (International Domain Name) UTF-8 version if it otherwise is a punycode version. If the punycode name cannot be converted to IDN correctly, libcurl returns *CURLUE_BAD_HOSTNAME*.

If libcurl is built without IDN capabilities, using this bit makes *curl_url_get(3)* return *CURLUE_LACKS_IDN* if the hostname is using punycode.

(Added in curl 8.3.0)

CURLU_GET_EMPTY

When this flag is used in *curl_url_get()*, it makes the function return empty query and fragments parts or when used in the full URL. By default, libcurl otherwise considers empty parts non-existing.

An empty query part is one where this is nothing following the question mark (before the possible fragment). An empty fragments part is one where there is nothing following the hash sign.

(Added in curl 8.8.0)

CURLU_NO_GUESS_SCHEME

When this flag is used in `curl_url_get()`, it treats the scheme as non-existing if it was set as a result of a previous guess; when `CURLU_GUESS_SCHEME` was used parsing a URL.

Using this flag when getting `CURLUPART_SCHEME` if the scheme was set as the result of a guess makes `curl_url_get()` return `CURLUE_NO_SCHEME`.

Using this flag when getting `CURLUPART_URL` if the scheme was set as the result of a guess makes `curl_url_get()` return the full URL without the scheme component. Such a URL can then only be parsed with `curl_url_set()` if `CURLU_GUESS_SCHEME` is used.

(Added in curl 8.9.0)

PARTS

CURLUPART_URL

When asked to return the full URL, `curl_url_get(3)` returns a normalized and possibly cleaned up version using all available URL parts.

We advise using the `CURLU_PUNYCODE` option to get the URL as "normalized" as possible since IDN allows hostnames to be written in many different ways that still end up the same punycode version.

Zero-length queries and fragments are excluded from the URL unless `CURLU_GET_EMPTY` is set.

CURLUPART_SCHEME

Scheme cannot be URL decoded on get.

CURLUPART_USER

CURLUPART_PASSWORD

CURLUPART_OPTIONS

The options field is an optional field that might follow the password in the userinfo part. It is only recognized/used when parsing URLs for the following schemes: pop3, smtp and imap. The URL

API still allows users to set and get this field independently of scheme when not parsing full URLs.

CURLUPART_HOST

The hostname. If it is an IPv6 numeric address, the zone id is not part of it but is provided separately in *CURLUPART_ZONEID*. IPv6 numerical addresses are returned within brackets ([]).

IPv6 names are normalized when set, which should make them as short as possible while maintaining correct syntax.

CURLUPART_ZONEID

If the hostname is a numeric IPv6 address, this field might also be set.

CURLUPART_PORT

A port cannot be URL decoded on get. This number is returned in a string just like all other parts. That string is guaranteed to hold a valid port number in ASCII using base 10.

CURLUPART_PATH

The *part* is always at least a slash ('/') even if no path was supplied in the URL. A URL path always starts with a slash.

CURLUPART_QUERY

The initial question mark that denotes the beginning of the query part is a delimiter only. It is not part of the query contents.

A not-present query returns *part* set to NULL.

A zero-length query returns *part* as NULL unless *CURLU_GET_EMPTY* is set.

The query part gets pluses converted to space when asked to URL decode on get with the *CURLU_URLDECODE* bit.

CURLUPART_FRAGMENT

The initial hash sign that denotes the beginning of the fragment is a delimiter only. It is not part of the fragment contents.

A not-present fragment returns *part* set to NULL.

A zero-length fragment returns *part* as NULL unless *CURLU_GET_EMPTY* is set.

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
    CURLUcode rc;
    CURLU *url = curl_url();
    rc = curl_url_set(url, CURLUPART_URL, "https://example.com", 0);
    if(!rc) {
        char *scheme;
        rc = curl_url_get(url, CURLUPART_SCHEME, &scheme, 0);
        if(!rc) {
            printf("the scheme is %s\n", scheme);
            curl_free(scheme);
        }
        curl_url_cleanup(url);
    }
}
```

AVAILABILITY

Added in curl 7.62.0

RETURN VALUE

Returns a CURLUcode error value, which is CURLUE_OK (0) if everything went fine. See the *libcurl-errors(3)* man page for the full list with descriptions.

If this function returns an error, no URL part is returned.

SEE ALSO

CURLOPT_CURLU(3), **curl_url(3)**, **curl_url_cleanup(3)**, **curl_url_dup(3)**, **curl_url_set(3)**, **curl_url_strerror(3)**