

**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\_URLENCODER

If set, *curl\_url\_get(3)* URL encodes the host name part when a full URL is retrieved. If not set (default), libcurl returns the URL with the host name "raw" to support IDN names to appear as-is. IDN host names 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 host name remains valid.

#### CURLU\_PUNYCODE

If set and *CURLU\_URLENCODER* 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 host name 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 host name 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 host name is using punycode.

(Added in curl 8.3.0)

### 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 host names to be written in many different ways that still end up the same punycode version.

#### 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 host name. 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 host name 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 a zero-length string.

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.

### 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 7.62.0. `CURLUPART_ZONEID` was added in 7.65.0.

### 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

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