

NAME

curl_getdate – convert date string to number of seconds

SYNOPSIS

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

```
time_t curl_getdate(const char *datestring, const time_t *now);
```

DESCRIPTION

curl_getdate(3) returns the number of seconds since the Epoch, January 1st 1970 00:00:00 in the UTC time zone, for the date and time that the *datestring* parameter specifies. The *now* parameter is not used, pass a NULL there.

This function works with valid dates and does not always detect and reject wrong dates, such as February 30.

PARSING DATES AND TIMES

A "date" is a string containing several items separated by whitespace. The order of the items is immaterial. A date string may contain many flavors of items:

calendar date items

Can be specified several ways. Month names can only be three-letter English abbreviations, numbers can be zero-prefixed and the year may use 2 or 4 digits. Examples: 06 Nov 1994, 06-Nov-94 and Nov-94 6.

If the year appears to be below 100 (two-digit), any year after 70 is assumed to be 1900 + the given year. All others are 2000 + the given year.

time of the day items

This string specifies the time on a given day. You must specify it with 6 digits with two colons: HH:MM:SS. If there is no time given in a provided date string, 00:00:00 is assumed. Example: 18:19:21.

time zone items

Specifies international time zone. There are a few acronyms supported, but in general you should instead use the specific relative time compared to UTC. Supported formats include: -1200, MST, +0100.

day of the week items

Specifies a day of the week. Days of the week may be spelled out in full (using English): 'Sunday', 'Monday', etc or they may be abbreviated to their first three letters. This is usually not info that adds anything.

pure numbers

If a decimal number of the form YYYYMMDD appears, then YYYY is read as the year, MM as the month number and DD as the day of the month, for the specified calendar date.

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
    time_t t;
    t = curl_getdate("Sun, 06 Nov 1994 08:49:37 GMT", NULL);
    t = curl_getdate("Sunday, 06-Nov-94 08:49:37 GMT", NULL);
    t = curl_getdate("Sun Nov 6 08:49:37 1994", NULL);
    t = curl_getdate("06 Nov 1994 08:49:37 GMT", NULL);
    t = curl_getdate("06-Nov-94 08:49:37 GMT", NULL);
    t = curl_getdate("Nov 6 08:49:37 1994", NULL);
}
```

```

t = curl_getdate("06 Nov 1994 08:49:37", NULL);
t = curl_getdate("06-Nov-94 08:49:37", NULL);
t = curl_getdate("1994 Nov 6 08:49:37", NULL);
t = curl_getdate("GMT 08:49:37 06-Nov-94 Sunday", NULL);
t = curl_getdate("94 6 Nov 08:49:37", NULL);
t = curl_getdate("1994 Nov 6", NULL);
t = curl_getdate("06-Nov-94", NULL);
t = curl_getdate("Sun Nov 6 94", NULL);
t = curl_getdate("1994.Nov.6", NULL);
t = curl_getdate("Sun/Nov/6/94/GMT", NULL);
t = curl_getdate("Sun, 06 Nov 1994 08:49:37 CET", NULL);
t = curl_getdate("06 Nov 1994 08:49:37 EST", NULL);
t = curl_getdate("Sun, 12 Sep 2004 15:05:58 -0700", NULL);
t = curl_getdate("Sat, 11 Sep 2004 21:32:11 +0200", NULL);
t = curl_getdate("20040912 15:05:58 -0700", NULL);
t = curl_getdate("20040911 +0200", NULL);
}

```

STANDARDS

This parser handles date formats specified in RFC 822 (including the update in RFC 1123) using time zone name or time zone delta and RFC 850 (obsoleted by RFC 1036) and ANSI C's *asctime()* format.

These formats are the only ones RFC 7231 says HTTP applications may use.

AVAILABILITY

Added in curl 7.1

RETURN VALUE

This function returns `-1` when it fails to parse the date string. Otherwise it returns the number of seconds as described.

On systems with a signed 32-bit `time_t`: if the year is larger than 2037 or less than 1903, this function returns `-1`.

On systems with an unsigned 32-bit `time_t`: if the year is larger than 2106 or less than 1970, this function returns `-1`.

On systems with 64-bit `time_t`: if the year is less than 1583, this function returns `-1`. (The Gregorian calendar was first introduced 1582 so no "real" dates in this way of doing dates existed before then.)

SEE ALSO

CURLOPT_TIMECONDITION(3), **CURLOPT_TIMEVALUE(3)**, **curl_easy_escape(3)**,
curl_easy_unescape(3)