

Search:

Not logged in

Reference <ctime> localtime

[register](#)[log in](#)

C++

[Information](#)
[Tutorials](#)
[Reference](#)
[Articles](#)
[Forum](#)

Reference

C library:

[<cassert> \(assert.h\)](#)
[<cctype> \(ctype.h\)](#)
[<cerrno> \(errno.h\)](#)
[<cfenv> \(fenv.h\)](#)
[<cfloat> \(float.h\)](#)
[<cinttypes> \(inttypes.h\)](#)
[<ciso646> \(iso646.h\)](#)
[<climits> \(limits.h\)](#)
[<locale> \(locale.h\)](#)
[<cmath> \(math.h\)](#)
[<csetjmp> \(setjmp.h\)](#)
[<csignal> \(signal.h\)](#)
[<csdarg> \(stdarg.h\)](#)
[<cstdlib> \(stdlib.h\)](#)
[<csddef> \(stddef.h\)](#)
[<csdint> \(stdint.h\)](#)
[<csdio> \(stdio.h\)](#)
[<csdlib> \(stdlib.h\)](#)
[<cstring> \(string.h\)](#)
[<ctgmath> \(tgmath.h\)](#)
[<ctime> \(time.h\)](#)
[<cuchar> \(uchar.h\)](#)
[<wchar> \(wchar.h\)](#)
[<cwctype> \(wctype.h\)](#)

Containers:

Input/Output:

Multi-threading:

Other:

<ctime> (time.h)

functions:

[asctime](#)
[clock](#)
[ctime](#)
[difftime](#)
[gmtime](#)
[localtime](#)
[mktime](#)
[strftime](#)
[time](#)

macros:

[CLOCKS_PER_SEC](#)
[NULL](#)

types:

[clock_t](#)
[size_t](#)
[time_t](#)
[struct tm](#)

GroupDocs.Viewer

.NET document viewer API



function

localtime

<ctime>

```
struct tm * localtime (const time_t * timer);
```

Convert time_t to tm as local time

Uses the value pointed by *timer* to fill a `tm` structure with the values that represent the corresponding time, expressed for the local timezone.

Parameters

timer

Pointer to an object of type `time_t` that contains a time value.

`time_t` is an alias of a fundamental *arithmetic type* capable of representing times as returned by function `time`.

Return Value

A pointer to a `tm` structure with its members filled with the values that correspond to the local time representation of *timer*.

The returned value points to an internal object whose validity or value may be altered by any subsequent call to `gmtime` or `localtime`.

Example

```
1 /* localtime example */
2 #include <stdio.h>      /* puts, printf */
3 #include <time.h>       /* time_t, struct tm, time, localtime */
4
5 int main ()
6 {
7     time_t rawtime;
8     struct tm * timeinfo;
9
10    time (&rawtime);
11    timeinfo = localtime (&rawtime);
12    printf ("Current local time and date: %s", asctime(timeinfo));
13
14    return 0;
15 }
```

[Edit & Run](#)

Output:

```
Current local time and date: Wed Feb 13 17:17:11 2013
```

Data races

The function accesses the object pointed by *timer*.

The function also accesses and modifies a shared internal object, which may introduce data races on concurrent calls to `gmtime` and `localtime`. Some libraries provide an alternative function that avoids this data race: `localtime_r` (non-portable).

Exceptions (C++)

No-throw guarantee: this function never throws exceptions.

See also

asctime	Convert tm structure to string (function)
ctime	Convert time_t value to string (function)
gmtime	Convert time_t to tm as UTC time (function)
mktime	Convert tm structure to time_t (function)
time	Get current time (function)

[Home page](#) | [Privacy policy](#)

© cplusplus.com, 2000-2021 - All rights reserved - v3.2

[Spotted an error? contact us](#)