

NAME

Date::Format - Date formatting subroutines

SYNOPSIS

```
use Date::Format;

@lt = localtime(time);

print time2str($template, time);
print strftime($template, @lt);

print time2str($template, time, $zone);
print strftime($template, @lt, $zone);

print ctime(time);
print asctime(@lt);

print ctime(time, $zone);
print asctime(@lt, $zone);
```

DESCRIPTION

This module provides routines to format dates into ASCII strings. They correspond to the C library routines `strftime` and `ctime`.

`time2str(TEMPLATE, TIME [, ZONE])`

`time2str` converts `TIME` into an ASCII string using the conversion specification given in `TEMPLATE`. `ZONE` if given specifies the zone which the output is required to be in, `ZONE` defaults to your current zone.

`strftime(TEMPLATE, TIME [, ZONE])`

`strftime` is similar to `time2str` with the exception that the time is passed as an array, such as the array returned by `localtime`.

`ctime(TIME [, ZONE])`

`ctime` calls `time2str` with the given arguments using the conversion specification "%a %b %e %T %Y\n"

`asctime(TIME [, ZONE])`

`asctime` calls `time2str` with the given arguments using the conversion specification "%a %b %e %T %Y\n"

MULTI-LANGUAGE SUPPORT

[Date::Format](#) is capable of formatting into several languages by creating a language specific object and calling methods, see [Date::Language](#)

```
my $lang = Date::Language->new('German');
$lang->time2str("%a %b %e %T %Y\n", time);
```

I am open to suggestions on this.

CONVERSION SPECIFICATION

Each conversion specification is replaced by appropriate characters as described in the following list. The appropriate characters are determined by the `LC_TIME` category of the program's locale.

```

%% PERCENT
%a day of the week abbr
%A day of the week
%b month abbr
%B month
%c MM/DD/YY HH:MM:SS
%C ctime format: Sat Nov 19 21:05:57 1994
%d numeric day of the month, with leading zeros (eg 01..31)
%e like %d, but a leading zero is replaced by a space (eg 1..32)
%D MM/DD/YY
%G GPS week number (weeks since January 6, 1980)
%h month abbr
%H hour, 24 hour clock, leading 0's
%I hour, 12 hour clock, leading 0's
%j day of the year
%k hour
%l hour, 12 hour clock
%L month number, starting with 1
%m month number, starting with 01
%M minute, leading 0's
%n NEWLINE
%o ornate day of month -- "1st", "2nd", "25th", etc.
%p AM or PM
%P am or pm (Yes %p and %P are backwards :)
%q Quarter number, starting with 1
%r time format: 09:05:57 PM
%R time format: 21:05
%s seconds since the Epoch, UCT
%S seconds, leading 0's
%t TAB
%T time format: 21:05:57
%U week number, Sunday as first day of week
%w day of the week, numerically, Sunday == 0
%W week number, Monday as first day of week
%x date format: 11/19/94
%X time format: 21:05:57
%y year (2 digits)
%Y year (4 digits)
%Z timezone in ascii. eg: PST
%z timezone in format -/+0000

```

%d, %e, %H, %I, %j, %k, %l, %m, %M, %q, %y and %Y can be output in Roman numerals by prefixing the letter with 0, e.g. %0Y will output the year as roman numerals.

LIMITATION

The functions in this module are limited to the time range that can be represented by the `time_t` data type, i.e. 1901-12-13 20:45:53 GMT to 2038-01-19 03:14:07 GMT.

AUTHOR

Graham Barr <gbarr@pobox.com>

COPYRIGHT

Copyright (c) 1995-2009 Graham Barr. This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself.