NAME

Time::gmtime - by-name interface to Perl's built-in gmtime() function

SYNOPSIS

```
use Time::gmtime;
$gm = gmtime();
printf "The day in Greenwich is %s\n",
(qw(Sun Mon Tue Wed Thu Fri Sat Sun))[ $gm->wday() ];

use Time::gmtime qw(:FIELDS);
gmtime();
printf "The day in Greenwich is %s\n",
(qw(Sun Mon Tue Wed Thu Fri Sat Sun))[ $tm_wday ];

$now = gmctime();

use Time::gmtime;
use File::stat;
$date_string = gmctime(stat($file)->mtime);
```

DESCRIPTION

This module's default exports override the core *gmtime()* function, replacing it with a version that returns "Time::tm" objects. This object has methods that return the similarly named structure field name from the C's tm structure from *time.h*; namely sec, min, hour, mday, mon, year, wday, yday, and isdst.

You may also import all the structure fields directly into your namespace as regular variables using the :FIELDS import tag. (Note that this still overrides your core functions.) Access these fields as variables named with a preceding `tm_` in front their method names. Thus, `$tm_obj->mday()` corresponds to `$tm_mday` if you import the fields.

The *gmctime()* function provides a way of getting at the scalar sense of the original *CORE::gmtime()* function.

To access this functionality without the core overrides, pass the `use` an empty import list, and then access function functions with their full qualified names. On the other hand, the built-ins are still available via the `CORE::` pseudo-package.

NOTE

While this class is currently implemented using the Class::Struct module to build a struct-like class, you shouldn't rely upon this.

AUTHOR

Tom Christiansen