

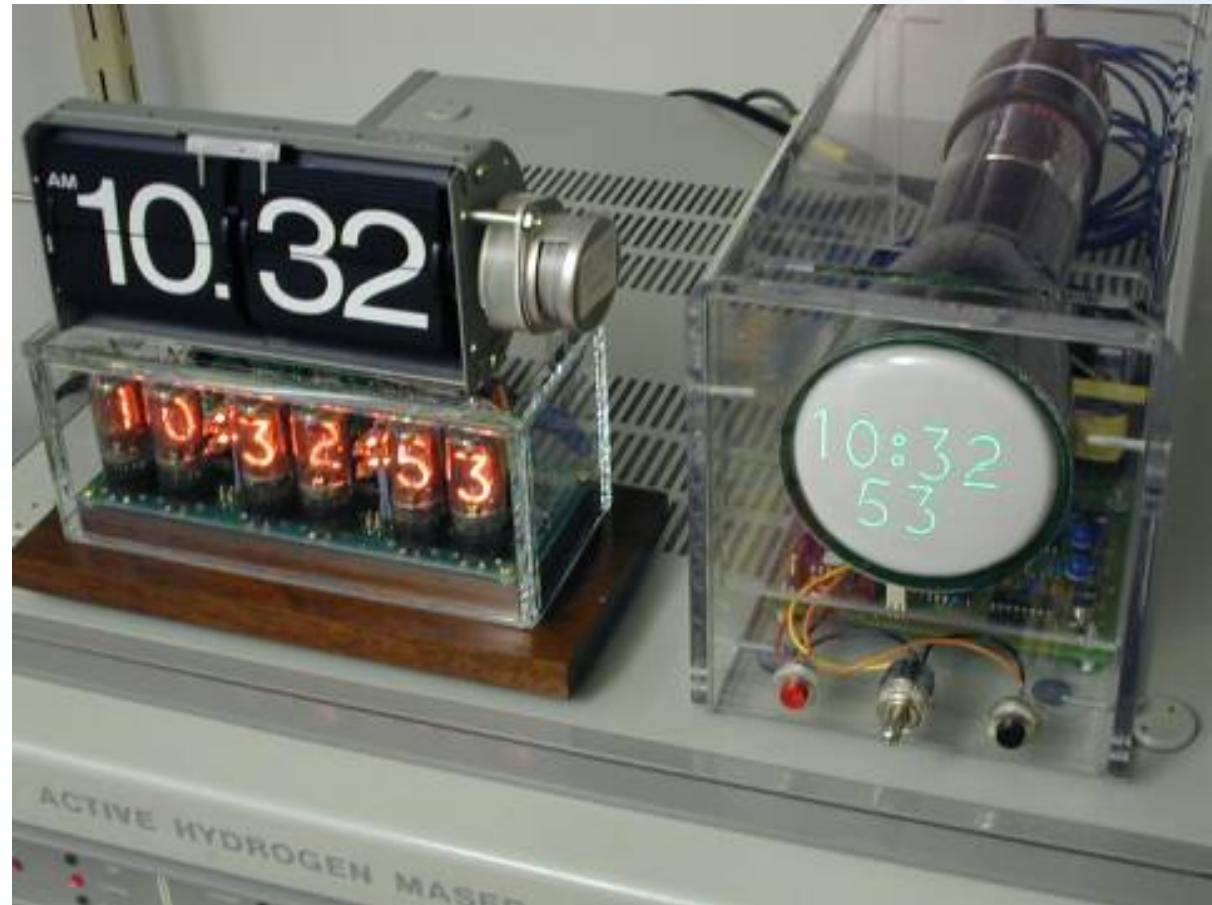
Welcome!

Share your information



What is Time?

This thing all things devours: Birds, beasts, trees, flowers;
Gnaws iron, bites steel, Grinds hard stones to meal; Slays king,
ruins town, And beats high mountain down.



What is Time?

- Time quantifies or measures the interval between events, or the duration of events. (wikipedia)
- The standard unit for time is the SI second.
- A second is the duration of 9,192,631,770 periods of the radiation corresponding to the transition between the two hyperfine levels of the ground state of the caesium-133 atom.
- We use larger units: minute, hour, day, week, month, year, century.
- Those additional units make things hard.

Problems with Dates and Time

Ambiguities:

06/08/04

- August 6th, 2004
- June 8th, 2004
- August 4th, 2006

Unreadable:

20040425010541

- April 25th, 2005, 01:05:41

"Weird" formats:

third saturday

2004-03-10 16:33:17.11403+01

2001-11-29T13:20:01.123-05:00

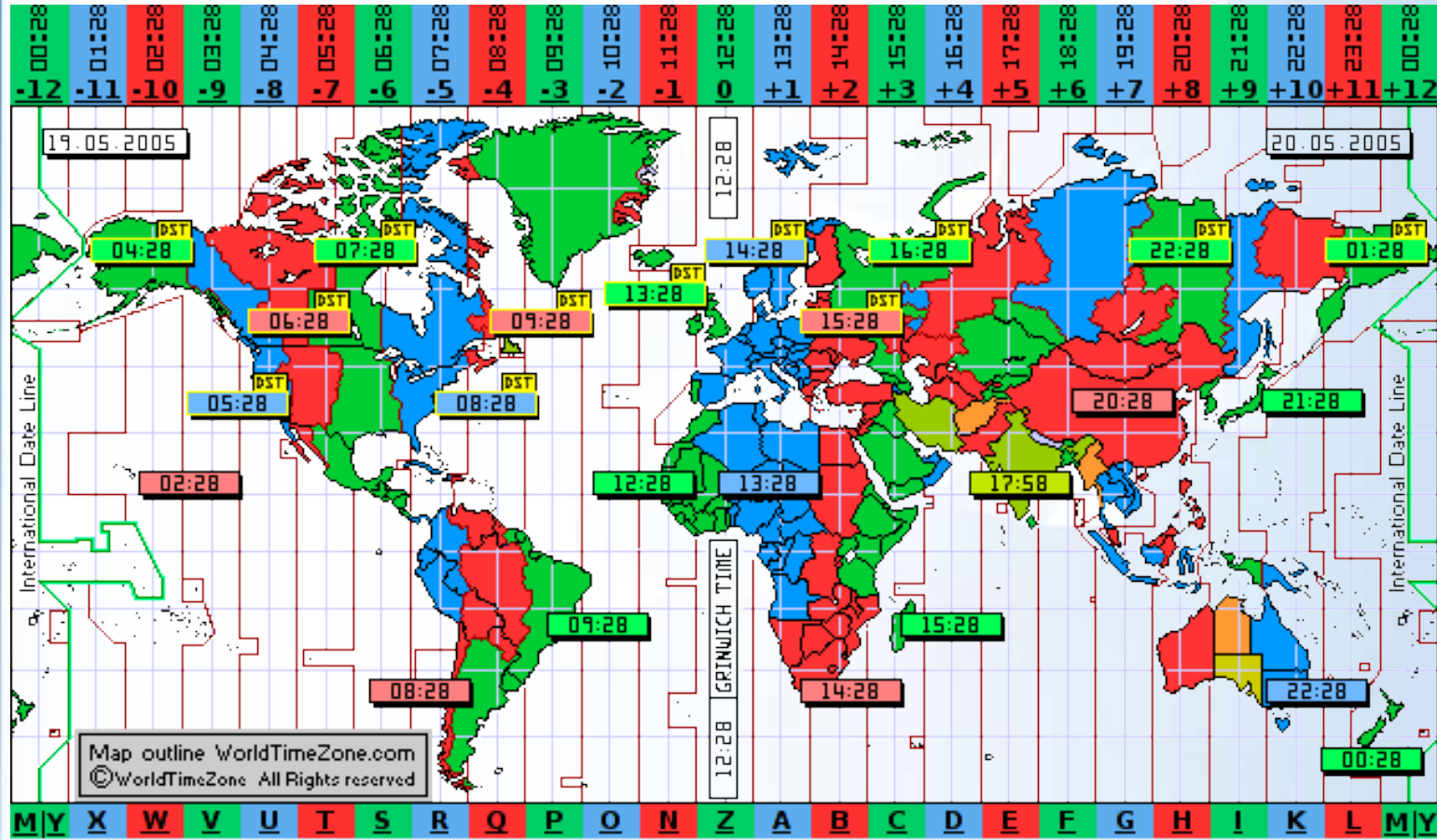
23:41:00.0Z

04:05:07.789 +0930

1999.238

Problems

Share your information



Problems

One identifier can mean different zones:

- PST: Pacific Standard Time, Pakistan Standard Time
- EST: Eastern Standard Time (USA), Eastern Standard Time (Australia) and Eastern Brazil Standard Time

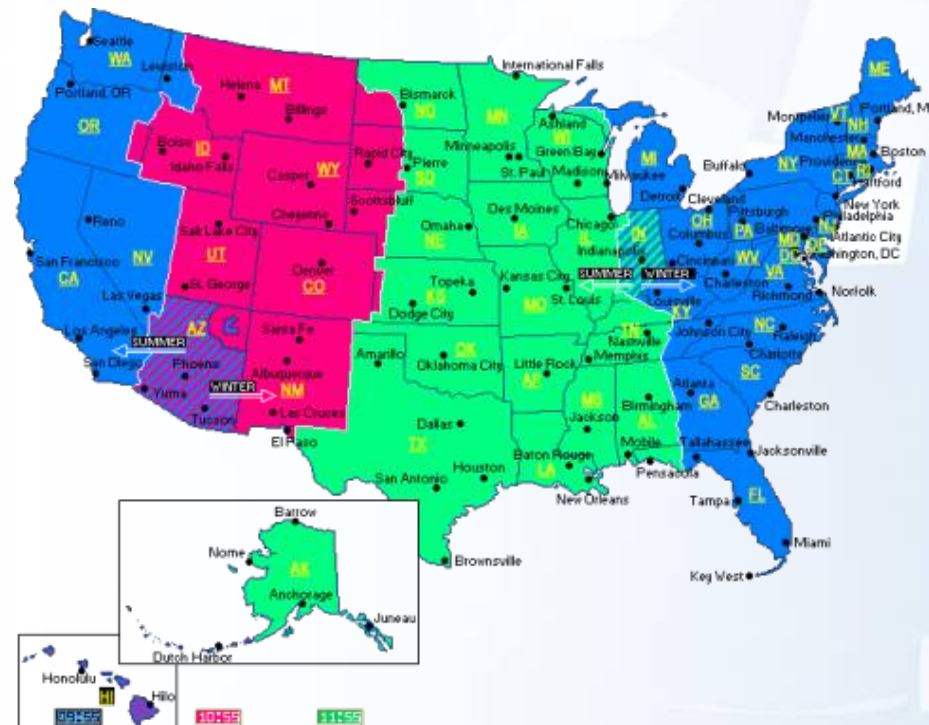
One zone can have multiple identifiers

- Central Europe Summer Time: CEST or CETDST

Names can be different on Operating Systems

Problems

- Artificial time offset to save "daylight"
- Not all countries/areas use it
- Switches are not done at the same time for all areas
- There are plenty of exceptions





Australia has vertical timezones

- Brazil changes change-over dates every year
- Lord Howe Island (Australia) moves only half an hour between DST and non-DST
- Nepal uses an quarter of an hour offsets
- And this continues for a while...

Date/Time Functions in PHP 4 and PHP

5.0

- `checkdate()` -- Validate a Gregorian date
- `date()` / `gmdate()` -- Format a local/GMT time/date
- `getdate()` -- Get date/time information (from a timestamp)
- `gettimeofday()` -- Get current time (as an array)
- `localtime()` -- Get the local time (as an array)
- `mktime()` / `gmmktime()` -- Get Unix timestamp for a date
- `strftime()` / `gmstrftime()` -- Format a local/GMT time/date according to locale settings
- `strtotime()` -- Parse about any English textual datetime description into a Unix timestamp

Date/Time Functions in PHP 4 and PHP

5.0

- Uses Unix timestamp as base unit (seconds since 1970-01-01, 00:00 GMT)
- Only 32 bit integers for timestamps (1902 to 2038)
- Limited to only positive numbers on some Operating Systems (1970 to 2038)
- `strtotime()` is buggy and very complex
- No way of dealing correctly with timezones
- Some functions are Operating System dependent

Date/Time Functions in PHP 5.1

- 64 bit timestamps
- strtotime() has been rewritten
- Nothing is Operating System dependent
- Full support for timezones, DST, date modifications
- New format modifiers: e for timezone identifier and o for ISO Year
- Advanced date handling functions

Date/Time Functions in PHP 5.1

"American" formats:

```
9/11                4:08 pm
12/22/78           8:51:00 am
```

"Combined" formats:

```
Sat, 24 Apr 2004 21:48:40 +0200
2001-11-29T13:20:01.123-05:00
```

Descriptive formats:

```
tomorrow           last saturday
four months ago    +40 days 2 hours
```

Textual formats:

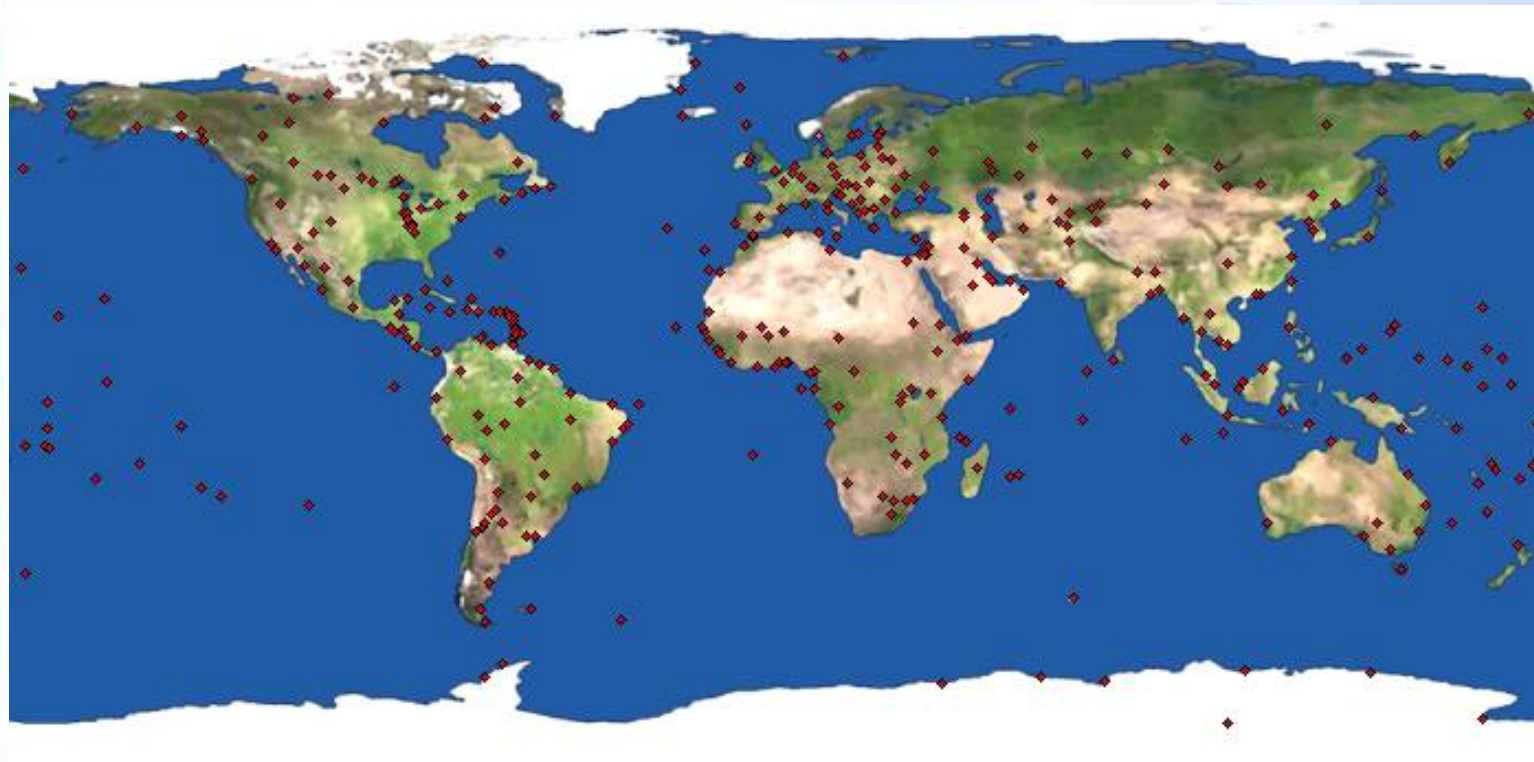
```
December 22. 1978    22-december-78
```

All ISO 8601 formats:

```
1978/12/22          1978-12-22          70-4-25
13:03:12.45678      15:57:41.0 pdt      13:03 CEST
13:03:12.45678 +0100 13:03:12.45678 CEST 04:05 -0930
15:57-8             231431 CEST         23:41F
```

Date/Time Functions in PHP 5.1

- Bundled timezone database with 538 zones
- Not dependent on timezone abbreviations
- Timezones have the format: Continent/Location or Continent/Location/Sublocation - Like: Europe/Amsterdam, America/Indiana/Knox



Parsing Dates

Parsing strings for date time information with the `strtotime()` function:

```
<?php
    $ts = strtotime("2005-07-11 22:16:50 CEST");
?>
```

With initial timestamp:

```
<?php
    $date = strtotime("2005-07-11 22:16:50 CEST");
    $ts = strtotime("next week", $date);
?>
```

The timestamps returned are still 32 bit signed integers as this is all that PHP supports.

Parsing Dates - Take #2

Parsing strings for date time information with the `date_create()` function:

```
<?php
    $ts = date_create("2005-07-11 22:16:50");
?>
```

This function will not return the timestamp as an integer, but instead returns a `Date` object which is a wrapper around a 64 bit integer (with some additional functionality ofcourse).

Formatting using format specifiers:

```
<?php
    date_default_timezone_set("Europe/Oslo");
    $ts = date_create("1979-12-31 09:15");
    echo date_format($ts, "D Y-m-d H:i:s - \I\S\O \W/\Y: W/o"), "<br/>\n";
```

All format modifiers as supported by date() are supported too.

Predefined formats:

```
<?php
    date_default_timezone_set("Europe/Oslo");
    $ts = date_create("December 22nd, 2005 15:41");
    echo date_format($ts, DATE_ISO8601), "<br/>\n";
    echo date_format($ts, DATE_RFC1036), "<br/>\n";
    echo date_format($ts, DATE_RSS), "<br/>\n";
```

Formatting Dates - Locale Aware

```
<?php
date_default_timezone_set("Europe/Oslo");
$ts = date_create("December 22nd, 2005 15:41");

$locales = array('en_US', 'fr_CA', 'nb_NO', 'ru_RU', 'ar_SA', 'ja_JP');
foreach ($locales as $locale) {
    i18n_loc_set_default($locale);
    echo date_format_locale($ts, DATE_RFC1036), "<br/>\n";
}
?>
```

Modifying dates and times:

```
<?php
    date_default_timezone_set("Europe/Oslo");
    $ts = date_create("now");
    echo $ts->format(DATE_RFC2822), "<br/>\n";

    echo $ts->modify("+2 days");
    echo $ts->format(DATE_RFC2822), "<br/>\n";

    echo $ts->modify("third month");
    echo $ts->format(DATE_RFC2822), "<br/>\n";

    echo $ts->modify("Friday +3 weeks");
    echo $ts->format(DATE_RFC2822), "<br/>\n";

    echo $ts->modify("next friday");
    echo $ts->format(DATE_RFC2822), "<br/>\n";
?>
```

Using Timezones

Specifying timezone abbreviation while parsing:

```
<?php
    $ts = date_create("1978-12-22 09:15 CET");
?>
```

Using timezone abbreviations is deprecated, one should always use either a default timezone, or the full identifier.

Specifying timezone identifier while parsing:

```
<?php
    $ts = date_create("1978-12-22 09:15 Europe/Oslo");
?>
```


Default Timezones

Setting a default timezone:

```
<?php
    date_default_timezone_set("Europe/Oslo");
    $ts = date_create("1978-12-22 09:15");
    echo date_format($ts, "e");
?>
```

Getting a default timezone:

```
<?php
    $default_identifier = date_default_timezone_get();
    echo $default_identifier;
?>
```

Default timezone is 'guessed' in the following order:

- `date_default_timezone_set()` value
- TZ environment variable
- php.ini's `date.timezone` setting
- System's rendering of timezone abbreviation

Using Timezones

Creating a timezone resource:

```
<?php
    $tz = timezone_open("Asia/Singapore");
?>
```

Using the timezone when parsing a string with a date representation:

```
<?php
    $tz = timezone_open("Pacific/Honolulu");
    $ts = date_create("1978-12-22 09:15", $tz);
?>
```

A passed timezone object does not override a parsed timezone:

```
<?php
    $tz = timezone_open("Pacific/Honolulu");
    $ts1 = date_create("1978-12-22 09:15 CET", $tz);
    $ts2 = date_create("1978-12-22 09:15 Europe/Amsterdam", $tz);
?>
```

Getting a timezone's name:

```
<?php
    $tz = timezone_open("Asia/Singapore");
    echo timezone_name_get($tz), ', ';

    $tz = timezone_open("CEST");
    echo $tz->getName();

?>
```

Getting the current offset to GMT with a timezone for a specific date:

```
<?php
    $tz = timezone_open("Europe/Amsterdam");
    $d = date_create("2005-01-22 09:15");
    echo timezone_offset_get($tz, $d), ', ';
    $d->modify("+6 months");
    echo $tz->getOffset($d);

?>
```

Using Timezones

Using the timezone when parsing a string with a date representation:

```
<?php
    $tz1 = timezone_open("Pacific/Honolulu");
    $tz2 = timezone_open("Europe/Amsterdam");
    $tz3 = timezone_open("Australia/Melbourne");

    $ts = date_create("1978-12-22 09:15", $tz1);
    echo $ts->getTimezone()->getName(), ': ',
        $ts->format(DATE_RFC822), "<br/>";

    $ts->setTimezone($tz2);
    echo $ts->getTimezone()->getName(), ': ',
        $ts->format(DATE_RFC822), "<br/>";

    date_timezone_set($ts, $tz3);
    echo timezone_name_get(date_timezone_get($ts)), ': ',
        date_format($ts, DATE_RFC822);
?>
```

Creating a timezone resource:

```
<?php
    $tz = timezone_open("Europe/Amsterdam");
    $trs = timezone_transitions_get($tz);
    $trs = $tz->getTransitions();

    echo "<pre>\n";
    foreach ($trs as $tr) {
        printf("%20s %7d %d %s\n",
            $tr['time'], $tr['offset'],
            $tr['isdst'], $tr['abbr']);
    }
?>
```

All supported timezone identifiers:

```
<?php
    $ids = timezone_identifiers_list();
    echo "Number of identifiers: ", count($ids), "<br/>";
    echo implode(", ", array_slice($ids, 0, 5)), '...';
    echo implode(", ", array_slice($ids, -5));
?>
```

All supported timezone identifiers:

```
<?php
    $abbrs = timezone_abbreviations_list();
    echo "<pre>\n";
    foreach ($abbrs as $abbr => $ids) {
        foreach ($ids as $id) {
            printf("%-6s %6d %d %s\n", strtoupper($abbr),
                $id['offset'], $id['dst'], $id['timezone_id']);
        }
    }
?>
```


Timezone Database

- Timezones change quicker than PHP releases
- pecl extension: timezonedb
- Can be used to override the built-in timezones with new rules

date

date/time support	enabled
Timezone Database Version	2006.1
Timezone Database	internal
Default timezone	Europe/Oslo

Directive	Local Value	Master Value
date.default_latitude	31.7667	31.7667
date.default_longitude	35.2333	35.2333
date.sunrise_zenith	90.583333	90.583333
date.sunset_zenith	90.583333	90.583333
date.timezone	Europe/Oslo	Europe/Oslo

Timezone and Date tips

- Store date/time in a timezone agnostic way: GMT
- Use a "Unix" timestamp, perhaps as string: `date("U");`

Sunset/Sunrise

```
<?php
date_default_timezone_set('America/Montreal');
if (isset($_POST['country'])) {
    $client = new SoapClient('http://geo.derickrethans.nl/geo.wsdl');
    $info = $client->Location($_POST['country'], $_POST['city']);
    $lat = $info['lat']; $lon = $info['lon'];
    $info = date_sun_info(time(), $lon, $lat);
    echo "Sunrise is at ", date(date::ATOM, $info['sunrise']), "<br/>";
    echo "Sunset is at ", date(date::RSS, $info['sunset']), "<br/><hr/>";
}
?>
<form method="post">
Country: <input name="country"/> City: <input name="city"/><br/>
<input type="submit" value="Check"/>
</form>
```

When Is The Additional Functionality Available

- In PHP 5.1 if you compile PHP with a special flag
- In PHP 5.2 by default

Compile like:

```
CFLAGS=-DEXPERIMENTAL_DATE_SUPPORT=1 ./configure
```

Resources

This presentation: <http://derickrethans.nl/talks.php>

World timezones: <http://www.worldtimezone.com/index24.php>

Questions?: dr@ez.no