[sha1_file »](#)
[« rtrim](#)

- [Manual de PHP](#)
- [Referencia de funciones](#)
- [Procesamiento de texto](#)
- [Strings](#)
- [Funciones de strings](#)

Change language: Spanish ⌄

[Submit a Pull Request](#) [Report a Bug](#)

setlocale

(PHP 4, PHP 5, PHP 7, PHP 8)

setlocale — Establecer la información del localismo

Descripción¶

setlocale(int $category, string $locale, string $... = ?): string
setlocale(int $category, array $locale): string

Establece la información del localismo.

Parámetros¶

category

> category es una constante con nombre que especifica la categoría de las funciones afectadas por el localismo:
>
> - LC_ALL para establecer todas las siguientes
> - LC_COLLATE para la comparación de cadenas; véase [strcoll()](#)
> - LC_CTYPE para la clasificación y conversión de caracteres, por ejemplo [strtoupper()](#)
> - LC_MONETARY para [localeconv()](#)
> - LC_NUMERIC para el separador decimal (véase también [localeconv()](#))
> - LC_TIME para el formato de fecha y hora con [strftime()](#)
> - LC_MESSAGES para las respuestas del sistema (disponible si PHP fue compilado con libintl)

locale

> Si locale es **null** o el string vacío "", los nombres de los localismos se establecerán a partir de los valores de las variables de entorno con los mismos nombres que las categorías anteriores, o desde "LANG".
>
> Si locale es "0", el localismo no se ve afectado, solamente devuelve la configuración actual.
>
> Si locale es un array o le siguen parámetros adicionales, cada elemento del array o parámetro se intenta establecer como nuevo localismo hasta que se tenga éxito. Esto es útil si un localismo se conoce con diferentes nombres en diferentes sistemas, o para proporcionar un respaldo para un localismo posiblemente no disponible.

...

> (String opcional o array de parámetros para probarlos como localismos hasta que se tenga éxito.)

Nota:

En Windows, setlocale(LC_ALL, '') establece los nombres del localismo desde la configuración regional o del lenguaje del sistema (accesible por medio del Panel de Control).

Valores devueltos ¶

Devuelve el nuevo localismo actual o `false` si la funcionalidad del localismo no está implementada en la plataforma, el localismo especificado no existe, o el nombre de la categoría no es válido.

Un nombre de categoría no válido también produce un mensaje de advertencia. Se pueden encontrar los nombres de las categorías/localismos en la » RFC 1766 y en la » ISO 639. Diferentes sistemas tienen diferentes nomenclaturas para los localismos.

Nota:

El valor de retorno de **setlocale()** depende del sistema donde se está ejecutando PHP. Devuelve exactamente lo que la función setlocale del sistema devuelve.

Historial de cambios ¶

Versión	Descripción
7.0.0	Se ha eliminado el soporte para el parámetro `category` pasado como string. En esta versión, solamente se pueden utilizar las constantes `LC_*`.
5.3.0	Está función ahora emite un aviso `E_DEPRECATED` si se pasa un string al parámetro `category` en lugar de una de las constantes `LC_*`.

Ejemplos ¶

Ejemplo #1 Ejemplos de setlocale()

```php
<?php
/* Establecer el localismo al holandés */
setlocale(LC_ALL, 'nl_NL');

/* Muestra: vrijdag 22 december 1978 */
echo strftime("%A %e %B %Y", mktime(0, 0, 0, 12, 22, 1978));

/* Probar diferentes nombres posibles de localismos para el alemán */
$loc_de = setlocale(LC_ALL, 'de_DE@euro', 'de_DE', 'de', 'ge');
echo "El localismo preferido para el alemán en este sistema es '$loc_de'";
?>
```

Ejemplo #2 Ejemplos para Windows de setlocale()

```php
<?php
/* Establecer el localismo al holandés */
setlocale(LC_ALL, 'nld_nld');

/* Muestra: vrijdag 22 december 1978 */
echo strftime("%A %d %B %Y", mktime(0, 0, 0, 12, 22, 1978));

/* Probar diferentes nombres posibles de localismos para el alemán */
$loc_de = setlocale(LC_ALL, 'de_DE@euro', 'de_DE', 'deu_deu');
echo "El localismo preferido para el alemán en este sistema es '$loc_de'";
?>
```

Notas ¶

Advertencia

La información del localismo se mantiene por proceso, no por hilo. Si se está ejecutanto PHP en una API de servidor multihilo como IIS, HHVM o Apache sobre Windows, se pueden experimentar cambios repentinos en la configuración local mientras se esté ejecutando un script, aunque el propio script nunca llame a **setlocale()**. Esto ocurre debido a que otros scripts ejecutándose en diferentes hilos de un mismo proceso, al mismo tiempo, cambian el localismo de todo el proceso con **setlocale()**.

Sugerencia

Los usuarios de Windows encontrarán información útil sobre string de `locale` en el sitio web MSDN de Microsoft. Los string de los idiomas soportados se pueden encontrar en la [» documentación de string de idiomas](#) y los string de paises/regiones en la [» documentación de string de países/regiones](#).

[+ add a note](#)

User Contributed Notes 33 notes

[up](#)
[down](#)
164
[*r dot nospam dot velseboer at quicknet dot nospam dot nl*](#) ¶
20 years ago
```
be careful with the LC_ALL setting, as it may introduce some unwanted conversions. For example, I used

setlocale (LC_ALL, "Dutch");

to get my weekdays in dutch on the page. From that moment on (as I found out many hours later) my
floating point values from MYSQL where interpreted as integers because the Dutch locale wants a
comma (,) instead of a point (.) before the decimals. I tried printf, number_format, floatval....
all to no avail. 1.50 was always printed as 1.00 :(

When I set my locale to :

setlocale (LC_TIME, "Dutch");

my weekdays are good now and my floating point values too.

I hope I can save some people the trouble of figuring this out by themselves.

Rob
```
[up](#)
[down](#)
52
[*russ at eatmymonkeydust dot com*](#) ¶
10 years ago
```
If you are looking for a getlocale() function simply pass 0 (zero) as the second parameter to
setlocale().

Beware though if you use the category LC_ALL and some of the locales differ as a string containing
all the locales is returned:

<?php
echo setlocale(LC_ALL, 0);
```

```
// LC_CTYPE=en_US.UTF-
8;LC_NUMERIC=C;LC_TIME=C;LC_COLLATE=C;LC_MONETARY=C;LC_MESSAGES=C;LC_PAPER=C;LC_NAME=C;
// LC_ADDRESS=C;LC_TELEPHONE=C;LC_MEASUREMENT=C;LC_IDENTIFICATION=C

echo setlocale(LC_CTYPE, 0);

// en_US.UTF-8

setlocale(LC_ALL, "en_US.UTF-8");
echo setlocale(LC_ALL, 0);

// en_US.UTF-8

?>
```

If you are looking to store and reset the locales you could do something like this:

```
<?php

$originalLocales = explode(";", setlocale(LC_ALL, 0));
setlocale(LC_ALL, "nb_NO.utf8");

// Do something

foreach ($originalLocales as $localeSetting) {
  if (strpos($localeSetting, "=") !== false) {
    list ($category, $locale) = explode("=", $localeSetting);
  }
  else {
    $category = LC_ALL;
    $locale   = $localeSetting;
  }
  setlocale($category, $locale);
}

?>
```

The above works here (Ubuntu Linux) but as the setlocale() function is just wrapping the equivalent system calls, your mileage may vary on the result.
up
down
9
epistomai at gmail dot com ¶
3 years ago
```
setlocale(LC_MONETARY, 'en_US') doesn't work anymore (at least in PHP Version 7.3.8).

I've used 'en_US.UTF-8' instead
```
up
down
16
Shashakhmetov Talgat ¶
7 years ago
```
//Fix encoding for russian locale on windows
$locale = setlocale(LC_ALL, 'ru_RU.CP1251', 'rus_RUS.CP1251', 'Russian_Russia.1251');

function strftime_fix($format, $locale, $timestamp = time()){
    // Fix %e for windows
```

```
    if (strtoupper(substr(PHP_OS, 0, 3)) == 'WIN') {
        $format = preg_replace('#(?<!%)((?:%%)*)%e#', '\1%#d', $format);
    }
    // convert
    $date_str = strftime($format, $timestamp);
    if (stripos($locale, "1251") !== false) {
      return iconv("windows-1251","utf-8", $date_str);
    } elseif (stripos($locale, "1252") !== false) {
      return iconv("windows-1252","utf-8", $date_str);
    } else {
      return $date_str;
    }
}
```

[up](#)
[down](#)
16
[brice/axice/be ¶](#)
13 years ago

```
Pay attention to the syntax.
- UTF8 without dash ('-')
- locale.codeset and not locale-codeset.

Stupid newbie error but worth knowing them when starting with gettext.

<?php
$codeset = "UTF8";  // warning ! not UTF-8 with dash '-'

// for windows compatibility (e.g. xampp) : theses 3 lines are useless for linux systems

putenv('LANG='.$lang.'.'.$codeset);
putenv('LANGUAGE='.$lang.'.'.$codeset);
bind_textdomain_codeset('mydomain', $codeset);

// set locale
bindtextdomain('mydomain', ABSPATH.'/locale/');
setlocale(LC_ALL, $lang.'.'.$codeset);
textdomain('mydomain');
?>

where directory structure of locale is (for example) :
locale/fr_FR/LC_MESSAGES/mydomain.mo
locale/en_US/LC_MESSAGES/mydomain.mo

and ABSPATH is the absolute path to the locale dir

further note, under linux systems, it seems to be necessary to create the locale at os level using
'locale-gen'.
```

[up](#)
[down](#)
19
[Kari Sderholm aka Haprog ¶](#)
13 years ago

```
It took me a while to figure out how to get a Finnish locale correctly set on Ubuntu Server with
Apache2 and PHP5.

At first the output for "locale -a" was this:
C
en_US.utf8
```

```
POSIX

I had to install a finnish language pack with
"sudo apt-get install language-pack-fi-base"

Now the output for "locale -a" is:
C
en_US.utf8
fi_FI.utf8
POSIX
```

The last thing you need to do after installing the correct language pack is restart Apache with "sudo apache2ctl restart". The locale "fi_FI.utf8" can then be used in PHP5 after restarting Apache.

For setting Finnish timezone and locale in PHP use:
```php
<?php
date_default_timezone_set('Europe/Helsinki');
setlocale(LC_ALL, array('fi_FI.UTF-8','fi_FI@euro','fi_FI','finnish'));
?>
```
up
down
9
[pigmeu at pigmeu dot net](¶)
18 years ago
```
!!WARNING!!

The "locale" always depend on the server configuration.

i.e.:
When trying to use "pt_BR" on some servers you will ALWAYS get false. Even with other languages.

The locale string need to be supported by the server. Sometimes there are diferents charsets for a language, like "pt_BR.utf-8" and "pt_BR.iso-8859-1", but there is no support for a _standard_ "pt_BR".

This problem occours in Windows platform too. Here you need to call "portuguese" or "spanish" or "german" or...

Maybe the only way to try to get success calling the function setlocale() is:
setlocale(LC_ALL, "pt_BR", "pt_BR.iso-8859-1", "pt_BR.utf-8", "portuguese", ...);

But NEVER trust on that when making functions like date conversions or number formating. The best way to make sure you are doing the right thing, is using the default "en_US" or "en_UK", by not calling the setlocale() function. Or, make sure that your server support the lang you want to use, with some tests.

Remember that: Using the default locale setings is the best way to "talk" with other applications, like dbs or rpc servers, too.

[]s

Pigmeu
```
up
down
4
[data dot ocean dot italia at gmail dot com](¶)
9 years ago

Instead, using php with IIS, I had to use this line for Italian language...

```php
<?php setlocale(LC_ALL, 'Italian_Italy.1250'); ?>
```
[up](#)
[down](#)
7
[jose dot nobile at gmail dot com ¶](#)
7 years ago
For Windows users complaining about setlocale.

The locale argument to the setlocale function takes the following form:
setlocale(LC_ALL, "<language>_<country>.<code_page>");

in short, if you want use for example: es_CO.UTF-8 it must be in Windows: Spanish_Colombia.1252

The code page 1252 is ISO-8859-1 (windows-1252   ANSI Latin 1; Western European (Windows)

Windows use different languages code from Unix, for example, es_CO becomes es-CO or
Spanish_Colombia, also it doesn't support UTF-8 charset as is shown in their website:
[https://msdn.microsoft.com/en-us/library/x99tb11d(v=vs.140)](#).aspx

"The set of available locale names, languages, country/region codes, and code pages includes all
those supported by the Windows NLS API except code pages that require more than two bytes per
character, such as UTF-7 and UTF-8. If you provide a code page value of UTF-7 or UTF-8, setlocale
will fail, returning NULL."

Please check the updated website of language and code pages:

[https://msdn.microsoft.com/en-us/library/39cwe7zf(v=vs.140)](#).aspx
and
[https://msdn.microsoft.com/en-us//goglobal/bb895996](#)

Here a copy paste in case the link is removed:

Afrikaans
Albanian
Arabic_Saudi_Arabia
Arabic_Iraq
Arabic_Egypt
Arabic_Libya
Arabic_Algeria
Arabic_Morocco
Arabic_Tunisia
Arabic_Oman
Arabic_Yemen
Arabic_Syria
Arabic_Jordan
Arabic_Lebanon
Arabic_Kuwait
Arabic_UAE
Arabic_Bahrain
Arabic_Qatar
Armenian
Azeri_Latin
Azeri_Cyrillic
Basque
Belarusian
Bengali_India

```
Bosnian_Latin
Bulgarian
Catalan
Chinese_Taiwan
Chinese_PRC
Chinese_Hong_Kong
Chinese_Singapore
Chinese_Macau
Croatian
Croatian_Bosnia_Herzegovina
Czech
Danish
Divehi
Dutch_Standard
Dutch_Belgian
English_United_States
English_United_Kingdom
English_Australian
English_Canadian
English_New_Zealand
English_Ireland
English_South_Africa
English_Jamaica
English_Caribbean
English_Belize
English_Trinidad
English_Zimbabwe
English_Philippines
Estonian
Faeroese
Farsi
Finnish
French_Standard
French_Belgian
French_Canadian
French_Swiss
French_Luxembourg
French_Monaco
Georgian
Galician
German_Standard
German_Swiss
German_Austrian
German_Luxembourg
German_Liechtenstein
Greek
Gujarati
Hebrew
Hindi
Hungarian
Icelandic
Indonesian
Italian_Standard
Italian_Swiss
Japanese
Kannada
Kazakh
Konkani
```

```
Korean
Kyrgyz
Latvian
Lithuanian
Macedonian
Malay_Malaysia
Malay_Brunei_Darussalam
Malayalam
Maltese
Maori
Marathi
Mongolian
Norwegian_Bokmal
Norwegian_Nynorsk
Polish
Portuguese_Brazilian
Portuguese_Standard
Punjabi
Quechua_Bolivia
Quechua_Ecuador
Quechua_Peru
Romanian
Russian
Sami_Inari
Sami_Lule_Norway
Sami_Lule_Sweden
Sami_Northern_Finland
Sami_Northern_Norway
Sami_Northern_Sweden
Sami_Skolt
Sami_Southern_Norway
Sami_Southern_Sweden
Sanskrit
Serbian_Latin
Serbian_Latin_Bosnia_Herzegovina
Serbian_Cyrillic
Serbian_Cyrillic_Bosnia_Herzegovina
Slovak
Slovenian
Spanish_Traditional_Sort
Spanish_Mexican
Spanish_Modern_Sort
Spanish_Guatemala
Spanish_Costa_Rica
Spanish_Panama
Spanish_Dominican_Republic
Spanish_Venezuela
Spanish_Colombia
Spanish_Peru
Spanish_Argentina
Spanish_Ecuador
Spanish_Chile
Spanish_Uruguay
Spanish_Paraguay
Spanish_Bolivia
Spanish_El_Salvador
Spanish_Honduras
Spanish_Nicaragua
```

```
Spanish_Puerto_Rico
Swahili
Swedish
Swedish_Finland
Syriac
Tamil
Tatar
Telugu
Thai
Tswana
Ukrainian
Turkish
Ukrainian
Urdu
Uzbek_Latin
Uzbek_Cyrillic
Vietnamese
Welsh
Xhosa
Zulu


The code pages identifiers:


https://msdn.microsoft.com/en-us/library/windows/desktop/dd317756(v=vs.85).aspx
```

up
down
3
Periklis ¶
15 years ago
In *some* Windows systems, setting LC_TIME only will not work, you must either set LC_ALL or both LC_CTYPE and LC_TIME. BUT if you have already set LC_TIME using setlocale earlier in the script, dates will not be affected! For example:
```php
<?php
setlocale(LC_TIME, 'greek');
setlocale(LC_CTYPE, 'greek');
?>
```
will not work, while
```php
<?php
setlocale(LC_CTYPE, 'greek');
setlocale(LC_TIME, 'greek');
?>
```
will do the job.
up
down
2
Sven K ¶
16 years ago
If your system doesn't show any installed locales by "locale -a", try installing them by "dpkg-reconfigure locales" (on debian).
up
down
3
Leigh Morresi ¶
14 years ago
Setting locale that is not supported by your system will result in some string operations returning a question mark "?" in your strings where it needs to perform transliteration.

1) Always check the return of setlocale() to ensure it has set to something supported

2) on Linux you can use the "locale -a" command to find a list of supported locales
[up](up)
down
1
RobQuist ¶
8 years ago
In addition to russ, about getting / backing up the locale:
I'm using this in unit-tests. I wanted to test something based on locale, and reset the locale
after the tests were done.

Yet there were some errors;
* setlocale doesn't like strings anymore. You need to use constants.
* Some contants don't exist anymore.

Here's an updated piece of code:

```php
<?php
$originalLocales = explode(";", setlocale(LC_ALL, 0));
setlocale(LC_ALL, 'nl_NL.UTF-8');

//Do something here

//Recover to the default setting
        $skipConstants = array( //these will be returned by setlocale(LC_ALL, 0), but don't exist
anymore.
            'LC_PAPER',
            'LC_NAME',
            'LC_ADDRESS',
            'LC_TELEPHONE',
            'LC_MEASUREMENT',
            'LC_IDENTIFICATION'
        );

        foreach ($originalLocales as $localeSetting) {
            if (strpos($localeSetting, "=") !== false) {
                list ($category, $locale) = explode("=", $localeSetting);
            } else {
                $category = LC_ALL;
                $locale   = $localeSetting;
            }

            if (!in_array($category, $skipConstants)) {
                setlocale(constant($category), $locale); //Using strings is deprecated.
            }
        }
?>
```
[up](up)
down
3
Anonymous ¶
16 years ago
The example from bruno dot cenou at revues dot org below shows the possibility, but I want to
spell it out: you can add charset info to setlocale.

Example:

Into my utf-8-encoded page I want to insert the name of the current month, which happens to be

March, in German "März" - with umlaut. If you use

```
    setlocale(LC_TIME, 'de_DE');
    echo strftime("%B");
```

this will return "März", but that html-entity will look like this on a utf-8 page: "M?rz".
Not what I want.

But if you use

```
    setlocale(LC_TIME, 'de_DE.UTF8');  // note the charset info !
    echo strftime("%B");
```

this returns "M√§rz", which, on utf-8, looks like it should: "März".
[up](#)
[down](#)
1
[bryn AT lunarvis DOT com](#) ¶
14 years ago
Posting this in the hope it might be useful to others, as I could find very little info anywhere.
If you want to use a Welsh locale and have the suitable language support installed, you pass 'cym'
(abbreviated form of Cymraeg) to setlocale:

```
<?php
setlocale(LC_TIME, 'cym');
$welsh= gmstrftime("%A, %B %Y - %H:%M",time());
echo $welsh;
?>
```

The above certainly applies to Windows systems, but should also apply to Unix if the required
support is installed.

Cheers,

Bryn.
[up](#)
[down](#)
1
[tomas dot hampl at gmail dot com](#) ¶
11 years ago
On Linux, setlocale() depends on the installed locales. To see which locales are available to PHP,
run this from the terminal:

"locale -a"

Provided list are all locales that are available on your server for PHP to use. To add a new one,
run

locale-gen <locale name> (this may need sudo / root permissions), for example to add a Czech
locale, run something like this:

"sudo locale-gen cs_CZ.utf8"

Then you can use this locale declaration:

```
setlocale(LC_ALL, 'cs_CZ.utf8');
```
[up](#)
[down](#)

1
szepeshazi at gmail dot com ¶
15 years ago
For those of you who are unfortunate enough (like me) to work in Windows environment, and try to set the locale to a language _and_ to UTF-8 charset, and were unable to do it, here is a workaround.

For example to output the date in hungarian with UTF-8 charset, this will work:

```
    $dateString = "%B %d., %A";
    setlocale(LC_ALL,'hungarian');
    $res=strftime($dateString);
    echo(iconv('ISO-8859-1', 'UTF-8', $res));
```

If anybody knows how to set the locale on Windows to the equivalent of "hu_HU.UTF-8" on unix, please do tell me.
up
down
1
divinity76 at gmail dot com ¶
5 months ago
if someone is looking for a getlocale(),

```
<?php
if(!function_exists("getlocale")){
    function getlocale(int $category)/*:string|false*/{
        return setlocale($category, 0);
    }
}
?>
```
up
down
1
birkholz at web dot de ¶
17 years ago
When i tried to get the current locale (e.g. after i set the lang to german with setlocale(LC_ALL, 'de_DE');), the following did not work on my suse linux 9.0-box:
$currentLocale = setlocale(LC_ALL, NULL);
This code did a reset to the server-setting.

$currentLocale = setlocale(LC_ALL, 0); works perfectly for me, but the manual says NULL and 0 are equal in this case, but NULL seems to act like "".
up
down
0
lingureanumanuel at yahoo dot com ¶
9 months ago
for windows

```
setlocale(LC_ALL, 'Greenlandic_Greenland.1252');
will return false
```

```
to make it work use
setlocale(LC_ALL, 'Kalaallisut_Greenland.1252');
```
up
down
1
mk at totu dot com ¶

18 years ago

Be carefull - setting a locale which uses commas instead of dots in numbers may cause a mysql db not to understand the query:

```php
<?php
setlocale(LC_ALL,"pl");
$price = 1234 / 100; // now the price looks like 12,34
$query = mysql_query("SELECT Id FROM table WHERE price='".$price."'");
?>
```

Even if there is a price 12.34 - nothing will be found

up
down
1

Un_passant ¶

14 years ago

For debian/ubuntu, don't forget the charset UFT8.


```
// Works on Ubuntu 8.04 Server
setlocale(LC_TIME, 'fr_FR.UTF8', 'fr.UTF8', 'fr_FR.UTF-8', 'fr.UTF-8');
```

up
down
0

tim dot peters at live dot com ¶

3 years ago

Maybe obvious, but I would expect that setlocale constantes (LC_*) would be bitwise, but they're not.

In example, doing this:

```php
<?php
    setlocale(LC_TIME + LC_COLLATE, 'nl');

    echo setlocale(LC_ALL, 0);
?>
```

would cause the following result:

LC_CTYPE=C;LC_NUMERIC=C;LC_TIME=C;LC_COLLATE=C;LC_MONETARY=C;LC_MESSAGES=nl;LC_PAPER=C;
LC_NAME=C;LC_ADDRESS=C;LC_TELEPHONE=C;LC_MEASUREMENT=C;LC_IDENTIFICATION=C

Note that LC_MESSAGES has changed, instead of LC_TIME and LC_COLLATE. (Because LC_TIME + LC_COLLATE = LC_MESSAGES).

Instead you would need to specify them individually, if you don't wish to use LC_ALL:

```php
<?php
    setlocale(LC_TIME, 'nl');
    setlocale(LC_COLLATE, 'nl');

    echo setlocale(LC_ALL, 0);
?> LC_CTYPE=C;LC_NUMERIC=C;LC_TIME=nl;LC_COLLATE=nl;LC_MONETARY=C;LC_MESSAGES=C;LC_PAPER=C;
```
LC_NAME=C;LC_ADDRESS=C;LC_TELEPHONE=C;LC_MEASUREMENT=C;LC_IDENTIFICATION=C

up
down
0

stepdate at gmail dot com ¶

4 years ago

If you have Locales installed and things won't work check the spelling: for German all the comments suggested "setlocale(LC_TIME, "de_DE.utf8")", but it has to be "setlocale(LC_TIME, "de_DE.UTF-8")"-> UTF-8 instead of utf8.

internationalist ¶

7 years ago

My script runs a loop that changes the locale (multilingual application). I've noticed that on some random occasion the locale still hasn't changed despite the setlocale() function being executed a step earlier. I had to add wait time for this condition. Interestingly enough, this was the case only with the 'nl_NL.UTF8' locale.

```php
<?php
//some code
$this->counter = 0;
// some code
$this->locale = 'nl_NL.UTF8';
setlocale(LC_ALL, $this->locale);
    $this->counter++;
    if ($this->locale !== setlocale(LC_CTYPE, 0)) { // Locale not changed yet.
      if ($this->counter > 10) {
        return;
      }
      sleep(1);
    }
// some code
?>
```

aaaaa976 at gmail dot com ¶

7 years ago

In Windows some times setlocale don't work, it return a empty array, buts the locale is set. I found that apache start before windows "load" locales, you must restart apache to solve this.

phcorp ¶

11 years ago

To find the locale of a Unix system:
```php
<?php system('locale -a') ?>
```

leif at neland dot dk ¶

12 years ago

Regarding dash'es in locale, it appears they should be omitted entirely.

In /etc/locale.gen I have

da_DK.ISO-8859-15 ISO-8859-15

but locale -a gives

da_DK.iso885915

which is the format setlocale() wants.

(Debian)

0
garygendron at yahoo dot com ¶
12 years ago
```
For a php Mysql query, you could also use, for french canadian, in this example :


$query = 'SET lc_time_names = "fr_CA"';
$result = mysql_query($query) or die("Query failed");


$query = 'SELECT @@lc_time_names';
$result = mysql_query($query) or die("Query failed");


$query = 'SELECT id, created, YEAR(created) as year, MONTH(created) as month,' .
' CONCAT_WS(" ", MONTHNAME(created), YEAR(created)) as archive' .
' FROM #__TABLE as e' .
' GROUP BY archive' .
' ORDER BY id DESC';


Your data will be displayed in any locale setting you want. You may even $_GET[lc_time_name] from
your multilanguage website.
```
0
michal dot kocarek at brainbox dot cz ¶
14 years ago
```
Note about using UTF-8 locale charset on Windows systems:


According to MSDN, Windows setlocale()'s implementation does not support UTF-8 encoding.


Citation from "MSDN setlocale, _wsetlocale" page (http://msdn.microsoft.com/en-
us/library/x99tb11d.aspx):
The set of available languages, country/region codes, and code pages includes all those supported
by the Win32 NLS API except code pages that require more than two bytes per character, such as
UTF-7 and UTF-8. If you provide a code page like UTF-7 or UTF-8, setlocale will fail, returning
NULL.


So basically, code like
<?php setlocale(LC_ALL, 'Czech_Czech Republic.65001'); // 65001 is UTF-8 codepage ?>
does not work on Windows at all.


(written in time of PHP 5.2.4)
```
-1
mvanbaak ¶
15 years ago
```
To complement Sven K's tip about debian:


You can also install the package locales-all
That one holds all the locales there are in compiled form.
```
-1
ostapk ¶
14 years ago
```
There is a new PECL extension under development called intl (it will be available in PHP5.3).
Meanwhile all who rely on the setlocale() and friends should be aware about the limitations of
them as covered in this post on the onPHP5.com blog: http://www.onphp5.com/article/22
```

up
down
-2
flavioacvalverde at gmail dot com ¶
11 years ago

For Portugal I had to use

```
<?php setlocale(LC_ALL, 'Portuguese_Portugal.1252'); ?>
```

using php with IIS on Windows server.

+ add a note

- Funciones de strings
    - addcslashes
    - addslashes
    - bin2hex
    - chop
    - chr
    - chunk_split
    - convert_uudecode
    - convert_uuencode
    - count_chars
    - crc32
    - crypt
    - echo
    - explode
    - fprintf
    - get_html_translation_table
    - hebrev
    - hex2bin
    - html_entity_decode
    - htmlentities
    - htmlspecialchars_decode
    - htmlspecialchars
    - implode
    - join
    - lcfirst
    - levenshtein
    - localeconv
    - ltrim
    - md5_file
    - md5
    - metaphone
    - money_format
    - nl_langinfo
    - nl2br
    - number_format
    - ord
    - parse_str
    - print
    - printf
    - quoted_printable_decode
    - quoted_printable_encode
    - quotemeta
    - rtrim
    - setlocale
    - sha1_file
    - sha1

- - hebrevc

- [Copyright © 2001-2022 The PHP Group](#)
- [My PHP.net](#)
- [Contact](#)
- [Other PHP.net sites](#)
- [Privacy policy](#)
- [View Source](#)