

# Internationalization

# Introduction

I18n (short for internationalization) are means of adapting an application for different languages and cultures. The scope of i18n support includes support for local date and time, currency, measures, number formatting and strings in different encoding. Native support for i18n and l10n (localization) appeared in PHP6.

Please note that in PHP6, POSIX locales are obsoleted by the set of locales included with the ICU library.

<b>Warning</b>
This extension is still in development and it isn't available to public yet.

# Installing/Configuring

## Requirements

The [Unicode Extension](#).

## Installation

There is no installation needed to use these functions; they are part of the PHP core.

## Runtime Configuration

This extension has no configuration directives defined in *php.ini*.

## Resource Types

This extension has no resource types defined.

# Predefined Constants

This extension has no constants defined.

# i18n Functions

# locale\_get\_default

locale\_get\_default -- Get the default Locale

## Description

string **locale\_get\_default** ( void )

This function returns the default Locale, which is used by PHP to localize certain features. Please note that this isn't influenced by [setlocale\(\)](#) or the system settings.

## Return Values

Returns a [string](#) with the current Locale.

## Examples

### Example #1 - A [locale\\_get\\_default\(\)](#) example

```
<?php

// get the default Locale
echo locale_get_default();

//set a new Locale...
locale_set_default('pt_PT');

// ... and print it
echo locale_get_default();

?>
```

The above example will output:

```
en_US_POSIX
pt_PT
```

## See Also

- [locale\\_set\\_default\(\)](#)

# locale\_set\_default

locale\_set\_default -- Set the default Locale

## Description

bool **locale\_set\_default** ( string \$name )

Sets the default Locale for PHP programs. Please note that this has nothing to do with [setlocale\(\)](#) nor with the system locale.

## Parameters

*name*

The new Locale name. A comprehensive list of the supported locales is available at.

## Return Values

Returns **TRUE** on success or **FALSE** on failure.

## Examples

### Example #2 - A [locale\\_set\\_default\(\)](#) example

This example demonstrates a possible usage of [locale\\_set\\_default\(\)](#) to localize the [sort\(\)](#) functions.

```
<?php

// the list of the strings to sort
$array = array(
    'caramelo',
    'cacto',
    'caçada'
);

// set our locale (Portuguese, in this case)
locale_set_default('pt_PT');

// sort using the locale we previously set
sort($array, SORT_LOCALE_STRING);

print_r($array);
?>
```

The above example will output:

```
Array
(
    [0] => caçada
    [1] => cacto
    [2] => caramelo
)
```

If we didn't use the locale, PHP would sort the string using the ASCII characters value, thus returning (wrongly):

```
Array
(
    [0] => cacto
    [1] => caramelo
    [2] => caçada
)
```

## See Also

- [locale\\_get\\_default\(\)](#)