

C library function - setlocale()

Description

The C library function **char *setlocale(int category, const char *locale)** sets or reads location dependent information.

Declaration

Following is the declaration for setlocale() function.

```
char *setlocale(int category, const char *locale)
```

Parameters

- **category** – This is a named constant specifying the category of the functions affected by the locale setting.
 - **LC_ALL** for all of the below.
 - **LC_COLLATE** for string comparison. See strcoll().
 - **LC_CTYPE** for character classification and conversion. For example – strtoupper().
 - **LC_MONETARY** for monetary formatting for localeconv().
 - **LC_NUMERIC** for decimal separator for localeconv().
 - **LC_TIME** for date and time formatting with strftime().
 - **LC_MESSAGES** for system responses.
- **locale** – If locale is NULL or the empty string "", the locale names will be set from the values of environment variables with the same names as the above categories.

Return Value

A successful call to setlocale() returns an opaque string that corresponds to the locale set. The return value is NULL if the request cannot be honored.

Example

The following example shows the usage of setlocale() function.

```
#include <locale.h>
#include <stdio.h>
```

[Live Demo](#)

```
#include <time.h>

int main () {
    time_t currttime;
    struct tm *timer;
    char buffer[80];

    time( &currttime );
    timer = localtime( &currttime );

    printf("Locale is: %s\n", setlocale(LC_ALL, "en_GB"));
    strftime(buffer,80,"%c", timer );
    printf("Date is: %s\n", buffer);

    printf("Locale is: %s\n", setlocale(LC_ALL, "de_DE"));
    strftime(buffer,80,"%c", timer );
    printf("Date is: %s\n", buffer);

    return(0);
}
```

Let us compile and run the above program that will produce the following result –

```
Locale is: en_GB
Date is: Fri 05 Dec 2014 10:35:02 UTC
Locale is: de_DE
Date is: Fr 05 Dez 2014 10:35:02 UTC
```