NAME setlocale – modify and query a program's locale

SYNOPSIS **#include <locale.h>**

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

MT-LEVEL Safe with exceptions

DESCRIPTION

setlocale( ) selects the appropriate piece of the program's locale as specified by the *category* and *locale* arguments. The *category* argument may have the following values: **LC_CTYPE**, **LC_NUMERIC**, **LC_TIME**, **LC_COLLATE**, **LC_MONETARY**, **LC_MESSAGES**, and **LC_ALL**. These names are defined in the **<locale.h>** header. **LC_ALL** names all of a program's locale categories.

LC_CTYPE affects the behavior of character handling functions such as **isdigit( )** and **tolower( )**, and multibyte character functions such as **mbtowc( )** and **wctomb( )**.

LC_NUMERIC affects the decimal point character and thousands separator character for the formatted input/output functions and string conversion functions.

LC_TIME affects the date and time format as delivered by **ascftime( )**, **cftime( )**, **getdate( )**, and **strftime( )**.

LC_COLLATE affects the sort order produced by **strcoll( )** and **strxfrm( )**.

LC_MONETARY affects the monetary formatted information returned by **localeconv( )**.

LC_MESSAGES affects the behavior of **dgettext( )**, **gettext( )**, and **gettxt( )**.

Each category corresponds to a set of databases which contain the relevant information for each defined locale. The location of a database is given by the following path, **/usr/lib/locale/***locale*/*category*, where *locale* and *category* are the names of locale and category, respectively. For example, the database for the **LC_CTYPE** category of the **de** (Deutsch or German) locale would be found in **/usr/lib/locale/de/LC_CTYPE**

A value of "C" for *locale* specifies the traditional UNIX system behavior. At program startup, the equivalent of

setlocale(LC_ALL, "C")

is executed. This has the effect of initializing each category to the locale described by the environment "C".

A value of "" for *locale* specifies that the locale should be taken from environment variables. The order in which the environment variables are checked for the various categories is given below:

Category	1st Env. Var.	2nd Env. Var.	3rd Env. Var
LC_CTYPE:	**LC_ALL**	**LC_CTYPE**	**LANG**
LC_COLLATE:	**LC_ALL**	**LC_COLLATE**	**LANG**
LC_TIME:	**LC_ALL**	**LC_TIME**	**LANG**
LC_NUMERIC:	**LC_ALL**	**LC_NUMERIC**	**LANG**
LC_MONETARY:	**LC_ALL**	**LC_MONETARY**	**LANG**
LC_MESSAGES:	**LC_ALL**	**LC_MESSAGES**	**LANG**

If a pointer to a string is given for *locale*, **setlocale( )** attempts to set the locale for the given category to *locale*. If **setlocale( )** succeeds, *locale* is returned. If **setlocale( )** fails, a null pointer is returned and the program's locale is not changed.

For category **LC_ALL**, the behavior is slightly different. If a pointer to a string is given for *locale* and **LC_ALL** is given for *category*, **setlocale( )** attempts to set the locale for all the categories to *locale*. The *locale* may be a simple locale, consisting of a single locale, or a composite locale. A composite locale is a string returned by a prior call to **setlocale(LC_ALL,0)**. This string will restore each category to the previous locale. If **setlocale( )** fails to set the locale for any category, a null pointer is returned and the program's locale for all categories is not changed. Otherwise, locale is returned.

A null pointer for *locale* causes **setlocale( )** to return the current locale associated with the *category*. The program's locale is not changed.

FILES

/usr/lib/locale/*locale***/LC_CTYPE**	character type database for *locale*
/usr/lib/locale/*locale***/LC_NUMERIC**	numeric format data for *locale*
/usr/lib/locale/*locale***/LC_TIME**	date and time formats for *locale*
/usr/lib/locale/*locale***/LC_COLLATE**	sort ordering information for *locale*
/usr/lib/locale/*locale***/LC_MESSAGES**	message catalogs for *locale*
/usr/lib/locale/*locale***/LC_MONETARY**	currency format data for *locale*

SEE ALSO

ctype(3C), **localeconv**(3C), **mbchar**(3C), **strcoll**(3C), **strftime**(3C), **gettext**(3I), **environ**(5)

NOTES

To change locale in a multi-thread application **setlocale** should be called prior to using any locale sensitive routine. Using **setlocale** to query the current locale is safe and can be used anywhere in a multi-thread application.

It is the user's responsibility to ensure that mixed locale categories are compatible. For example, setting **LC_CTYPE=C** and **LC_TIME=ja** (where **ja** indicates Japanese) will not work, because Japanese time cannot be represented in the "C" locale's ASCII codeset.