NAME localeconv – get numeric formatting information

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

struct lconv *localeconv(void);

MT-LEVEL Safe with exceptions

DESCRIPTION **localeconv( )** sets the components of an object with type **struct lconv** (defined in **<locale.h>**) with the values appropriate for the formatting of numeric quantities (monetary and otherwise) according to the rules of the current locale (see **setlocale**(3C)). The definition of **struct lconv** is given below (the values for the fields in the "C" locale are given in comments).

```
char *decimal_point;        /* "." */
char *thousands_sep;        /* "" (zero length string) */
char *grouping;             /* "" */
char *int_curr_symbol;      /* "" */
char *currency_symbol;      /* "" */
char *mon_decimal_point;    /* "" */
char *mon_thousands_sep;    /* "" */
char *mon_grouping;         /* "" */
char *positive_sign;        /* "" */
char *negative_sign;        /* "" */
char int_frac_digits;       /* CHAR_MAX */
char frac_digits;           /* CHAR_MAX */
char p_cs_precedes;         /* CHAR_MAX */
char p_sep_by_space;        /* CHAR_MAX */
char n_cs_precedes;         /* CHAR_MAX */
char n_sep_by_space;        /* CHAR_MAX */
char p_sign_posn;           /* CHAR_MAX */
char n_sign_posn;           /* CHAR_MAX */
```

The members of the structure with type **char** * are strings, any of which (except **decimal_point**) can point to a null string (""), to indicate that the value is not available in the current locale or is of zero length. The members with type **char** are nonnegative numbers, any of which can be **CHAR_MAX** (defined in the **<limits.h>** header) to indicate that the value is not available in the current locale. The members are the following:

char *decimal_point
The decimal-point character used to format non-monetary quantities.

char *thousands_sep
The character used to separate groups of digits to the left of the decimal-point character in formatted non-monetary quantities.

char *grouping
A string in which each element is taken as an integer that indicates the number of digits that comprise the current group in a formatted non-monetary quantity. The elements of **grouping** are interpreted according to the following:

CHAR_MAX	No further grouping is to be performed.
0	The previous element is to be repeatedly used for the remainder of the digits.
other	The value is the number of digits that comprise the current group. The next element is examined to determine the size of the next group of digits to the left of the current group.

char *int_curr_symbol
The international currency symbol applicable to the current locale, left-justified within a four-character space-padded field. The character sequences should match with those specified in *ISO 4217 Codes for the Representation of Currency and Funds*.

char *currency_symbol
The local currency symbol applicable to the current locale.

char *mon_decimal_point
The decimal point used to format monetary quantities.

char *mon_thousands_sep
The separator for groups of digits to the left of the decimal point in formatted monetary quantities.

char *mon_grouping
A string in which each element is taken as an integer that indicates the number of digits that comprise the current group in a formatted monetary quantity. The elements of **mon_grouping** are interpreted according to the rules described under **grouping**.

char *positive_sign
The string used to indicate a nonnegative-valued formatted monetary quantity.

char *negative_sign
The string used to indicate a negative-valued formatted monetary quantity.

char int_frac_digits
The number of fractional digits (those to the right of the decimal point) to be displayed in an internationally formatted monetary quantity.

char frac_digits
The number of fractional digits (those to the right of the decimal point) to be displayed in a formatted monetary quantity.

char p_cs_precedes
Set to 1 or 0 if the **currency_symbol** respectively precedes or succeeds the value for a nonnegative formatted monetary quantity.

char p_sep_by_space
Set to 1 or 0 if the **currency_symbol** respectively is or is not separated by a space from the value for a nonnegative formatted monetary quantity.

char n_cs_precedes
Set to 1 or 0 if the **currency_symbol** respectively precedes or succeeds the value for a negative formatted monetary quantity.

char n_sep_by_space
Set to 1 or 0 if the **currency_symbol** respectively is or is not separated by a space from the value for a negative formatted monetary quantity.

char p_sign_posn
Set to a value indicating the positioning of the **positive_sign** for a nonnegative formatted monetary quantity. The value of **p_sign_posn** is interpreted according to the following:

0 Parentheses surround the quantity and **currency_symbol**.

1 The sign string precedes the quantity and **currency_symbol**.

2 The sign string succeeds the quantity and **currency_symbol**.

3 The sign string immediately precedes the **currency_symbol**.

4 The sign string immediately succeeds the **currency_symbol**.

char n_sign_posn
Set to a value indicating the positioning of the **negative_sign** for a negative formatted monetary quantity. The value of **n_sign_posn** is interpreted according to the rules described under **p_sign_posn**.

RETURN VALUES

localeconv( ) returns a pointer to the filled-in object. The structure pointed to by the return value may be overwritten by a subsequent call to **localeconv( )**.

EXAMPLES

The following table illustrates the rules used by four countries to format monetary quantities.

Country	Positive format	Negative format	International format
Italy	L.1.234	-L.1.234	ITL.1.234
Netherlands	F 1.234,56	F -1.234,56	NLG 1.234,56
Norway	kr1.234,56	kr1.234,56-	NOK 1.234,56
Switzerland	SFrs.1,234.56	SFrs.1,234.56C	CHF 1,234.56

For these four countries, the respective values for the monetary members of the structure returned by **localeconv** are as follows:

	Italy	Netherlands	Norway	Switzerland
int_curr_symbol	"ITL."	"NLG "	"NOK "	"CHF "
currency_symbol	"L."	"F"	"kr"	"SFrs."
mon_decimal_point	""	","	","	"."
mon_thousands_sep	"."	"."	"."	","
mon_grouping	"\3"	"\3"	"\3"	"\3"
positive_sign	""	""	""	""

negative_sign	"-"	"-"	"-"	"C"
int_frac_digits	0	2	2	2
frac_digits	0	2	2	2
p_cs_precedes	1	1	1	1
p_sep_by_space	0	1	0	0
n_cs_precedes	1	1	1	1
n_sep_by_space	0	1	0	0
p_sign_posn	1	1	1	1
n_sign_posn	1	4	2	2

FILES

/usr/lib/locale/*locale***/LC_MONETARY/monetary**
LC_MONETARY database for *locale*

/usr/lib/locale/*locale***/LC_NUMERIC/numeric**
LC_NUMERIC database for *locale*

SEE ALSO

chrtbl(1M), **montbl**(1M), **setlocale**(3C)

NOTES

localeconv( ) can be used safely in a multi-thread application, as long as **setlocale**(3C) is not being called to change the locale.

LC_MONETARY
Determines how monetary formats are handled. In the "C" locale, monetary handling follows the U.S. rules.

LC_NUMERIC
Determines how numeric formats are handled. In the "C" locale, numeric handling follows the U.S. rules.