



Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'localeconv.3p' command

\$ man localeconv.3p

LOCALECONV(3P) POSIX Programmer's Manual LOCALECONV(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

localeconv ? return locale-specific information

SYNOPSIS

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

```
struct lconv *localeconv(void);
```

DESCRIPTION

The functionality described on this reference page is aligned with the ISO C standard. Any conflict between the requirements described here and the ISO C standard is unintentional. This volume of POSIX.1?2017 defers to the ISO C standard.

The localeconv() function shall set the components of an object with the type struct lconv with the values appropriate for the formatting of numeric quantities (monetary and otherwise) according to the rules of the current locale.

The members of the structure with type char * are pointers to strings, any of which (except decimal_point) can point to "", to indicate that the value is not available in the current locale or is of zero length.

The members with type `char` are non-negative numbers, any of which can be `{CHAR_MAX}` to indicate that the value is not available in the current locale.

The members include the following:

`char *decimal_point`

The radix character used to format non-monetary quantities.

`char *thousands_sep`

The character used to separate groups of digits before the decimal-point character in formatted non-monetary quantities.

`char *grouping`

A string whose elements taken as one-byte integer values indicate the size of each group of digits in formatted non-monetary quantities.

`char *int_curr_symbol`

The international currency symbol applicable to the current locale. The first three characters contain the alphabetic international currency symbol in accordance with those specified in the ISO 4217:2001 standard. The fourth character (immediately preceding the null byte) is the character used to separate the international currency symbol from the monetary quantity.

`char *currency_symbol`

The local currency symbol applicable to the current locale.

`char *mon_decimal_point`

The radix character used to format monetary quantities.

`char *mon_thousands_sep`

The separator for groups of digits before the decimal-point in formatted monetary quantities.

`char *mon_grouping`

A string whose elements taken as one-byte integer values indicate the size of each group of digits in formatted monetary quantities.

`char *positive_sign`

The string used to indicate a non-negative valued formatted monetary quantity.

tary 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 after the decimal-point) to be displayed in an internationally formatted monetary quantity.

char frac_digits

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

char p_cs_precedes

Set to 1 if the currency_symbol precedes the value for a non-negative formatted monetary quantity. Set to 0 if the symbol succeeds the value.

char p_sep_by_space

Set to a value indicating the separation of the currency_symbol, the sign string, and the value for a non-negative formatted monetary quantity.

char n_cs_precedes

Set to 1 if the currency_symbol precedes the value for a negative formatted monetary quantity. Set to 0 if the symbol succeeds the value.

char n_sep_by_space

Set to a value indicating the separation of the currency_symbol, the sign string, and 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 non-negative formatted monetary quantity.

char n_sign_posn

Set to a value indicating the positioning of the negative_sign for a negative formatted monetary quantity.

char int_p_cs_precedes

Set to 1 or 0 if the int_curr_symbol respectively precedes or succeeds the value for a non-negative internationally formatted monetary quantity.

char int_n_cs_precedes

Set to 1 or 0 if the int_curr_symbol respectively precedes or succeeds the value for a negative internationally formatted monetary quantity.

char int_p_sep_by_space

Set to a value indicating the separation of the int_curr_symbol, the sign string, and the value for a non-negative internationally formatted monetary quantity.

char int_n_sep_by_space

Set to a value indicating the separation of the int_curr_symbol, the sign string, and the value for a negative internationally formatted monetary quantity.

char int_p_sign_posn

Set to a value indicating the positioning of the positive_sign for a non-negative internationally formatted monetary quantity.

char int_n_sign_posn

Set to a value indicating the positioning of the negative_sign for a negative internationally formatted monetary quantity.

The elements of grouping and mon_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 integer 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 before the current group.

The values of p_sep_by_space, n_sep_by_space, int_p_sep_by_space, and int_n_sep_by_space are interpreted according to the following:

- 0 No space separates the currency symbol and value.
- 1 If the currency symbol and sign string are adjacent, a space separates them from the value; otherwise, a space separates the currency symbol from the value.
- 2 If the currency symbol and sign string are adjacent, a space separates them; otherwise, a space separates the sign string from the value.

For `int_p_sep_by_space` and `int_n_sep_by_space`, the fourth character of `int_curr_symbol` is used instead of a space.

The values of `p_sign_posn`, `n_sign_posn`, `int_p_sign_posn`, and `int_n_sign_posn` are interpreted according to the following:

- 0 Parentheses surround the quantity and currency_symbol or `int_curr_symbol`.
- 1 The sign string precedes the quantity and currency_symbol or `int_curr_symbol`.
- 2 The sign string succeeds the quantity and currency_symbol or `int_curr_symbol`.
- 3 The sign string immediately precedes the currency_symbol or `int_curr_symbol`.
- 4 The sign string immediately succeeds the currency_symbol or `int_curr_symbol`.

The implementation shall behave as if no function in this volume of POSIX.1?2017 calls `localeconv()`.

The `localeconv()` function need not be thread-safe.

RETURN VALUE

The `localeconv()` function shall return a pointer to the filled-in object. The application shall not modify the structure to which the return value points, nor any storage areas pointed to by pointers within the structure. The returned pointer, and pointers within the structure, might be invalidated or the structure or the storage areas might be overwritten by a subsequent call to `localeconv()`. In addition, the returned pointer, and pointers within the structure, might be invalidated or the structure or the storage areas might be overwritten by subse?


```

?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 ?
?int_p_cs_precedes ? 1 ? 1 ? 1 ? 1 ?
?int_n_cs_precedes ? 1 ? 1 ? 1 ? 1 ?
?int_p_sep_by_space ? 0 ? 0 ? 0 ? 0 ?
?int_n_sep_by_space ? 0 ? 0 ? 0 ? 0 ?
?int_p_sign_posn ? 1 ? 1 ? 1 ? 1 ?
?int_n_sign_posn ? 1 ? 4 ? 4 ? 2 ?
????????????????????????????????????????????????????????????????????

```

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

fprintf(), fscanf(), isalpha(), isascii(), nl_langinfo(), setlocale(),
strcat(), strchr(), strcmp(), strcoll(), strcpy(), strftime(),
strlen(), strpbrk(), strspn(), strtok(), strxfrm(), strtod(), uselo?
cale()

The Base Definitions volume of POSIX.1?2017, <langinfo.h>, <locale.h>

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard

is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html> .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.kernel.org/doc/man-pages/reporting_bugs.html .

IEEE/The Open Group

2017

LOCALECONV(3P)