NAME

nl_langinfo, nl_langinfo_l − query language and locale information

SYNOPSIS

#include <langinfo.h>

char *nl_langinfo(nl_item *item***);**

char *nl_langinfo_l(nl_item *item***, locale_t** *locale***);**

Feature Test Macro Requirements for glibc (see **feature_test_macros**(7)):

nl_langinfo_l():
Since glibc 2.24:
_POSIX_C_SOURCE >= 200809L
Glibc 2.23 and earlier:
_POSIX_C_SOURCE >= 200112L

DESCRIPTION

The **nl_langinfo**() and **nl_langinfo_l**() functions provide access to locale information in a more flexible way than **localeconv**(3). **nl_langinfo**() returns a string which is the value corresponding to *item* in the program's current global locale. **nl_langinfo**() returns a string which is the value corresponding to *item* for the locale identified by the locale object *locale*, which was previously created by **newlocale**(1). Individual and additional elements of the locale categories can be queried. **setlocale**(3) needs to be executed with proper arguments before.

Examples for the locale elements that can be specified in *item* using the constants defined in *<langinfo.h>* are:

CODESET (LC_CTYPE)
Return a string with the name of the character encoding used in the selected locale, such as "UTF-8", "ISO-8859-1", or "ANSI_X3.4-1968" (better known as US-ASCII). This is the same string that you get with "locale charmap". For a list of character encoding names, try "locale −m" (see **locale**(1)).

D_T_FMT (LC_TIME)
Return a string that can be used as a format string for **strftime**(3) to represent time and date in a locale-specific way.

D_FMT (LC_TIME)
Return a string that can be used as a format string for **strftime**(3) to represent a date in a locale-specific way.

T_FMT (LC_TIME)
Return a string that can be used as a format string for **strftime**(3) to represent a time in a locale-specific way.

DAY_{1–7} (LC_TIME)
Return name of the *n*-th day of the week. [Warning: this follows the US convention DAY_1 = Sunday, not the international convention (ISO 8601) that Monday is the first day of the week.]

ABDAY_{1–7} (LC_TIME)
Return abbreviated name of the *n*-th day of the week.

MON_{1–12} (LC_TIME)
Return name of the *n*-th month.

ABMON_{1–12} (LC_TIME)
Return abbreviated name of the *n*-th month.

RADIXCHAR (LC_NUMERIC)
Return radix character (decimal dot, decimal comma, etc.).

THOUSEP (LC_NUMERIC)
Return separator character for thousands (groups of three digits).

YESEXPR (LC_MESSAGES)
Return a regular expression that can be used with the **regex**(3) function to recognize a positive response to a yes/no question.

NOEXPR (LC_MESSAGES)
Return a regular expression that can be used with the **regex**(3) function to recognize a negative response to a yes/no question.

CRNCYSTR (LC_MONETARY)
Return the currency symbol, preceded by "−" if the symbol should appear before the value, "+" if the symbol should appear after the value, or "." if the symbol should replace the radix character.

The above list covers just some examples of items that can be requested. For a more detailed list, consult *The GNU C Library Reference Manual*.

RETURN VALUE

On success, these functions return a pointer to a string which is the value corresponding to *item* in the specified locale.

If no locale has been selected by **setlocale**(3) for the appropriate category, **nl_langinfo**() return a pointer to the corresponding string in the "C" locale. The same is true of **nl_langinfo_l**() if *locale* specifies a locale where *langinfo* data is not defined.

If *item* is not valid, a pointer to an empty string is returned.

The pointer returned by these functions may point to static data that may be overwritten, or the pointer itself may be invalidated, by a subsequent call to **nl_langinfo**(), **nl_langinfo_l**(), or **setlocale**(3). The same statements apply to **nl_langinfo_l**() if the locale object referred to by *locale* is freed or modified by **freelocale**(3) or **newlocale**(3).

POSIX specifies that the application may not modify the string returned by these functions.

ATTRIBUTES

For an explanation of the terms used in this section, see **attributes**(7).

Interface	Attribute	Value
nl_langinfo()	Thread safety	MT-Safe locale

CONFORMING TO

POSIX.1-2001, POSIX.1-2008, SUSv2.

NOTES

The behavior of **nl_langinfo_l**() is undefined if *locale* is the special locale object **LC_GLOBAL_LOCALE** or is not a valid locale object handle.

EXAMPLE

The following program sets the character type and the numeric locale according to the environment and queries the terminal character set and the radix character.

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

int
main(int argc, char *argv[])
{
    setlocale(LC_CTYPE, "");
    setlocale(LC_NUMERIC, "");
```

```
        printf("%s\n", nl_langinfo(CODESET));
        printf("%s\n", nl_langinfo(RADIXCHAR));

        exit(EXIT_SUCCESS);
    }
```

SEE ALSO

locale(1), **localeconv**(3), **setlocale**(3), **charsets**(7), **locale**(7)

The GNU C Library Reference Manual

COLOPHON

This page is part of release 5.02 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at https://www.kernel.org/doc/man−pages/.