

NAME

xlocale - Thread-safe extended locale support

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <xlocale.h>

DESCRIPTION

The extended locale support includes a set of functions for setting thread-local locales, as well as convenience functions for performing locale-aware calls with a specified locale.

The core of the xlocale API is the *locale_t* type. This is an opaque type encapsulating a locale. Instances of this can be either set as the locale for a specific thread or passed directly to the *_l* suffixed variants of various standard C functions. Two special *locale_t* values are available:

- NULL refers to the current locale for the thread, or to the global locale if no locale has been set for this thread.
- LC_GLOBAL_LOCALE refers to the global locale.

The global locale is the locale set with the `setlocale(3)` function.

SEE ALSO

`duplocale(3)`, `freelocale(3)`, `localeconv(3)`, `newlocale(3)`, `querylocale(3)`, `uselocale(3)`

CONVENIENCE FUNCTIONS

The xlocale API includes a number of *_l* suffixed convenience functions. These are variants of standard C functions that have been modified to take an explicit *locale_t* parameter as the final argument or, in the case of variadic functions, as an additional argument directly before the format string. Each of these functions accepts either NULL or LC_GLOBAL_LOCALE. In these functions, NULL refers to the C locale, rather than the thread's current locale. If you wish to use the thread's current locale, then use the unsuffixed version of the function.

These functions are exposed by including <xlocale.h> *after* including the relevant headers for the standard variant. For example, the `strtol_l(3)` function is exposed by including <xlocale.h> after <stdlib.h>, which defines `strtol(3)`.

For reference, a complete list of the locale-aware functions that are available in this form, along with the

headers that expose them, is provided here:

- <wctype.h>* iswalnum_l(3), iswalphal_l(3), iswcntrl_l(3), iswctype_l(3), iswdigit_l(3), iswgraph_l(3), iswlower_l(3), iswprint_l(3), iswpunct_l(3), iswspace_l(3), iswupper_l(3), iswxdigit_l(3), tolower_l(3), towupper_l(3), wctype_l(3),
- <ctype.h>* digitoint_l(3), isalnum_l(3), isalpha_l(3), isblank_l(3), iscntrl_l(3), isdigit_l(3), isgraph_l(3), ishexnumber_l(3), isideogram_l(3), islower_l(3), isnumber_l(3), isphonogram_l(3), isprint_l(3), ispunct_l(3), isrune_l(3), isspace_l(3), isspecial_l(3), isupper_l(3), isxdigit_l(3), tolower_l(3), toupper_l(3)
- <inttypes.h>* strtoumax_l(3), strtoumax_l(3), wctoumax_l(3), wctoumax_l(3)
- <langinfo.h>* nl_langinfo_l(3)
- <monetary.h>* strfmon_l(3)
- <stdio.h>* asprintf_l(3), fprintf_l(3), fscanf_l(3), printf_l(3), scanf_l(3), snprintf_l(3), sprintf_l(3), sscanf_l(3), vasprintf_l(3), vfprintf_l(3), vfscanf_l(3), vprintf_l(3), vscanf_l(3), vsnprintf_l(3), vsprintf_l(3), vsscanf_l(3)
- <stdlib.h>* atof_l(3), atoi_l(3), atol_l(3), atoll_l(3), mblen_l(3), mbstowcs_l(3), mbtowc_l(3), strtod_l(3), strtol_l(3), strtold_l(3), strtoll_l(3), strtoul_l(3), strtoull_l(3), wctombs_l(3), wctomb_l(3)
- <string.h>* strcoll_l(3), strxfrm_l(3), strcasecmp_l(3), strcasestr_l(3), strncasecmp_l(3)
- <time.h>* strftime_l(3) strptime_l(3)
- <wchar.h>* btowc_l(3), fgetwc_l(3), fgetws_l(3), fputwc_l(3), fputws_l(3), fwprintf_l(3), fwscanf_l(3), getwc_l(3), getwchar_l(3), mbrlen_l(3), mbrtowc_l(3), mbsinit_l(3), mbsnrtowcs_l(3), mbsrtowcs_l(3), putwc_l(3), putwchar_l(3), swprintf_l(3), swscanf_l(3), ungetwc_l(3), vfwprintf_l(3), vfwscanf_l(3), vswprintf_l(3), vswscanf_l(3), vwprintf_l(3), vwscanf_l(3), wctomb_l(3), wcsoll_l(3), wcsftime_l(3), wcsnrtombs_l(3), wcsrtombs_l(3), wcstod_l(3), wcstof_l(3), wcstol_l(3), wcstold_l(3), wcstoll_l(3), wcstoul_l(3), wcstoull_l(3), wcswidth_l(3), wcsxfrm_l(3), wctob_l(3), wcwidth_l(3), wprintf_l(3), wscanf_l(3)
- <wctype.h>* iswblank_l(3), iswhexnumber_l(3), iswideogram_l(3), iswnumber_l(3), iswphonogram_l(3), iswrune_l(3), iswspecial_l(3), nextwctype_l(3), towctrans_l(3),

wctrans_l(3)

<locale.h> localeconv_l(3)

STANDARDS

The functions conform to IEEE Std 1003.1-2008 ("POSIX.1").

HISTORY

The xlocale APIs first appeared in Darwin 8.0. This implementation was written by David Chisnall, under sponsorship from the FreeBSD Foundation and first appeared in FreeBSD 9.1.

CAVEATS

The setlocale(3) function, and others in the family, refer to the global locale. Other functions that depend on the locale, however, will take the thread-local locale if one has been set. This means that the idiom of setting the locale using setlocale(3), calling a locale-dependent function, and then restoring the locale will not have the expected behavior if the current thread has had a locale set using uselocale(3). You should avoid this idiom and prefer to use the *_l* suffixed versions instead.