

**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.