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 'newlocale.3p' command

$ man newlocale.3p

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

newlocale ? create or modify a locale object

SYNOPSIS

```
#include <locale.h>
locale_t newlocale(int category_mask, const char *locale,
    locale_t base);
```

DESCRIPTION

The newlocale() function shall create a new locale object or modify an existing one. If the base argument is (locale_t)0, a new locale object shall be created. It is unspecified whether the locale object pointed to by base shall be modified, or freed and a new locale object created.

The category_mask argument specifies the locale categories to be set or modified. Values for category_mask shall be constructed by a bitwise-inclusive OR of the symbolic constants LC_CTYPE_MASK, LC_NUMERIC_MASK, LC_TIME_MASK, LC_COLLATE_MASK, LC_MONETARY_MASK, and LC_MESSAGES_MASK, or any of the implementation-defined mask values defined in <locale.h>.

For each category with the corresponding bit set in category_mask the

data from the locale named by locale shall be used. In the case of mod?
ifying an existing locale object, the data from the locale named by lo?
cale shall replace the existing data within the locale object. If a
completely new locale object is created, the data for all sections not
requested by category_mask shall be taken from the default locale.
The following preset values of locale are defined for all settings of
category_mask:

"POSIX" Specifies the minimal environment for C-language transla?
tion called the POSIX locale.

"C" Equivalent to "POSIX".

"" Specifies an implementation-defined native environment.
This corresponds to the value of the associated environment
variables, LC_* and LANG; see the Base Definitions volume
of POSIX.1?2017, Chapter 7, Locale and Chapter 8, Environ?
ment Variables.

If the base argument is not (locale_t)0 and the newlocale() function
call succeeds, the contents of base are unspecified. Applications shall
ensure that they stop using base as a locale object before calling
newlocale(). If the function call fails and the base argument is not
(locale_t)0, the contents of base shall remain valid and unchanged.
The behavior is undefined if the base argument is the special locale
object LC_GLOBAL_LOCALE, or is not a valid locale object handle and is
not (locale_t)0.

RETURN VALUE

Upon successful completion, the newlocale() function shall return a
handle which the caller may use on subsequent calls to duplocale(),
freelocale(), and other functions taking a locale_t argument.
Upon failure, the newlocale() function shall return (locale_t)0 and set
errno to indicate the error.

ERRORS

The newlocale() function shall fail if:

ENOMEM There is not enough memory available to create the locale object
or load the locale data.

EINVAL The category_mask contains a bit that does not correspond to a valid category.

ENOENT For any of the categories in category_mask, the locale data is not available.

The newlocale() function may fail if:

EINVAL The locale argument is not a valid string pointer.

The following sections are informative.

EXAMPLES

Constructing a Locale Object from Different Locales

The following example shows the construction of a locale where the LC_CTYPE category data comes from a locale loc1 and the LC_TIME cate? gory data from a locale loc2:

```
#include <locale.h>
...
locale_t loc, new_loc;
/* Get the "loc1" data. */
loc = newlocale (LC_CTYPE_MASK, "loc1", (locale_t)0);
if (loc == (locale_t) 0)
    abort ();
/* Get the "loc2" data. */
new_loc = newlocale (LC_TIME_MASK, "loc2", loc);
if (new_loc != (locale_t) 0)
    /* We don t abort if this fails. In this case this
       simply used to unchanged locale object. */
    loc = new_loc;
...
```

Freeing up a Locale Object

The following example shows a code fragment to free a locale object created by newlocale():

```
#include <locale.h>
...
/* Every locale object allocated with newlocale() should be
 * freed using freelocale():
```

```
 */
locale_t loc;
/* Get the locale. */
loc = newlocale (LC_CTYPE_MASK | LC_TIME_MASK, "locname", (locale_t)0);
/* ... Use the locale object ... */
...
/* Free the locale object resources. */
freelocale (loc);
```

APPLICATION USAGE

Handles for locale objects created by the newlocale() function should either be released by a corresponding call to freelocale(), or be used as a base locale to another newlocale() call.

The special locale object LC_GLOBAL_LOCALE must not be passed for the base argument, even when returned by the uselocale() function.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

duplocale(), freelocale(), uselocale()

The Base Definitions volume of POSIX.1?2017, Chapter 7, Locale, Chapter 8, Environment Variables, <locale.h>

COPYRIGHT

most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.ker?nel.org/doc/man-pages/reporting_bugs.html .