NAME

Locale::gettext - message handling functions

SYNOPSIS

```
use Locale::gettext;
use POSIX; # Needed for setlocale()

setlocale(LC_MESSAGES, "");

# OO interface
my $d = Locale::gettext->domain("my_program");

print $d->get("Welcome to my program"), "\n";
# (printed in the local language)

# Direct access to C functions
textdomain("my_program");

print gettext("Welcome to my program"), "\n";
# (printed in the local language)
```

DESCRIPTION

The gettext module permits access from perl to the *gettext()* family of functions for retrieving message strings from databases constructed to internationalize software.

`$d` = Locale::gettext->domain(DOMAIN)
`$d` = Locale::gettext->domain_raw(DOMAIN)

Creates a new object for retrieving strings in the domain **DOMAIN** and returns it. `domain` requests that strings be returned as Perl strings (possibly with wide characters) if possible while `domain_raw` requests that octet strings directly from functions like `dgettext()`.

`$d`->get(MSGID)

Calls `dgettext()` to return the translated string for the given **MSGID**.

`$d`->cget(MSGID, CATEGORY)

Calls `dcgettext()` to return the translated string for the given **MSGID** in the given **CATEGORY**.

`$d`->nget(MSGID, MSGID_PLURAL, N)

Calls `dngettext()` to return the translated string for the given **MSGID** or **MSGID_PLURAL** depending on **N**.

`$d`->ncget(MSGID, MSGID_PLURAL, N, CATEGORY)

Calls `dngettext()` to return the translated string for the given **MSGID** or **MSGID_PLURAL** depending on **N** in the given **CATEGORY**.

`$d`->dir([NEWDIR])

If **NEWDIR** is given, calls `bindtextdomain` to set the name of the directory where messages for the domain represented by `$d` are found. Returns the (possibly changed) current directory name.

`$d`->codeset([NEWCODE])

For instances created with `Locale::gettext->domain_raw`, manuiplates the character set of the returned strings. If **NEWCODE** is given, calls `bind_textdomain_codeset` to set the character encoding in which messages for the domain represented by `$d` are returned. Returns the (possibly changed) current encoding name.

gettext(), *dgettext()*, and *dcgettext()* attempt to retrieve a string matching their `msgid` parameter within the context of the current locale. *dcgettext()* takes the message's category and the text domain as parameters while *dgettext()* defaults to the LC_MESSAGES category and *gettext()*

defaults to LC_MESSAGES and uses the current text domain. If the string is not found in the database, then `msgid` is returned.

ngettext(), *dngettext()*, and *dcngettext()* function similarily but implement differentiation of messages between singular and plural. See the documentation for the corresponding C functions for details.

textdomain() sets the current text domain and returns the previously active domain.

*bindtextdomain(domain, dirname)*instructs the retriev al functions to look for the databases belonging to domain `domain` in the directory `dirname`

bind_textdomain_codeset(domain, codeset) instructs the retrieval functions to translate the returned messages to the character encoding given by **codeset** if the encoding of the message catalog is known.

NOTES

Not all platforms provide all of the functions. Functions that are not available in the underlying C library will not be available in Perl either.

Perl programs should use the object interface. In addition to being able to return native Perl wide character strings, `bind_textdomain_codeset` will be emulated if the C library does not provide it.

VERSION

1.05.

SEE ALSO

gettext(3i), *gettext(1)*, *msgfmt(1)*

AUTHOR

Phillip Vandry <vandry@TZoNE.ORG>