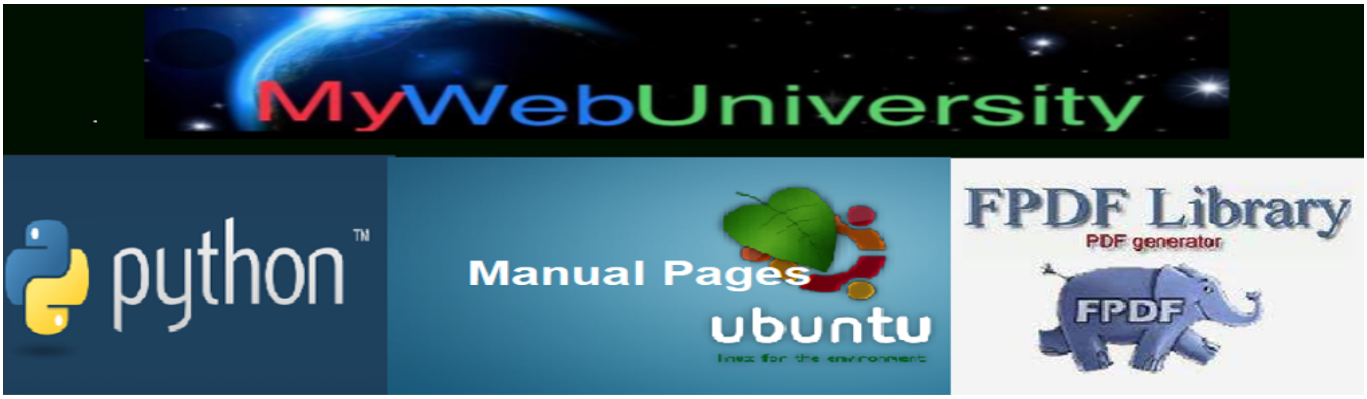

Rocky Enterprise Linux 9.2 Manual Pages on command 'I18N::Langinfo.3perl'

$ man I18N::Langinfo.3perl

NAME

I18N::Langinfo - query locale information

SYNOPSIS

```
use I18N::Langinfo;
```

DESCRIPTION

The langinfo() function queries various locale information that can be used to localize output and user interfaces. It uses the current underlying locale, regardless of whether or not it was called from within the scope of "use?locale". The langinfo() function requires one numeric argument that identifies the locale constant to query: if no argument is supplied, $_ is used. The numeric constants appropriate to be used as arguments are exportable from I18N::Langinfo.

The following example will import the langinfo() function itself and three constants to be used as arguments to langinfo(): a constant for the abbreviated first day of the week (the numbering starts from Sunday = 1) and two more constants for the affirmative and negative answers for a yes/no question in the current locale.

```
use I18N::Langinfo qw(langinfo ABDAY_1 YESSTR NOSTR);
```

```
my ($abday_1, $yesstr, $nostr) =
    map { langinfo($_) } (ABDAY_1, YESSTR, NOSTR);

print "$abday_1? [$yesstr/$nostr] ";
```

In other words, in the "C" (or English) locale the above will probably print something like:

```
Sun? [yes/no]
```

but under a French locale

```
dim? [oui/non]
```

The usually available constants are as follows.

? For abbreviated and full length days of the week and months of the year:

```
ABDAY_1 ABDAY_2 ABDAY_3 ABDAY_4 ABDAY_5 ABDAY_6 ABDAY_7
ABMON_1 ABMON_2 ABMON_3 ABMON_4 ABMON_5 ABMON_6
ABMON_7 ABMON_8 ABMON_9 ABMON_10 ABMON_11 ABMON_12
DAY_1 DAY_2 DAY_3 DAY_4 DAY_5 DAY_6 DAY_7
MON_1 MON_2 MON_3 MON_4 MON_5 MON_6
MON_7 MON_8 MON_9 MON_10 MON_11 MON_12
```

? For the date-time, date, and time formats used by the strftime() function (see POSIX):

```
D_T_FMT D_FMT T_FMT
```

? For the locales for which it makes sense to have ante meridiem and post meridiem time formats:

```
AM_STR PM_STR T_FMT_AMPM
```

? For the character code set being used (such as "ISO8859-1", "cp850", "koi8-r", "sjis", "utf8", etc.), and for the currency string:

CODESET CRNCYSTR

? For an alternate representation of digits, for the radix character used between the integer and the fractional part of decimal numbers, the group separator string for large-ish floating point numbers (yes, the final two are redundant with POSIX::localeconv()):

ALT_DIGITS RADIXCHAR THOUSEP

? For the affirmative and negative responses and expressions:

YESSTR YESEXPR NOSTR NOEXPR

? For the eras based on typically some ruler, such as the Japanese Emperor (naturally only defined in the appropriate locales):

ERA ERA_D_FMT ERA_D_T_FMT ERA_T_FMT

For systems without "nl_langinfo"

Starting in Perl 5.28, this module is available even on systems that lack a native "nl_langinfo". On such systems, it uses various methods to construct what that function, if present, would return. But there are potential glitches. These are the items that could be different:

"ERA"

Unimplemented, so returns "".

"CODESET"

Unimplemented, except on Windows, due to the vagaries of vendor locale names,

returning "" on non-Windows.

"YESEXPR"

"YESSTR"

"NOEXPR"

"NOSTR"

Only the values for English are returned. "YESSTR" and "NOSTR" have been removed from POSIX 2008, and are retained here for backwards compatibility. Your platform's "nl_langinfo" may not support them.

"D_FMT"

Always evaluates to %x, the locale's appropriate date representation.

"T_FMT"

Always evaluates to %X, the locale's appropriate time representation.

"D_T_FMT"

Always evaluates to %c, the locale's appropriate date and time representation.

"CRNCYSTR"

The return may be incorrect for those rare locales where the currency symbol replaces the radix character. Send email to <mailto:perlbug@perl.org> if you have examples of it needing to work differently.

"ALT_DIGITS"

Currently this gives the same results as Linux does. Send email to <mailto:perlbug@perl.org> if you have examples of it needing to work differently.

"ERA_D_FMT"

"ERA_T_FMT"

"ERA_D_T_FMT"

"T_FMT_AMPM"

These are derived by using "strftime()", and not all versions of that function know

about them. "" is returned for these on such systems.

See your nl_langinfo(3) for more information about the available constants. (Often this means having to look directly at the langinfo.h C header file.)

EXPORT

By default only the "langinfo()" function is exported.

BUGS

Before Perl 5.28, the returned values are unreliable for the "RADIXCHAR" and "THOUSEP" locale constants.

Starting in 5.28, changing locales on threaded builds is supported on systems that offer thread-safe locale functions. These include POSIX 2008 systems and Windows starting with Visual Studio 2005, and this module will work properly in such situations. However, on threaded builds on Windows prior to Visual Studio 2015, retrieving the items "CRNCYSTR" and "THOUSEP" can result in a race with a thread that has converted to use the global locale. It is quite uncommon for a thread to have done this. It would be possible to construct a workaround for this; patches welcome: see "switch_to_global_locale" in perlapi.

SEE ALSO

perllocale, "localeconv" in POSIX, "setlocale" in POSIX, nl_langinfo(3).

The langinfo() function is just a wrapper for the C nl_langinfo() interface.

AUTHOR

Jarkko Hietaniemi, <jhi@hut.fi>. Now maintained by Perl 5 porters.

COPYRIGHT AND LICENSE

terms as Perl itself.