

Locale programming

Read the sample locale output below and try to generate comparable output with your system. The notes below give some ideas for where to find information on locale APIs that can help generate the output.

Your preferred locale might not be available on your system. In that case, you should install it. (See the references for this section.)

Sample output

```
Program start. No locales set yet.
Testing: messages locale = C

Setting locale to empty string.
Testing: messages locale = en_US.UTF-8

Setting all locales to en_GB.UTF-8.
Testing: messages locale = en_GB.UTF-8

Setting all locales to en_US.UTF-8.
Testing: Messages locale = en_US.UTF-8

Language name with region: American English
Currency symbol: -$
Date format: %m/%d/%Y
Time format: %r
Date/time format: %a %d %b %Y %r %Z
First day of the week: Sunday
Time zone: America/Los_Angeles
Number format: 1,234,567,890.012345
```

Notes

Almost all of the information displayed above can be accessed with Linux and Unix APIs. (The time zone information above must be accessed directly.) The most useful single source of information on these APIs is *The GNU C Library* (see below).

References

- The GNU C Library
- The GNU C Library: The Elegant and Fast Way

- Time, Clock, and Calendar Programming In C
- linux - How to install/change locale on Debian? - Server Fault

No locales set

Program start. No locales set yet.
Testing: messages locale = C

Notes

Quoting *The GNU C Library*, “This is the standard C locale. The attributes and behavior it provides are specified in the ISO C standard. When your program starts up, it initially uses this locale by default.”

What the standard C locale amounts to is the **en_US** locale with other locale information that ISO considers standard.

The messages locale is simply the language in which the system displays messages. It’s a good test for what the overall locale of a system is.

References

- The GNU C Library: Standard Locales

Empty locale

Setting locale to empty string.
Testing: messages locale = en_US.UTF-8

Notes

When you set the empty locale with "", your program takes its locale information from your program’s environment (its shell variables and so on).

References

- The GNU C Library: Standard Locales

Setting specific locales

Setting all locales to `en_GB.UTF-8`.
Testing: `messages locale = en_GB.UTF-8`

Setting all locales to `en_US.UTF-8`.
Testing: `Messages locale = en_US.UTF-8`

Notes

The third main alternative, besides the default C locale, and the empty locale, which takes its values from your program's environment, is setting your program's environment yourself manually to a locale name like `en_US.UTF-8`.

References

- The GNU C Library: Standard Locales

Long locale name

Language name with region: American English

Notes

On Linux systems, the system displays the long name for the region (**American**) together with the long name for the language (**English**). If the locale were `en_GB` instead of `en_US`, the system would display **British English**.

References

- Howto get the language name for a given locale in linux - Stack Overflow

Currency symbol

Currency symbol: `-$`

Notes

The currency symbol shows what character is used to indicate money in the current locale, such as a dollar sign or euro sign. In this example, the `-` character indicates that the dollar sign comes before the monetary value rather than after.

References

- `nl_langinfo(3)`: query language/locale info - Linux man page

Date format

Date format: `%m/%d/%Y`

Notes

The date format shows how the system formats the year, month, and day in your current locale – that is, in what order they appear, how the system separates them, and so on.

References

- The GNU C Library: The Elegant and Fast Way

Time format

Time format: `%r`

Notes

The time format of `%r` in the example above specifies the time in AM/PM notation. A capital `%R` specifies 24-hour notation.

References

- The GNU C Library: The Elegant and Fast Way
- `strftime(3)`: format date/time - Linux man page

Date/time format

Date/time format: `%a %d %b %Y %r %Z`

Notes

The date/time format string, as shown above, specifies how the system should display both date and time together.

References

- The GNU C Library: The Elegant and Fast Way
- strftime(3): format date/time - Linux man page

Days of the week

First day of the week: Sunday

Notes

The names of the days of the week are different in different locales. In some calendars, the week even starts on a different day. For example, in the ISO 8601 calendar, the first day of the week is Monday.

References

- The GNU C Library: The Elegant and Fast Way

Time zone

Time zone: America/Los_Angeles

Notes

Time zone configuration and display can often be confusing on Unix and Linux systems. However, there is one useful file on Debian and Ubuntu systems that the system uses to store human-readable timezone information. It is configured when the system is installed.

References

- date - Timezone setting in Linux - Unix & Linux Stack Exchange
- UbuntuTime

Number format

Number format: 1,234,567,890.012345

Notes

The number format of a locale is simply how it displays numbers. For example, the `en_US` locale uses commas to separate groups of three digits and a period character (.) before the decimal part of a number.

The GNU C Library contains an API for heavy-duty number formatting, but you may find that your language of choice already does all the formatting you need.

References

- The GNU C Library: Formatting Numbers
- formatting - How to format a number from 1123456789 to 1,123,456,789 in C? - Stack Overflow