3.3.3 Localization Facts

This lesson covers the following topics:

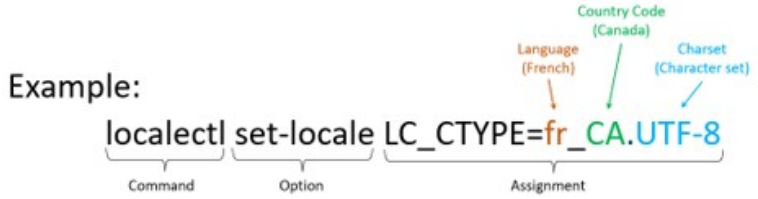
- Locale overview
- Character sets
- Locale environment variable definitions
- Locale variable precedence rules
- Miscellaneous commands
- Keyboard settings/mappings

Locale Overview

Linux uses locale environment variables to determine the conventions for the date and time formatting, character display, and other elements. Programs that support this technology use these environment variables accordingly.

Locales:

- Determine the way data is displayed on a computer, including:
  - The language and encoding of the text displayed on screen
  - Character classes
  - Sort order
  - Number formatting
  - Currency type and format
  - Date and time display
- Use configuration files that are part of the system library. On most distributions these are located in **/usr/share/locale**.
- Use language codes specified in ISO-639 and country codes specified in ISO-3166.
- Use the format **localectl *option assignment*** (such as ***language_COUNTRY.charsets*)**
  Note: The language (such as en for English) must be in all lowercase, while the COUNTRY (such as GB for Great Britain) must be in all uppercase.



As an alternative, you can first temporarily change the environment variable and then make it permanent using the export command. For example, to set the same LC_CTYPE command, you would complete the following:

```
# LC_CTYPE="fr_CA.UTF-8"
# export LC_CTYPE
```

Character Sets

Unicode (ISO 10646)
Unicode is a standard that tries to clearly represent every character in every human language (currently supporting 137,374+ characters). It defines how individual characters are represented in such things as text files and webpages. Character encodings (such as UTF-8) are then used to access the character set.

ASCII
ASCII (American Standard Code For Information Interchange) is a seven-bit encoding technique that assigns a number to each of the 128 characters used most frequently in American English. It is currently described by the ISO 646:1991 IRV (International Reference Version) standard. Since Unicode (when using UTF-8) is ASCII-compatible, plain ASCII text still renders properly on modern UTF-8 using systems. However, ASCII does not include symbols frequently used in other countries, such as the British pound symbol (£).

UTF-8
UTF-8 is a variable-length encoding standard of Unicode and is capable of encoding all 1,112,064 valid code points in Unicode using one to four 8-bit bytes. UTF-8 can support many languages and can accommodate pages and forms in any mixture of those languages. Its use also eliminates the need for server-side logic to individually determine the character encoding for each page served or each incoming form submission.

Locale Environment Variable Definitions

The following table lists the configurable environment variables used to control a Linux system's locale:

Variable name	Explanation
LANG	Defines all locale settings at once while allowing further individual customization via the **LC_*** settings. When this variable is set to **LANG=C**, programs will display their output without passing it through locale translations. This is helpful when the output is being corrupted by the locale and helps avoid some types of problems.
LC_CTYPE	Defines the character handling properties for the computer. This determines whether characters are recognized as alphabetical, numeric, etc. This also determines the character set used (if applicable).
LC_MESSAGES	Specifies localizations for applications that use a message-based localization scheme.
LC_COLLATE	Defines the alphabetical ordering of strings (for example, the output of sorted directory listings).
LC_NUMERIC	Controls formatting for numeric values that are not monetary (for example, which character to use as the thousands separator and the decimal separator).
LC_MONETARY	Identifies currency units and formatting of currency type and numeric values.
LC_TIME	Defines formatting for dates and times, such as whether to use a 24-hour clock versus a 12-hour clock.
LC_PAPER	Designates the default paper size (for example, US letter versus A4).
LC_NAME	Denotes personal name format (for example, whether the surname comes first or last).
LC_ADDRESS	Specifies address formatting.
LC_TELEPHONE	Defines telephone number format.
LC_MEASUREMENT	Determines which measurement units are used.
LANGUAGE	Overrides the LC_MESSAGES settings.
LC_ALL	Sets all locales to the same setting. This is used for overriding all other settings.

The locale settings are changed using the command: **localectl set-locale** LOCALE (where LOCALE is the environment variable to be changed with its options).

Examples:
localectl set-locale LC_PAPER=fr_CA.UTF-8
The system will be set to use the Canadian French paper size using the UTF-8 character set.

localectl set-locale LC_TIME=en_US.UTF-8
The system will be set to user the English 12 hr. time format used in the US using the UTF-8 character set.

Before you attempt to change the locale, you should verify that the desired locale options are available. This is accomplished using the 'list-locales' option. Example: **localectl list-locales**

If the locale is not available, install the desired language set.

In most cases, your system will need to be restarted before the permanent changes made using the **localectl** command can be used.

Locale Variable Precedence Rules

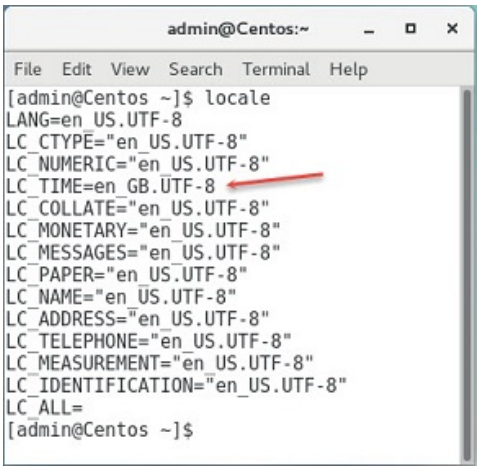
Not all of the LC_ environment variables have the same level of precedence. Linux uses the following rules to determine which locale settings to use. This helps determine what to do if conflicting values are placed in the locale variables.

Precedence Rules

1. If the LC_ALL variable is defined (non-null), its value is used, and the values assigned to all other LC_ variables are not checked.
2. If the LC_ALL variable is undefined (null), the specific LC_ variable in question is checked. If the specific LC_ variable has a value, it is used.
3. If the LC variable in question has a null value, the LANG environment variable is used.

Miscellaneous Commands

The following table describes a few additional locale related commands and options.

Command	Description
echo	From a shell prompt (terminal), the echo command can be used to view the current locale settings if one has been set. This is accomplished by typing 'echo' followed by a dollar sign ($) and then the name of the environment variable that you want to view. Example: **echo $LC_CTYPE**
locale	From a shell prompt (terminal), this command displays the current locale settings for the computer. If the option shown has quotes, it is obtaining the setting by inheritance. Settings without quotes are explicitly assigned.  Addition options include the following: ▪ **charmap** displays the available charmap (character set description files) used for character encoding. ▪ **-a** displays a list of the installed locales which are available for use. ▪ **-m** displays a list of the installed character encoding options which are available for use. Examples: **locale charmap** **locale -a** **locale -m** The locale command is located in /usr/bin.
iconv	The iconv program reads in text using one type of encoding and outputs the text in using another encoding type. If no input file is provided, or if it is given as a dash (-), iconv read from the standard input. If no output file is given, iconv writes to the standard output. Converts encoding from one encoding type to another. ▪ **-f** specifies the old encoding type. ▪ **-t** specifies the new encoding type. ▪ **-o** specifies the input and output file. As an example, the following command will convert text from the ISO 8859-15 character encoding to the UTF-8 encoding, where ***input.txt*** is the name of the input file: **iconv -f ISO-8859-15 -t UTF-8 *input.txt* output.txt**

Keyboard Settings/Mappings

The system locale also controls the keyboard settings (or mappings). The keyboard layout settings enable the user to control the layout used on the text console and graphical user interfaces. One way to accomplished this is by using the **localectl** command.

The following table list a few of the uses of the **localectl** commands available on most distributions. To simplify the table, only the options are show. Each option must be proceeded by the **localectl** command, such as **localectl status**.

Command	Description
status	Displays the current setting of the system locale and keyboard mapping.
list-keymaps	Lists the available keyboard mappings that can be configured on your system.

	Use the **grep** command to limit the keyboard mappings shown. Example: **localectl list-keymaps | grep us** The results may include the following: ▪ si-us ▪ sunt5-us-cz ▪ us ▪ us-euro
set-keymap *map*	Sets the default keyboard layout to use. Replace **map** with the desired option as found when using the **list-keymaps** option. Example: **localectl set-keymaps us-euro** The **--no-convert** option prevents the console from being converted to X11 or X11 to the console. Example: **localectl set-keymaps us-euro --no-convert** Different keymaps can be applied to using the **--no-convert** command. Example: **localectl --no-convert set-x11-keymap fe** In this example, the console remains as it is while the X11 changes to French.

TestOut Corporation All rights reserved.