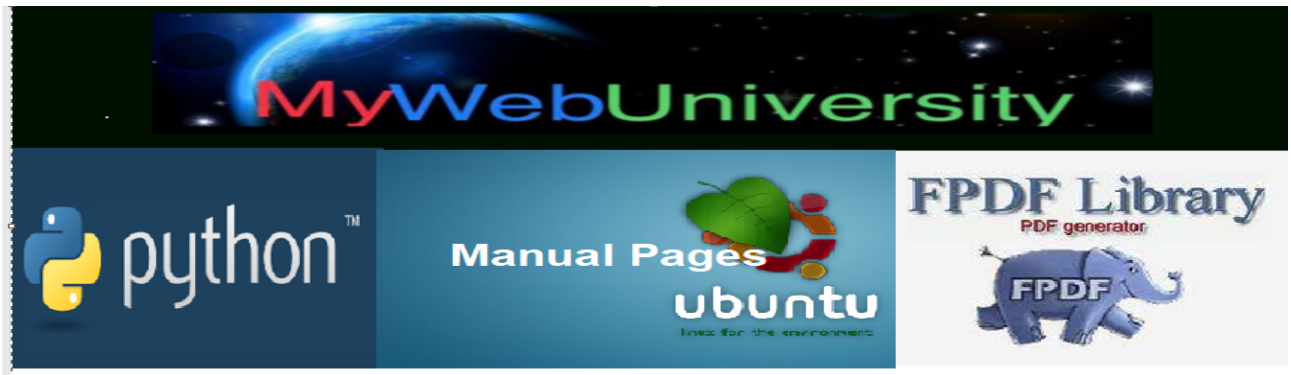

Full credit is given to the above companies including the OS that this PDF file was generated!

Linux Ubuntu 26.04 Manual Pages on command 'luit.1'

$ man luit.1

LUIT(1) General Commands Manual LUIT(1)

NAME

luit - Locale and ISO 2022 support for Unicode terminals

SYNOPSIS

luit [options] [--] [program [args]]

DESCRIPTION

Luit is a filter that can be run between an arbitrary application and a UTF-8 terminal emu? lator. It will convert application output from the locale's encoding into UTF-8, and con? vert terminal input from UTF-8 into the locale's encoding.

Luit reads its input from the child process, i.e., an application running in the terminal. Luit writes its output to the terminal. The two (input and output) can have different en? codings.

An application may also request switching to a different output encoding using ISO 2022 and ISO 6429 escape sequences. Use of this feature is discouraged: multilingual applications should be modified to directly generate UTF-8 instead.

Luit is usually invoked transparently by the terminal emulator. For information about run? ning luit from the command line, see EXAMPLES below.

OPTIONS

-V Print luit's version and quit.

-alias filename

the locale alias file

(default: unknown).

-argv0 name

Set the child's name (as passed in argv[0]).

-c Function as a simple converter from standard input to standard output.

-encoding encoding

Set up luit to use encoding rather than the current locale's encoding.

-g0 charset

Set the output charset initially selected in G0. The default depends on the locale, but is usually ASCII.

-g1 charset

Set the output charset initially selected in G1. The default depends on the locale.

-g2 charset

Set the output charset initially selected in G2. The default depends on the locale.

-g3 charset

Set the output charset initially selected in G3. The default depends on the locale.

-gl gn Set the initial assignment of GL in the output. The argument should be one of g0, g1, g2 or g3. The default depends on the locale, but is usually g0.

-gr gk Set the initial assignment of GR in the output. The default depends on the locale, and is usually g2 except for EUC locales, where it is g1.

-h Display a usage and options message on the standard output and quit.

-ilog filename

Log into filename all the bytes received from the child.

-k7 Generate seven-bit characters for keyboard input.

-kg0 charset

Set the input charset initially selected in G0. The default depends on the locale, but is usually ASCII.

-kg1 charset

Set the input charset initially selected in G1. The default depends on the locale.

-kg2 charset

Set the input charset initially selected in G2. The default depends on the locale.

-kg3 charset

Set the input charset initially selected in G3. The default depends on the locale.

-kgl gn

Set the initial assignment of GL in the input. The argument should be one of g0, g1, g2 or g3. The default depends on the locale, but is usually g0.

-kgr gk

Set the initial assignment of GR in the input. The default depends on the locale, and is usually g2 except for EUC locales, where it is g1.

-kls Generate locking shifts (SO/SI) for keyboard input.

+kss Disable generation of single-shifts for keyboard input.

+kssgr Use GL codes after a single shift for keyboard input. By default, GR codes are gen? erated after a single shift when generating eight-bit keyboard input.

-list List the supported charsets and encodings, then quit. Luit uses its internal tables for this, which are based on the fontenc library.

-list-builtin

List the built-in encodings used as a fallback when data from iconv or fontenc is missing.

This option relies on luit being configured to use iconv, since the fontenc library does not supply a list of built-in encodings.

-list-fontenc

List the encodings provided by ?.enc? files originally distributed with the fontenc library.

-list-iconv

List the encodings and locales supported by the iconv library. Luit adapts its in? ternal tables of fontenc names to iconv encodings.

To make scripting simpler, luit ignores spaces, underscores and ASCII minus-signs (dash) embedded in the names. Luit also ignores case when matching charset and en? coding names.

This option lists only the encodings which are associated with the locales supported on the current operating system. The portable iconv application provides a list of its supported encodings with the -l option. Other implementations may provide simi? lar functionality. There is no portable library call by which an application can ob? tain the same information.

-olog filename

Log into filename all the bytes sent to the terminal emulator.

+ols Disable interpretation of locking shifts in application output.

+osl Disable interpretation of character set selection sequences in application output.

+oss Disable interpretation of single shifts in application output.

+ot Disable interpretation of all sequences and pass all sequences in application output to the terminal unchanged. This may lead to interesting results.

-p In startup, establish a handshake between parent and child processes. This is needed for some older systems, e.g., to successfully copy the terminal settings to the pseudo-terminal.

-prefer list

Set the lookup-order preference for character set information. The parameter is a comma-separated list of keywords. The default order (listing all keywords) is

fontenc,builtin,iconv,posix

The default order uses fontenc first because this allows luit to start more rapidly (about 0.1 seconds) than using iconv for complex encodings such as eucJP. However, you may find that the iconv implementation is more accurate or complete. In that case, you can use the -show-iconv option to obtain a text file which can be used as an encoding with the fontenc configuration.

This option relies on luit being configured to use iconv, since the fontenc library does not provide this choice.

-show-builtin encoding

Show a built-in encoding, e.g., from a ?.enc? file using the ?.enc? format.

This option relies on luit being configured to use iconv, since the fontenc library does not supply a list of built-in encodings.

-show-fontenc encoding

Show a given encoding, e.g., from a ?.enc? file using the ?.enc? format. If luit is configured to use the fontenc library, it obtains the information using that library. Otherwise luit reads the file directly.

Some of fontenc's encodings are built into the library. The fontenc library uses those in preference to an external file. Use the -show-builtin option to provide similar information when luit is configured to use iconv.

-show-iconv encoding

Show a given encoding, using the ?.enc? format. If luit is configured to use iconv, it obtains the information using that interface. If iconv cannot supply the informa? tion, luit may use a built-in table.

-t Initialize luit using the locale and command-line options, but do not open a pty con? nection. This option is used for testing luit's configuration. It will exit with

success if no errors were detected. Repeat the -t option to cause warning messages to be treated as errors.

-v Be verbose. Repeating the option, e.g., ?-v -v? makes it more verbose. Luit does not use getopt, so ?-vv? does not work.

-x Exit as soon as the child dies. This may cause luit to lose data at the end of the child's output.

-- End of options.

ENVIRONMENT

Luit uses these environment variables:

FONT_ENCODINGS_DIRECTORY

overrides the location of the ?encodings.dir? file, which lists encodings in external ?.enc? files.

LC_ALL

LC_CTYPE

LANG During initialization, luit calls setlocale to check if the user's locale is sup? ported by the operating system. If setlocale returns a failure, luit looks instead at these variables in succession to obtain any clues from the user's environment for locale preference.

NCURSES_NO_UTF8_ACS

Luit sets this to tell ncurses to not rely upon VT100 SI/SO controls for line-draw? ing.

SHELL This is normally set by shells other than the Bourne shell, as a convention. Luit will use this value (rather than the user's entry in /etc/passwd) to decide which shell to execute. If SHELL is not set, luit executes /bin/sh.

FILES

unknown

The file mapping locales to locale encodings.

BUGS

Limitations

None of this complexity should be necessary. Stateless UTF-8 throughout the system is the way to go.

Charsets with a non-trivial intermediary byte are not yet supported.

Selecting alternate sets of control characters is not supported and will never be.

Security

On systems with SVR4 (?Unix-98?) ptys (Linux version 2.2 and later, SVR4), luit should be run as the invoking user.

On systems without SVR4 (?Unix-98?) ptys (notably BSD variants), running luit as an ordinary user will leave the tty world-writable; this is a security hole, and luit will generate a warning (but still accept to run). A possible solution is to make luit suid root; luit should drop privileges sufficiently early to make this safe. However, the startup code has not been exhaustively audited, and the author takes no responsibility for any resulting se? curity issues.

Luit will refuse to run if it is installed setuid and cannot safely drop privileges.

EXAMPLES

The most typical use of luit is to adapt an instance of XTerm to the locale's encoding. Current versions of XTerm invoke luit automatically when it is needed. If you are using an older release of XTerm, or a different terminal emulator, you may invoke luit manually:

```
$ xterm -u8 -e luit
```

If you are running in a UTF-8 locale but need to access a remote machine that doesn't sup? port UTF-8, luit can adapt the remote output to your terminal:

```
$ LC_ALL=fr_FR luit ssh legacy-machine
```

Luit is also useful with applications that hard-wire an encoding that is different from the one normally used on the system or want to use legacy escape sequences for multilingual out? put. In particular, versions of Emacs that do not speak UTF-8 well can use luit for multi? lingual output:

```
$ luit -encoding 'ISO 8859-1' emacs -nw
```

And then, in Emacs,

```
M-x set-terminal-coding-system RET iso-2022-8bit-ss2 RET
```

AUTHORS

Luit was written by Juliusz Chroboczek <jch@pps.jussieu.fr> for the XFree86 project.

Thomas E. Dickey has maintained luit for use by xterm since 2006.

SEE ALSO

These are portable:

? xterm(1),

? ncurses(3X).

These are Linux-specific:

? unicode(1),

? utf-8(1),

? charsets(1).

These are particularly useful:

? Character Code Structure and Extension Techniques (ISO 2022, ECMA-35)

? Control Functions for Coded Character Sets (ISO 6429, ECMA-48)

? http://czyborra.com/charsets/