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 emulator. It will convert application output from the locale's encoding into UTF-8, and convert terminal input from UTF-8 into the locale's encoding.

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 running **luit** from the command line, see EXAMPLES below.

OPTIONS

-h Display some summary help and quit.

-list List the supported charsets and encodings, then quit.

-V Print luit's version and quit.

-v Be verbose.

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

-p In startup, establish a handshake between parent and child processes. This is needed for some systems, e.g., FreeBSD.

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

-argv0 *name*
Set the child's name (as passed in argv[0]).

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

+oss Disable interpretation of single shifts in application output.

+ols Disable interpretation of locking shifts in application output.

+osl Disable interpretation of character set selection sequences 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.

-k7 Generate seven-bit characters 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 generated after a single shift when generating eight-bit keyboard input.

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

-gl *gn* Set the initial assignment of GL. 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. The default depends on the locale, and is usually **g2** except for EUC locales, where it is **g1**.

-g0 *charset*
Set the charset initially selected in G0. The default depends on the locale, but is usually **ASCII**.

-g1 *charset*
: Set the charset initially selected in G1. The default depends on the locale.

-g2 *charset*
: Set the charset initially selected in G2. The default depends on the locale.

-g3 *charset*
: Set the charset initially selected in G3. The default depends on the locale.

-ilog *filename*
: Log into *filename* all the bytes received from the child.

-olog *filename*
: Log into *filename* all the bytes sent to the terminal emulator.

-alias *filename*
: the locale alias file
(default: /usr/share/X11/locale/locale.alias).

--
: End of options.

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 support 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 output. In particular, versions of **Emacs** that do not speak UTF-8 well can use **luit** for multilingual output:

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

And then, in **Emacs**,

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

FILES

/usr/share/X11/locale/locale.alias
: The file mapping locales to locale encodings.

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 security issues.

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

BUGS

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.

SEE ALSO

xterm(1), unicode(7), utf-8(7), charsets(7)
Character Code Structure and Extension Techniques (ISO 2022, ECMA-35).
Control Functions for Coded Character Sets (ISO 6429, ECMA-48).

AUTHOR

The version of **Luit** included in this X.Org Foundation release was originally written by Juliusz Chroboczek <jch@freedesktop.org> for the XFree86 Project and includes additional contributions from Thomas E. Dickey required for newer releases of xterm(1)