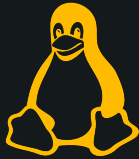




<https://portacle.github.io>





File Edit Options Buffers Tools Development SLIME REPL Presentations Lisp Trace Help

```
1 ;;; Welcome to Portacle, the Portable Common Lisp Environment.
2 ;; For information on Portacle and how to use it, please read the website at
3 ;; https://github.com/Shinmera/portacle
4 ;; or see the *portacle-help* buffer. You can switch to it by pressing this:
5 ;; Ctrl+h h
6 ;; or by clicking on the *scratch* field below until it changes to read
7 ;; *portacle-help*.
8 ;;
9 ;; You can use this buffer for notes and tinkering with small pieces of code.
10
```

□

- 448 U: *scratch* Lisp Interaction company Portacle 11 : 0 All

```
1 ; SLIME 2.19
2 CL-USER> □
```

* 22 U: *slime-repl sbcl* REPL company Paredit Portacle adoc 2 : 9 All



<https://portacle.github.io>

Lang OS or Unifying system policies as Lisp code

Michael Raskin, raskin@mccme.ru

LaBRI, Université de Bordeaux

April 15, 2018

Lisp System Environment

Control the things I *want* to control in the system via Lisp code

Implementing entire system in Lisp: unrealistic

Implementing all policies in Lisp: getting there

Why not XYZ

Normal GNU/Linux distributions? scsh? GuixSD?

Wanted:

- Complicated but easy to apply «package deals» for settings from CPU frequency to instant messenger status line
- Policy I can read, diff and version-control
- Plausible path to single suitable programming language for all configuration (happens to be Common Lisp...)

Wanted in practice: WiFi cards with firmware blobs

Not Wanted: too many DSLs, weak DSLs, GUI-only management tools, parsing JSON in pure Bash

What I really use?

StumpWM

URxvt

Screen

Editor

SBCL

Browser

TeXLive

PDF viewers

Random moderately popular Linux programs here and there

What I really use?

- * StumpWM

- * URxvt

- * Screen

- * Editor

- * SBCL

Browser

- * TeXLive

PDF viewers

Random moderately popular Linux programs here and there

- * programs I actually expect to work

What I have now

Nix package of entire system (and most software from Nixpkgs' tens of thousands of packages runs as well as on NixOS)

Lisp daemon — to start other daemons, instead of sudo — privileged system management tasks

(Doesn't monopolise computer, can be restarted)

List-of-lists-and-strings protocol — hopefully safe...

A Lisp function for every situation where I want to reconfigure the laptop

Support for launching things in configurable isolation

(All Firefox instances are isolated)

StumpWM integration

Two machines running all this

<https://github.com/7c6f434c/lang-os>

Still needed

Take over interaction between two machines

Replace more old Bash scripts

Generate parts of Nix code by Lisp

Control programs' access to X session (Xpra?)

Web content manipulation API: pure Lisp for simple things,

Firefox/Marionette when site needs more (but same code for data extraction for both)



Unicode for Common Lisp

Thomas Bakketun




Itches

- Encoding/decoding
 - Flexible error handling
- Normalization
- Collation (alphabetic order)
 - Different for each locale
- UTF-32 wastes memory
- Differences between implementations



Solutions

- Agnostic memory representation
 - Generic interface for UTF-8/16/32 text
 - Little overhead (hopefully)
- Backwards compatible
 - CL:STRING is also of type Unicode text
- Modular
 - Lightweight base

- 
- Initial work
 - <https://github.com/bakketun/unicode-for-common-lisp>
 - thomas@bakketun.net

Quickref

Antoine Martin
Didier Verna

Quickref

Reference manuals for Quicklisp libraries.

Quicklisp version 2010-02-20

Documentation generated with [Dedl 2.0](#) "Robert April".

Index Entry

#

[1am](#)

[3bmd](#)

[cl-6502](#)

[3b-swf](#)

[3d-matrices](#)

[3bgl-shader](#)

[3d-vectors](#)

A

[a-cl-logger](#)

[cl-anonfun](#)

[asdf-dependency-grovel](#)

[able](#)

[cl-ansi-term](#)

[asdf-encodings](#)

[cl-abnf](#)

[cl-ansi-text](#)

[asdf-finalizers](#)

[cl-abstract-classes](#)

[antik](#)

[asdf-flu](#)

<http://quickref.common-lisp.net/>

Global documentation project for Common Lisp libraries
Declt + Quicklisp

Quickref

Reference manuals for Quicklisp libraries.

Quicklisp version 2010-02-20

Documentation generated with Dedit 2.0 "Robert April".

Index Entry

#

[1am](#)

[3bmd](#)

[cl-6502](#)

[3b-swf](#)

[3d-matrices](#)

[3bgl-shader](#)

[3d-vectors](#)

A

[a-cl-logger](#)

[cl-anonfun](#)

[asdf-dependency-grovel](#)

[able](#)

[cl-ansi-term](#)

[asdf-encodings](#)

[cl-abnf](#)

[cl-ansi-text](#)

[asdf-finalizers](#)

[cl-abstract-classes](#)

[antik](#)

[asdf-flu](#)

- **Actively maintained**
- **New features coming up**
- **Code released soon**

Quickref

Reference manuals for Quicklisp libraries.

Quicklisp version 2010-02-20

Documentation generated with Dedit 2.0 "Robert April".

Index Entry

#

[1am](#)

[3bmd](#)

[cl-6502](#)

[3b-swf](#)

[3d-matrices](#)

[3bgl-shader](#)

[3d-vectors](#)

A

[a-cl-logger](#)

[cl-anonfun](#)

[asdf-dependency-grovel](#)

[able](#)

[cl-ansi-term](#)

[asdf-encodings](#)

[cl-abnf](#)

[cl-ansi-text](#)

[asdf-finalizers](#)

[cl-abstract-classes](#)

[antik](#)

[asdf-flu](#)

```
docker run --name quickref quickref/quickref
docker cp quickref:/home/quickref/quickref .
```