

Quicklisp

Get started with Common Lisp libraries, quickly

Quicklisp is a project for getting started with a rich set of open source Common Lisp libraries developed by the CL community.

Of course, there is no community.

clbuild, LibCL, Lispy,
repo-install, cl-librarian,
LAIR, Desire, etc.

There are a lot of projects along these lines already. clbuild is the most popular one I know of – many people I know use and recommend it, but there are others, like LibCL, Lispy, repo-install, etc.

I won't mention asdf-install
if you won't.

Quicklisp goals

There are some useful areas where Quicklisp improves on what's available.

Insanely easy to get and use

First, Quicklisp should be easy to get and use.

Two steps is one too many

Quicklisp is three - sorry!

To get started with Quicklisp, you download and load a file.

Work everywhere

Second, it should work in any CL on any platform.

Common Lisp is not Unix!

Don't rely on Unix programs.

Spawning gzip, tar, gpg, etc. is easy only on Unix, and only on certain Unixes. You can do those things in de facto portable plain CL, and it will work on Windows.

Highly available & stable

Third, quicklisp shouldn't break because someone else's site breaks.

Oh, crap, time to renew
cliki.net again.

Quicklisp archives are published on Amazon's S3 storage system. If S3 goes down, quicklisp is probably the last thing you'll notice – a lot of the Internet will be on fire.

Oh, crap, I can't reach
flaky.net's darcs!

For projects that are primarily available in version control, quicklisp takes snapshots. It also uses versioned releases for projects that use that style. It makes all the project archives available via HTTP. It's firewall and proxy friendly.

Extend freely

Make your own libraries available

Only takes three text files and the archive tarballs

You don't have to coordinate with me to publish a set of project archives.

Add support for a new CL yourself

Only takes a handful of new methods

You don't have to coordinate with me to add support for a new Common Lisp implementation.

Implement a completely new idea

Only takes a handful of new methods

You don't have to coordinate with me to use quicklisp protocols that implement a different style of management, like pulling from a VCS directly or from other sources of metadata. It's meant to be extensible.

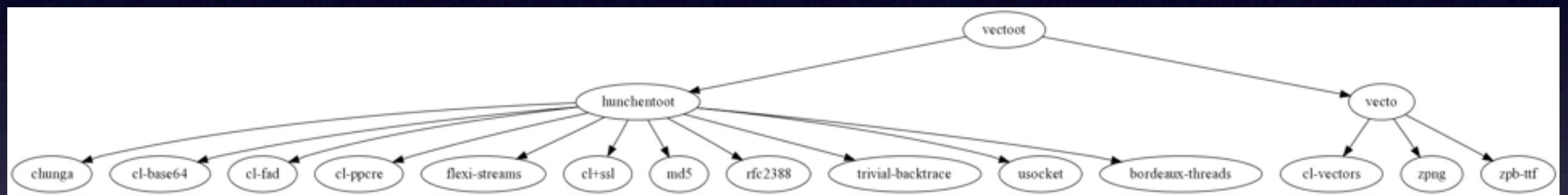
Build on the metadata

- Automated testing
- Bundling for delivery - build your application without Quicklisp
- Static package, symbol databases - “What packages does project Foo define? What symbols are exported?”

You don't have to coordinate with me to do interesting stuff with the archives and metadata.

Demo time!

Here's a demo of a simple web application that makes gradient PNG files. You could use this for CSS or something. It requires two Common Lisp libraries, Hunchentoot and Vecto.



If you were starting CL today and wanted to try something like this, you'd have to figure out how to get and install over a dozen libraries. The veterans know how to do it, but it can be a challenge.

Thanks to Nathan Froyd for making this idea possible
with chipz, ironclad, and more.