

# To Hell And Back Again

Stefan Kamphausen\*

<http://www.skamphausen.de/misc/writings/>

January 1, 2006

## 1 To Hell And Back Again

*A highly emotional rant about Common Lisp of someone who later coded happily in lisp which was not meant for the public initially. You have been warned ;-)*

### 1.1 To Hell And Back Again

Speaking as a programmer to an audience of programmers I think I can safely assume that we've all been there: in (programming) hell. Speaking for myself I've been at least in NewDeleteHell (AKA ConstructorDestructorOperatorHell AKA AggregationPointerDeepCloneHell AKA C++Hell), AbstractionHell (AKA WhereAreThingsActuallyDoneHell AKA JavaHell) and BadCodeHell (AKA IHaveWrittenThat?!Hell).

I've spent a good deal of time developing in emacs-lisp (xemacs-lisp to be more precise) and that gave me an almost pleasant feeling while coding, not the itching and weird feelings you get from languages like C or Java or .... Thus it came I got interested in CL, just as almost every serious programmer sooner or later does. Then reading almost the full PaulGrahamWebsite and finally browsing the CLIKI I decided to go ahead and install CL...

### 1.2 Installing CL

Uhm, yeah, "install CL" ... which one? First I tried CLISP and it installed just fine. I typed clisp at the prompt and up came a REP-client. It came with readline support, fine. Then I waited for a project to come by to be implemented in CL.

---

\*mail@skamphausen.de

In the meantime I read more about CL on the 'net and got the blurry feeling that CMUCL might be better suited for me. Another install.

And this is the first hell of CL.

OK, it was all well documented and I have been told beforehand that this is not configure-make-install-land. I've been in Makefile-land, in RPM-land, in DEB-land and even in manual-compile-land before but if it weren't for the step-by-step example in the BUILDING-file I think I'd just given up on CMUCL.

But this wasn't the end. By far.

### 1.3 Installing Additional Libraries

Of course I wanted to install some library or the other. Ah how easy this was in emacs. Load the file, do a M-x eval-current-buffer and try some functions. If the new library came in more than one file, simply add the directory to the load-path for the moment and try then. Neat. Almost always good enough to test. Not so in the case of CL libraries.

First you got to find out whether this lib will work in your special implementation of CL. It has always been an issue to find out whether an elisp-lib would work in my emacs (which is xemacs) or not and that was annoying. Figuring whether a CL-lib will work in one of the N major C-lisps is disgusting.

OK, I finally got asdf-install (funny name ;-)) to work and just for testing purposes installed cl-pcre. Although I didn't really see where which part of the install (the download, the source, the build, the final lib) would go, I decided to accept the defaults and configure that to my taste later. Since my first asdf-install I always get some screenful of compilation messages from cl-pcre whenever I start cmucl.

OK, I wanted to try closure, the webbrowser. Impossible to install for me.

OK, I wanted to try McCLIM, I needed CLX first. Yeah, but which? There's a bunch of different CLX'es out there. More reading lead to an installed clx-library.x86f together with some other I forgot. Back to McCLIM... Next requirement gray-streams (whatever that may be), hm, I find the files in the CMUCL-sources, is that what I need? Wil try. Next requirement defsystem. And in the installation manual there is a strange warning about defsystem from CLOCC and defsystem from PCL and whatnot. Hm, I find a defsystem in CMUCL sources in a subdir called PCL? Aha? Anyway I use them and compile them as I have been told to. Then I need to start a "fresh lisp" (as opposed to my current polluted one? ugly!) to build a new lisp-core? Huh? Will that be my future? Whenever I install I lib it goes directly to the core which will grow larger and larger? Well anyway, loading a "system.lisp" doesn't work so I go back a step and try to install the defsystem from CLOCC (do I know `_what_` exactly went wrong? How could I?)

Here I pass out when I am told to edit the logical host information in a strange file I which I find nothing like a hostname at all and which seems to be adresssing the Allegro-system in general.

This is where stop installing and start this writing.

## 1.4 Stop yelling at me

Do you remember the old days when the internet was young, free of layers and A O L was just beginning it's business? Ah, how we were disgusted by those newbies from A O L SCREAMING at us ALL THE TIME. Very, very unpolite they were.

Do you remember reading a software licence for a commercial software product the last time? The part when they get to the real business: DON'D DO THIS OR WE'LL SUE YOU 'TIL YOU WISH YOU'D NEVER BEEN BORN, YOU FILTHY LITTLE USER, YOU! Ah, unpolite they are.

And CL? Why does it yell at me all the time. I mean, it is bad enough that the language is case-insensitive but couldn't it -please- just use the small letters?

## 2 The Road goes ever on and on

It is to say that the rest of this article was written several months later, after a few thousand lines of CL coding.

## 2.1 Computer Science Theories

The lack of modules has been a major issue for many lisp users as it seems. OK, CPAN ist definetely a big plus for Perl but then most of the time you end up using the same standard modules all over again and again. And a CPAN installable module is not easily installed on a customers machine so I try to not use them anyway.

In lisp I've found many libs that seemed to have started well. In anstoundingly short time and little code people were able to get things working and I have been down that road, too ([3]).

But then works stops.

At least in my impression many libraries are in a proof-of-concept stage or in a I-need-only-this-subset-and-thats-done-stage. This may be fine for the original programmer of that lib but it shouldn't remain at that.

Maybe just a lack of ressources?

## 2.2 And I must follow, if I can

Well, despite all the issues I had at the beginning I pursued that road with eager feet nevertheless. The long-time programming experience seems to say that *every* language has it's issues. The issues of Common Lisp are itching and, since the initial mountain is so hard to climb, keep the user base small. But then, the developing experience is definetely a good one and coding in a REPL can be very comfy.

I will stay on this road if it's possible for me.

## 2.3 And whither then? I cannot say

Please note that most of the writings above were not meant for the public. It was more a rant for myself letting of steam. I just thought it might be an interesting or even funny read for those connected to the CL Gardeners ([2]) and one posting ([1]) on the mailing list made me clean this thing up a little so that it may be published.

## 3 Legalese

This document is ©Copyright 2005-2006 by Stefan Kamphausen. This document is considered Free Speech and is issued under the terms of the The Free Speech License ([4]).

## References

- [1] Peter Seibel. Common Lisp Gardeners Posting 2005-000234. <http://www.lispniks.com/pipermail/gardeners/2005-December/000234.html>.
- [2] Peter Seibel and The Helpful Gardeners. Common Lisp Gardeners. <http://www.lispniks.com/cl-gardeners/>.
- [3] Stefan Kamphausen. Common Lisp IPC. <http://common-lisp.net/project/cl-ipc/>.
- [4] Stefan Kamphausen. Free Speech License. <http://skamphausen.de/misc/writings/freespeel.pdf>, <http://skamphausen.de/misc/writings/freespeel.html>.