

Portable Lisp Programming

Gene Michael Stover

created 13 April 2004
updated 14 April 2004

Copyright © 2004 Gene Michael Stover. All rights reserved. Permission to copy, store, & view this document unmodified & in its entirety is granted.

Contents

1	Introduction	5
2	Logical Pathnames	7
3	Files	9
3.1	Opening a file for output	9
3.1.1	Test cases	9
3.1.2	Discussion	9
A	Other File Formats	11

Chapter 1

Introduction

Currently (14 April 2004), this article is in the form of notes. And not many of them so far. I'm adding slowly & inconsistently; this article isn't of high priority at the moment. It's below write a Lisp compiler, replace my lost driver's license, and figure out how the hell I'm going to pay rent in May, but it's of higher priority than "send an endless stream of profane letters to President Bush".

Chapter 2

Logical Pathnames

Here are some quick notes that deserve discussion.

clisp does not by default automatically detect a logical pathname when you use one as an argument to `OPEN` or another function which uses pathnames. You must explicitly call `TRANSLATE-LOGICAL-PATHNAME`. This is technically incorrect behaviour, but I agree with clisp's authors that it is the only unambiguous behaviour because pathnames on Microsoft Windows, when containing a driver letter, could be mistaken for logical pathnames. I need to document a technique for dealing with this.

When clisp translates a logical pathname to a native one, the native pathname may be relative, but with `sbc1`, the native pathname is always absolute even if you express it as a relative one. This means that logical pathname translations should always be absolute.

It seems that every Common Lisp implementation has its own system for specifying logical pathname translations. Some implementations borrow a technique from other implementations, but all are basically different. The most portable way to set logical pathname translations is to `SETF LOGICAL-PATHNAME-TRANSLATIONS`. The technique is not so limiting as it might seem. The pre-installation configuration program can create the `SETF` form in a file. Then the installation program can load that file, then all the other files, many of which will be specified with logical pathnames.

Chapter 3

Files

Currently (14 April 2004), I'm assuming text files. I'll have to extend this chapter to include binary files later.

3.1 Opening a file for output

If you want to open a text file for output with these semantics:

Open the specified file for textual output. If the file does not exist, create it. If it does exist, truncate it. (Basically the same semantics as “`fopen (pathname, "w")`” in Standard C)

avoid “`:IF-EXISTS :NEW-VERSION`” because it causes an error on SBCL if the file exists. Figure 3.1 is an example using `:RENAME-AND-DELETE`.

3.1.1 Test cases

I have tested this on clisp 2.33 & on sbcl 0.8.8, both on various versions of Gnu/Linux. I used the example from Figure 3.1 for the tests.

If the test uses `:NEW-VERSION`, it succeeds on clisp but fails on sbcl because the file already exists, so `:NEW-VERSION` should be avoided.

3.1.2 Discussion

I believe sbcl's behaviour is undesirable, though not incorrect.

The rationale behind sbcl's behaviour is probably that unix file systems are not normally versioned, but this breaks programs that were written for versioned file systems if they are ported to sbcl.

A more useful behaviour for sbcl would be the behaviour that clisp uses. If the file is opened with “`:IF-EXISTS :NEW-VERSION`” on an unversioned file system, the Lisp could rename the existing file before creating a new file with

```
(defvar *pn* (make-pathname :name "moo" :type "txt"
                           :version :newest))

(defun moo (pn)
  (with-open-file (strm pn :direction :output
                    :if-exists :rename-and-delete)
    (princ (get-universal-time) strm)))

;; Ensure the file doesn't exist.
(and (probe-file *pn*) (delete-file *pn*))

;; This works though the file doesn't exist.
(moo *pn*)

;; This works though the file does exist.
(moo *pn*)

;; Cleanup the file
(delete-file *pn*)
```

Figure 3.1: Example of a portable way to open a text file for output

the requested name. If the target filename for the rename operation already existed, it could be deleted.

This behaviour is allowed, in fact recommended, on page 650 of [Ste90]. Not only would this behaviour work with Lisp programs written for versioned file systems, but it would also provide an approximation of versioned behaviour on a non-versioned file system.

Appendix A

Other File Formats

- This document is available in multi-file HTML format at <http://cybertiggyr.com/gene/plp/>.
- This document is available in Pointless Document Format (PDF) at <http://cybertiggyr.com/gene/plp/plp.pdf>.

Bibliography

- [Ste90] Guy L. Steele. *Common Lisp: The Language*. Butterworth-Heinemann, second edition, 1990. ISBN 1-55558-041-6.