

2021 Medley Interlisp Annual Report

November 15, 2021

Introduction

To “revive” something is to make it live again. Making Medley Interlisp live again means putting the system in order so that others without a previous deep background in Interlisp can use and appreciate it (if only as a virtual antique).

The Medley Interlisp project started in earnest in March of 2020 (at the beginning of the pandemic). This report focuses on activities and accomplishments since the December 2020 virtual meeting of [LispNYC](#) (recording at: https://www.youtube.com/watch?v=x6-b_hazcyk).

In this document, work in the project is broken down into three main categories:

- Work on the software itself -- debugging, adapting, etc.
- Documentation for the software
- Building an organization and developing a community

Software

“Software Archeology” is a process like putting together a functional bowl from pottery shards. We have not completed the task, but we have a stable base.

The software is old: it had been developed between the '70s and the 90s, with many revisions, by many different people, working with little internal documentation. There were many “excursions” to support systems or options that are no longer available. Over the last 25 years, software standards evolved for C, Common Lisp, character codes, and operating systems.

Maiko

Maiko is the implementation, in C, of the Medley Interlisp virtual machine. It can be found at <https://github.com/interlisp/maiko>.

- **Conformance:** We have been making Maiko conform to modern C compiler expectations. Removing support for outdated systems simplified the task.
- **SDL:** Support is underway for SDL, a multi-platform (Windows, Mac OS X, Linux, iOS, and Android) graphics toolkit as an X11 replacement. SDL can be used instead of X11 as the shim to the native window system, which (we hope) will allow the system to run on Windows without separately installing X11, Docker Desktop, or WSL2.

Because of the cleanup work, the code is much more portable than ever before. Systems we've tested on or regularly build for include

- Operating systems: macOS, Linux, FreeBSD, Solaris 11.4
- CPU architectures: i386, x86_64, arm7, aarch64, SPARC-32 and -64

This covers Raspberry Pi (linux.arm7) and Windows 10 (with [WSL2](#) or Docker). and Windows 11 (includes WSL2). Performance is outstanding. A \$40 Raspberry Pi completes Lisp tasks 250-1000 times faster than the Dorado (the \$90k high-end Xerox 1132 Lisp machine).

Medley

The rest of the system is implemented in Lisp: Interlisp and Common Lisp interpreters, compilers, debugging tools, editors, window system, graphics, device drivers. In the last year:

- **Organization:** We have been cleaning out and organizing, comparing dates on files, and deciding which to keep; building maintenance scripts and Lisp utilities.
- **EOL and character conversion:** the code for handling end-of-line and character codes was generalized, on the way to full Unicode support. (See Medley Issue [#2](#))
- **Git integration:** Interlisp's file manager was designed for versioned files with version numbers. GitHub has a different graph model of history. We built "restore-versions" to restore numbered versions from git logs, but the problems are deeper. If the Interlisp style of programming is going to be supported, we have to solve these problems. (see Medley issues [#265](#), [#226](#))
- **Loadups:** We recreated [the method of building a system from scratch](#), which used to be an hours-long manual process.
- **Debugging:** We found and fixed a variety of bugs: a few Y2K problems, some instances of "bit rot" and smashed files, incomplete implementations, and code patched at "wrong level".
- **Modernization:** Window manipulation by title bar dragging and corner selection, mouse wheel scrolling, clipboard--these seem to be necessary enhancements for modern users.
- **Release process & automation:** GitHub Actions are now used to automate the build process for Medley releases and also a [Docker](#) container, deployed to [Docker Hub](#) as Interlisp/Medley. This gives a way to run Medley in the cloud (on Amazon Web Services, Azure, Google Cloud) or on the Windows Desktop with remote access in a browser.

Interlisp Applications

One of the reasons for restoring Medley Interlisp is to support the revival of applications written for it.

- **Notecards:** Notecards is a hypertext system developed well before the web or Apple's Hypercard. <http://notecards.online> was built with the Medley/Docker release, running Medley "in the cloud," with web browser access. We are adopting this to work for other Medley Interlisp systems, and for others to build.
- **Rooms:** Rooms is a Lisp desktop manager for less cluttered window access with interesting innovations. In the Medley repository. Loads but untested.
- **CLOS:** The Common Lisp Object System adds a style of Common Lisp class structures added after Medley's Common Lisp implementation to the ANSI standard. Also loads but untested.
- **LOOPS:** An object-oriented Lisp addition (to Interlisp) prior to CLOS. Of interest because of AI applications built using it, including "Truckin" -- a demonstration of "knowledge

programming [see [paper](#)]. The source code for the Truckin' application exists and will be used to test LOOPS once it fully loads and is compilable.

- **LFG**: the Grammar Writer's Workbench,
- **LispUsers**: User-contributed software (before "open source" was a thing). We've [checked out over half of the LispUsers files](#).

Documentation

Books

- ***Interlisp: The Language and Its Usage***, by Steve Kaisler, originally published by John Wiley & Sons, scanned and converted to a compact PDF. This book describes the core features of the Interlisp language. Now at <https://interlisp.org/docs/1986-Interlisp-language-book-1.pdf>.
- ***Medley Interlisp: The Interactive Programming Environment***, by Steve Kaisler, describes the Interlisp-D Interactive Programming Environment as implemented on Xerox D-machines and now running on several modern platforms. Now at <https://interlisp.org/docs/2021-interlisp-book-2.pdf>.
- ***Medley Interlisp: Tools and Utilities***, by Steve Kaisler (undergoing editing) describes the editors and tools to be used for program development. It is expected to be completed and uploaded to Interlisp.org in early December 2021.

Software Documentation

We've converted TEdit source documents to PDF (see [medley-pdfs](#)). We have the IRM (Interlisp Reference Manual) with links to online help (DInfo), User Guides, Release Notes, Primer. We still need to convert and publish documentation for some Interlisp applications and to organize and update to make it easier for newcomers.

Bibliography

Zotero is an Internet bibliographic service. We have set up an [Interlisp Zotero](#) of Lisp-related material, many items with the source material as PDFs. Our goal is to have a comprehensive source of information about Interlisp-related technical papers, technical reports, and manuals for various versions and implementations of Interlisp (and, for comparison) contemporaneous "Lisp Machine" competitors.

Working older versions

In addition to modern Medley, we have begun to organize other emulations of older versions, and a "history" repository, including files from Interlisp-10, the "DarkStar" emulator of the Xerox 1108 / 8010 hardware, and sysouts for many different releases of Interlisp-D. We're working on making an earlier version of Medley (running on an emulated DosBox); currently available with instructions in the Interlisp/DOS repository.

Outreach

To attract users and supporters, we need to make the project known.

- We developed and installed a GitHub “code of conduct”, “contributor’s guide” and issue templates (although we haven’t been using the templates and need to review them).
- Maintained additional web resources
 - <https://interlisp.org> web site, and its Bibliography from Zotero.
 - README’s and other repository documents
 - [Interlisp/medley Wiki \(github.com\)](https://github.com/Interlisp/medley/wiki)
- We track mentions of “interlisp” in social media (and we will post a link to this document)
 - Twitter (@interlisp8)
 - LinkedIn
 - Reddit
 - Facebook: Xerox Workstation Alums, LISPERs groups.
 - IRC #Interlisp on Libera.chat
 - Meetup & Lisp groups: [LispNYC](#), [Bay Area Lisp](#):
online-lisp-meets@common-lisp.net
- Conferences to announce to?
 - [European Lisp Symposium \(european-lisp-symposium.org\)](http://european-lisp-symposium.org)
- Computer History Museum (CHM) -- we’re hoping to provide access to Interlisp via an analogous setup for [the CHM’s Smalltalk Zoo](#).
- [Software Preservation Network \(SPN\) | advance software preservation through collective action](#) through their [Emulation-as-a-Service Infrastructure](#) is a possible way of getting Medley access available to researchers.

Who are we?

In this document, we use “we” to mean our contributors:

- as members of the lispcore@googlegroups.com mailing list/group ([Medley Interlisp core - Google Groups](#))
- as participants in a weekly zoom call
 - Meetings recorded but not public
 - Agendas: [Meeting Agendas and Notes](#)
- as members of the GitHub [Interlisp organization](#)
- as contributors to an Open Source project
 - 384 issues, 133 closed, 151 open [Issues](#). 54 [discussions](#)
 - 191 Medley [pull requests](#), 387 Maiko [pull requests](#)
- as a (non-profit) corporation: [InterlispORG Inc.](#)

We depend on volunteers to help. If you’d like to help but aren’t sure how to, ask.

Future activities

There were some tasks/projects listed here; that have been moved to GitHub [issues](#).

Acknowledgments

We've gotten a lot done, but there's quite a bit more to do. Thank you to Abe, Alexander, Arun, Bill, Blake, Frank, Herb, John C, Larry, Michele, Nick, Paul, Peter, Ron, Wayne, Zoe.

A moment in memoriam to past contributors, including John Sybalsky, Warren Teitelman, Danny Bobrow.
