Building Capacities in Digital Archaeology

Study Leave Report

January 1 – June 30, 2016

John Durno
University of Victoria Libraries

Table of Contents

Appendices

Overview

Increasingly our cultural memory is embodied in digital objects, but preserving and providing access to them presents interesting challenges as technologies evolve. These challenges are not uniform but instead vary depending on the nature of the technological environments from which the objects emerged. During the past six months I have undertaken a number of interrelated projects intended to build the Libraries' capacity to work productively with digital artifacts from earlier computing eras, primarily the 1980s through the mid-1990s. As outlined in my study leave proposal (Appendix I) I have:

- Worked with Archives staff to identify at-risk digital content in our collections
- Developed strategies for preservation and access of content on obsolete media
- Created software environments for working with old file formats
- Completed a project to restore the digital artworks in the Glenn Howarth archive
- Evaluated the ability of Archivematica to ingest and manage floppy disk images

During the course of my leave I expanded the scope of my activities to include the following projects and initiatives that were not in the original proposal, but contributed to its overall goal:

- Installed, configured and tested emulators for a number of historic computing systems
- Organized and promoted the Libraries' Ideafest 2016 event, “Retro Computing in the Library”
- Imaged and processed over 300 floppy disks from Archives and Special Collections for long-term preservation
- Worked with researchers from the university and the community to retrieve content from old media
- Presented on the Glenn Howarth digital art restoration project at the Archives Association of BC annual conference and at Ideafest
- Published a detailed technical note on the restoration of NAPLPS graphics in our institutional repository (Appendix II)
- Written a comprehensive paper on techniques for retrieving data from 1980s floppy disks (Appendix III, provisionally accepted for publication in Code4Lib journal)
- Acquired, inventoried, evaluated and partially reconditioned a collection of historic computing equipment from the 1980s and 1990s
- Participated in an SSHRC-funded project studying the challenges of future-proofing and preserving digital humanities projects

In addition to the above I continued to participate in the University Senate and Senate Agenda Committee during my leave.

Detailed description of activities and projects

Digital Media Inventory

At the time I submitted my study leave proposal no inventory had yet been made identifying at-risk digital media in our collections. As the existence of such an inventory was a necessary precondition of further work in this area, I included its creation as part of my study leave work plan. Happily, that inventory was created by Archives staff prior to the beginning of my leave, as part of the research undertaken to inform a Digital Preservation Plan being developed by the Libraries' Digital Preservation Working Group. This enabled me to move directly to the other projects on my work plan, as outlined below.

Floppy Disk Imaging

In 2014 Jerry Trofimchuk and I completed a project to build two workstations capable of imaging 3.5" and 5.25" floppy disks. Following some initial testing we were able to use these workstations to successfully retrieve content from over 100 floppy disks drawn primarily from the Glenn Howarth archive. However early on we became aware that, owing to fundamental constraints in PC floppy drive circuitry and drive hardware, these workstations had three significant limitations:

1) They could not image 3.5" single & double density (400/800K) Mac floppies
2) They could not image the second side of so-called "flippy disks" (floppy disks intended by the manufacturer to be single sided but which had data written on both sides anyway)
3) They had limited capability for recovering data from damaged disks

In order to address these limitations, in the summer of 2015 we procured a Kryoflux controller and specially modified 5.25" flippy drive. The Kryoflux kit consists of specialized and relatively expensive circuitry and software for forensic floppy disk imaging. It worked well in its initial testing phase but we did not have an opportunity to work with further it until my study leave. The Kryoflux requires some technical ability to use effectively and is also easy to damage. I opted to do the imaging work myself so as to ensure I fully understood its operational requirements before delegating the work to staff.

During my study leave I used the Kryoflux to image well over 300 floppy disks of varying descriptions. This encompassed two major data recovery projects and a number of minor ones. The two major projects were:

1) 90 floppy disks from the collection of Pat Salmon. This work was done at the request of Robert Amos, art critic for the Times Colonist. Pat Salmon was a longtime friend of renowned BC artist E.J. Hughes and acted as his agent from the early 1990s up to his death in 2007.[1] Documents recovered from the disks will provide important primary material for a planned series of monographs on Hughes. All the documents were created in Appleworks on an Apple II system running ProDOS. Data retrieval was done with the Kryoflux and Apple Commander, an open source program for working with Apple II disk images.

1 Pat Salmon was also at one time a librarian and art history instructor at the University of Victoria

2) 200 floppy disks from the Rona Murray fonds in Special Collections. Rona Murray was an award-winning author of poetry, plays and non-fiction who lived for most of her life in British Columbia. Special Collections acquired the disks in two separate acquisitions, one in 1995 and the other in 2005. Documents on the disks consist of drafts of poetry and fiction, biographical materials, and letters. Although there was no immediate need to access the content, imaging the disks was necessary in order to preserve copies of their contents on redundant, non-volatile storage. It also afforded a test case for developing floppy disk imaging workflows. Disks were mostly of two varieties: single density 5.25” floppies containing Wordstar files created on a Kaypro II running CP/M, and 3.5” HD floppies containing Wordstar and WordPerfect files created on a Windows PC. Data retrieval was done with the Kryoflux and open source utilities for working with CP/M and MS DOS disk images.

The minor projects included smaller collections of floppy disks in Special Collections, for example ten floppy disks in the Ann Hansen fonds; as well as work arising from the Floppy Disk clinic at Ideafest retrieving data off of a small number of disks from university researchers and the general community. I also spent some time attempting to retrieve data from a dozen problem disks in the Glenn Howarth fonds: these disks initially appeared unreadable, but further research indicated they were in fact in an unusual sector format. I was not completely successful in my attempt to recover their contents, but I did manage to retrieve a number of files, including a few digital artworks not replicated elsewhere in the collection.

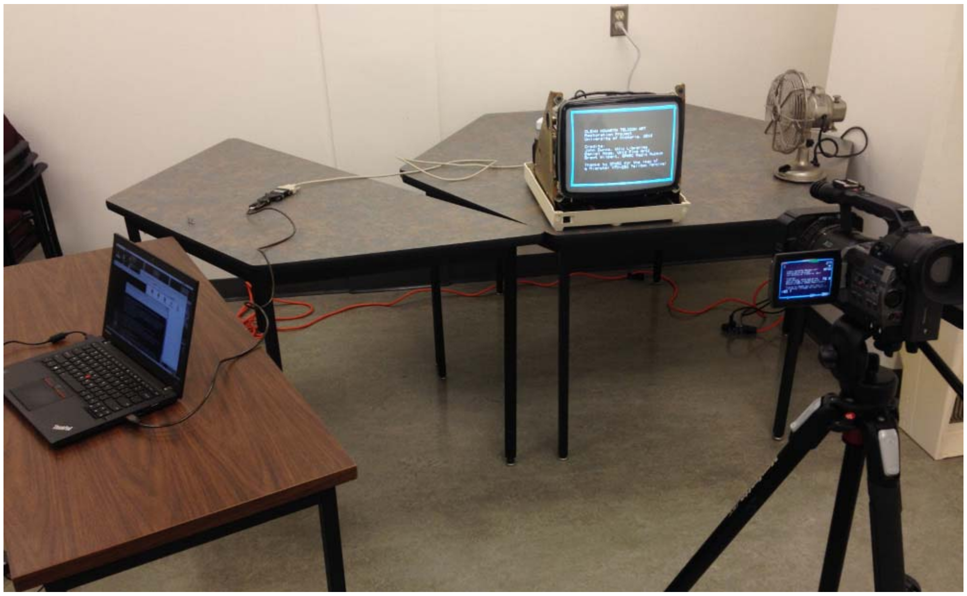
Floppy Disk Paper/Archivematica Evaluation

One of the goals for my study leave was to evaluate our archival repository software, Archivematica, with regard to its ability to ingest the kinds of forensic disk images I was creating with the Kryoflux and our other forensic workstations. My initial intention was to write a short technical paper outlining its strengths and shortcomings.

To this end I set up a test instance of Archivematica and attempted to ingest some sample disk images, without great success. In almost every case Archivematica failed to recognize the image type and was therefore unable to extract its contents or run any post-ingestion workflows. Further research into the software components Archivematica uses to manage disk image ingestion led me to realize that none of its components were designed to handle non-PC floppy formats from the 1980s. Further, this turned out to be true not only of Archivematica, but also of other toolkits like BitCurator and Islandora. They all use the same basic components for handling disk images.

I concluded that the best way forward was to create a workflow for floppy disk processing outside the Archivematica framework, closely documenting the tools and techniques I had been using in my imaging & data retrieval work to date. This would serve both as a reference for future work and a proposal for managing floppy disk imaging in Archives & Special Collections going forward. In the belief that my work might be of interest to other organizations facing similar challenges, I wrote up my work as a paper intended for publication in our professional literature. In the process my intended short technical paper grew into a comprehensive 25 page document, entitled “Digital Archaeology and/or Forensics: Working with floppy disks from the 1980s.” It has been provisionally accepted for publication in Code4Lib journal and is included as Appendix III to this study leave report.

Glenn Howarth Digital Art Restoration Project



Recording Telidon artworks, April 2016

In 2012 the Library received a donation of materials from the estate of the late Glenn Howarth, RCA. Included in the donation were the only remaining copies of digital works created by the artist in the early 1980s, in collaboration Dr. Ernest Chang and Dr. David Godfrey, both from the University of Victoria. I began a project to recover the artworks in January 2015 at the request of Caroline Riedel, curator of the University Art Collections, and Lara Wilson, University Archivist.

The artworks and the software that had been used to create and display them had been saved on two dozen 5.25” floppy disks. The files were recovered and transferred to modern, redundant storage and the reconstruction project got underway. It was determined that the artworks were created using a graphics protocol called Telidon which originated out of a Canadian effort to build a networked commercial computing utility in the late 1970s and early 1980s. Telidon was ultimately unsuccessful and the technology never gained widespread adoption. Unfortunately this had the downstream effect of limiting the current availability of software and hardware required for the restoration work. Further, we determined that despite its short-lived existence Telidon encompassed two generations of software and hardware, and the second generation was not backwards compatible with the first. Glenn Howarth's artworks spanned both generations.

After further research I worked out a method for displaying the second generation artworks (Telidon 709/NAPLPS) on modern computers using an emulated DOS environment (DOSBox) and shareware from the now defunct Simtel archives (MicroStar PP3). During my study leave I created a technical note describing the process (see Appendix II) so no further detail is presented in this overview.[2]

2 Also published in UVicSpace, https://dspace.library.uvic.ca/handle/1828/7340

However the first generation works (Telidon 699) presented a more difficult problem. As far as I could determine, it was not to possible to render Telidon 699 images purely in software.[3] Due to the relative sophistication of Telidon graphics (for the time) and the limitations of consumer-grade computers in the early 1980s, a peripheral called a “Telidon decoder”, was required to display the images. Again thanks to the underwhelming adoption of Telidon very few of these units were produced; records show that approximately 250 Telidon Terminals were used during the BC field trials in the early 1980s.[4] It was therefore extremely fortunate that I was able to track down a functioning Telidon decoder in the collection of the SPARC Radio Museum in Coquitlam, along with a highly capable volunteer at the museum (Brent Hilpert) willing to assist with the operation and maintenance of the 35 year old piece of equipment.[5]

Restoration of the Telidon 699 works began in earnest during my study leave. In late January I made a trip over to the SPARC museum and spent a couple of hours working with Brent to connect my laptop to the decoder and ensure I could transmit image files. This was not entirely straightforward as it required multiple adapters and a specially configured null modem device to send data from a USB port on my laptop to the DB25 serial port on the decoder.

I transported the decoder back to Victoria and the following week began developing the necessary software to transmit the images to the decoder in the correct sequence and with the correct timings. Although I was unable to use the original display software (called “SHOW”) to send the images to the decoder I was able to run it in an Apple II emulator and analyze its behaviour, which was to parse a file called “com.text” included along with each set of images that specified both the order and the timings of the image display. I was then able to replicate SHOW's functionality using the python scripting language and its associated serial communications module.

I then needed to determine why the text content in the images was not rendering correctly: multiple lines of text would wrap on top of themselves or display outside of their intended blocks. I traced the problem to the control character sequences in the image files, and was then able to write a script to convert the incorrect control sequences to the correct ones. I am uncertain why this was necessary; it is possible that Howarth used a model of decoder that implemented the standard differently than our Microtel unit.

Recording the output of the decoder presented another challenge. Unlike modern computing devices, the decoder offers no obvious way of capturing the screen output in software. My research indicated that the original Telidon artists (including Glenn Howarth) had used the expedient of recording their artworks using a video camera pointed at the CRT display, suggesting a way forward here. However the camera needed to be specially modified so its refresh rate would correspond to the CRT refresh rate, otherwise the recording would be marred by a rolling shadow effect. As this kind of work was well outside my area of expertise, I enlisted Daniel Hogg, a highly competent filmmaker from the Fine Arts department to do the video recording.

---

3 Or more accurately: the software to do it never existed, and creating it now would be a non-trivial project

4 Parkhill D. 1987. The beginning of a beginning. Ottawa: Department of Communications

5 For detailed technical specifications, see Hilpert B. Microtel VTX-202 Telidon Decoder. Retrieved from: http://www.cs.ubc.ca/~hilpert/e/telidon/index.html

Unfortunately there was a setback. Just before the recording sessions were scheduled to begin, the decoder developed a display problem: the top two centimetres of its built-in CRT display collapsed. This was beyond my ability to fix and in any case it would have been unwise to tamper with equipment from the collection of another organization. The decoder needed to be transported back to Coquitlam for repair work. Brent's repairs (capacitor replacement) were ultimately successful but this delayed the project for approximately two months.

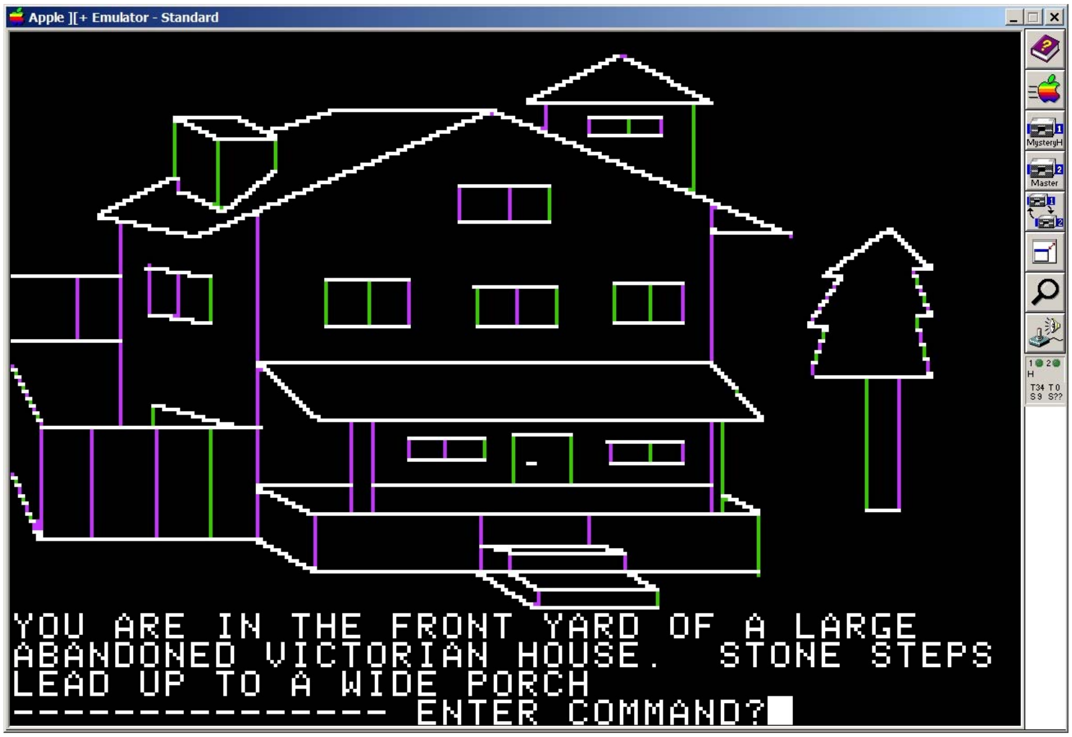
The decoder returned to the Library toward the end of April. Recording of the artworks began in earnest on April 29th and continued through the first week of May. I am pleased to report the resulting videos (approximately 2 hours in total) are both faithful recordings of Glenn Howarth's artworks and some of the best Telidon art recordings ever made, thanks in large part to Daniel's abilities and also to the fact that video recording technologies have progressed somewhat since the early 1980s.

Following the sessions the recordings were edited by Daniel, who prepared the display for a show of Glenn Howarth's artworks that opened in the Legacy Maltwood Gallery on July 30th. I sent him a screen recording of the second generation artworks to be included as well, and I wrote some exhibition notes that accompany the gallery and website displays (included here as Appendix IV). The recordings will eventually be kept along with other Glenn Howarth materials in our Archives.

The exhibition received a highly positive review in the Times Colonist on August 7th, 2016. The project to restore the digital artworks was described as a “heroic effort”.

I have presented on this project a number of times since it began last year. During my study leave I presented on it twice: once at the Archives Association of BC conference in Vancouver, and once at Ideafest 2016.

Emulation Environments



Mystery House running in AppleWin Apple][emulator

Broadly speaking, there are two ways to ensure the longevity of digital objects when their original computing environments are no longer available: emulation and format migration. Emulation involves recreating earlier generations of computing hardware in software, making it possible to run old operating systems and software on modern computers. Format migration involves translating documents from obsolete file formats to their contemporary equivalents. For example, confronted with a document in the obsolete Wordstar 3.3 format, an emulation based strategy would set up a DOS emulator such as DOSBox, install a copy of Wordstar software on it, and open the document in its original format. A format migration strategy would convert the document to a modern, well-supported format like PDF.

In earlier years (late 1990s/early 2000s) the two strategies were often pitted against each other[6], and for a time format migration was the more widely accepted of the two, forming the basis of the OAIS standard and then operationalized in software implementing that standard, for example Archivematica. More recently however emulation has been the focus of significant interest in the preservation community[7], in part because emulation has proven to be easier than early theorists supposed it would

6 For a wonderfully misguided example, see Bearman, D. 1999. Realities and chimeras in digital preservation. D-lib magazine 5(4). Available from: http://www.dlib.org/dlib/april99/bearman/04bearman.html

7 See Rieger O, Murray T, Casad M, Alexander D, Dietrich D, Kovari J, Muller L, Paolillo M, Mericle D. 2015. Preserving and Emulating Digital Art Objects. eCommons (Cornell University). Available from: https://ecommons.cornell.edu/handle/1813/41368 ; and Dietrich D, Kim J, McKeehan M, Rhonemus A. 2016. How to Party Like it's 1999: Emulation for Everyone. code{4}lib journal 32. Available from:

be; in part because format migration tends not to work so well for complex, interactive artifacts like artworks, e-literature, or games; and in part due to a growing realization that the two strategies are complementary, not conflicting. As an example of their complementarity I could mention our current strategy for converting Wordstar documents to PDF involves opening them in WordPerfect 12 running on a virtualized instance of Windows 2000 and printing to file via Adobe Acrobat 8 running in the same environment.

It is in the context of this renewed interest in emulation that I undertook a small research project to explore the challenges involved in setting up a variety of historic computing environments on contemporary hardware. I was gratified to learn that although the level of difficulty varied from one environment to the next, none posed any insurmountable problems. I chose to recreate the following environments:

Emulator	Emulated Hardware	Operating System	Software
DOSBox	PC 286, 386	DOS	Wordstar, PP3, etc
AppleWin	Apple II and variants	Apple DOS	Mystery House, SHOW, etc
Mini vMac	Mac Motorola 68K	Mac OS 6	Agrippa, Afternoon, etc
Basilisk II	Mac PowerPC	Mac OS 7.5	Shakespeare's Life and Times, Hypercard
Kegs	Apple IIgs	Apple ProDOS	Teach
Atari800	Atari XE	Atari OS	AtariWriter
VirtualBox	PC 32-bit	Windows 2000, Windows XP	WordPerfect, ClarisWorks, Netscape, Myst, etc
SIMH	DEC PDP-10	TOPS-10	Adventure
WinUAE*	Commodore Amiga (all versions)	Amiga OS	Arkanoid, etc
VICE*	Commodore 64, Vic 20	Commodore Kernal/Basic	Zork, etc

* Commercial version, packaged/licensed by Cloanto

These environments were chosen either because they were required for restoring digital objects in our collection (DOSbox, AppleWin, Mini vMac and VirtualBox), because they were required to restore digital objects from community researchers (Atari800, Kegs), or in a few cases simply because they were of historic interest (WinUAE, VICE and SIMH).

As noted above, none of the emulators I tested involved any insurmountable technical challenges and in some cases posed no challenges at all. The fact that I was able to use them to restore digital artifacts from both our Archives and our general collection testifies to their utility in the context of digital preservation.[8] However, there are some issues to be aware of when it comes to emulation:

http://journal.code4lib.org/articles/11386

8 I have already mentioned the use of DOSbox and AppleWin in the context of the Glenn Howarth project. See the Atari XE section of my paper on Floppy Disks for an example of the use of emulation to restore materials for a community researcher

1. Emulation environments are of variable quality. In many cases the systems emulators seek to re-implement were never publicly documented, which means that emulation software developers must reverse-engineer the original systems. Emulated systems may therefore be more erratic or unstable than the original systems. Of the systems I tested, Basilisk II was the most failure-prone, and I encountered at least one case where DOSbox was unable to run software that ran well in native DOS.

2. Emulators do not exist for some systems. For example, there are few if any emulators that can run Mac OS 9, and none that officially support Mac OS X (although it can be done in VirtualBox). Emulating Windows 95 can also be challenging.

3. There are legal issues: it is not possible to legally emulate a system for which you do not have the necessary rights. While some emulators like DOSBox emulate both the hardware and operating system, other emulators only emulate the hardware layer. These typically require two components in addition to the emulator itself: a boot ROM from the system you are seeking to emulate, and a copy of the original operating system. Emulators are generally freely and legally available as they are usually developed as open source projects by volunteer communities. However boot ROMs and Operating Systems are still under copyright to the original companies that developed them or their successors. In order to be able to use them in an emulated context you need to own licensed copies, which typically means owning the original hardware. Fortunately at the UVic Libraries we have a developing collection of computing hardware from the 80s and 90s and can legally emulate all of the systems listed above.

Ideafest 2016: Retro Computing in the Library

Dr. Ernest Chang & Campbell Good, creators of the software Glenn Howarth used to make his digital art, beside a Telidon terminal at Ideafest 2016 (photo by Robert Amos)

My proposal for our Ideafest event "Retro Computing in the Library" was put forward in the summer of 2015, and so was not included in my original study leave proposal that was submitted several weeks earlier. Nevertheless I believed such an event would have the potential to focus and augment my study leave activities so I was pleased when the proposal was accepted in the fall.

The event was held on March 10, 2016 from 1:00-4:00pm. There was a good turnout (approximately 100 people over the course of 3 hours) and positive feedback from the attendees.

My involvement with Ideafest can be categorized in the following ways:

1. Coordination: I worked with the University's Ideafest coordinators, Tara Todesco and Nicholas Wood, to ensure they had all of the information they required to schedule and promote our event. I worked with Libraries Facilities and Communications to identify appropriate space for our event and develop a communications strategy. I also coordinated the various exhibits, exhibitors, and facilitators whose contributions greatly helped to make the event the success that it was.

2. Promotion: I gave two interviews to local media. An article based on the first interview appeared in the Times Colonist the Sunday before Ideafest[9], and a recording of the second interview was broadcast twice on CBC's “On the Island” morning show, once on March 9th and again on the 10th. Both the article and the interview helped to boost attendance. I also created the digitally-modified photograph that was used for the Ideafest poster and web site.

3. Exhibits: There were 19 exhibits at the event, ranging from emulated environments to period hardware to artworks. I was directly responsible for creating many of them. Most of the computing environments I enumerated in the section “Emulation Environments” above were featured in the show, along with information sheets explaining their significance (included in Appendix V). In addition I set up two displays of Glenn Howarth's Telidon graphics and put together a small display of historic computer storage technologies using artifacts from our own collection as well as items borrowed from Humanities Computing and University Systems.

There were also a number of exhibits which I did not myself create. These included the following:

- DOS Workstation prepared by Jerry Trofimchuk
- Windows 95 workstation prepared by Jerry Trofimchuk
- Windows 98 workstation prepared by Jerry Trofimchuk
- Apple IIc computer from the personal collection of Rich McCue
- Arcade Emulator created by Rich McCue
- Nintendo Entertainment System from the personal collection of Jennifer McClintick

4. Presentations: I presented twice at Ideafest. The first presentation was a short welcome/introduction to the event; the second was a longer talk on the Glenn Howarth art restoration project. Rich McCue also presented on the construction of his Arcade Emulator.

5. Floppy disk clinic: Following the presentations, the second half of our Ideafest event included a floppy disk clinic where members of the community were encouraged to bring in their floppy disks to see how much data we could recover from them. Five people did. I did the data recovery work. Disks included Mac HD 3.5” floppies, PC floppies of both the 3.5 and 5.25” varieties, and a couple of Atari disks. Some of the work was done at the event and some was done afterward. Everyone's data was recovered.

9 Amos R. (2016) Retro artworks, high-tech lace. Times Colonist March 15, 2016. Available from: http://www.timescolonist.com/life/islander/robert-amos-retro-artworks-high-tech-lace-1.2191339 A slightly revised version of the same article was published in Whitehot Magazine, August 2016: http://whitehotmagazine.com/articles/one-victoria-s-leading-artists/3459

Historic Computing Equipment

Digital Equipment Corporation VT125 Terminal (1984) recovered from Elliot Building

Developing a collection of historic computing equipment was not part of the original plan for my study leave, but when the opportunity arose it seemed another good way to build our capacities in digital archaeology. Just prior to Ideafest, and directly as a result of the publicity we were getting, I was contacted by Stephen Wylie, Surplus Coordinator in Purchasing Services, who indicated that he had some old equipment on hand that might interest us. There was not enough time to evaluate the equipment prior to Ideafest, but we made an appointment for Jerry Trofimchuk and I to visit the Purchasing warehouse a couple of weeks afterward.

Over the past 10 years Stephen has been setting aside some of the more interesting discards that came his way. These included eight compact Macs from the late 1980s and early 90s, four 100 series Mac Powerbooks, two G4 Cubes, three DOS laptops from the late 1980s/early 90s, a NeXT workstation, a micro TRS-80, and a wide range of peripherals, add-on cards, documentation, software and related materials. Jerry and I did a preliminary evaluation and arranged for the most interesting artifacts to be delivered to the Library.

The artifacts arrived on five carts on April 12th. Over the course of the next two months (and in between working on other projects) I compiled an inventory of the equipment we received (Appendix VI), evaluated its operational condition (not all of the equipment was functional), determined what we should keep, and found storage space for it. Subsequent to that delivery a few more items have surfaced on campus and have been added to the inventory, including a Windows 95 workstation, a Sun Ultra Creator 1 workstation, and a DEC VT125 terminal.

Following the initial evaluation Jerry Trofimchuk and I did some basic maintenance work on the hardware, including:

- Cleaning, dust removal
- Removal/replacement of old CMOS batteries
- Procurement of a 9” Torx T-15 screwdriver to open the compact Mac cases
- Creation of a set of boot floppies for the Mac Plus computers
- Procurement of an original Mac Plus keyboard and cord (via Ebay)
- Procurement of an LCD monitor adapter for the Sun Ultra
- Procurement of an S-Video/Composite adapter for the Commodore 64

Currently four of the compact Macs are fully operational, along with one of the G4 cubes and the Windows 95 workstation. Other equipment is in various states of partial functionality, or missing key components such as power supplies so its condition is uncertain. Some of the non-functional equipment is being returned to Purchasing Services for final disposal, in other cases it is being kept for spare parts or considered for restoration.

The utility of the equipment in the Library context has yet to be fully determined. Here are some possibilities:

1. Accessing content on older media. While the Kryoflux remains the single most capable device we have for retrieving content from floppy disks, I have encountered a number of instances where other devices worked better for retrieving content off of specific disks. There is significant variation not only between floppy disks created on different kinds of systems, but in fact between floppy disks created on different instances of the same system, due to varying drive alignments and manufacturing variances. It is therefore advisable to have a variety of options when it comes to retrieving data from obsolete media.

2. Digital history workshops. The success of the Library's Ideafest exhibit clearly demonstrates an interest in historic computing technologies across the University and broader community. Through workshops and related events the Library has the opportunity to further scholarly and community engagement with our digital past. Given the interest in this area in digital humanities, there may be a natural fit with DHSI.[10]

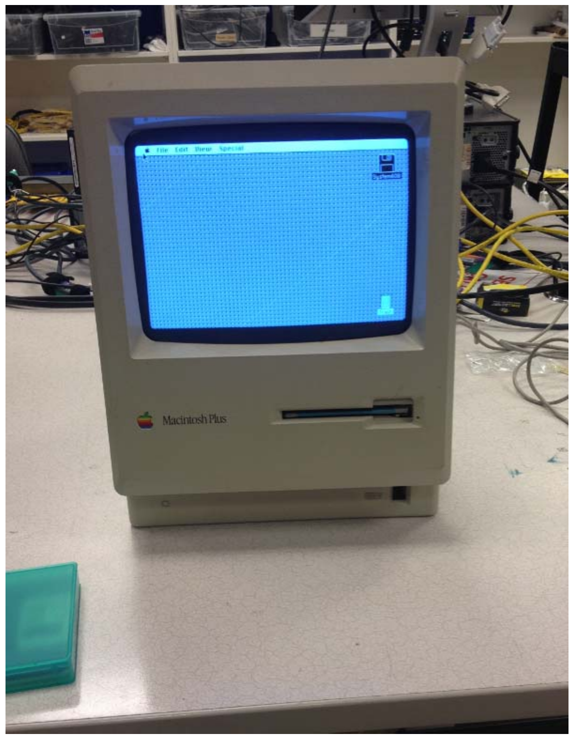
3. Study/exhibition of digital artifacts in their original context. Some communities of scholars, notably the E-Literature community, maintain collections of early computing equipment in order to experience the objects of their study in their original context.[11] We have some examples of e-lit and other digital artworks in our collection, and I believe there is some interest in broadening the scope of that collection going forward. This equipment could serve to showcase that work as it was originally intended to be seen.

---

10 See for example the work of Matthew Kirschenbaum, specifically his monographs *Mechanisms* (2007) and *Track Changes (2016).* Also see MIT's Platform Studies series, https://mitpress.mit.edu/books/series/platform-studies

11 One example being the Electronic Literature Lab (ELL) at Washington State University: http://dtc-wsuv.org/wp/ell/

4. Emulation. As noted above, in order to legally emulate some computing systems (eg. Macs) it is necessary to own the original hardware. Examples given above clearly demonstrate the utility of emulation for preserving and accessing older digital content.



Mac Plus booting System 6.08 from an 800K floppy disk

Outcomes/Next steps

There are five immediate outcomes stemming from the work described above:

1. The Glenn Howarth exhibition opened July 30th in the Legacy Maltwood in McPherson and runs until October 23rd. Three sequences of his newly-restored digital artworks are displayed in a loop; total running time is approximately 45 minutes. Clips from the videos will be featured in an associated web exhibition.

2. I will meet with the Libraries' Digital Preservation Working Group to go over the findings from my study leave and discuss how best to integrate the processes I have worked out with the preservation of legacy digital objects in our collection. We will also consider how the hundreds of floppy disk images I have acquired and processed during my leave should best be stored for long term preservation. Currently they reside on the university's storage area network, but should likely go into Archivematica at some point. There are still several floppy disks and CDs in the collection which need to be imaged although by far the largest part of that work has now been done.

3. I will revise my paper *Digital Archaeology and/or Forensics* in consultation with the editorial team of Code4Lib Journal, to prepare it for publication in the forthcoming October issue

4. I will continue to participate in the SSHRC-funded Endings Project as part of a team studying how best to preserve digital humanities projects for the long term. A possible sub-project is being discussed regarding the use of in-browser emulation to preserve an early work of digital scholarship, “Shakespeare's Life and Times,” the CD ROM precursor to the Internet Shakespeare editions.

5. I am in discussion with David Leach, Writing department chair and Director of the Technology & Society program, to determine whether an introduction to emulation and/or historic hardware would be useful to students of *WRIT 324: Writing Interactive Narrative* as they explore early examples of electronic literature and related digital artifacts.

Longer term we will need to consider how closely aligned developing and maintaining a centre of expertise in historic computing is with the Libraries' strategic objectives. I believe such a centre would enhance our ability to contribute to the cultural mission of the Libraries and the broader university, and strengthen connections with communities of scholars particularly in the digital humanities and related disciplines. It would also be unique in the region, although some precedents exist elsewhere.[12] However it would require a modest commitment of resources, particularly with regard to storing and maintaining legacy equipment.

---

12 See for exampe retroTech at Georgia Tech Library, http://retrotech.library.gatech.edu. Our collection of vintage equipment surpasses theirs, although they are further along in program development.

Appendix I
Study leave proposal, January 1 – June 30 2016
John Durno
Application initiated May 8, 2015

Leave Abstract:	I will work with early computer media in our Archives, working with Archives staff to identify at-risk digital content and develop strategies for preservation and access over the long term. My main focus will be on creating software environments for working with obsolete file formats. As a case study I will focus on the restoration of digital artworks in the Glenn Howarth artists archive, which present interesting challenges as they are in an obsolete, partially hardware-dependent file format stored on a variety of media types. These images are unique content of historic significance to the University and the larger community and will be featured in an exhibition at the Legacy Maltwood in the spring of 2016. I also intend to evaluate the ability of our digital preservation software, Archivematica, to ingest and manage forensic disk images derived from early computer media. I believe this leave proposal to be directly aligned with the strategic objectives of the Library in the area of digital preservation and community engagement, and also with my professional focus on the intersection of libraries and computing technologies.
Planned Itinerary:	All of this work can be done locally. Time permitting I may undertake site visits to other cognate organizations working with similar technologies.
Leave Collaboration:	This project requires collaboration with the staff of University Archives and Special Collections in the McPherson Library. I have consulted with the Director, Special Collections & University Archivist, the Associate Archivist, and the Associate Director of Special Collections and have confirmed they are amenable to this proposed project.
Funding Opportunities:	This project does not require external funding.

Appendix II

Displaying NAPLPS (Telidon 709) graphics on a modern computer: Technical note

John Durno (jdurno@uvic.ca)
University of Victoria Libraries
January 2016
Version 1.0

Background

Telidon was a project of the Communications Research Centre of the Canadian Federal Government Department of Communications. Lasting from 1978 through 1985, it promoted the creation of consumer Videotex networks using Canadian technology for the transmission of high quality (for the time) graphics and text in the context of highly limited bandwidth and processing power. [Wikipedia, n.d]

During the project's existence its image/text encoding standard went through two major versions. The first version, Telidon 699 (named for CRC Technical Note 699) dated from November 1979. The second version, Telidon 709, superseded 699 in 1982. Telidon 709 became NAPLPS, the North American Presentation Level Protocol Syntax. As its name implies, NAPLPS was a North America-wide standard, due to the involvement of US entities such as AT&T who saw potential in the Canadian technology.

A significant amount of content was created for the Telidon project. Educators, artists, writers and businesses developed Telidon content across a wide range of disciplines, covering topics as diverse as Business Education, Computer Literacy, Fine Arts, Geography, Math and the Social Sciences [Consortel Catalogue, 1985]. It is uncertain how much of this content still exists.

NAPLPS content continued to be created in the years immediately following the end of the Telidon project, as NAPLPS had a post-Telidon life in the Bulletin Board Systems that were popular during the latter 1980s and early 1990s. NAPLPS graphics proved to be well-suited for 1200 BPS transmission speeds and low-powered home systems common in that era of computing. [Hughes, 1993]

In 2015, the University of Victoria Archives began work on restoring some early-80s Telidon graphics created by local artist Glenn Howarth. During the course of that work we determined that it is still (as of 2016) possible to render NAPLPS graphics on contemporary computing systems using readily available shareware (Microstar PP3) running in an open source DOS emulator (DOSbox). Due to the cross-platform compatibility of DOSbox it is possible to display NAPLPS graphics on all major desktop operating systems currently in use (Windows, Linux and Mac OS). This technical note documents our method in sufficient detail that others seeking to display NAPLPS graphics should be able to do so by following the instructions below.

Note that this method does not work for Telidon 699 graphics, as the 709/NAPLPS standard was not fully backwards compatible. As of this writing, no software decoder for Telidon 699 graphics is known to exist, and hardware decoders are exceedingly rare.

Please note: Versions of DOSBox, PP3 and Stackey3 may have been distributed with this paper. However the version of DOSBox included here may be out of date; it is advisable to use the most recent version available.

Overview

The method described herein relies on the following third-party software programs:

1. DOSBox, an open source x86 emulator with DOS.

2. Microstar Personality Plus 3 (PP3), a shareware bulletin board client.

3 Optionally, Stackey3, a shareware DOS macro utility

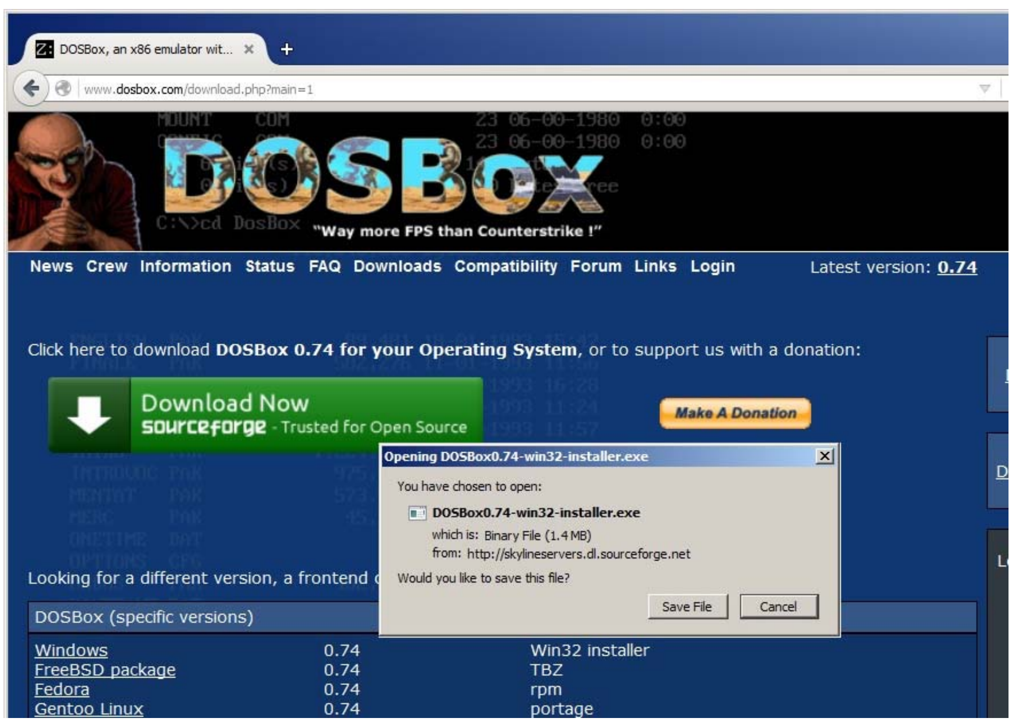
Briefly, the method involves downloading and installing DOSBox, downloading PP3, running PP3 from within DOSBox, and then entering a combination of PP3 and DOSBox commands to display your NAPLPS files optimally. You will likely need to rename your NAPLPS files prior to displaying them if you want to invoke PP3's slideshow capability. You can automate the process of starting up a slideshow using Stackey3 and the DOSBox config file.

Step by step

1. Download and install DOSBox.

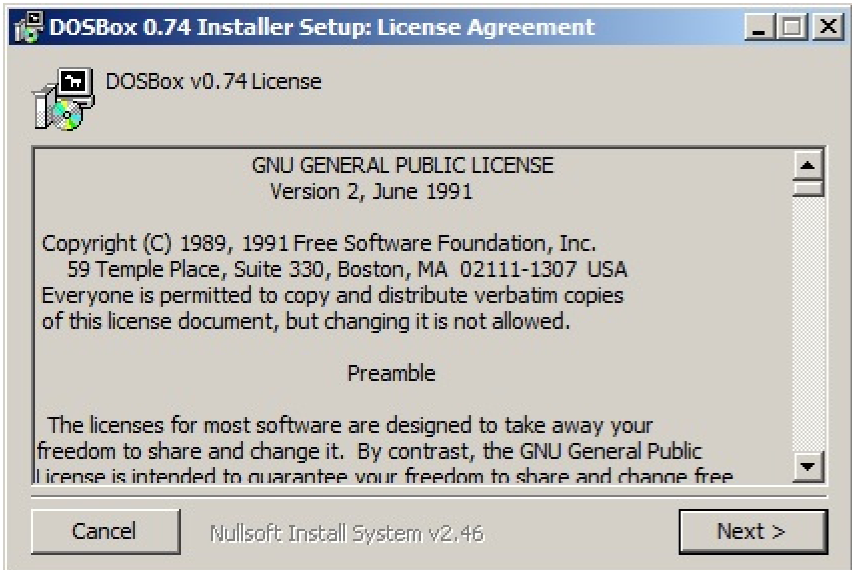
DOSBox is available from http://www.dosbox.com. It is available for all common operating systems including Windows, Mac OSX and Linux. This technical note describes a typical Windows installation.

Download the correct version for your operating system from the DOSBox downloads page:

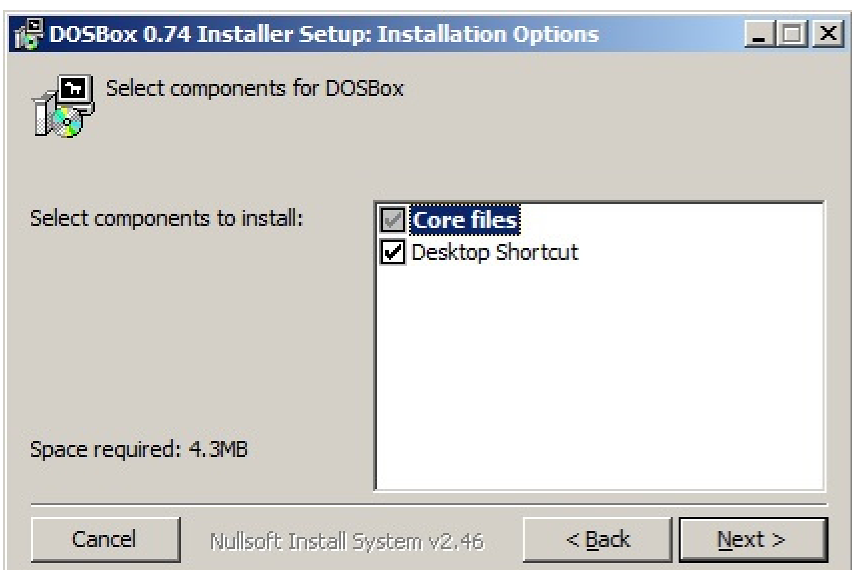


Once the file has been downloaded, click on it to begin the installation process. You will likely be prompted for your Windows administrator password at this point.

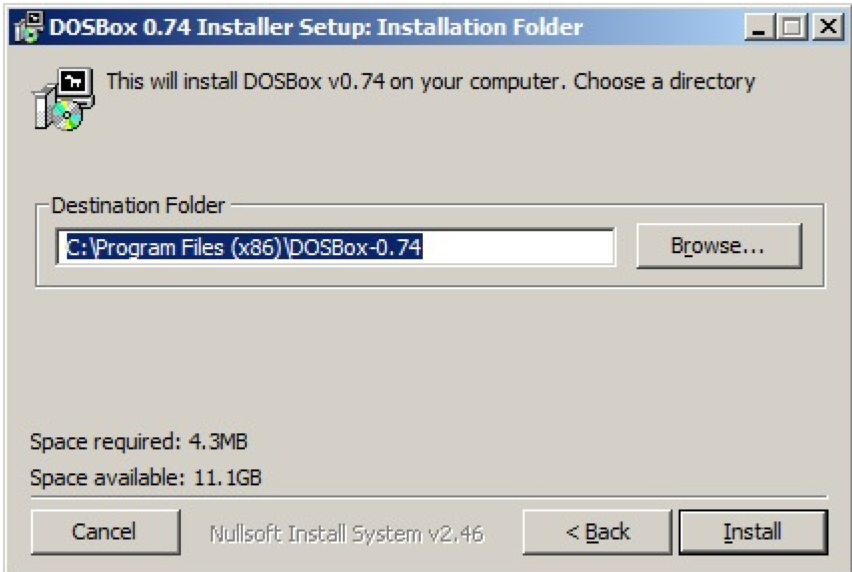
The first dialogue box is the license agreement. DOSBox is licensed under the GNU GPL 2, a standard open source license. Click Next to accept the license terms:



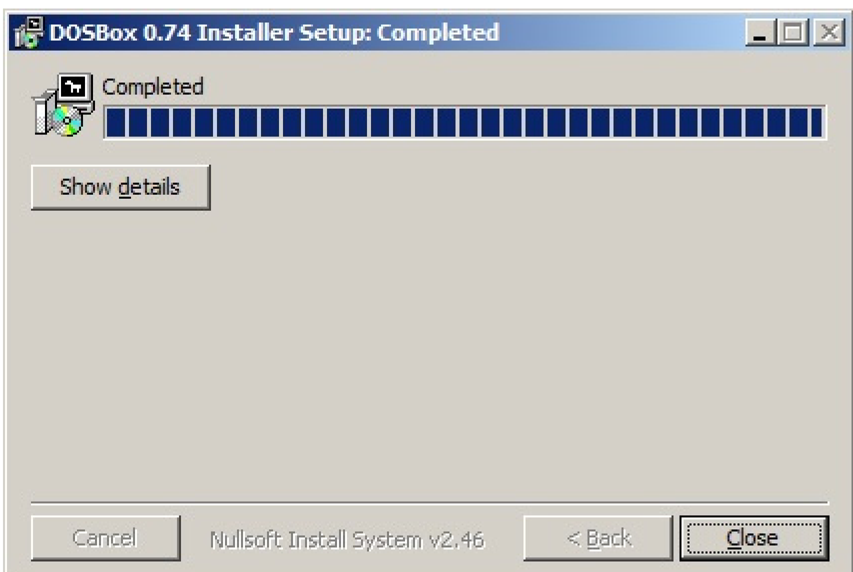
There is only one meaningful installation option. I want to add a DOSBox shortcut to the desktop, so I will leave that option checked, and then click *Next*



Next choose the installation folder. The default should be fine. Click *Install*.



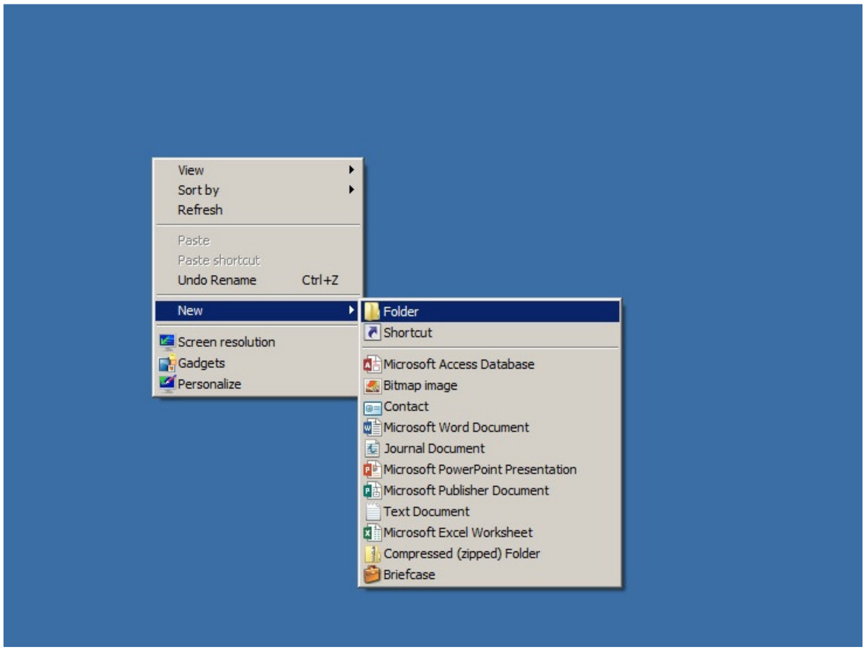
DOSBox installation is complete. Click Close to close the dialogue box.



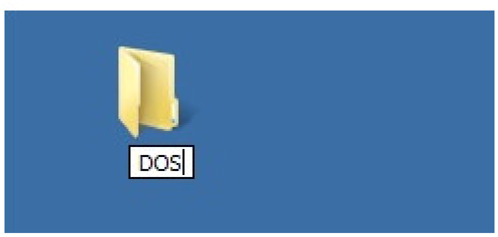
2. Create a suitable directory to contain the PP3 software and any NAPLPS files you want to display.

You will need to create a directory that is accessible to DOSBox. The directory can be called anything you want and be located anywhere you have permissions to read and write files. In this case, we will create a directory called “DOS” located on the desktop.

Right-click on the Desktop and select New->Folder from the pop-up menu.

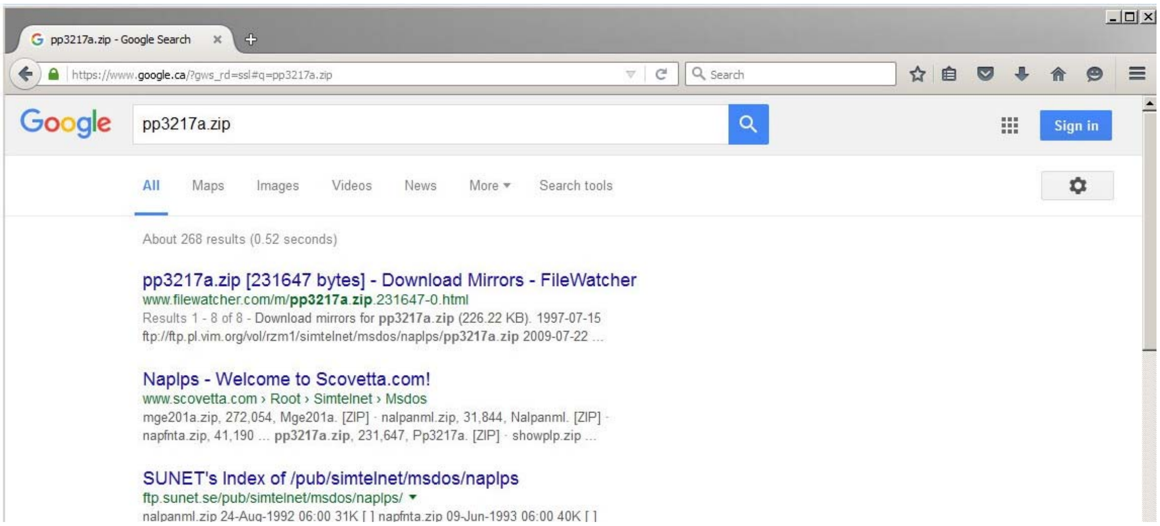


Name the new folder *DOS*:

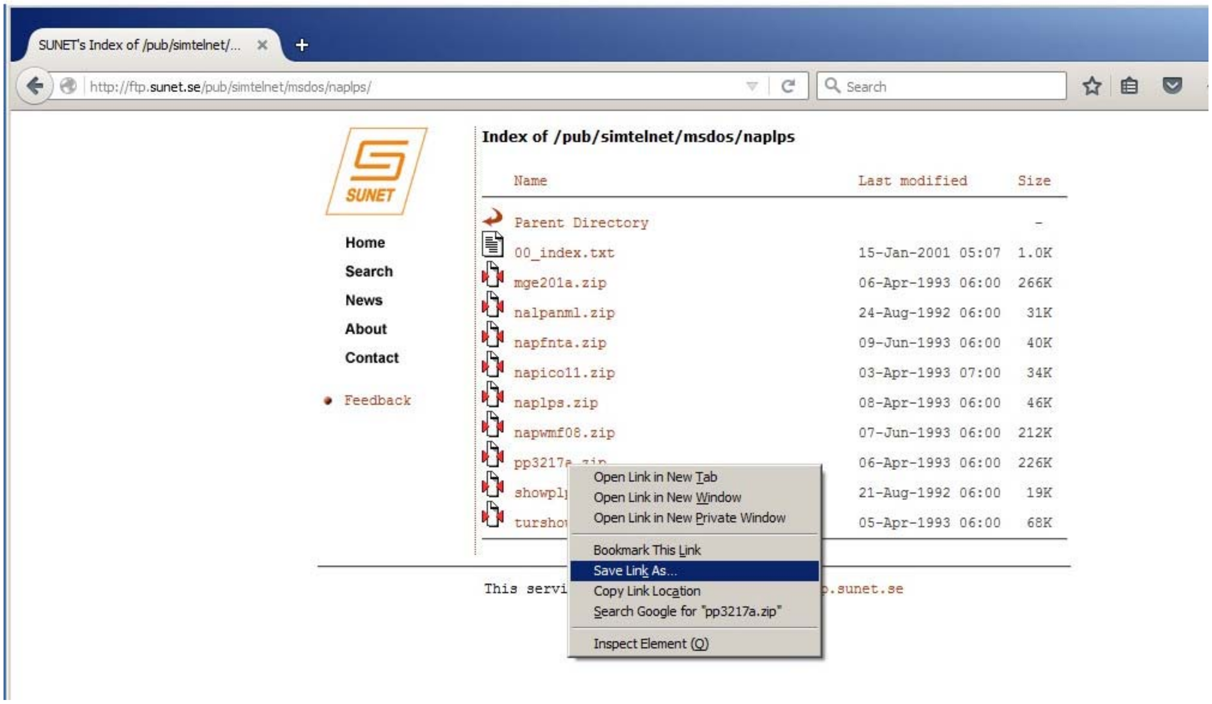


3. Download and unzip PP3 in the DOS directory.

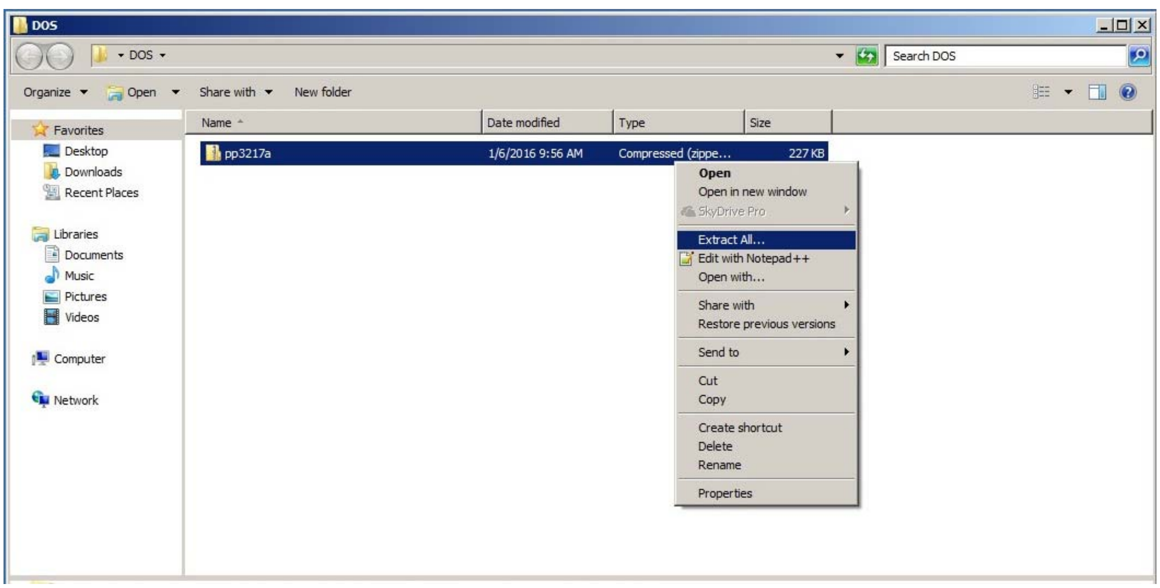
In its time (the early 1990s) PP3 was widely distributed via the Simtel DOS shareware archive. The Simtel archives are no longer being actively maintained or officially hosted, but mirrors still exist. You can try http://cd.textfiles.com/simtel/simtel20/MSDOS/NAPLPS/.index.html or if that's not working the easiest way to find the software is to search for its filename (pp3217a.zip) in Google:



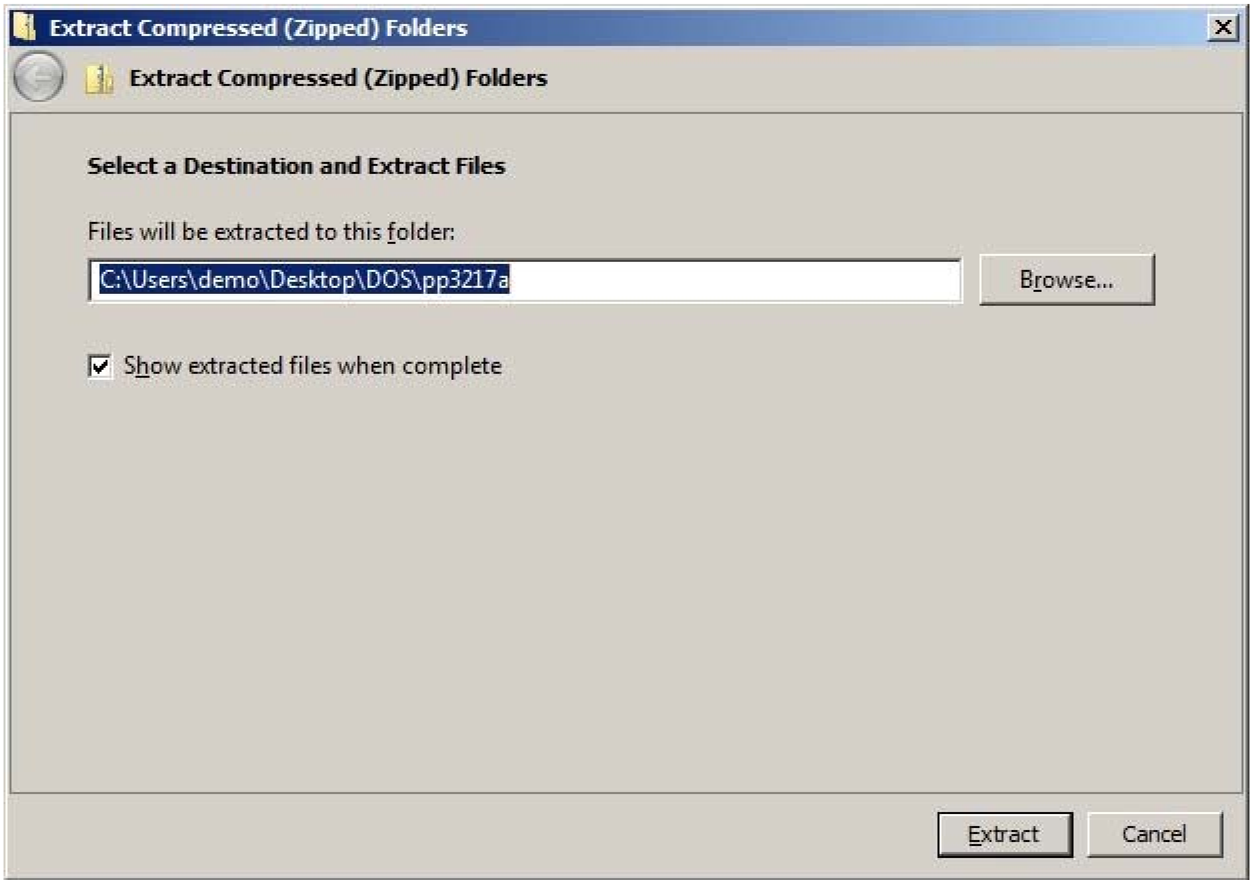
Connect to one of the mirror sites you found and download the file pp3217a.zip to the DOS directory you created in the previous step. If you're using Firefox, you can right-click on the file and select "Save Link As ..." from the pop up menu. This will give you the option to select the DOS directory on your Desktop as the save location.



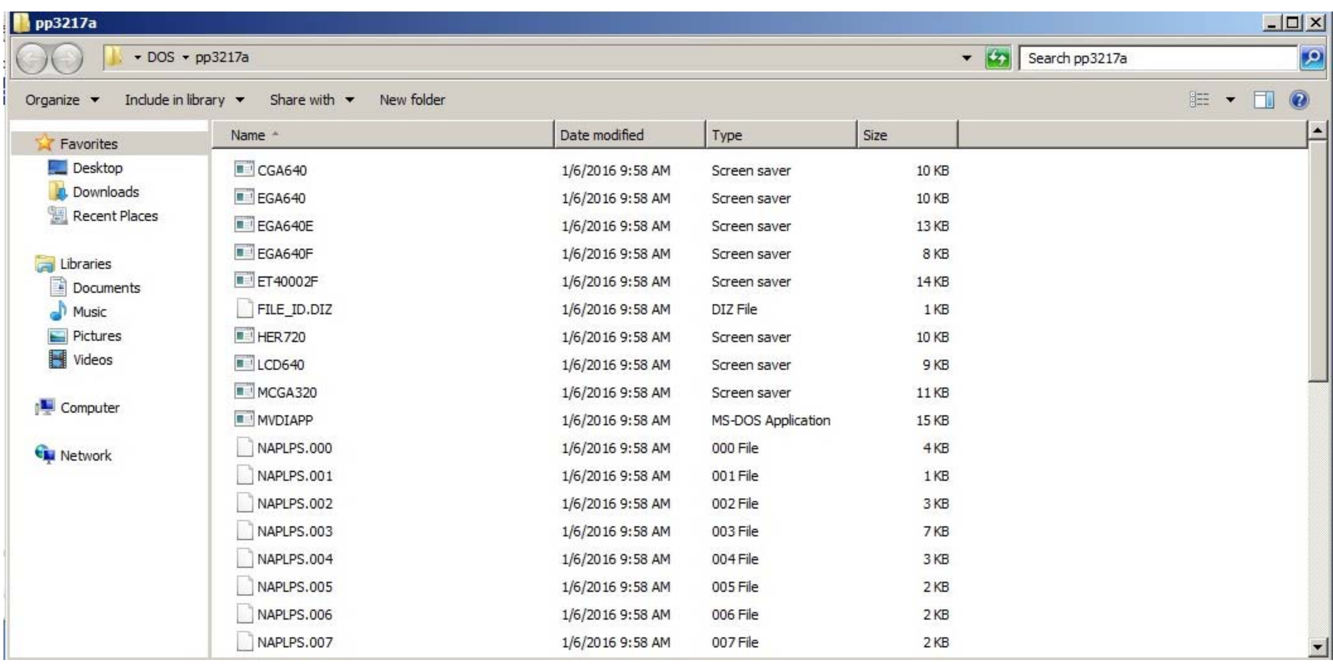
After the file has been downloaded to the DOS directory, you will need to uncompress it. Right-click the pp3217a.zip file and select “Extract all” from the pop-up menu.



You will be prompted to select a location for the uncompressed files. The default should be fine. Click “Extract” to complete the operation.



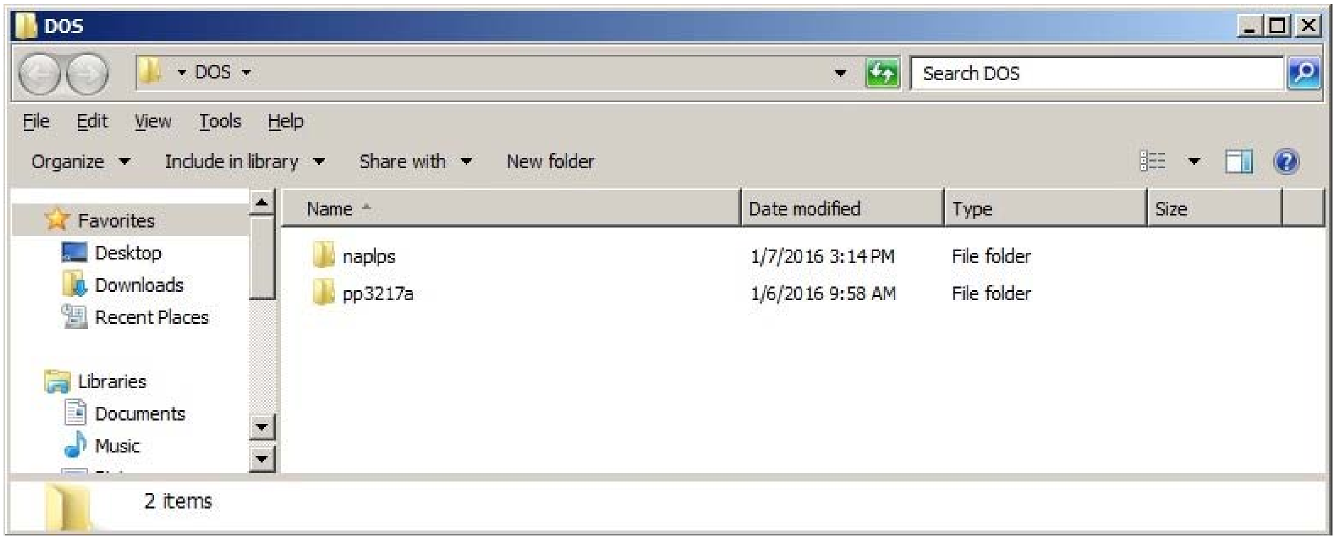
This operation will create a folder called pp3217a in the DOS folder on your Desktop. It contains the files for the PP3 application.



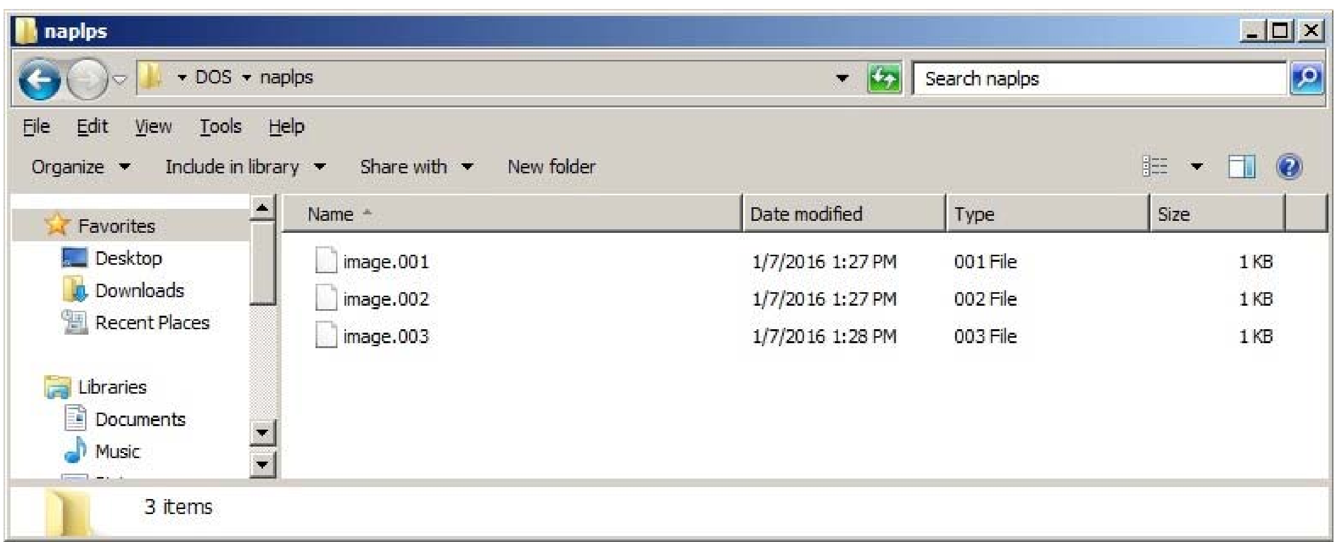
You can remove the compressed file pp3217a.zip from the DOS directory at this point. We will have no further need of it.

4. Put your NAPLPS images in one or more directories adjacent to the PP3 directory.

In this example, our NAPLPS files will be located in a folder called “naplps”.



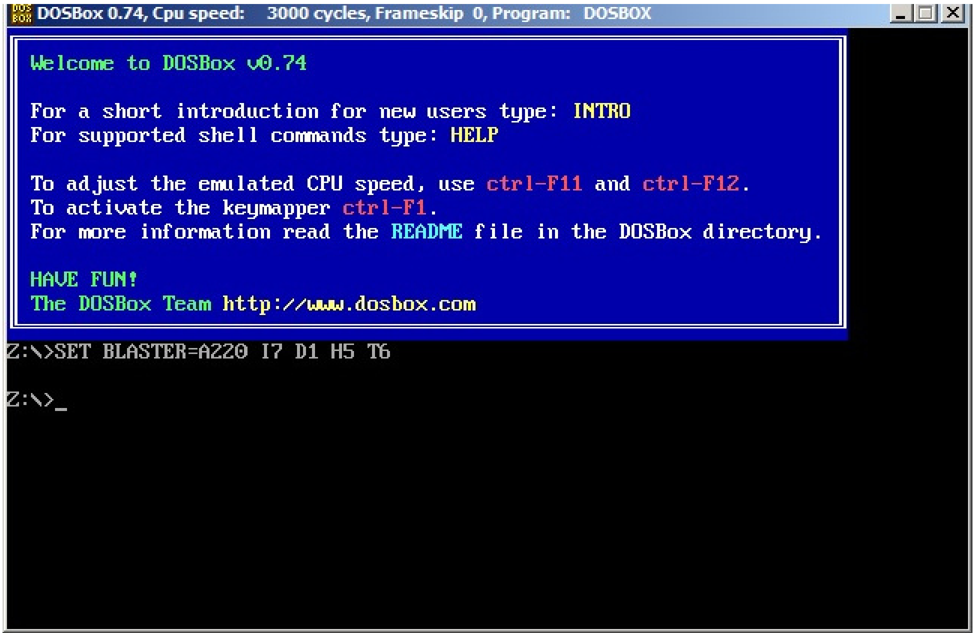
Inside the folder there are three files, named image.001, image.002 and image.003. The files are named according to the convention expected by the PP3 slideshow function: all the files must have a maximum 8 character filename followed by a three-digit numeric extension in ascending order.



Since this is a non-standard way of naming NAPLPS files, it is likely you will need to rename your files before you can display them in PP3. You can either do this manually, or use one of many available tools for automating the renaming operation. Coverage of those tools is beyond the scope of this technical note.

5. Run DOSBox

If you followed the DOSbox installation method above, you will have a DOSBox icon on your Desktop. Click that to launch DOSBox. When DOSBox starts up, it looks like this:



Initially you have no access to your local hard drive. You must first mount part of your local drive so DOSBox can see it. In this example, we will mount the DOS directory we created in step 2 above. We do this by typing the following command at the Z:\> prompt:

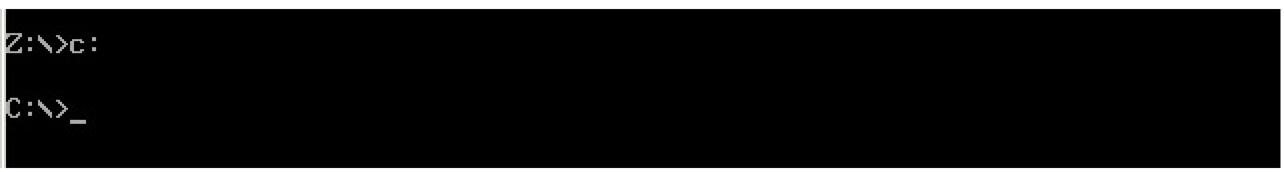
mount c c:\users\demo\Desktop\DOS

Then press enter. DOSBox will tell you if the mount operation was successful.

In this example, my home directory is called 'demo'. You will of course need to substitute the name of your home directory to run the command on your own system, like so:

mount c c:\users\<your home directory>\Desktop\DOS

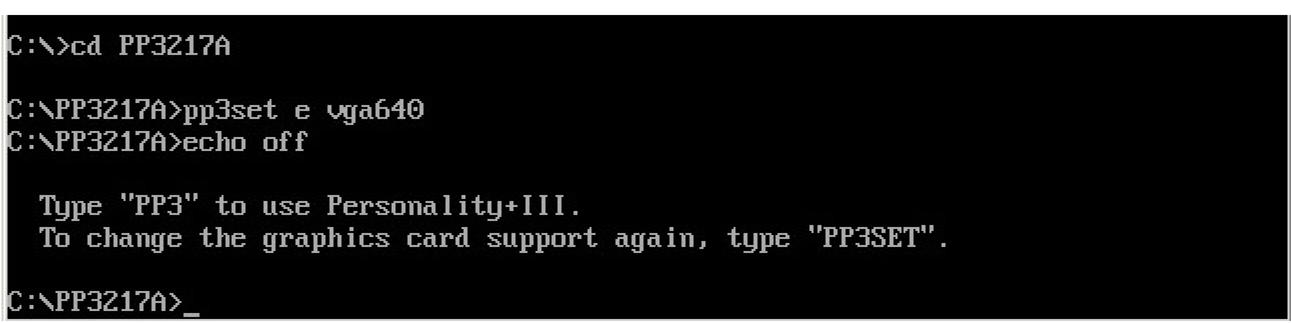
Switch to your newly mounted C: drive by typing c: at the Z:\> prompt. This will change the Z:\> prompt to a C:\> prompt.



6. Run PP3 from within DOSBox

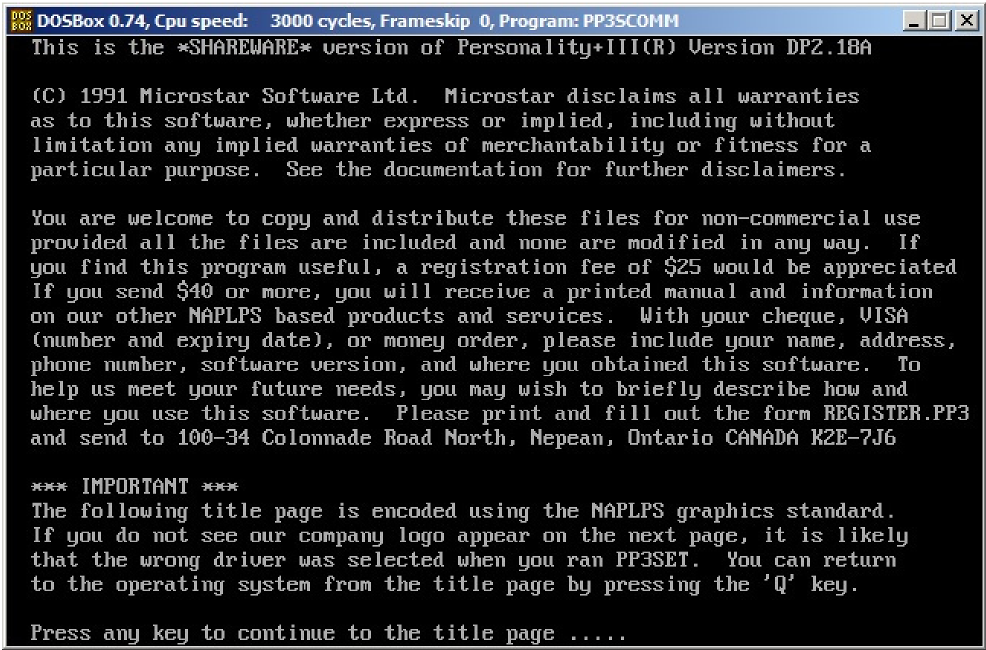
The PP3 program is located within the directory 'pp3217a' at the root of our C: drive. Change into the pp3217a directory by typing 'cd PP3217A' at the prompt.

If this is the first time you have run PP3, you will need to first set the video mode before you can run the program. You can do that by typing 'pp3set e vga640' at the prompt after you've changed into the pp3217a directory. The program will respond with a couple of comments as below:



Now you can run PP3. Type 'pp3' at the C: prompt:

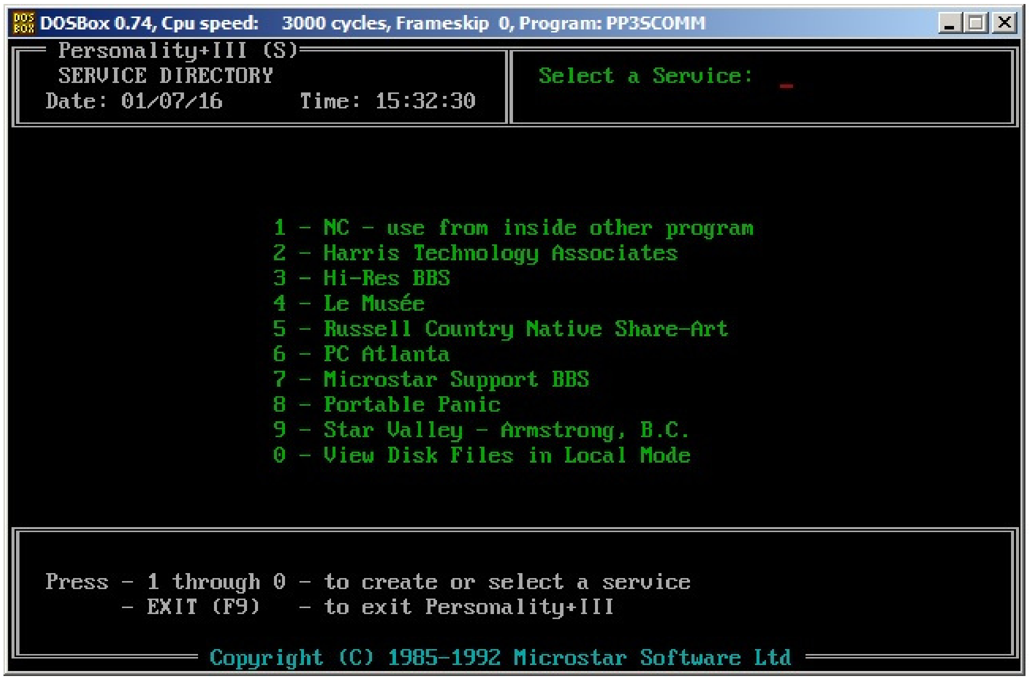
The program launches by displaying the licensing terms. Press any key to go to the next screen.



The next screen is the splash screen. Press any key to continue.

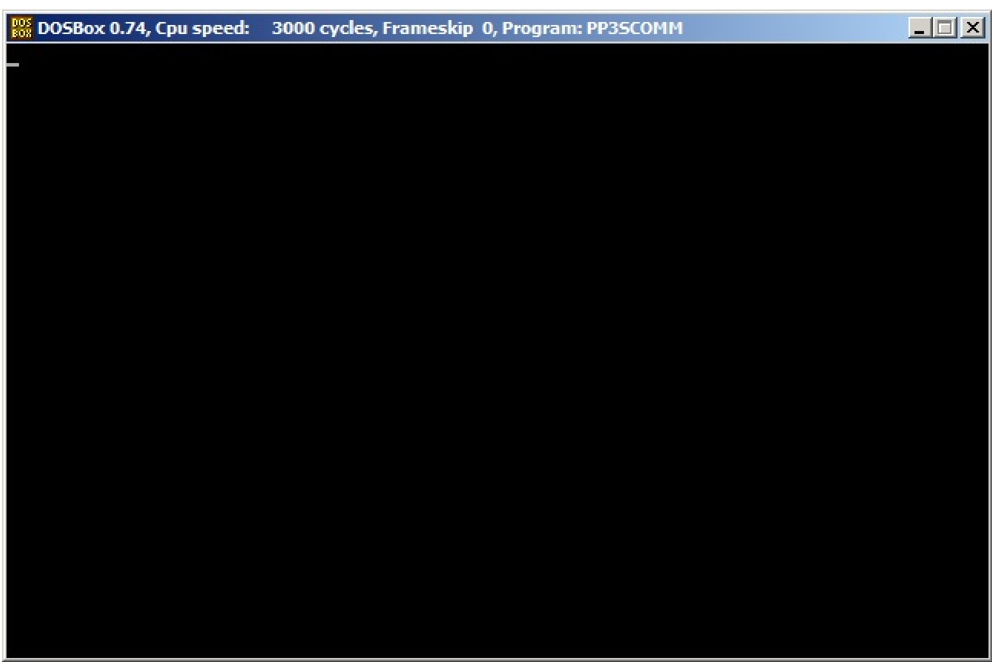
7. Displaying NAPLPS Graphics

Once past the splash screen, PP3 displays a menu of options:

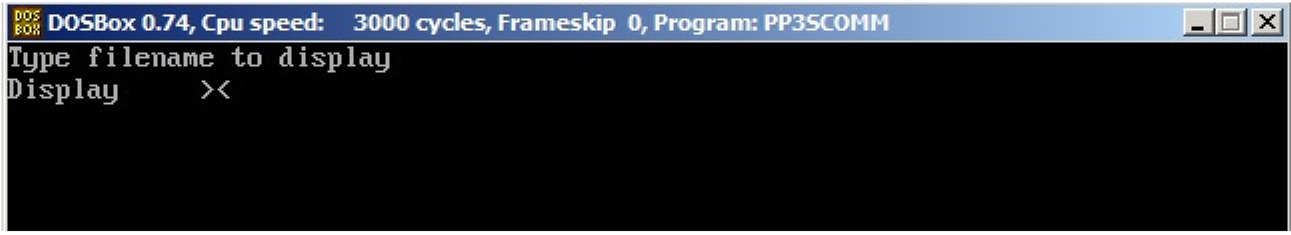


The last option, "View Disk Files in Local Mode," is the one we're interested in. Press 0 to select it.

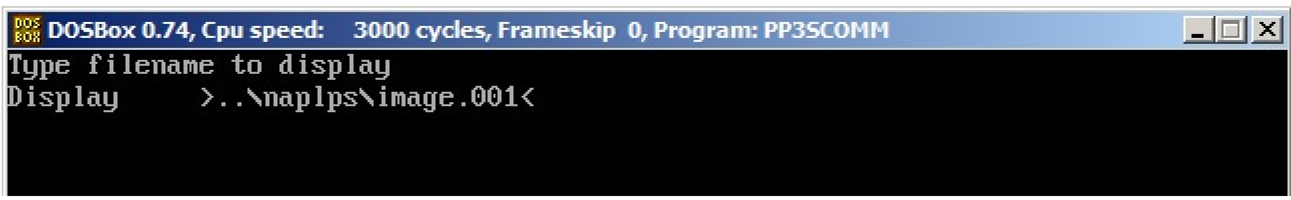
You will be greeted with a blank screen and a blinking cursor. This is normal:



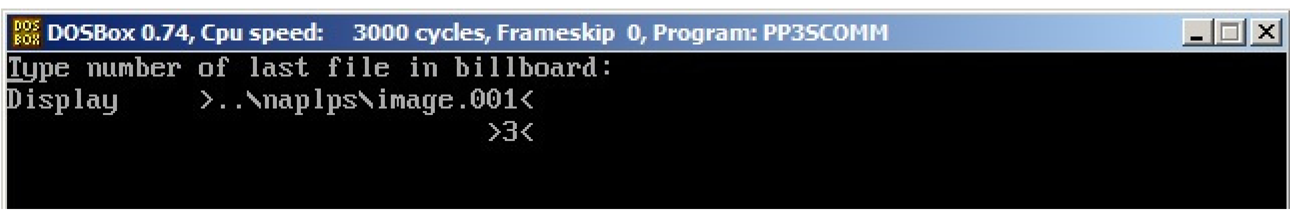
Press the control key (Ctrl) and the F2 key together and the program will ask you to specify an image to display:



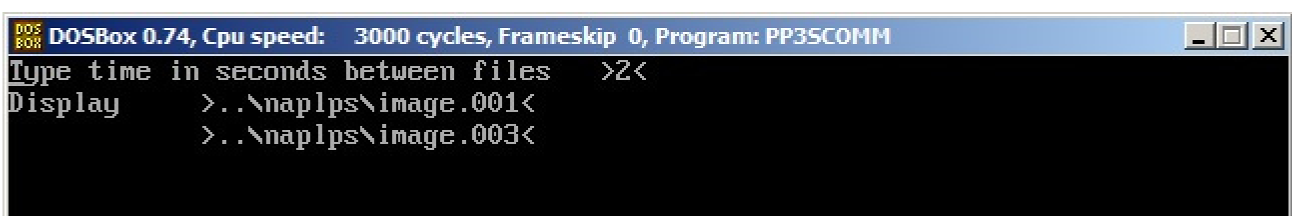
Enter the name of the first image in the sequence you wish to display. In this example, the images are located in a directory called 'naplps' that sits at the root of our C: drive, one level up from the pp3217a directory. We can specify this location using a relative directory path: ..\naplps\image.001, like so:



Hit enter and it will prompt you to indicate the last number of the image in the sequence. In this case, the last image we want to display is named image.003, so here we will enter '3':

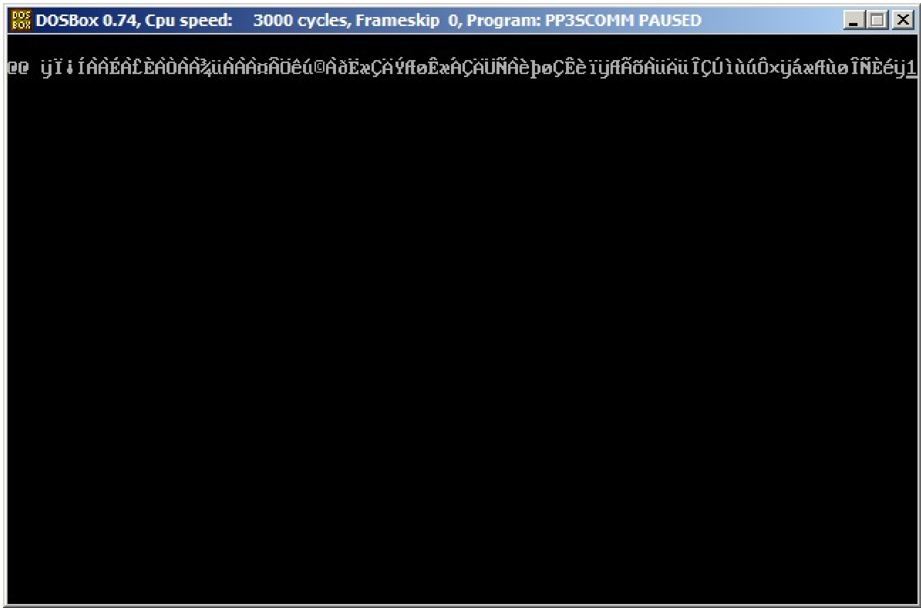


Finally we will be prompted to indicate the number of seconds to pause between images in the slideshow. In this example, we specify 2 seconds. Note that PP3 has automatically filled in the full path to the last image in the slideshow:

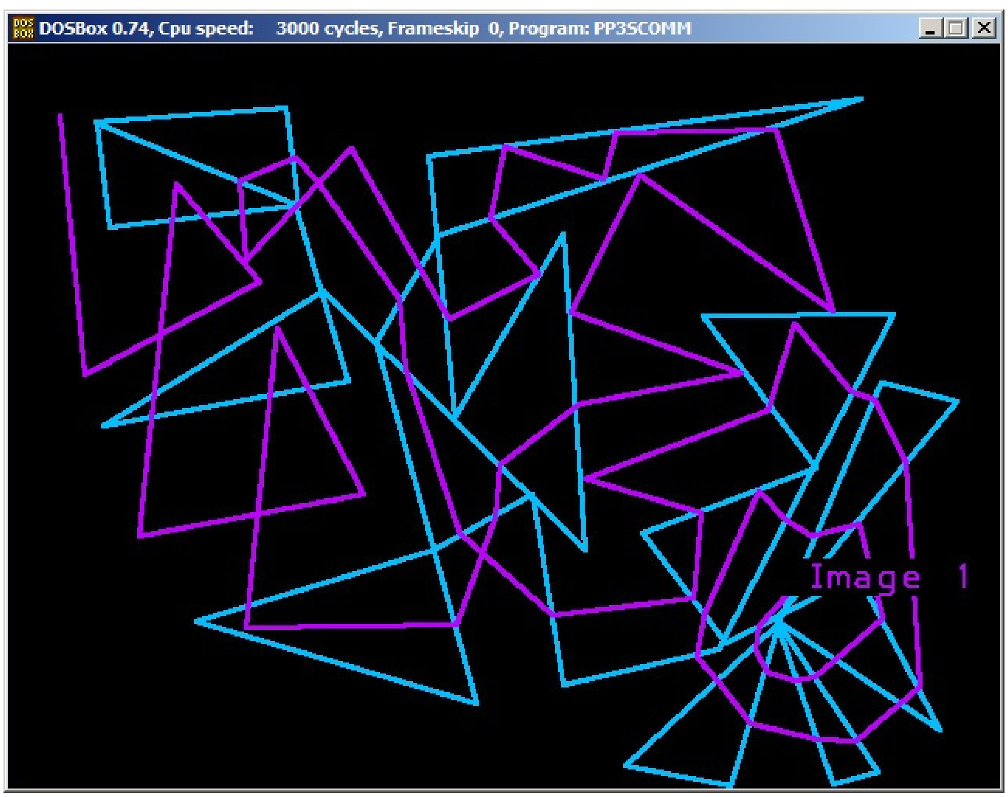


Press Enter.

You will see a screen displaying a line of text rather than the image you were expecting:



This is because PP3 is displaying your image file in ASCII mode rather than NAPLPS mode. To change to NAPLPS mode, press the Alt key and F9 together. The next image in the sequence will display in NAPLPS mode:



The slideshow will continue to cycle until you tell it to stop. Pressing Alt and F10 will return you to the PP3 menu. From there, pressing F9 will allow you to exit PP3 and return to the DOS prompt. Control/F9 exits DOSBox, or you can just close the window.

For more detailed instructions on PP3, including how to display single images rather than multiple images in a slideshow format, see the file README.PP3 in the pp3217a directory.

8. Speeding Up/Slowing Down

By default, DOSBox runs at a speed that will cause your images to render significantly faster than they would have on the hardware that was available in the mid 1980s. DOSBox gives you the ability to vary processing speed. Press Control & F11 to slow down your rendering speed (multiple times if necessary), and Control/F12 to speed it up. Number of cycles will display in DOSBox window bar.

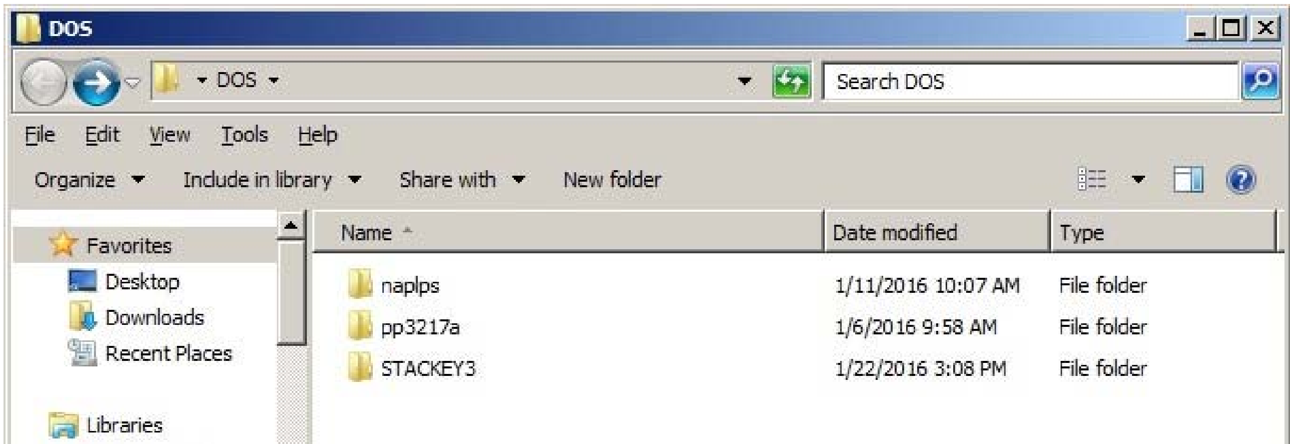
Alternatively you can set the speed to a defined value in the DOSBox config file before starting DOSBox. See the section below, "Automating Startup" for more details.

9. Automating Startup (Optional)

As we have seen, starting up a NAPLPS slideshow in DOSBox/PP3 requires entering several commands in sequence. This may not always be convenient. It is possible to automate the process using a shareware utility called Stackey3 and calling it from within the DOSBox config file.

First you will need to download and install Stackey3. This is similar to installing PP3, in that you will need to download and unzip the program files into the DOS directory we created in Step 2. As with PP3, Stackey3 is also available from various Simtel archive mirror sites. You can try downloading the file STACKEY3.ZIP from http://cd.textfiles.com/simtel/simtel20/MSDOS/BATUTL/.index.html or if that's not available simply search for the filename in Google.

The process for downloading and installing Stackey3 is the same as for PP3, so you can refer to section 3 of this document if you need detailed notes. You should install (unzip) it at the top level of your DOS directory, same as for PP3 and the NAPLPS files. When you're done, your DOS directory should look like this:

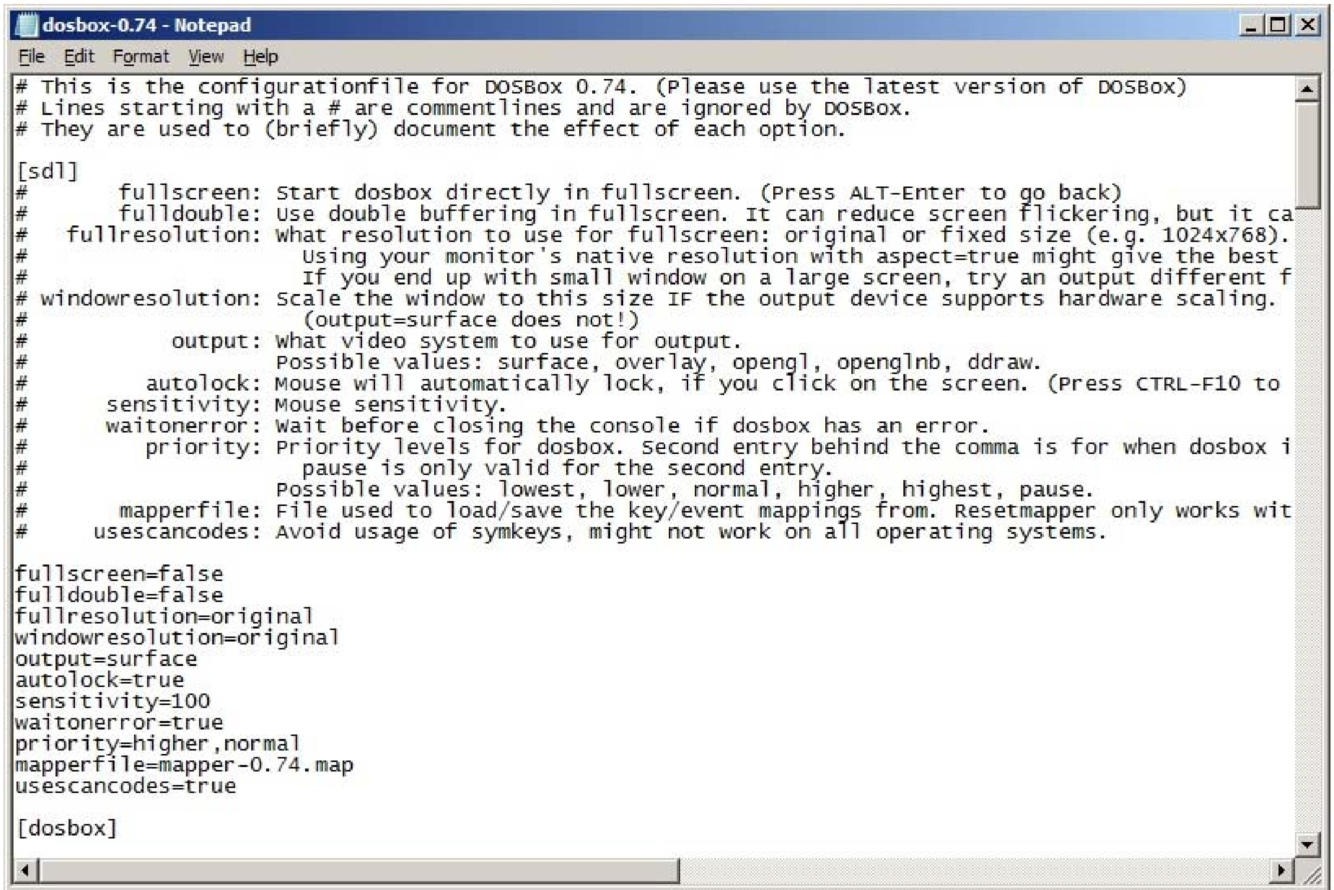


Next you will need to modify your DOSBox config file to set the necessary parameters and activate Stackey3. The config file is located in different places on different systems. According to the DOSBox documentation, on Windows Vista, 7 and 8 systems the config file is located in:

{system drive}:\Users*{username}*\AppData\Local\DOSBox\dosbox-{version}.conf

Since we are using a Windows 7 system and DOSBox version 0.74, and our username is "demo," our config file is located in C:\Users\demo\AppData\Local\DOSBox\dosbox-0.74.conf

Open your config file in a text editor and have a look at it. Use Notepad (under the Accessories menu) or another text editor if you prefer. (Do not use Word). You can leave most of the file unchanged but there are two or three sections you might want to modify. The config file looks like this when you open it in Notepad:




```
# This is the configurationfile for DOSBox 0.74. (Please use the latest version of DOSBox)
# Lines starting with a # are commentlines and are ignored by DOSBox.
# They are used to (briefly) document the effect of each option.

[sdl]
#       fullscreen: Start dosbox directly in fullscreen. (Press ALT-Enter to go back)
#       fulldouble: Use double buffering in fullscreen. It can reduce screen flickering, but it ca
#   fullresolution: What resolution to use for fullscreen: original or fixed size (e.g. 1024x768).
#                     Using your monitor's native resolution with aspect=true might give the best
#                     If you end up with small window on a large screen, try an output different f
# windowresolution: Scale the window to this size IF the output device supports hardware scaling.
#                     (output=surface does not!)
#           output: What video system to use for output.
#                   Possible values: surface, overlay, opengl, openglnb, ddraw.
#         autolock: Mouse will automatically lock, if you click on the screen. (Press CTRL-F10 to
#      sensitivity: Mouse sensitivity.
#      waitonerror: Wait before closing the console if dosbox has an error.
#         priority: Priority levels for dosbox. Second entry behind the comma is for when dosbox i
#                     pause is only valid for the second entry.
#                   Possible values: lowest, lower, normal, higher, highest, pause.
#       mapperfile: File used to load/save the key/event mappings from. Resetmapper only works wit
#     usescancodes: Avoid usage of symkeys, might not work on all operating systems.

fullscreen=false
fulldouble=false
fullresolution=original
windowresolution=original
output=surface
autolock=true
sensitivity=100
waitonerror=true
priority=higher,normal
mapperfile=mapper-0.74.map
usescancodes=true

[dosbox]
```

Sections of the comfig file are denoted by the words in square brackets. Look for the section called [cpu].

By default, the [cpu] section contains the following settings and values:

core=auto
cputype=auto
cycles=auto
cycleup=10
cycledown=20

As noted above, the default CPUspeed in DOSBox renders NAPLPS graphics much more quickly than they would have rendered on 1980s-era hardware. To slow down the CPU you can change the processor type to a slower model, and set the cycles to a defined value. For this example, we will change the CPU type to 386 and set the cycles to 400. The CPU values now look like this:

core=auto

cputype=386
cycles=400
cycleup=10
cycledown=20

Even if you think you might want different values leave them set this way for now. The following modifications depend to a certain extent on the processing speed.

The next section we need to modify is down at the bottom of the DOSBox config file, called [autoexec]. It should be empty except for a couple of comments. Enter the following values:

```
mount c C:\Users\Demo\Desktop\DOS
c:
cd pp3217a
..\stackey3\stackey W36 CR W36 CR W36 "2" W36 "1" W36 "0" CR W36 C2 W36
        "..\naplps\image.001" CR W36 "3" CR W36 "2" CR W54 A9
PP3
```

Important: This line beginning with “ ..\stackey3\stackey W36 CR W36 ” needs to be on a single line. It only wraps here because it exceeds the length of the page margin.

Assuming your system is set up exactly the same way mine is, your demo should now start automatically. However it is likely you will need to change some things. Lets step through the commands in the autoexec section to see what they are doing.

mount c C:\Users\Demo\Desktop\DOS
Mount the designated folder as your C: drive. You will likely need to change this with the appropriate values for your system

c:
Make c: the active drive

cd pp3217a
Change to the directory where the PP3 program is located

..\stackey3\stackey W36 CR W36 CR W36 "0" CR W36 C2 W36 "..\naplps\image.001" CR W36 "3" CR W36 "2" CR W54 A9
Initialize the stackey macro utility with the commands you want to run in PP3. We'll come back to this in a minute. Again, this all needs to be on one line in the DOSBox config file.

pp3
Start PP3

About Stackey3 Commands

Stackey3 is a utility that enables you to automate the entry of keyboard commands in DOS. The stackey3 line in the config file plus the following pp3 line automate the steps outlined in section 6 above.

The trick to making it work successfully is to ensure that wait times between commands are sufficient to allow the previous commands to complete before entering the next one. A “wait” command in Stackey3 is indicated by 'W' followed by an integer. W18 indicates Stackey3 should wait one second before entering the next command. W36 indicates a two-second wait is required.

The line above tells Stackey3 to enter the following sequence of commands

W36 - wait two seconds while the PP3 intro screen loads
CR – enter a carriage return to move to the next screen
W36 – wait two seconds while the PP3 splash screen loads
CR – enter a carriage return to move to the next screen
W36 – wait two seconds while the menu selection screen loads
“0”– select service number zero
CR – enter a carriage return
W36 – wait two seconds while the next screen loads
C2 – enter “Control – F2”
W36 wait two seconds
"..\naplps\image.001" – enter the location and file name of the first image in the sequence
CR – enter a carriage return
W36 – wait two seconds
“3” - enter number 3 (the number of the last image in the sequence)
CR – enter a carriage return
“2” - enter the interval to wait between images
CR – enter a carriage return
W54 – wait 3 seconds while the first image loads
A9 – enter “Alt-F9” to switch into NAPLPS mode

As noted above, the processor speed and the wait times between commands are somewhat interdependent. The sequence will fail if commands are entered before the results of the previous command have been processed. You may need to play around with the values in the sequence for optimal results on your system.

10. Full screen mode

Finally, you may want to display your images larger than the default size of DOSBox on your system. You can modify the the size of the display in the [sdl] section of the DOSBox config file. The fullscreen, fullresolution, and output settings may all need to be set. Change fullscreen to 'true', 'fullresolution' to your monitor resolution, and as noted in the comment lines for the [sdl] section, change the output value to something other than 'surface' if you wind up with a small window on a large black background. Possible settings for 'output' are identified in the comment section. They include surface, overlay, opengl, and openglnb

References

Consortel Catalogue. 1985. Vancouver, BC: Consortel.

Hughes, D. 1993. One BBSCon 1993: NAPLPS: Universal Graphics for BBSs to the Internet. Retrieved from https://archive.org/details/93bbscon-naplps

Wikipedia, n.d. Telidon. Retrieved from https://en.wikipedia.org/wiki/Telidon

code{4}lib JOURNAL

Issue 34, 2016-10-25 ISSN 1940-5758

Digital Archaeology and/or Forensics: Working with Floppy Disks from the 1980s

While software originating from the domain of digital forensics has demonstrated utility for data recovery from contemporary storage media, it is not as effective for working with floppy disks from the 1980s. This paper details alternative strategies for recovering data from floppy disks employing software originating from the software preservation and retro computing communities. Imaging hardware, storage formats and processing workflows are also discussed.

John Durno, University of Victoria Libraries

Introduction

In this paper I will discuss tools and techniques I have employed to retrieve content on floppy disks from various 8- and 16-bit computer systems dating from the 1980s. In the context of several projects completed over the past two years I have recovered data from approximately 500 floppy disks. Disks from different sources exhibited many unique characteristics and required specialized processing to extract the content in a usable form.

Specifically, this paper describes the tools and techniques I have used to recover content saved to 5.25″ and 3.5″ floppy media from the following systems:

- IBM PC MS-DOS
- Kaypro II CP/M
- Mac OS 7
- Apple II ProDOS
- Apple II p-System & IBM PC p-System
- Atari 130 XE

Floppy disks were sourced by way of donations to our Archives, from materials in the Library collection, and from researchers seeking to access materials on early computing media from their own collections. The different requirements of these use cases also affected the processes employed to retrieve information. The researchers' interests were best served by simple file recovery, sometimes involving format conversion, while materials destined for archival preservation required a more structured and consistent process.

While my main reason for writing this paper is the hope that my methods may prove useful to others confronted with similar problems, these case studies are also intended to support a larger argument. As evidenced by disk imaging toolkits such as those found in Archivematica and BitCurator much of our current practice reflects the assumption that tools derived from the domain of digital forensics can productively be applied to all types of digital media, from 1980s floppy disks through contemporary hard drives. I would argue instead that floppy disks from the 1980s are a separate problem domain requiring specialized tools from outside the digital forensics community.

Relative to earlier eras, today we enjoy a high level of standardization in our computing environments and routinely expect digital content to be intelligible across multiple platforms. The major challenge is managing the sheer volume of data that is stored on modern systems. In the 1980s however these conditions were reversed. While the volumes of data were often miniscule by contemporary standards, a wide range of computing systems developed by competing vendors implemented proprietary technologies up and down the stack, from disk encodings to file systems to applications and their associated file formats. Far more than in our own time, digital information from the 1980s was deeply enmeshed in the particular platforms, operating systems and applications that created it. We should not therefore expect to approach the very small amounts of boutique data found on floppy disks with the same tools we use to wrangle the vast amounts of content on newer hard drives.[1]

The observation “Forensic techniques and tools will not eliminate the problems presented by older media, but they can make certain parts of the preservation process more efficient and more secure”[2] points in the direction my argument will take, but I would argue it does not sufficiently stress the distinction between forensic tools and forensic techniques. While forensic techniques are still, broadly speaking, applicable, in most cases it does not make sense to apply tools developed by the digital forensics community to the problem of data retrieval from floppy disks. Often it is not even possible. Tools developed by the software preservation and retro-computing communities are generally far more useful as the cases described below will illustrate.

Disk Imaging

The first step in any kind of data recovery from floppy media is to create a disk image, a sector by sector copy of the entire disk, including the parts of the disk not normally seen by users (boot sector, file table, sector numbering, data marked deleted but not yet overwritten). Any further manipulation of the disk contents (such as content extraction) is done from a copy of the disk image. This has the benefit of minimizing the handling of the original floppy media, and provides a baseline known good state to control for inadvertent changes that could be introduced by subsequent manipulation of the contents. Write blocking is critical in the imaging process in order to prevent inadvertent alterations to the source media. In this my approach differs not at all from standard digital forensics methodology.

In most cases, the right tools make imaging disks relatively straightforward. In addition to drives of the requisite size (typically 3.5″ and 5.25″), one also needs a controller (circuitry to control the drive) and appropriate software to run the imaging process. Historically, floppy drive circuitry was often built into computer motherboards but most modern motherboards have limited capability in this regard. Devices like the Kryoflux and FC5025 supply external USB-attached drive controllers, with sophisticated circuitry enabling them to bypass the limitations of older disk controllers (for example, they can read disks with both GCR and MFM encodings).

At the University of Victoria we have three devices capable of imaging floppy disks, each with their own strengths and weaknesses.

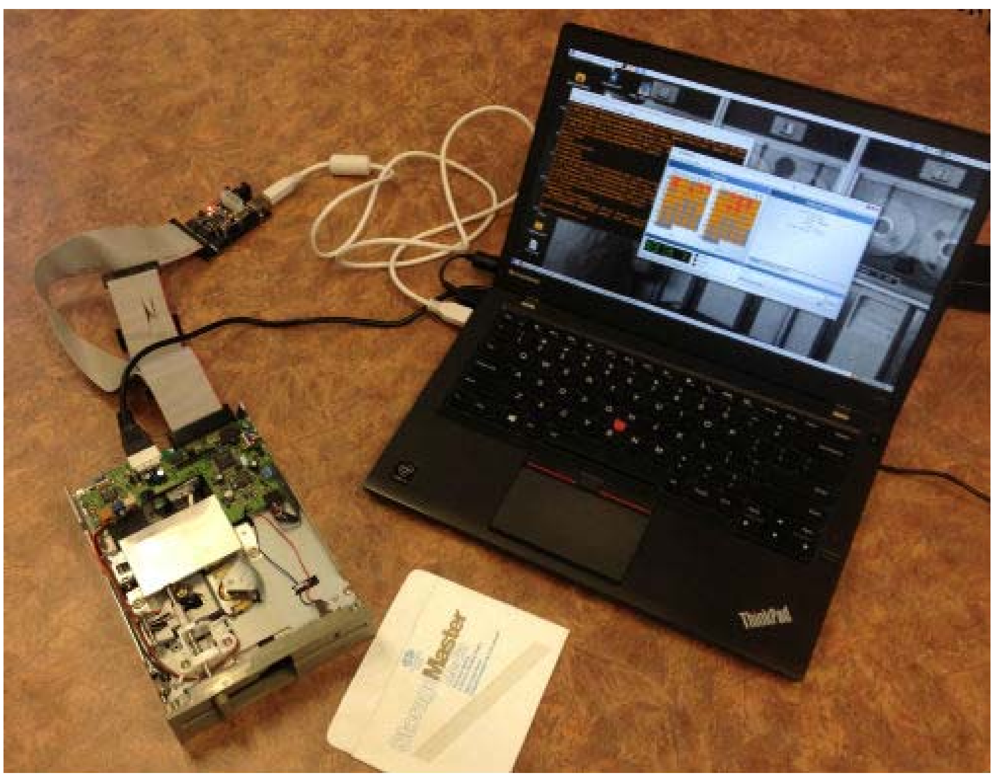


Figure 1. Kryoflux controller connected to 5.25′ Floppy Drive and Laptop (via USB)

1. Kryoflux. Without a doubt, the Kryoflux is the most versatile device we have for imaging floppy disks. It consists of a controller, cabling, and software. With appropriate drives it is capable of imaging both 3.5″ and 5.25″ disks and can handle a wide range of encodings and sector formats. It has hardware-level write-blocking. The software has both GUI and command line modes (the command line is harder to use but more flexible). The Kryoflux is also the most capable of our three devices for recovering data from damaged disks, and is the only device we have that is capable (with a modified drive) of recovering data from ‘flippy’ disks

(single-sided floppies with data written on both sides). However it is easy to damage and can only be handled by experienced technical staff.

2. Windows 7 workstation. This is a purpose built workstation with a DeviceSide Data FC 5025 USB floppy controller (for 5.25″ disks), a motherboard-attached 3.5″ floppy drive, and standard optical media drive. The FC 5025 controller has proven to be surprisingly capable given its low cost (relative to the Kryoflux) and can handle a wide range of disk formats. The 3.5″ floppy drive has been under-utilized but will work with any Windows imaging software; for example, FTK Imager. The FC5025 unit employs hardware level write blocking, while write blocking for other devices is enabled at the level of the Windows registry. This approach is also useful for acquiring content from non-floppy devices, such as external hard drives, CDs and thumb drives.

3. Industrial motherboard workstation. Another purpose-built workstation, this one has a 32 bit industrial motherboard and can boot into DOS, Windows XP, and Linux (CentOS). 3.5″, 5.25″ and optical drives are all directly attached to the motherboard. Of our three workstations we have used it the least for imaging, but it has come in handy for edge cases (see the section on PC p-System disks below). Imaging software includes standard open source tools like Guymager and dd on the Linux partition, the versatile OmniFlop[3] on the Windows XP partition, and Dave Dunfield's ImageDisk[4] in DOS. Write blocking is implemented at the application level and by using write protection features on the disks themselves.[5]

Given its versatility, the Kryoflux is the generally preferred option for floppy disk imaging. However, the other two devices have occasionally provided better results in accessing the contents of copy-protected disks and disks with unusual sector formats. And one should always be conscious that floppy drives can silently go out of alignment, resulting in disk images that cannot be read, or worse still, contain corrupted files. Having multiple drives and devices available makes it possible to periodically run sanity checks on your imaging output.[6]

Most imaging software comes with built-in settings for known disk types. However it is not always easy – particularly in the case of posthumous donations of archival materials – to determine what particular combination of hardware and operating system was used to write to the disk in the first place. In some cases the relevant information is noted on the disk label, but in others a certain amount of trial and error may be required.

Image formats

Tools developed by the digital forensics community typically save disk images as either AFF (Advanced Forensics Format) or EWF (Encase/Expert Witness Format). The main advantages of these formats are seen to be that they store metadata, including checksums, within the disk image file itself. That provides some assurance that the metadata is accurate and the image has not subsequently been tampered with.

Unfortunately, AFF up to version 3 has been deprecated by its creator[7] , and EWF is proprietary (albeit well-documented and with some open source support)[8] . For these reasons, and because neither format is well supported outside the forensics community, there is some cause to doubt their suitability as long-term preservation formats.

Sector image formats would appear to be more suitable for floppy disk preservation, for the following reasons:

- They are the standard output of imaging tools such as the Kryoflux and FC5025.
- They are widely supported by a range of open source content extraction utilities,
- Their contents are accessible under emulation, and
- They have been in existence longer than most preservation formats, which bodes well for their continued longevity.

The forensics community designs tools for criminal investigators, not for digital archaeologists. Tools that support AFF and EWF (Guymager, fiwalk and The Sleuth Kit for example), while undoubtedly valuable in their problem domain, were not designed to recover and preserve content from computer media from the 1980s. Most of the tools that can recover content from 1980s floppy disks either output or expect to work with standard sector image formats.

Sector images vs. stream files

Stream files are a proprietary but publicly-documented format output by the Kryoflux controller. They contain a great deal of information and are the easiest to acquire: they contain all the low-level information the Kryoflux generates while reading flux transitions, and can be made without knowing any technical details (encodings, sector geometries) about the source media beforehand. However, stream files are quite large in comparison to sector images (roughly 100 times larger, so a stream file for a 360K disk will come out to around 40 MB), and need to be converted to sector images in order to access their data in humanparsable format. The Kryoflux software, DTC, can be used to create images from stream files. They are not a preservation format, as noted in the documentation: 'Stream files are hardware specific (to the KryoFlux device) and therefore are not intended for long term preservation.'[9]

It is worth noting that acquiring only stream files would be a workable strategy for obtaining floppy disk images in situations where it is necessary to use semi-skilled labour to do the imaging work. In such cases the work of converting streams to sector images could be deferred until later.

Sector images invert the characteristics of stream files: they are typically the same size as the original disk, accessible to emulators and content extraction tools, non-proprietary in most cases, and are not a complete recording of all the low level data the controller was able to read from the disk, such as flux reversals and index blocks.

Processing images for information storage and access

While creating disk images is relatively straightforward, making their contents intelligible to humans can be more challenging. Fortunately a wide range of readily available tools exist to facilitate the process. I will be describing some of them below.

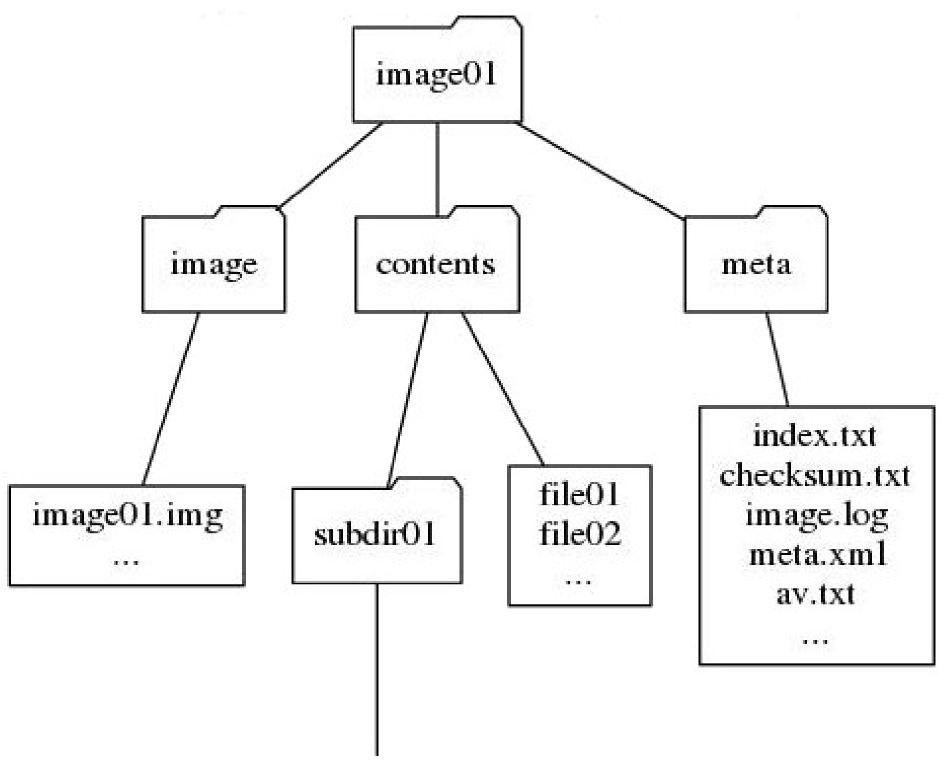
Except where otherwise noted, all of the work was done on a modern Lenovo Thinkpad 450s running Ubuntu Linux 15.10. Most open source content extraction tools and emulators work well in Ubuntu. Many of these tools have a command line interface, which makes it possible to script the repetitive parts of the job.

Also, working in Linux provides added security against computer viruses. Floppies were an attack vector for viruses in their day and many of the viruses from 25 years ago are still perfectly capable of infecting a modern Windows computer. For this reason, files recovered from floppy disk images should be immediately scanned for viruses prior to further handling. I use the open source ClamAV[10] for this purpose. The images themselves should also be scanned to check for boot sector viruses.

While one could preserve disk images only and not extract their contents until such time as they were required, running the extraction process up front facilitates an examination of the image contents in order to ensure the imaging process was successful.

Structured output

As noted above, the desired end product of the content extraction process varies depending on the nature of the project. For digital preservation, one reasonable option would be creating a structured directory containing the disk image, the extracted contents of the disk image, and metadata files:



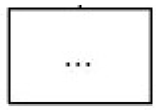


Figure 2. An example data structure for the image and output of a floppy disk

Non-exhaustively, the meta folder may contain:

- index.txt: A recursive directory listing of the contents of the disk image
- checksum.txt: An MD5 checksum of the disk image, generated immediately after the image was acquired
- image.log: A log of the image creation process generated by the Kryoflux or other imaging software, noting the condition of disk sectors
- meta.xml: Metadata recorded in accordance with archival policy, for example control numbers, date the image was made, text from the disk label, disk image format, tools used to extract the files and so on.
- av.txt: Output from the antivirus scanner

Time and resources permitting, photos of the disks could also be included in the meta folder. Floppy disk labels often contain useful metadata.

Case Study 1: PC/MS-DOS Floppy Disks (3.5″/5.25″)

Floppy disks created on PC/MS-DOS systems were not as prevalent in our processing queues as we originally anticipated. Due to their ubiquity in other contexts, these are the easiest kinds of disks to work with, and the only one of the case studies in this paper that digital forensics tools like guymager, The Sleuth Kit, and fiwalk can handle.

Disk imaging: All standard imaging tools come with settings for PC/MS-DOS disks. Because the Kryoflux works at the level of the disk encoding rather than the file system, the MFM setting (i4) is appropriate here. For double density disks, the step setting -k2 (indicating 40 cylinders) should be used, for example:
dtc -fimage.img -k2 -i4

File Extraction: Again, multiple options exist for copying individual files from the disk image to the local file system. I typically use the set of utilities collectively known as mtools, which enable Unix systems to work with MS-DOS file systems.

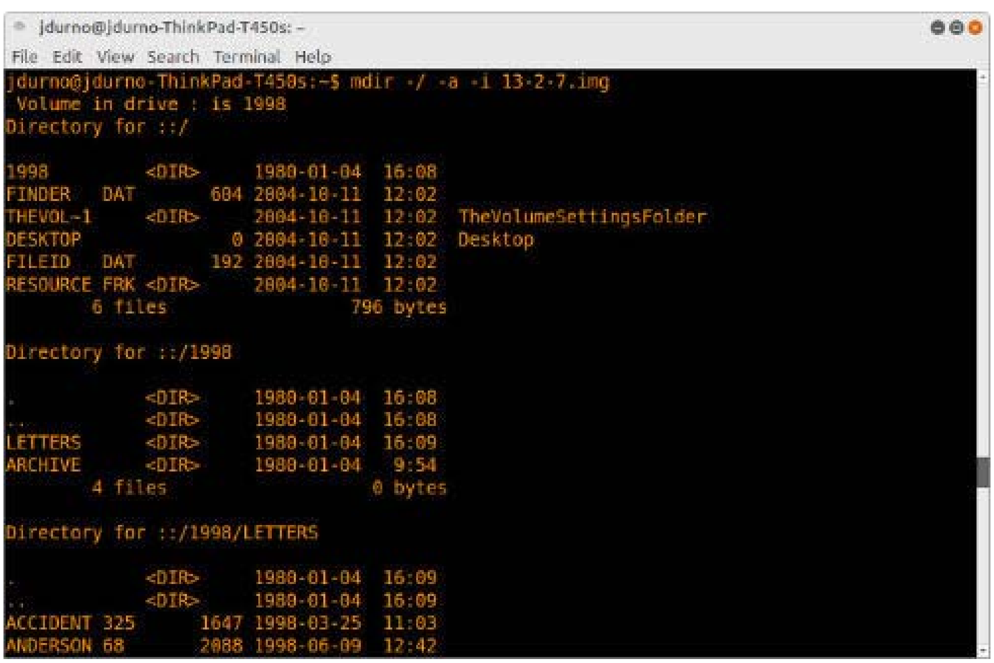


Figure 3. Partial output of `mdir` command

The following commands produce the desired output, for a disk image with the name “image.img”:

Recursive directory listing including hidden files, output to the file “index.txt”:

```
mdir -/ -a -i image.img > index.txt
```

Recursive content extraction output to subfolder “content”, preserving modification times and file attributes:

```
mcopy -p -s -m -i image.img ::* content
```

In a small number of cases where mdir and mcopy failed to read a disk image one may mount the image in the linux file system and extract the contents that way. Use:

```
sudo mount -t msdos -o ro,loop image.img mountpoint/
```

… where “mountpoint/” is the name of the empty directory where you want the image to mount. Standard unix utilities (ls, cp) can be used to work with the files once the image has been mounted.

Case Study 2: Kaypro II CP/M (5.25″)

Disk imaging: The Kryoflux MFM setting (i4) works for Kaypro II CP/M disks. The FC5025 floppy controller has a setting specifically for KayPro II CP/M disks.

File extraction: I used Michael Haardt’s cpmtools package[11] , which functions similarly to mtools, but for CP/M file systems. One significant difference is that the disk format must be specified, because the CP/M operating system ran on a variety of systems, not just PC-compatibles. In this example, the command line switch `-f kpii` (for Kaypro II) is added.

Disk formats are defined in a disk definitions file which, on Linux systems is usually located in /usr/local/share/diskdefs. The file contains human-readable text specifying floppy disk geometries (sector lengths, number of tracks, block sizes, and so on) employed by the many different different systems that ran CP/M. A comment at the beginning of most stanzas identifies the system to which it applies. For example, the disk definition for the Kaypro II is contained in the following stanza:

```
#Kaypro II
diskdef kpii
seclen 512
    tracks 40
      sectrk 10
      blocksize 1024
      maxdir 64
      skew 0
      boottrk 1
      os 2.2
end
```

Most versions of CP/M did not implement directories for file storage, so recursive output is not necessary. Instead, CP/M structured file storage around what were called user spaces, numbered areas where different users could store their files. As most CP/M home computers were single user systems usually there will only be one user area, numbered 0 indicating that files were accessible to all users.

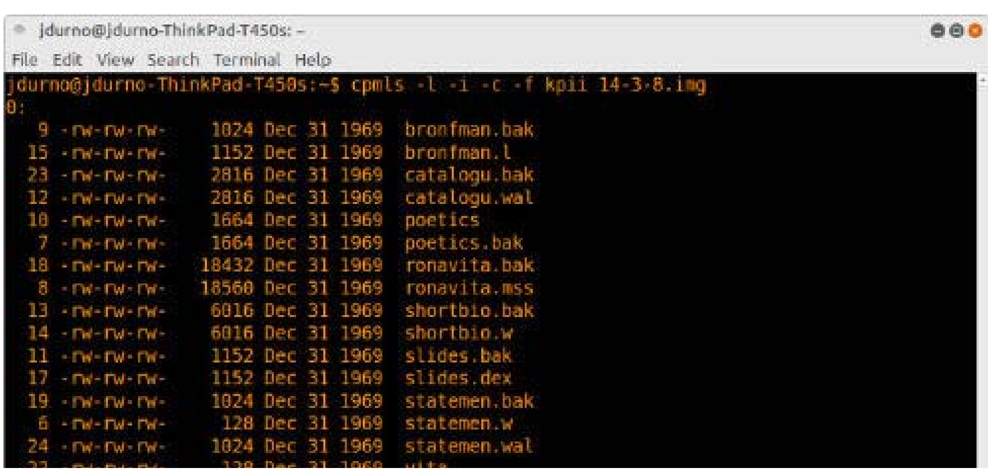


Figure 4. Output of `cpmls` command

For a directory listing a number of different output options are available. The “-l” flag produces Unix style output, with permissions, file modification times (not always trustworthy, as illustrated above) and file sizes.

```
cpmls -l -i -c -f kpii image.dsk > index.txt
```

This command extracts all the files on “image.dsk” in user area 0 to the subdirectory “content”, preserving original time stamps.

```
cpmcp -p -f kpii image.dsk 0:* content/
```

Case Study 3: Mac OS, 3.5″ Double sided, double density (800K)

Mac floppy disks from the mid-1980s through the mid-1990s differed from PC disks in two significant ways:

1. The file system, HFS (Hierarchical File System)
2. The disk encoding scheme, Apple GCR

There are a couple of exceptions: the earliest Mac 400K floppies had a different file system (MFS, for “Macintosh File System”) and the later high density floppies used MFM encoding, not GCR, for PC compatibility. Another aspect which may cause confusion is Apple’s successor to HFS, developed for OS X, called HFS Plus (or sometimes, incorrectly, “HFS Extended”) and sometimes referred to as HFS. Software like The Sleuth Kit that supports the later variant of HFS do not necessarily support the earlier one.[12]

These complications notwithstanding, if you need to work with double density 3.5″ Mac floppy disks you will require tools capable of handling GCR encoding and HFS file systems.[13]

Disk imaging

The Kryoflux can image 3.5″ double density disks using the “Apple DOS 400/800K sector image” setting. As PC disk controllers cannot read GCR encodings and the FC5025 can only handle 5.25″ disks, the Kryoflux is the only option for imaging Mac 3.5″ disks up to 800K unless you have a vintage Mac with a working floppy drive.

Because the later Mac HD (1.44K) floppies were MFM-encoded, there is a much broader range of options for obtaining Mac HD floppy disk images. Most PC floppy disk controllers can handle them, and most PC-compatible disk imaging software (eg. Omniflop, FTKImager, or Guymager in Linux) includes the appropriate settings.

File extraction

My toolkit of choice for working with HFS file systems is hfsutils, a collection of utilities for Unix systems that has been in development since 1996[14] . It works similarly to mtools and cpmtools, except that the disk image needs to be mounted before it can be read. The following command mounts an HFS disk image named “image.dmg”:

```
hmount image.dmg
```

If the command was successful, hmount will return some information about the disk including the volume name, time created and last modified, and number of bytes free. One can capture that output by modifying the above command to:

```
hmount image.dmg > hmount.txt
```

To recursively list directory contents, including hidden files and catalog ids, in long format, output to the file “index.txt”, use:

```
hls -i -a -l -R > index.txt
```

Note that the disk image file does not need to be specified, because it was mounted in the previous step.

The HFS file system includes resource forks and data forks. Every file in the file system may have both. Data forks contain unstructured data, while resource forks include structured data relating to the file, including icon bitmaps, positioning of windows, and so on. While much of this information is irrelevant to viewing standard file types on modern systems, data in the resource fork sometimes extends beyond system-specific metadata including, for example, the embedded images in a word processing document.[15]

While it is therefore important to preserve resource forks, it is debatable whether resource forks need to be included among the files extracted from the disk image as a routine procedure. If the goal is to read the extracted files on non-Mac systems, then resource forks are unnecessary as other systems cannot use them. If the goal is to read the files on a Mac, the best approach is probably to mount the disk image and access the files that way.

To extract files without the resource fork, to a subdirectory named ‘content’, use:

```
hcopy -r :* content/
```

To extract files with resource forks, use:

```
hcopy -m :* content/
```

After working with a disk image, one must unmount it before mounting another one. The following command unmounts the disk:

```
humount
```

Case Study 4: Apple II

The Apple II series was in production from 1977 through 1993, during which time between five and six million were sold.[16] Apple II hardware and software evolved significantly during that period, and even models of the same era could boot into different operating systems. No single approach works for all variants when it comes to processing Apple II disks. In this section I discuss two variants that I have come across.[17]

Apple II ProDOS (5.25″)

ProDOS was first introduced in late 1983 to overcome a number of shortcomings associated with the previous Apple DOS version, 3.3. It was eventually renamed ‘ProDOS 8’ after a 16 bit version was released.

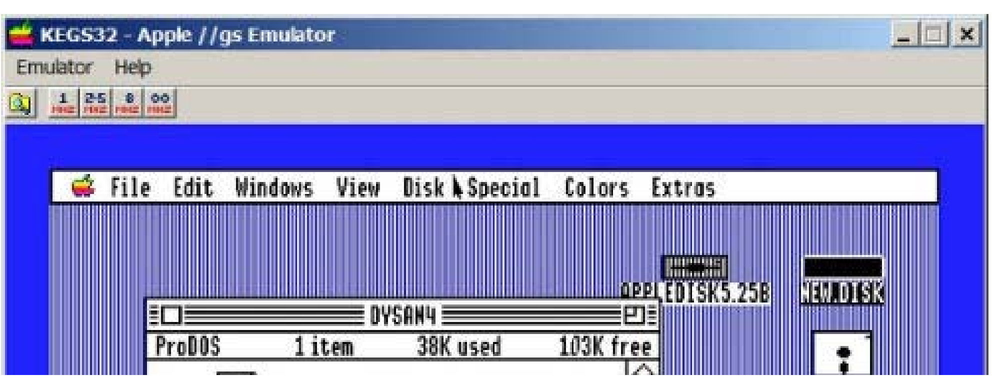
Disk imaging

Apple DOS 3.3 and ProDOS use the same 16 sector format for 140K disks, so the Kryoflux setting for “DSK, DOS 3.3 Interleave” was appropriate here. The FC5025 has an “Apple ProDOS” setting. The Kryoflux command line requires the following parameters for an optimal read of Apple DOS 3.3 disks:

```
dtc -f<imagename> -x0 -i8 -l8 -dd1p
```

File extraction

As HFS was specific to Macs, hfsutils cannot be used to extract content from Apple II disks, which had their own file system. Initially I mounted a few sample disk images in the KEGS Apple IIgs emulator to examine their contents, which as it turned out consisted primarily of files written in an early version of Appleworks.[18] While it was possible to read the Appleworks files in the emulator, the process was awkward and slow. The sheer volume of content (hundreds of files on more than 90 floppy disk images) would have rendered that approach problematic for anyone wishing to read more than a handful of documents.



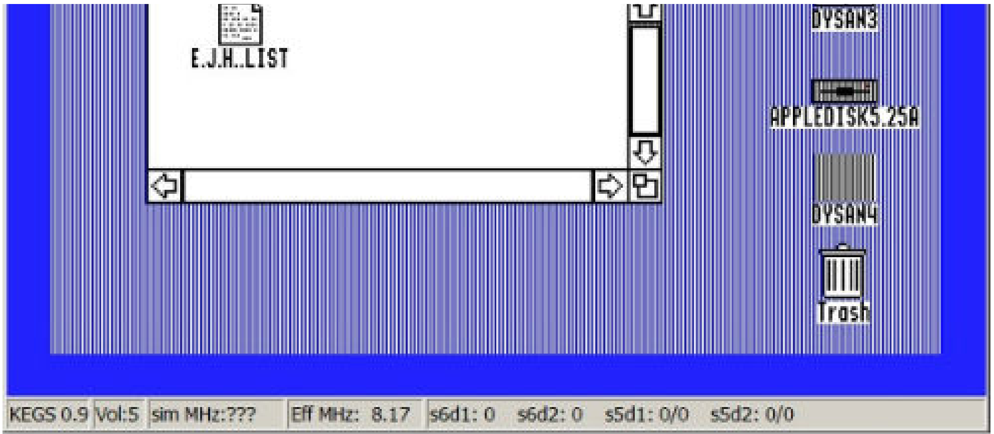


Figure 5. Viewing files in the KEGS Apple IIgs Emulator

Instead I used AppleCommander to extract and convert the contents of the disk images[19] . AppleCommander is an open source java program with both GUI and command line interfaces. In addition to ProDOS it can handle a range of common Apple II disk image formats, versions of DOS 3.3, Apple Pascal, and CP/M among them.

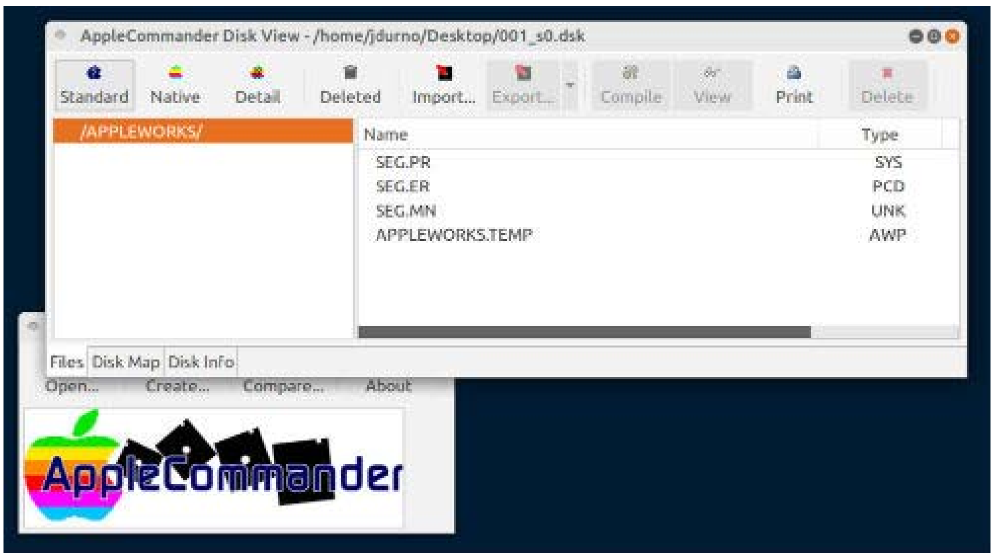


Figure 6. AppleCommander GUI

To write a directory listing into 'index.txt', use:

```
java -jar ac.jar -ll disk.dsk > index.txt
```

… where "ac.jar" is the name of the Apple Commander jar file and "disk.dsk" is the name of the disk image you want the directory listing from. The switch "-ll" outputs a long file listing with file name, creation and modification dates, and file sizes.

The current stable release version of AppleCommander (1.3.5) does not have the ability to extract all the files from a disk image via the command line interface, but the interim release candidate (1.3.5.13) does, using the command:

```
java -jar ac.jar -x disk.dsk content
```

This command extracts all the files on the disk image to the subdirectory "content." It automatically applies a conversion filter to certain types of documents; for example AppleWorks documents are extracted as text files. While text conversion was acceptable for our project, it would be problematic in cases where it was important to retain formatting. In a curious reversal of the norm, the AppleCommander GUI provides a greater range of bulk extraction options than the command line, including text, RTF, HTML, and no conversion (binary extraction).

Case Study 5: Apple II p-System & IBM PC p-System 5.25″

The p-System OS was developed in the late 1970s at the University of California, San Diego. Like CP/M, was designed to be portable across different varieties of hardware, including the Apple II, the PDP 11, and later the IBM PC. Two variants of p-System disks (Apple II and IBM PC) were donated to our archives, corresponding to two phases of a project undertaken by our Computer Science department collaborating with the Victoria-based artist Glenn Howarth in the first half of the 1980s.

Disk imaging

p-System disks used the encoding scheme appropriate for the hardware on which they were running, MFM on PC and GCR on Apple.

For imaging the Apple disks, the built-in "DSK, DOS 3.3 Interleave" Kryoflux setting produced disk images readable by the file extraction utilities described below.

Imaging the PC disks proved to be more of a challenge. The PC version of the p-System OS was a rarity in its day. As a consequence of its relative obscurity, there are no built-in settings for it in any of my imaging tools. For example, it is not included in the list of ~150 formats supported by Omniflop, nor is it a known format that is supported by the FC5025 and Kryoflux controllers. The Kryoflux was, of course, able to obtain stream files but my attempts to produce sector images resulted in files that were oddly twice the size they should have been (720K images for a 360K floppy).

Further analysis of these disks using a program called Anadisk indicated the sector numbering was unusual. Following a recommendation on the Kryoflux forums I reimaged the disks using a DOS utility called ImageDisk and converted the resulting IMD files to IMG.[20] These worked somewhat better, allowing me to extract some, but not all, of the files on the disks. I believe the remaining problems have to do with how the UCSD p-System handled sector interleaving, but have not yet confirmed this.

File extraction

As with other test cases, specialized tools were required to process the p-System disks. In this case, the same tools and settings could be used for both the Apple II and IBM PC disk images. I used Peter Miller's ucsd-psystem-fs (http://ucsd-psystem-fs.sourceforge.net/), a set of utilities for manipulating UCSD p-System disk images.

For both PC and Apple disk images, the following command outputs a directory listing from the disk image "disk.img" into the text file "index.txt":

```
ucsdpsys_disk -f disk.img -l > index.txt
```

The following command extracts the files on the disk image "disk.img" into the subfolder "contents":

```
ucsdpsys_disk -f disk.img -g contents/
```

Case Study 6: Atari 130XE 5.25″

Atari manufactured their line of 8-bit home computers from 1979 through 1992. The mass-market 130XE model dates from the latter half of that period. Atari systems were popular for gaming but were also used for office applications. In this case the disks came from the personal collection of a researcher, not from our Archives. The researcher was able to tell us both the model of computer and the software (AtariWriter) that was used to create the files on the disks.

Disk imaging

Early Atari floppy disks were typically FM encoded, single sided, and had 40 tracks with 18 sectors each. Later disk drives used MFM encoding and additional tracks per sector, permitting increased storage density. The Kryoflux has settings for both types. The "MFM XFD, Atari 8-bit" GUI setting worked for these disks.

File extraction

I was unable to locate command line tools to work with Atari disk images. Instead, we configured an instance of the Atari800 emulator, and used AtariWriter to read the files from the disk images mounted as virtual floppies.[21] It proved possible to send files as postscript from AtariWriter to a print queue on the host system. In this case the emulator was installed on the researcher's OS X laptop and configured to print files to PDF. This would not be an optimal technique for converting hundreds of files as it involves touching each file individually, but it addressed our requirements in this particular case.

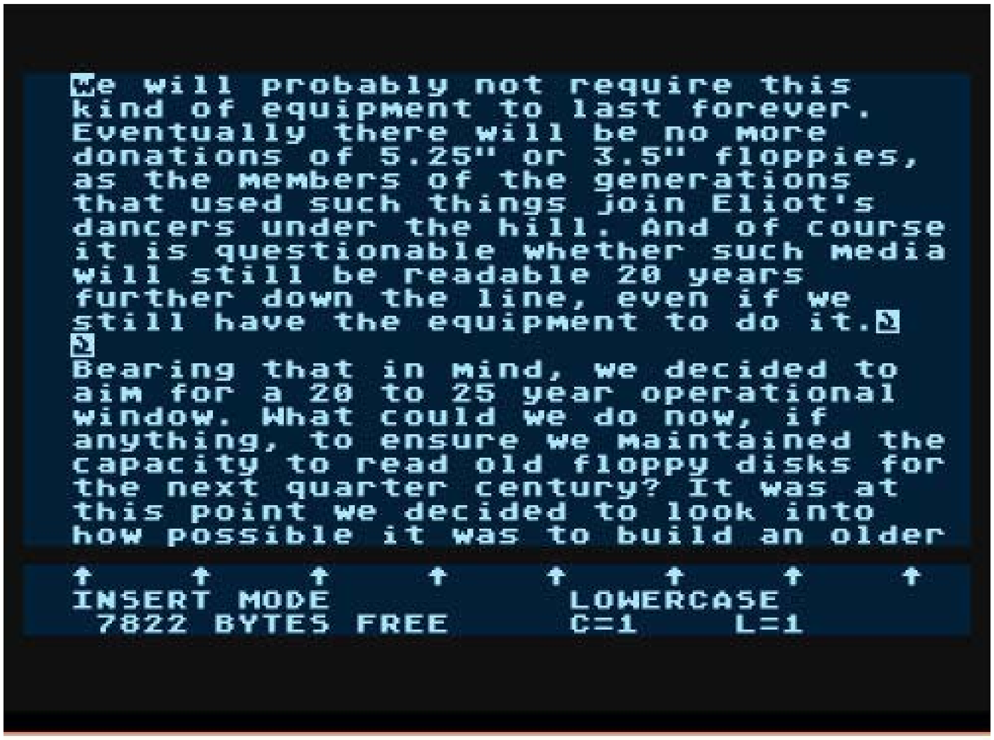


Figure 7. AtariWriter document rendered in Atari800 emulator

Automating the process

The work of imaging and processing floppy disks typically falls into two distinct phases. In the first phase, the disks are examined in order to determine the optimal imaging and content extraction strategies. Once the technical analysis phase is over, however, the work quickly becomes rote: issuing the same commands over and over again as the individual disks are processed. Further analysis and decision making is typically only required in the case of damaged disks or in the event anomalous disks are encountered.

It is therefore advantageous to script as much of the work as possible. To automate the workflow described above, I have written scripts that reduce the effort required per disk to two brief commands, one to create the image and associated log file; and the other to create the directory structure (as shown in 'Structured output' above), move the disk image and log to their correct locations, checksum the disk image, extract the contents and directory listing, and run the antivirus scan and log its output to the correct location.

The sections above outline the basic building blocks of these scripts. Examples of my CP/M and DOS content extraction scripts are found in Github[22] . They are not in any way sophisticated examples of the art of programming; the good news is they don't need to be.

Rendering files on modern systems

The topic of rendering obsolete file formats on modern systems is vast and the challenges are many. The following are general observations from my own experience.

Speaking very broadly, there are two ways to go about accessing files with formats that are not supported on modern systems. One may convert the obsolete formats to their contemporary equivalents, or else recreate a period software environment either on old hardware or more usually via emulation. These approaches are not mutually exclusive; in practice they may be combined as in the Atari example above.[23]

One of the common characteristics I have observed across multiple sets of floppies is that disks from the same source typically exhibit little variety in terms of file formats. Unlike the computers of our own time with their plethora of different applications and associated file formats, personal computers of the 1980s were typically used for a much smaller range of tasks and ran a much smaller range of productivity software. In my experience, it is not uncommon for dozens of disks from the same source to contain nothing but files from a single word processing application.

Here again, the tools developed for enumerating file types in the context of a forensic investigation are of little use. They typically have problems identifying file types from 1980s systems, even common formats like Wordstar and AppleWorks. There is also less utility in enumerating file types in a set of floppy disks that contain only one or two types of files.

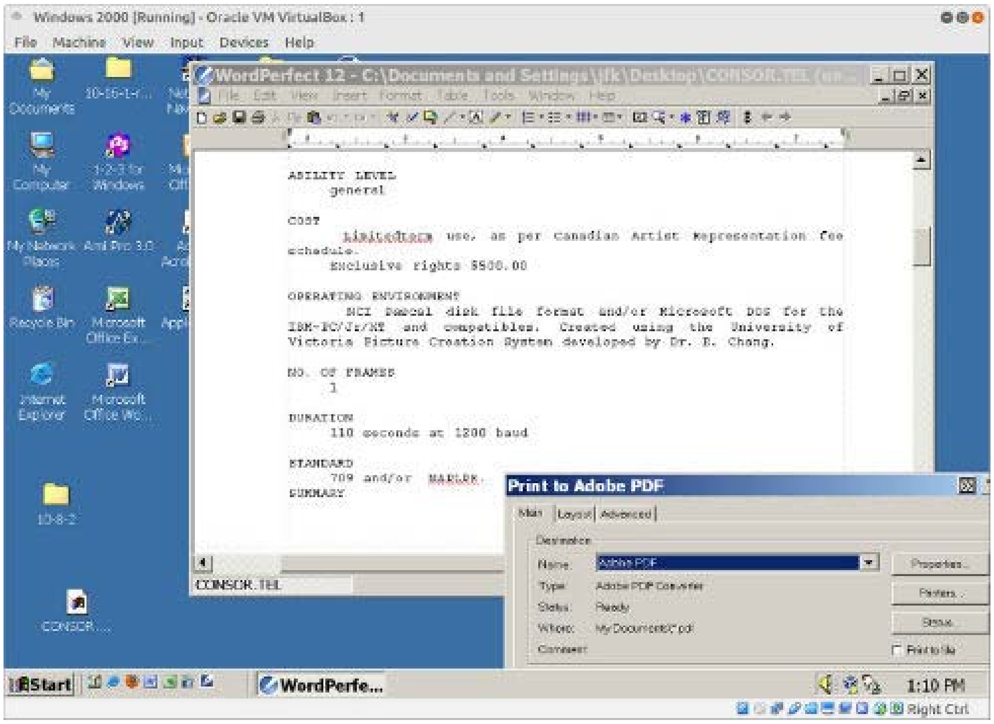


Figure 8. Converting Wordstar to PDF via WordPerfect 12 running on Windows 2000 in Virtualbox

Bulk format migration tools are also problematic. In many cases they don't exist for the formats in question or are expensive and proprietary. This is particularly troublesome in the case of Wordstar files, since Wordstar was widely used on multiple computing platforms in the first half of the 80s. Our current strategy for converting Wordstar documents to PDF involves opening them in WordPerfect 12 installed in a virtualized (Virtual Box) instance of Windows 2000 and rendering them as PDF files via Adobe Acrobat Professional 8. WordPerfect 12 has Wordstar conversion capabilities back to version 3.3. This approach respects the formatting of the original document but is time-consuming as documents must be converted individually by hand. It is far simpler and faster to script a bulk conversion from Wordstar to plain text (8-bit ASCII) if that will suffice.[24]

Damaged/Problem disks

In general, the 5.25′ floppy disks I have encountered have held up well over time, calling into question some of the more alarmist claims about the rate of deterioration of data on floppy media. I have seen numerous examples of disks stored in very average conditions that were readable 30+ years after they were first created. The exceptions have largely been disks containing application files, presumably because these were used heavily, whereas disks meant primarily for saving data would have been used less frequently. The relative proportion of damaged 3.5′ floppies has been higher, despite being on average 10 years newer than their 5.25′ predecessors.

The disposition of damaged disks requires careful consideration, not least because disks may be damaged in multiple ways:

1. The disk itself may be damaged beyond the point where it can be imaged
2. The protective case may be damaged or bent, but the disk inside might still be viable
3. The disk can be imaged, but with errors indicating bad sectors, leaving the resulting image in one of two states:
    1. It is possible to extract files from the disk image, but some of the contents might be corrupted
    2. It is not possible to extract files using the standard tools
4. The disk might be copy-protected, have an unknown sector format (as with the PC p-System disks mentioned above), or be a hard sector disk

With enough effort many of these problems can be overcome or at least mitigated. However doing so requires expertise that might not be readily at hand, and time, which is always in short supply. For example, if the protective case is damaged one may try cutting it open and carefully transferring the disk inside to an undamaged case. If the disk is copy protected or has an unknown sector format, there are tools that can be used to analyze its idiosyncrasies and help determine a solution. But, if the data on the disks is of uncertain value, it may not be worthwhile to expend the time necessary to address these problems, and in some cases problems will persist no matter how much effort is expended. Here it becomes a case of preserving what one can.

If a disk is truly unreadable, the only question is whether to preserve the physical media against the day when another attempt might be made. If the disk can be imaged but with bad sectors, then it seems reasonable to preserve that image along with a log file indicating where damage was encountered. If it is possible to extract files, that should also be done, however the possibility of damage should be clearly indicated in the metadata accompanying the image and extracted files.

Conclusion: Forensics or Archaeology?

This paper was originally envisioned as a technical note exploring the disk image ingestion capabilities of our library's digital preservation storage system, Archivematica. However during the early stages of research it became apparent that Archivematica could only perform a bit-level ingestion of the floppy disk images I had been acquiring in the course of my data recovery projects. Also, even had it been able to extract individual files, Archivematica did not recognize common 1980s formats like Wordstar and Appleworks, so was limited in its capacity to migrate their contents to archival formats.[25]

Archivematica relies heavily on open source tools developed by the digital forensics community. As noted above, these tools (for example, The Sleuth Kit and fiwalk) are not designed to handle the peculiarities of 1980s floppy disk image formats. The comparative recency of the emergence of digital forensics as a professional discipline seems a likely explanation. Most digital forensics software in common use dates from the late-1990s onward and reflects a disciplinary focus on retrieving data from contemporary storage media, not media long considered obsolete.

Archivematica's disk image ingestion capabilities appear to have been optimized to work with images created by BitCurator[26] , or more accurately with its forensic imaging software Guymager. Guymager is limited to imaging within the constraints of the disk controller on the physical hardware of the underlying system, which in a typical configuration would be a PC controller. This means it lacks a number of the specialized abilities of the Kryoflux mentioned in the section "Disk Imaging" above, making it a less than optimal choice for floppy disk imaging.

It should be noted that in addition to the standard suite of forensics tools, BitCurator includes a number of tools that originated outside the digital forensics community. Specifically, it includes mtools and hfsutils, both mentioned above, and HFS Explorer for accessing the contents of older Mac floppies. It would be straightforward to further expand BitCurator to encompass most of the other software discussed in this paper, and this could in fact be done by the end user.

Building similar content handling capabilities into Archivematica would be considerably more effort given the more tightly coupled nature of its tool chain, and it is possible to question whether it would be worth the effort. In my experience, recovering content from floppy disks involves enough variables to sufficiently confound any attempt to fully automate the process.

As the case studies above illustrate, employing a variety of specialized tools represents a more viable approach to content recovery from 1980s floppy disks than relying on the software found in archival toolkits. Given the long involvement of the retro-computing and software preservation communities with this problem domain, it is not surprising that the most effective tools originate from those communities as distinct from the realm of digital forensics.

About the Author

John Durno is Head of Library Systems at the University of Victoria (UVic), where he manages the team responsible for building and maintaining the Libraries' computing environment. For the past two years he been working with UVic Archives and Special Collections to develop preservation strategies for at risk digital materials on volatile media. Prior to joining UVic in 2006 John coordinated province-wide library technology projects for the British Columbia Electronic Library Network.

Footnotes

[1] Limitations of digital forensic software have been noted within the forensics community itself. See Beebe, N. Digital Forensic Research: The Good, The Bad and the Unaddressed. Advances in Digital Forensics V, IFIP AICT 306. 2009. doi:10.1007/978-3-642-04155-6_2: 'One of the successes identified in the previous section was the collective ability to archaeologically identify, excavate and examine digital artifacts. The problem, however, is that knowledge and expertise are

heavily biased toward Windows, and to a lesser extent, standard Linux distributions. The FAT12/16/32, NTFS and EXT2/3 file systems, the operating systems that implement them (Windows 9X/ME/NT/XP/Vista and various Linux distributions), and common user applications installed on them (Microsoft Internet Explorer and Outlook, Mozilla Firefox and Thunderbird, etc.) have been well studied. Researchers have paid insufficient attention to other operating systems, file systems and user applications, especially in light of current market trends.'

[2] In Kirschenbaum M, Ovenden R, Redwine G. 2010. Digital Forensics and Born-Digital Content in Cultural Heritage Collections. Available from https://www.clir.org/pubs/reports/pub149/pub149.pdf

[3] Omniflop disk imaging tool available from: http://www.shlock.co.uk/Utils/OmniFlop/

[4] Dunfield, D. Disk Image/Software Image Archive. Available from: http://www.classiccmp.org/dunfield/img/index.htm

[5] For a full description of this workstation and its construction, see Durno, J, & Trofimchuk, J. 2015. Digital forensics on a shoestring: a case study from the University of Victoria. code{4}lib journal 27. Available from: http://journal.code4lib.org/articles/10279

[6] See, for example, Formatted HD DOS floppies reading as unformatted. 2016. Kryoflux Support Forums. Available from: http://forum.kryoflux.com/viewtopic.php?f=3&t=1166

[7] AFF format deprecated. Available from: https://sourceforge.net/p/guymager/wiki/AFF%20format%20deprecated/

[8] Forensics Wiki. Encase image file format. Available from: http://www.forensicswiki.org/wiki/Encase_image_file_format

[9] Louis-Guérin, J. 2013. Kryoflux Stream File Documentation. Revision 1.1. Available from: http://www.kryoflux.com/download/kryoflux_stream_protocol_rev1.1.pdf

[10] ClamAV. Available from https://www.clamav.net

[11] Haardt, M. Cpmtools 2.0. Available from: http://www.moria.de/~michael/cpmtools/

[12] SleuthKitWiki. HFS. Available from: http://wiki.sleuthkit.org/index.php?title=HFS

[13] For further discussion of the challenges of working with early Mac floppies, see Purity, S. Working with Macintosh floppy disks in the new millenium. Available from: http://siber-sonic.com/mac/newmillfloppy.html

[14] Leslie, R. HFS Utilities. Available from: http://www.mars.org/home/rob/proj/hfs/

[15] Wikipedia. Resource fork. Available from https://en.wikipedia.org/wiki/Resource_fork

[16] 'Apple II' series, available from https://en.wikipedia.org/wiki/Apple_II_series

[17] Information about the evolution of Apple II operating systems is widely available, see for example 'DOS 3.3, ProDOS & Beyond' available from http://apple2history.org/history/ah15/

[18] KEGS Apple IIgs emulator. Available from: http://kegs.sourceforge.net

[19] Green, R. AppleCommander. Available from http://applecommander.sourceforge.net/

[20] For more information on this topic, see UCSD p-System disks (PC version). 2016. Kryoflux Support Forums. Available from http://forum.kryoflux.com/viewtopic.php?f=3&t=1153

[21] Atari800. Available from http://atari800.sourceforge.net/

[22] Sample CP/M and DOS ingestion scripts, see https://github.com/jdurno/floppy-utils

[23] For an excellent discussion of the challenges and opportunities afforded by emulation as an access strategy, see Dietrich D, Kim J, McKeehan M, & Rhonemus A. 2016. How to Party Like it's 1999: Emulation for Everyone. code{4}lib journal 32. Available from: http://journal.code4lib.org/articles/11386

[24] The basic priniciple is described in: Just solve the file format problem. WordStar. Available from http://fileformats.archiveteam.org/wiki/Wordstar. . Also see my implementation of the principle at: https://github.com/jdurno/floppy-utils/blob/master/wsconv

[25] While Wordstar is listed in the PRONOM registry that Archivematica relies on for file format identification, its ability to recognize the Wordstar format appears to be based on the file extension. Unfortunately, appending file extensions to identify file

types was not as common in the 1980s: while I have encountered hundreds of Wordstar files in my work to date I have never seen one with a .ws, .ws3 or equivalent extension.

[26] Reference "Wherever possible, use BitCurator packages for forensics tools". Archivematica Wiki. https://wiki.archivematica.org/Digital_forensics_image_ingest

Works Cited

Atari800. Available from http://atari800.sourceforge.net/

Beebe, N. 2009. Digital Forensic Research: The Good, The Bad and the Unaddressed. Advances in Digital Forensics V, IFIP AICT 306. doi:10.1007/978-3-642-04155-6_2

Chessman S. 1996. dd. Linux Journal 32. Available from: http://www.linuxjournal.com/article/1320

Diamond E. 1994. The Archivist as Forensic Scientist – Seeing ourselves in a different way. Archivaria 38. Available from: http://journals.sfu.ca/archivar/index.php/archivaria/article/view/12031/1300

Dietrich D, Kim J, McKeehan M, & Rhonemus A. 2016. How to Party Like it's 1999: Emulation for Everyone. code{4}lib journal 32. Available from: http://journal.code4lib.org/articles/11386

Durno J. 2016. Floppy-utils. Available from: https://github.com/jdurno/floppy-utils

Durno J, Trofimchuk J. 2015. Digital forensics on a shoestring: a case study from the University of Victoria. code{4}lib journal 27. Available from: http://journal.code4lib.org/articles/10279

Forensics Wiki. Encase image file format. Available from: http://www.forensicswiki.org/wiki/Encase_image_file_format

Green, R. AppleCommander. Available from http://applecommander.sourceforge.net/

Guymager Wiki. AFF format deprecated. Available from: https://sourceforge.net/p/guymager/wiki/AFF%20format%20deprecated/

Haardt, M. Cpmtools 2.0. Available from: http://www.moria.de/~michael/cpmtools/

Just solve the file format problem. WordStar. Available from http://fileformats.archiveteam.org/wiki/Wordstar

Kirschenbaum M, Ovenden R, Redwine G. 2010. Digital Forensics and Born-Digital Content in Cultural Heritage Collections. Available from https://www.clir.org/pubs/reports/pub149/pub149.pdf

KEGS Apple IIgs emulator. Available from: http://kegs.sourceforge.net

Kryoflux Support Forums. 2016. Formatted HD DOS floppies reading as unformatted. Available from: http://forum.kryoflux.com/viewtopic.php?f=3&t=1166

Kryoflux Support Forums. 2016. UCSD p-System disks (PC version). Available from: http://forum.kryoflux.com/viewtopic.php?f=3&t=1153

Leslie, R. HFS Utilities. Available from: http://www.mars.org/home/rob/proj/hfs/

Louis-Guérin, J. 2013. Kryoflux Stream File Documentation. Revision 1.1. Available from: http://www.kryoflux.com/download/kryoflux_stream_protocol_rev1.1.pdf

Pollitt M. 2010. A History of Digital Forensics. Advances in Digital Forensics VI, IFIP AICT 337

Purity, S. Working with Macintosh floppy disks in the new millenium. Available from: http://siber-sonic.com/mac/newmillfloppy.html

Ross S, Gow A. 1999. Digital Archaeology: Rescuing Neglected and Damaged Data Resources. Available from http://www.ukoln.ac.uk/services/elib/papers/supporting/pdf/p2.pdf

SleuthKitWiki. HFS. Available from: http://wiki.sleuthkit.org/index.php?title=HFS

Weyhrich, Steven. DOS 3.3, ProDOS & Beyond. Available from: http://apple2history.org/history/ah15/

Wikipedia. Resource fork. Available from https://en.wikipedia.org/wiki/Resource_fork

Wolverton, M. 2016. Digital Forensics in the Library. Nature. Vol 534: 139-140. Available from: http://www.nature.com/polopoly_fs/1.19998!/menu/main/topColumns/topLeftColumn/pdf/534139a.pdf

Subscribe to comments: For this article | For all articles

Appendix IV

Descriptive texts written for The Averted Eye Sees: The Life and Work of Glenn Howarth. Exhibition held July 30 – October 12th, 2016
Legacy Maltwood Gallery

Glenn Howarth's Telidon Art
[Gallery wall text]

"Telidon was machine stitched into a corner of the Canadian modern age flag" -- Glenn Howarth.
Telidon began in 1978 as a project of the Canadian federal Deparment of Communications to develop networked information services using Videotex technology. Telidon offered significantly higher quality graphics than competing systems developed by France and the United Kingdom.

Glenn Howarth first encountered Telidon in 1981, when he met two University of Victoria faculty members working on projects to develop Telidon systems, Dr. David Godfrey in Creative Writing and Dr. Ernest Chang in Computer Science. Howarth's painting was not going well at the time and he leapt at the chance to explore a new medium: "I had discovered the electronic paint brush I had dreamed of," he wrote.

Installation view, Legacy Maltwood Gallery

He soon came to believe his work had a larger significance, writing "The computer can be used for political control, and it can be used to strengthen and liberate the human spirit. Humanism + the creative imagination must struggle to appropriate computer technology."

Howarth continued working in computer graphics for over three years, gaining international recognition for his efforts. At the apex of his involvement he represented Canada as a Telidon artist at the 1983 Biennale in Sao Paulo, Brazil.

Following this period of success however Howarth grew increasingly dissatisfied with the limitations of the medium and the difficulties of working with intractable technology. He formally declared his "emigration" from Telidon and return to painting on March 9th, 1985. His disillusionment paralleled the failure of Telidon systems to gain a foothold in the marketplace, and preceded the withdrawal of federal funding for the Telidon project by less than a month. The last major exhibition of Glenn Howarth's Telidon art was at Expo 86 in Vancouver.

Restoring Glenn Howarth's Telidon Art
[Gallery wall text]

In 2012 the University of Victoria Archives received a donation from the estate of the late Glenn Howarth, RCA. Included in the donation were approximately two dozen 5.25" floppy disks containing his digital artworks and related material. A formal project to restore these works began in January 2015, led by John Durno (UVic Libraries).

From writings in the archive we were able to determine that Glenn Howarth had used software developed by the University of Victoria Computer Science department. The software, called Picture Creation System (PCS), enabled users to create graphics files for display on Telidon systems. Telidon was a project of the Canadian Federal Department of Communications to build a networked computer information utility. Lasting from 1978 through 1985, it was a uniquely Canadian effort with significant involvement from both the public and private sectors.

During the late 70s and early 80s personal computers were not yet capable of displaying the sophisticated (for the time) graphics that were the hallmark of Telidon. Telidon systems required dedicated decoder boxes to interpret graphics files which would then be output to a consumer-grade television set.

This meant the restoration of Glenn Howarth's artworks depended on locating an obscure and still functioning piece of computing hardware from the early 1980s. While initial efforts met with little success, in the fall of 2015 we determined that such a device existed in the collection of SPARC, a volunteer-run radio museum in Coquitlam, BC.

With the help of SPARC volunteer Brent Hilpert, we were able to connect a modern laptop to the decoder and display Howarth's Telidon artworks on its built-in CRT monitor. A digital recording of the artworks was made by Daniel Hogg (UVic Fine Arts), enabling Howarth's works to be viewed on modern computing systems.

Descriptive Texts for Glenn Howarth Web Exhibits

Computer Baby Pictures from GREE part one

“I went to computer graphics with a baby on my mind.” Glenn Howarth’s involvement with Telidon graphics coincided with the birth of his only child in 1982, and the responsibilities of new parenthood provided some of the impetus for his transition to working in the new medium. “Since painting and drawing were going poorly and the opportunity arose to work in Telidon computer graphics, I put aside my freedom, dressed in a jacket and tie, and did what I thought responsible Fathers to be should do, walking down to the office each day, working in full colour with an electric pen.” As Artist-in-Residence at the Victoria-based company Softwords, Howarth undertook a variety of commercial and self-directed projects: “The writer-owner [Dave Godfrey] is delighted to have me there making pictures, anything I choose for whatever reason, commercial or otherwise … “

Computer Baby Pictures was likely made during the early part of Howarth’s Softwords residency in 1982/83 and is part of a larger set of images Howarth made as part of his art practice. The landscape that builds up in the latter half of the sequence has been identified by Robert Amos as the view from Howarth’s Rockland apartment. Local landmarks such as Christ Church Cathedral are visible in the background.

Vegetarian Nightmare from GREE part two

Vegetarian Nightmare dates from slightly later than *Computer Baby Pictures*, likely early 1983. Glenn Howarth described the genesis of the Vegetarian Nightmare sequence in an undated slide lecture, a recording of which is in the artist’s archive. He recalled: “I hadn’t eaten meat in about seven years. And I went to a formal dinner, and I had a bleeding steak … and the nightmares I had all week … never again. These are some of the nightmares, like a steak laying on top of a coal bucket. The cow jumped over the moon and the moon was a steak, and […] pseudo animation allowed me to get that steak to drip blood.”

Starfish Crucifixion from 1984 NAPLPS

During its brief existence, the Telidon standard went through two major versions. The first version, Telidon 699, was superseded in 1983 by a second version. Called variously Telidon 709 and NAPLPS, the second version significantly increased the range of colours available and introduced a number of other new features.

In 1983/84 Glenn Howarth was awarded a Canada Council grant funding his transition from 699 to NAPLPS. During this period he worked as an artist in residence in the Computer Science department at the University of Victoria, providing user feedback to a team developing NAPLPS image creation software.

Starfish Crucifixion is one of three NAPLPS works by Howarth listed in Consortel, a catalogue of Telidon resources published in March 1985. For that reason we can date the work more precisely than most of Howarth's other works (May 1984). Howarth describes the work as follows:

1. A red-pink starfish appears on the fringe of a beach towel beside a cream and tan thigh
2. From above looking down the starfish is on the thigh
3. From a worm's-eye perspective a flesh-coloured crucifix rises steeply against a blue-green sky. Four extra crossarms at forty-five degrees radiate out between the horizontal and vertical members. The starfish is layed out on the six radials and pinned there by multicoloured spikes.

Appendix V

IdeaFest 2016: Retro Computing in the Library - Computer Exhibits
Date of event: March 10, 2016

Imaging Workstations

1. FC5025 Content acquisition station. For 5.25” floppies, 3.5” DOS floppies

2. Old computer built with new parts. Use for 3.5” floppies, with Guymager

Howarth stations:

3. Microtel Telidon Terminal

4. NAPLPS slideshow. (Ubuntu + DOSBox + PP3). Glenn Howarth's NAPLPS graphics

5. Telidon 699 slideshow (Ubuntu/VLC playing on loop)

Kirschenbaum Corner:

6. Windows computer with MiniVmac and *Afternoon* (Michael Joyce) installed.

7. Windows computer with MiniVmac and *Agrippa* installed

8. Windows computer with AppleWin and *Mystery House* installed.

Arcade:

9. Rich McCue’s arcade emulator

10. Jennifer McClintick's original Nintendo Entertainment System

Pre-Internet Shakespeare

11. Old iMac running OS 9 with copy of original CD

12. Emulated Mac (Basilisk II on Ubuntu, running System 7.5) with CD image loaded

13. Windows 95 PC running *Scenario*, from A Shakespeare Suite (early Humanities Computing project)

Old & New

14. PC 286 staff workstation with DOS, WordPerfect 5.1.

15. New computer that looks old: Linux workstation configured with Common Desktop Environment, Mosaic browser.

16. Windows computer with Commodore Amiga emulator

17. Windows computer with Commodore 64 emulator

18. Windows 98 PC running *Myst*

19. Apple IIc computer from the personal collection of Rich McCue

Descriptions that accompanied the exhibits are included on the following pages. Note that exhibits #1-#3 and exhibits #9 and #19 did not have accompanying descriptions.

Glenn Howarth Telidon/NAPLPS Art [Exhibit #4]

About this exhibit

This exhibit showcases a selection of early computer art by Victoria artist Glenn Howarth, RCA. To create these images Glenn Howarth used locally developed software that implemented the NAPLPS (North American Presentation Level Protocol Syntax) standard developed by the research division of the Canadian federal Department of Communications. It is likely that he created this set of artworks while he was an artist in residence in the Computer Science department of the University of Victoria in 1984. The Picture Creation System software that he used was developed at UVic by a team led by Dr. Ernest Chang.

While these images may look primitive now, they pushed well beyond the limits of what was possible with generally available computing technology at the time they were made. Glenn Howarth gained widespread recognition for his work in Telidon art, and represented Canada in a prestigious international art fair, the Sao Paulo Biennale, in 1983.

Copies of these artworks and others were donated to the University Archives in 2012 on the original source media (5 1/4" floppy disks). Restoration work has been ongoing for the past year. These artworks and others will be featured in an exhibition in the Legacy Maltwood gallery on the lower level of the Mearns Centre/McPherson Library beginning in June 2016.

About Telidon and NAPLPS

Telidon (from the Greek words τῆλε, tele "at a distance" and ἰδών, idon "seeing") was a videotex service developed by the Canadian Communications Research Centre (CRC) during the late 1970s and supported by commercial enterprises led by Infomart in the early 1980s. Videotex services were early examples of networked information utilities. They had some uptake in Europe (particularly in France and the UK) but ultimately did not gain widespread adoption in North America.

The CRC referred to Telidon as a "second generation" system, offering improved performance, 2D color graphics, multi-lingual support and a number of different interactivity options supported on various hardware. With additional features added by AT&T Corporation and others, Telidon was redefined as a protocol and became the NAPLPS standard in 1983.

Telidon failed to garner sufficient uptake to become profitable, and government support for the project wound down in 1985. The NAPLPS standard outlived Telidon for a few years, finding a second life as an image encoding standard in the Bulletin Board Systems that were common in the late 1980s and early 1990s before the web made them obsolete. Its last widespread use was as the basis for the image transmission technology of the Prodigy Online service.

About the Software

Glenn Howarth's NAPLPS artworks were originally displayed on purpose-built hardware that is very difficult to source now. The software we are using to render his images is from the latter part of the NAPLPS era. Called "PP3", it was developed as a Bulletin Board System client by a Nepean, Ontario company named MicroStar and released as shareware in 1993. PP3 is running in an open source DOS emulator called DOSBox, with its processing speed adjusted downward to approximate the rendering capabilities of 1980s hardware.

Glenn Howarth Telidon 699 Art [Exhibit #5]

About this exhibit

This exhibit showcases a selection of early computer art by Victoria artist Glenn Howarth, RCA. To create these images Glenn Howarth used locally developed software that implemented the Telidon 699 standard developed by the research division of the Canadian federal Department of Communications. It is likely that he created this set of artworks while he was an artist in residence at a local software company, Softwords, in 1982-83. The Picture Creation System software that he used was developed at UVic by a team led by Dr. Ernest Chang. It ran on an Apple IIe and required specialized hardware (a Telidon decoder) to display images.

While these images may look primitive now, they pushed well beyond the limits of what was possible with generally available computing technology at the time they were made. Glenn Howarth gained widespread recognition for his work in Telidon art, and represented Canada in a prestigious international art fair, the Sao Paulo Biennale, in 1983.

Copies of these artworks and others were donated to the University Archives in 2012 on the original source media (5 1/4" floppy disks). Restoration work has been ongoing for the past year. These artworks and others will be featured in an exhibition in the Legacy Maltwood gallery on the lower level of the Mearns Centre/McPherson Library beginning in June 2016.

About Telidon

Telidon (from the Greek words τῆλε, tele "at a distance" and ἰδών, idon "seeing") was a videotex service developed by the Canadian Communications Research Centre (CRC) during the late 1970s and supported by commercial enterprises led by Infomart in the early 1980s. Videotex services were early examples of networked information utilities. They had some uptake in Europe (particularly in France and the UK) but ultimately did not gain widespread adoption in North America.

The CRC referred to Telidon as a "second generation" system, offering improved performance, 2D color graphics, multi-lingual support and a number of different interactivity options supported on various hardware. With additional features added by AT&T Corporation and others, Telidon was redefined as a protocol and became the NAPLPS standard in 1983.

Telidon failed to garner sufficient uptake to become profitable, and government support for the project wound down in 1985. A subsequent version of Telidon, called '709', became the basis for an image encoding standard called NAPLPS that was subsequently used in a variety of other applications in the late 80s and early 90s.

About the Display

Glenn Howarth's Telidon 699 artworks were originally displayed on purpose-built hardware that is very difficult to source now. These images were played back on a rare Telidon 699 decoder (a Microtel VTX-202, on loan from the SPARC Radio Museum in Coquitlam), and recorded with a modern digital video camera. This recording was made by Daniel Hogg in the University of Victoria Department of Fine Arts. It is a work in progress - the shadows that periodically creep across the screen will be eliminated in the finished version.

Afternoon, a story by Michael Joyce [Exhibit #6]

About this exhibit

This exhibit (*Afternoon, a story* by Michael Joyce) is one of three relating to the book *Mechanisms* by Matthew Kirschenbaum, a scholarly study based on forensic analysis of three classic works of digital literature. The other two exhibits are:

- Agrippa (A Book of the Dead) by William Gibson
- Mystery House by Sierra OnLine

About Afternoon

"*Afternoon, a story* is a work of electronic literature written in 1987 by American author Michael Joyce. It was published by Eastgate Systems in 1990 and is known as the first hypertext fiction. Afternoon was first offered to the public as a demonstration of the hypertext authoring system Storyspace, announced in 1987 at the first Association for Computing Machinery Hypertext conference in a paper by Michael Joyce and Jay David Bolter. In 1990, it was published on diskette and distributed in the same form by Eastgate Systems. This is one of the most-discussed works of electronic literature, and many articles have been written about it. Espen J. Aarseth devotes a chapter of his book *Cybertext* to Afternoon, calling it a classic example of modernist literature. It is more often thought of as a work of Postmodern literature, as evidenced by its inclusion in the Norton Anthology of Postmodern American Fiction." Source: Wikipedia (edited and abridged)

The version of Afternoon in this exhibition is the third edition, released by Eastgate Systems in 1992. It was recovered from a floppy disk in the University of Victoria Libraries' collection.

How to view Afternoon

Afternoon runs in a Mac System 7 emulator called "Mini vMac". Click on the "afternoon" icon on the Desktop. Mini vMac will boot up and the Mac Desktop will appear. Click on the "Eastgate Systems" icon and then click on the "Afternoon, a story" icon.

About Mini vMac

MiniVmac is an open source emulator for Mac System 6 and System 7 (Classic OS). It emulates the hardware of an old Mac only and requires a Mac system ROM and operating system disk image to work. Assuming you have those components it is easy to install and run.

Agrippa (A Book of the Dead) by William Gibson [Exhibit #7]

About this exhibit

This exhibit (Agrippa, by William Gibson) is one of three relating to the book *Mechanisms* by Matthew Kirschenbaum, a scholarly study based on forensic analysis of three classic works of digital literature. The other two exhibits are:

- Afternoon, a story by Michael Joyce
- Mystery House by Sierra OnLine

About Agrippa

“Agrippa (A Book of the Dead) is a work of art created by science fiction novelist William Gibson, artist Dennis Ashbaugh and publisher Kevin Begos Jr. in 1992. The work consists of a 300-line semi-autobiographical electronic poem by Gibson, embedded in an artist's book by Ashbaugh. Gibson's text focused on the ethereal nature of memories (the title is taken from a photo album). Its principal notoriety arose from the fact that the poem, stored on a 3.5" floppy disk, was programmed to encrypt itself after a single use.” source: Wikipedia

One of the ironies of Agrippa is, as Matthew Kirschenbaum notes, “in Gibson's 'Agrippa' we have an electronic text that is volatile and ephemeral *by design,* which nonetheless turns out to be one of the most persistent and available literary artifacts on the Web.”

The Agrippa disk image and much more is available from *The Agrippa Files*, a scholarly site that provides a unique collection of materials relating to the work's development, release and reception. http://agrippa.english.ucsb.edu/

How to view Agrippa

Agrippa runs in a Mac System 6 emulator called “Mini vMac.” If the Mini vMac window is open you will need to close it before you can launch Agrippa. You can close Mini vMac in one of two ways:

1. Select “Shut down” from under the “Special” menu on the desktop, and then click the x button in the upper left side of the screen.

2. If the “Special” menu is not visible (for example, if Agrippa is currently running) you can force Mini vMac to quit by holding down Control and Q, and then pressing Y.

To launch Agrippa, click on the “agrippa” icon on the Desktop. The Mini vMac emulator will boot, and then you will see the Mac Desktop appear. Click on the “Agrippa (a book of the dead)” icon, and then click on the “Agrippa” icon in the folder that has just opened. A splash screen will appear, followed by a graphic image. Be patient – the text of the poem will appear and begin scrolling approximately three minutes later.

The emulation will pause if you click outside the MiniVmac window. Click inside the window to restart it.

Mystery House (Hi-Res Adventure #1) by Sierra OnLine [Exhibit #8]

About this exhibit

This exhibit (Mystery House by Sierra OnLine) is one of three relating to the book *Mechanisms* by Matthew Kirschenbaum, a scholarly study based on forensic analysis of three classic works of digital literature. The other two exhibits are:

- Afternoon, a story by Michael Joyce
- Agrippa, by William Gibson

About Mystery House

"Mystery House is an adventure game released by On-Line Systems in 1980. It was designed, written and illustrated by Roberta Williams and programmed by Ken Williams for the Apple II. Mystery House is the first graphical adventure game and the first game produced by On-Line Systems, the company which would evolve into Sierra On-Line ... The game went on to sell 80,000 units worldwide, making it one of the best-selling computer games at the time. GamePro named Mystery House the 51st most important game of all time in 2007, for introducing a visual component to adventure games and for featuring graphics at a time when most computer games did not." Source: Wikipedia (abridged and adapted)

Text adventure games like Mystery House (alongside other examples like Zork and Adventure) are considered to be early examples of digital literature and have been much written about in scholarly journals and books pertaining to that field of study.

Mystery House was released into the public domain in 1987 and is freely available from various web archives. This copy was downloaded from:
http://mirrors.apple2.org.za/ftp.apple.asimov.net/images/games/adventure/mystery_house.dsk

How to play Mystery House

Click the Apple icon on the desktop to start AppleWin, an Apple II emulator. Then click the Apple icon near the top right of the application window to start Mystery House. Follow the on screen instructions that appear after the game has loaded. As it can take months to solve Mystery House, for the purposes of this exhibition you may wish to use the cheat sheat provided with this handout to move through the game, rather than attempting to guess the correct commands.

If you need to restart the game, click the Apple icon near the top right of the application window (the same one you click to start the game) to 'reboot' the emulator.

About AppleWin

AppleWin is an open source Apple II+ emulator for Windows. It has been in development since 1994 and is freely downloadable from https://github.com/AppleWin/AppleWin

Nintendo Entertainment System (1983-1995) [Exhibit #10]

About this exhibit

Video games are legitimate objects of enquiry as any other cultural artifact, and many scholarly articles and books have been written about them. In other to study them however it is necessary to be able to experience them. As technologies go obsolete, it becomes increasingly difficult to experience video games in anything like their original context.

This exhibit showcases a Nintendo Entertainment System (NES) from the mid-1980s. It is becoming increasingly unusable however as it does not work with modern LCD television sets. It requires a CRT monitor for its display.

About the NES

The Nintendo Entertainment System is an 8-bit home video game console that was developed and manufactured by Nintendo. It was initially released in Japan as the Family Computer on July 15, 1983, and was later released in North America during 1985, in Europe during 1986, and Australia in 1987. It was succeeded by the Super Nintendo Entertainment System.

The best-selling gaming console of its time, the NES helped revitalize the US video game industry following the video game crash of 1983. With the NES, Nintendo introduced a now-standard business model of licensing third-party developers, authorizing them to produce and distribute titles for Nintendo's platform.

In 2009, the Nintendo Entertainment System was named the single greatest video game console in history by IGN, in a list of 25. It is the second greatest console behind only the Sega Dreamcast in PC Magazine's "Top 10 Video Game Consoles of All Time".

Adapted from: https://en.wikipedia.org/wiki/Nintendo_Entertainment_System

Pre-Internet Shakespeare [Exhibits #11 & #12]

About this exhibit

This exhibit showcases an early version of what became the Internet Shakespeare Editions, an internationally renowned website developed by Dr. Michael Best, hosted and partly sponsored by the University of Victoria. Prior to its emergence on the web in 1996, it was published (as "Shakespeare's Life and Times") on floppy disks as a Hypercard stack for Apple MacIntosh computers in 1991. A revised edition on CD ROM was released in 1994.

While the Internet Shakespeare Editions continues grow and evolve on the web, its precursor no longer runs on modern computers. This exhibit showcases Shakespeare's Life and Times in two computing environments:

1. The CD running on an Apple iMac dating from the late 1990s
2. A disk image of the CD running on the Basilisk II, an emulator for Mac OS 7/8 systems from the mid 1990s, installed on Linux

Are there any significant differences to experiencing the content on period hardware versus an emulation?

About Internet Shakespeare Editions/Shakespeare's Life and Times

"The Internet Shakespeare Editions is a non-profit organization that produces a website devoted to William Shakespeare and his works. The organization is an associate member of the Shakespeare Theatre Association of America, under the classification of theatre service provider, and is supported by the University of Victoria and the Social Sciences and Humanities Research Council of Canada.

The website includes a variety of Shakespeare-related resources, including fully annotated texts of his plays and poems, multimedia materials and records of his plays in performance, and historical information about Shakespeare's life and the Renaissance."
Source: https://en.wikipedia.org/wiki/Internet_Shakespeare_Editions

"The Internet Shakespeare Editions project began in 1991 when Dr. Michael Best, a Renaissance scholar at British Columbia's University of Victoria (UVic), developed a HyperCard multimedia resource, Shakespeare's Life and Times, with design by graphic design specialist Roberta Livingstone. The work was published by Intellimation Inc. (Santa Barbara, California) on eight floppy disks. Within three years, a revolution took place in new media, and after further development, Shakespeare's Life and Times was published by the same company on CD ROM."
Source: http://internetshakespeare.uvic.ca/Foyer/history/

How to view Shakespeare's Life and Times

Click on the desktop icon to launch the application, and follow the instructions on-screen.

About Basilisk II

Basilisk II is an open source emulator for 680x0 Apple Macintosh computers. It was first released in 1997. Its official website is at http://basilisk.cebix.net/

Scenario, from A Shakespeare Suite [Exhibit #13]

This exhibit showcases software called Scenario, developed 1998-2002 by Martin Holmes and Stewart Arneil in the Humanities Computing and Media Centre, based on a concept by Michael Best. Scenario is an application that allows you to try out different stagings of Shakespeare plays.

It was published on CD-ROM in 2002 as part of "A Shakespeare Suite." It is the third of three exhibits showcasing work relating to the Internet Shakespeare Editions.

About the Internet Shakespeare Editions

“The Internet Shakespeare Editions is a non-profit organization that produces a website devoted to William Shakespeare and his works. The organization is an associate member of the Shakespeare Theatre Association of America, under the classification of theatre service provider, and is supported by the University of Victoria and the Social Sciences and Humanities Research Council of Canada.

The website includes a variety of Shakespeare-related resources, including fully annotated texts of his plays and poems, multimedia materials and records of his plays in performance, and historical information about Shakespeare's life and the Renaissance.”
Source: https://en.wikipedia.org/wiki/Internet_Shakespeare_Editions

“The Internet Shakespeare Editions project began in 1991 when Dr. Michael Best, a Renaissance scholar at British Columbia's University of Victoria (UVic), developed a HyperCard multimedia resource, Shakespeare's Life and Times."

About Windows 95

Scenario is running on Microsoft Windows 95, one of the most widely used computer operating systems of all time. It represented a major milestone in the popularization of the graphical user interface (GUI), a mode of computer interaction involving windows, icons and mice that is ubiquitous now.

Although Macs and Amigas had adopted a graphical user interface as much as 10 years previously, when it was released in 1995 Windows 95 marked the first time Microsoft fully embraced GUI concepts. Given Microsoft's market share at the time, Windows 95 represented the point at which the GUI hit the consumer mainstream.

The first edition of Windows 95 was promoted with much fanfare upon release and sold millions of copies in the first few weeks. It initially shipped without Internet Explorer and did not have internet capability out of the box.

This workstation is a typical configuration for the mid-late 1990s, somewhat upgraded from its original specifications. Its processor, a Pentium 100, first came on the market in 1994. More specs for this workstation are on the other side of this page.

Specs:

Hardware:

- Pentium 100 CPU
- 64 MB RAM (upgrade)
- S3Trio64 video card with 2 MB Video RAM
- Sound Blaster Awe 64 sound card (upgrade)
- DEC Digital DE500-BA (211143-PC) network card (upgrade)
- 80 GB IDE hard drive (upgrade)

Software:

- Windows 95 installed (the original release with the FAT 16 file system)
- Internet Explorer 5.5 SP2
- Netscape Navigator 4.80
- Microsoft Office 97 Pro
- Word Perfect Suite 8
- Claris Home Page 3.0
- Front Page 97
- Shakespeare Scenario
- An assortment of utilities (WinZip, WSFTP, etc.)

Library Staff Computer, c. 1990-95 [Exhibit #14]

About this exhibit

Over the years many thousands of computers have passed through the Library. The typical service life of a computer is five years, after which they tend to go out of warranty, develop mechanical problems and it becomes cost-ineffective to continue supporting them. Most of the Library's computers from years gone by have long since left the building, but this one lingered on in an obscure basement storage room, where it was only recently rediscovered. Appropriately enough, it was used by staff in our Archives and Special Collections unit.

Technical specs

This computer has a 286/16 mhz processor running MS DOS 6.2. It was likely upgraded from its initial configuration, as it has a whopping 250MB hard drive that would have been cost prohibitive to purchase in 1989. It has 2MB RAM, which was considered to be more than adequate for a DOS computer of this era. (In 2016, new UVic desktop computers have 4000 times that much RAM). It has two floppy drives, 5.25" and the newer 3.5" format. The 3.5" drive still works, but the 5.25" drive is broken.

About the manufacturer

This computer was manufactured in June 1989 by the Hewitt Rand Corporation of Richmond BC. The company was one of several BC computer manufacturers in the 80s and 90s, of which the best known was probably Seanix. Hewitt Rand was founded in 1983 and was acquired by Merit Distributing in 1999. It appears to have dropped off the face of the earth shortly thereafter.

"Hewitt Rand" reflects a practice that was common among small-scale computer manufacturers in the 1980s, combining words from the names of other, more prestigious technology companies (in this case, Hewlett-Packard and Remington Rand) to create a name with an impressive-sounding pedigree. "Packard Bell" is another example of a company that took this approach.

What Can I Do With This Computer?

Not much! Our staff workstations were fairly limited in 1990. This one didn't even have a network port, just a parallel port for an attached printer. In its day it was primarily used for word processing and had the defacto standard software, WordPerfect 5.1, installed.

You can start WordPerfect by typing "wp" at the DOS prompt (C:\>). From there, you can activate the on-screen menu by right-clicking with your mouse, and then left-clicking on the option you want to select. Or you can refer to the handy, very large manual to learn about more advanced functionality. If you want to take your work home, the 3.5" floppy drive is your only option, but sadly, floppy disks are a bit hard to come by these days.

Common Desktop Environment/Mosaic Web Browser [Exhibit #15]

About this exhibit

Most of our retro computing Ideafest exhibits take one of two forms: either they are original computing hardware or they are modern systems emulating original hardware. This exhibit is neither. It illustrates another way to keep historic software going: adapt it to run on a contemporary operating system. This Dell computer is running a modern version of Ubuntu Linux, but its desktop environment software and Mosaic web browser date from the Unix world of the mid 1990s.

About the Common Desktop Environment

“The Common Desktop Environment (CDE) is a desktop environment for Unix and OpenVMS, based on the Motif widget toolkit. It was part of the UNIX98 Workstation Product Standard, and was long [1994-2000] the "classic" Unix desktop associated with commercial Unix workstations.”
Source: Wikipedia

About the Mosaic Web Browser

“NCSA Mosaic, or simply Mosaic, is a discontinued early web browser. It has been credited with popularizing the World Wide Web. Its intuitive interface, reliability, Windows port and simple installation all contributed to its popularity within the web ... Mosaic was also the first browser to display images inline with text instead of displaying images in a separate window.

“Mosaic was developed at the National Center for Supercomputing Applications (NCSA) at the University of Illinois Urbana-Champaign beginning in late 1992. NCSA released the browser in 1993, and officially discontinued development and support on January 7, 1997.”
Source: Wikipedia

How to use this Exhibit

Feel free to click around and explore the features of the Common Desktop Environment and the Mosaic web browser. The home page of the Mosaic Browser has been set to a local mirror of a collection of historical web pages at CERN, the European nuclear research facility that is also the birthplace of the World Wide Web. While the Mosaic browser capably renders pages from this site, it cannot adequately render most modern web pages. The software still runs, but the web has evolved beyond its abilities.

Because this workstation is running a modern operating system under the hood, it is capable of running a modern web browser (Firefox in this case). You can use Firefox to find out more about the Common Desktop Environment.

As this computer is not on a network, you cannot use either browser for general web browsing.

About the Software
Both Mosaic and the Common Desktop Environment are being maintained in their current incarnations by volunteers, and are freely available on Github and Sourceforge, respectively:

Mosaic: https://github.com/alandipert/ncsa-mosaic
CDE: https://sourceforge.net/projects/cdesktopenv/

Commodore Amiga Emulator [Exhibit #16]

This exhibit showcases an emulator for Amiga systems, *Amiga Forever*, packaged and sold online by the software company Cloanto. The package makes extensive use of an open source Amiga emulation program called UAE, the “Universal Amiga Emulator”.

While it would in theory be possible to put together a similar package using software freely downloadable from a variety of web archives, it would be hard to do so legally, because most of the software you would need is still covered by restrictive licensing agreements. The legal challenges of emulating older systems can sometimes outweigh the technical ones.

Besides convenience, the main value of the Cloanto package therefore is that the vendor has negotiated licenses for all of the software they include.

About the Amiga

While Apple is typically given all the credit for popularizing the point and click graphical user interface, in reality the early Apple Macs were quite limited and expensive. The Commodore Amiga, first released commercially in 1985 (a year after the first Macs) had many features that were missing from the Macs of that era: a colour display with an extensive palette, stereo sound, and true pre-emptive multitasking (the ability to run more than one program at a time). They were the first true multimedia computers, and were highly popular with artists and other creative types, and video game enthusiasts. Remarkably, they also cost significantly less than Macs and IBM PCs, their main competitors.

Consequently early models sold very well, mainly to the home market. Unfortunately, bad marketing hampered sales of later models, and technological stagnation allowed both PCs and Macs to eventually catch up with the Amiga. The last Amiga model (the CD32) was released in 1993. Not a lot of people bought them. Commodore went bankrupt in 1994.

What can I do with Amiga Forever?

Click on the “Amiga Forever” icon to launch the Amiga Forever browser. From there you can select the “Systems” folder to choose from among a variety of Amiga systems. Double click on a system to boot it up. Early systems like the Amiga 500 attempt to replicate the experience of booting on older hardware, right down to the noise of the floppy drive!

If you want to play a game you can choose one of the many games from the “Games” folder. Note that a number of these games will not play correctly as they expect you to have peripherals (such as a joystick) that are not included in this display. If you would like to play a game that works with a keyboard and mouse, follow the instructions for Arkanoid on the following page.

To view the first 30 pages of a scholarly history of Amiga systems, click on the PDF document “The Future Was Here” on the desktop. The author relied heavily on emulation for his research, stating “Indeed, some of the deepest software explorations in this book would have been virtually impossible if I had been restricted to real Amiga hardware, without access to the additional tools UAE provides the digital archaeologist.” [p. Xii] The whole book is available online via the University of Victoria Libraries' catalogue.

How to Play Arkanoid

If the Amiga explorer window is not already open, click on the “Amiga Forever” icon on the Desktop to open it. Then click on the “Games” folder and double-click on “Arkanoid”.

After the application window opens, click on the small icon with the 4 arrows to expand it to full screen. (To make it smaller again, press the Alt and Esc keys at the same time).

After an initial Amiga DOS loading screen, some large bouncing text will appear. Click the left mouse button to make it go away.

The Arkanoid splash screen will appear. Click the left mouse button again to get to the main game window.

Press F1 to start the game in one player mode.

After the intro, It will ask you to select a round. You can click the left mouse button to start at round one, or move your mouse left and right to go to different rounds. Then click the left mouse button to begin playing.

Once the round starts, you can control the left and right movement of the paddle with your mouse. Left clicking triggers certain events (launching the ball, firing when in weaponized mode).

The goal of the game is to clear the coloured blocks from each level. Gameplay changes (typically becomes easier in some way) when your paddle contacts one of the losenges that intermittently fall from the blocks.

Arkanoid follows the classic “3 strikes” rule (miss the ball 3 times and the game is over).

About Arkanoid

Arkanoid (アルカノイド Arukanoido) is an arcade game developed by Taito in 1986. It expanded upon Atari's Breakout games of the 1970s by adding power-ups, different types of bricks, and a variety of level layouts. The title refers to a doomed "mothership" from which the player's ship, the Vaus, escapes.

Upon release, the arcade game was critically acclaimed ... The home versions were also well received. Computer Gaming World stated in 1988 that Arkanoid on the Amiga was "a perfect version of the arcade game ... incredible!"

Source: Wikipedia
https://en.wikipedia.org/wiki/Arkanoid

Commodore 64 Emulator [Exhibit #17]

Like “Amiga Forever”, “C64 Forever” is a commercial software package published by Cloanto, originally a developer of software for Commodore 64 and Amiga Systems. It is based on the open source VICE emulator and can emulate a variety of Commodore computer models including the Commodore 64, the Vic 20, and the Commodore PET, among others.

About the Commodore 64

“The Commodore 64, also known as the C64, is an 8-bit home computer introduced in January 1982 by Commodore International. It is listed in the Guinness World Records as the highest-selling single computer model of all time, with independent estimates placing the number sold between 10 and 17 million units.

“Volume production started in early 1982, with machines being released on to the market in August at a price of US$595 (roughly equivalent to $1,500 in 2016). Preceded by the Commodore VIC-20 and Commodore PET, the C64 takes its name from its 64 kilobytes (65,536 bytes) of RAM, and has technologically superior sound and graphical specifications when compared to some earlier systems such as the Apple II and Atari 800, with multi-color sprites and a more advanced sound processor.

“The C64 dominated the low-end computer market for most of the 1980s. For a substantial period (1983–1986), the C64 had between 30% and 40% share of the US market and two million units sold per year.”

Source: Wikipedia

What Can I Do With This Computer?

Click on the “C64 Forever” icon on the desktop to open the C64 Explorer window. Click on the Systems folder to choose from a variety of Commodore operating systems. Double click on C64 to boot into Commodore 64. (Note that if you choose a different operating system, the rest of these instructions might not work).

You might try a simple program in BASIC. Type the following:

```
10 INPUT “WHAT IS YOUR NAME? “;A$
20 PRINT “YOUR NAME IS: ”;A$
```

Then type RUN to run your program.

If that gets boring, you can refer to the Commodore 64 Manual on the Desktop for more ideas, or see the following page for instructions on how to play Zork 1, one of the earliest text adventure games.

Playing Zork 1

Zork 1, one of the earliest text adventure games, has been attached to the C64 emulator as a floppy disk image. Don't worry if you don't know what this means.

To load Zork, you will need to type the following command:

LOAD”*”,8

Then press the “Enter” key. You will see the following message:

SEARCHING FOR *

LOADING

READY.

Type the word RUN, and then press Enter. You will be instructed to wait for one and a half minutes while the game loads. Refer to the Zork manual on the Desktop for further instructions.

About Zork

“Zork is one of the earliest interactive fiction computer games, with roots drawn from the original genre game, Colossal Cave Adventure. The first version of Zork was written in 1977–1979 using the MDL programming language on a DEC PDP-10 computer. The authors—Tim Anderson, Marc Blank, Bruce Daniels, and Dave Lebling—were members of the MIT Dynamic Modelling Group.

“When Zork was published commercially, it was split up into three games: Zork: The Great Underground Empire - Part I (later known as Zork I), Zork II: The Wizard of Frobozz, and Zork III: The Dungeon Master.

“Zork distinguished itself in its genre as an especially rich game, in terms of both the quality of the storytelling and the sophistication of its text parser, which was not limited to simple verb-noun commands ("hit troll"), but recognized some prepositions and conjunctions ("hit the troll with the Elvish sword").”

Source: Wikipedia

Other Games and Operating Systems

C64 Forever emulates many different Commodore operating systems. You can find more systems to boot up and explore in the “Systems” folder. It also comes with a variety of old computer games, as you can see in the “Games” folder. Feel free to try any of these, but note that many of the games will not work well as they typically expect additional peripherals (such as a joystick) to be attached to the computer.

Windows 98/Myst [Exhibit #18]

About Myst

Myst is a graphic adventure puzzle video game designed and directed by the brothers Robyn and Rand Miller. It was developed by Cyan, Inc., published by Brøderbund, and initially released on the Macintosh platform on September 24, 1993. Remakes and ports of the game have been released for platforms including Sega Saturn, PlayStation, 3DO, Microsoft Windows, Atari Jaguar CD, CD-i, AmigaOS, PlayStation Portable, Nintendo DS, iOS, and OS X. The game puts the player in the role of the Stranger, who uses a special book to travel to the island of Myst. There, the player solves puzzles and travels to other worlds known as "Ages". Clues found in each of these Ages help to reveal the back-story of the game's characters. The game has several endings, depending on the course of action the player takes.

Upon release, Myst was a surprise hit, with critics lauding the ability of the game to immerse players in the fictional world. The game was the best-selling PC game until The Sims exceeded its sales in 2002. Myst helped drive adoption of the then-nascent CD-ROM format. Myst's success spawned four direct video game sequels as well as several spin-off games and novels.

Source: https://en.wikipedia.org/wiki/Myst

About Windows 98

Windows 98 (codenamed Memphis while in development) is a graphical operating system by Microsoft. It is the second major release in the Windows 9x line of operating systems and the successor to Windows 95. It was released to manufacturing on May 15, 1998 and to retail on June 25, 1998.

Like its predecessor, Windows 98 is a hybrid 16-bit/32-bit[4] monolithic product with an MS-DOS based boot stage. Windows 98 was succeeded by Windows 98 Second Edition on May 5, 1999, which in turn was succeeded by Windows ME on June 19, 2000. Microsoft ended mainstream support for Windows 98 and 98 SE on June 30, 2002, and extended support on July 11, 2006.

Unlike the first release of Windows 95, Windows 98 shipped with a web browser and full internet capability, reflecting the huge growth of the web in the second half of the 1990s.

Source: https://en.wikipedia.org/wiki/Windows_98

Retro Computing Inventory
2016.07.31

Description	Powers On	Boots	Notes	Maintenance Performed
Mac Desktops				
Macintosh Classic	Yes	Yes		
Macintosh Plus 1Mb	Yes	Yes	Requires boot floppy	New battery, basic cleaning
MacIntosh SE/30	Yes	No	Serial mouse. HD might be dead. Radius full page display expansion card	
Macintosh Classic	Yes	No	Only two screws in back. Doesn't boot, but sees mouse. Save for parts.	Battery removed 1/2 AA 3.6 volts. Adjusted monitor brightness
Macintosh Classic II	Yes	Not tested	Dead monitor?	Battery removed 1/2 AA 3.6 volts
Macintosh SE	Yes	Yes	Perfect. System 7. 10MB Hard drive. MacWrite II, licensed to Janos Sandor (Hungarian/Canadian conductor)	
Macintosh Classic	Yes	Yes	Works well. System 7.0.1. Works and MS Word 5	Battery removed 1/2 AA 3.6 volts
Macintosh Plus	Yes	Yes	1988 vintage. Needs boot floppy. Monitor blanks out intermittently	New battery, basic cleaning
Power Mac G4 Cube	Yes	No	Hard drive appears to have been wiped. Tried installing OS X 10.1 and it hung before completion and then shut itself down. Mechanical problems most likely, also the optical drive doesn't eject properly	
Power Mac G4 Cube	Yes	Yes	Appears to be in excellent working order. Boots OS 9.1	
iMac (Bondi blue)	Yes	Yes	Boots OS 9. Works but noisy drive suggesting possible mechanical failure imminent.	
iMac(Blueberry)	Yes	Yes	OS 9. Good working order.	
Power Mac G3 desktop	Yes	Yes	Excellent mid-1990s desktop. Networked, working 3.5” floppy drive	
PowerPC Performa	?	?		
G3 Tower	Yes	Yes		
G4 Tower	Yes	Yes		
Mac Laptops				
Mac Powerbook Duo 270c	Yes	No	Screen shows light, but not even a Mac question mark	
Mac Powerbook Duo Power supply	Yes	Yes	Works but exposed wires on connector	
Mac Powerbook Duo 270c Floppy adapter	?	?	Need to get the Powerbook Duo working	
Mac Powerbook Duo 3.5” Floppy drive	?	?	Need to get the Powerbook Duo working	
Mac Powerbook Duo Microdock	?	?	Need to get the Powerbook Duo working	
Mac Powerbook 180	Yes	Yes	Worked once. System 7.1. Word 5. Hypercard 2.1 player. Think power supply died after first use, but could be something else	
Mac Powerbook 165	?	?	Need power supply	
Mac Powerbook 170	?	?	Need power supply	
Mac Powerbook 145b	?	?	No power supply, no hard drive. Note from Greg says it works	
Mac Accessiories/Peripherals				
M0110a Keyboard for Mac Plus			Requires special cord, which we have. Some keys don't work: M, X maybe others.	
M0110a Keyboard for Mac Plus			Requires special cord, which we have. Acquired via Ebay. All keys work.	
Apple High Resolution Monochrome Monitor	Yes	N/A	Not sure if working	
Apple Monitor	Yes	N/A	A little washed out, but not bad	
5 Expansion cards for 1979 Apple II graphics tablet			The tablets are super rare, and these cards alone go for $60 US and up on Ebay. We'll need these if we ever want to get the Ernie Chang's Telidon 699 software going again (PCS)	
Apple computer 1981 super serial card			Also needed to resurrect PCS	
3 MO100 Apple Serial Mice				
2 A9M0331 Apple ADB Mice				

Apple Extended Keyboard III ADB				
Apple computer 1981 parallel interface card				
Apple M2706 ADB Mouse				
Early Apple connector cable				
Apple Quicktake 100 camera				
Apple HDI SCSI Dock adapter				
Harmon Kardon speakers for G4 Cube				
Carry bag for compact Mac form factor				
PC Desktops				
Hewitt Rand 286, 1989 vintage	Yes	Yes	Everything works. Former staff computer, comes with WordPerfect installed.	Imaged hard drive
IBM PS/2	Yes	Yes	Works but real time clock needs repair	see Jerry
Dell Pentium 3	Yes	Yes	Boots Windows 98, Windows 2000	see Jerry
Dell Pentium 100	Yes	Yes	Boots Windows 95 (original version)	see Jerry
Dell Optiplex 520	Yes	Yes	Boots DOS, XP, CentOS	see Jerry
PC Laptops				
Zenith Data Systems ZFL 181 92 luggable laptop. Two pop-up 3.5” floppy drives.			Need power supply. Built like a tank. Would be great to get this going if possible	
Toshiba Sattelite 2500 CDS. Windows NT/98 compatible. 2 PCMCIA slots. 3.5” FD and CD ROM drives.			Missing power supply. Pretty beat up. Might be useful if not too hard to get going.	
Toshiba T1000. Half-size screen. 3.5” floppy drive			Missing power supply. Probably least useful of these PC laptops. Discard?	
Zenith Data Systems Supersport. 3.5” floppy drive			Built like an even bigger tank than the other Zenith. Battery case but no battery. Neat looking but probably not useful. Discard?	
PC Accessories/Peripherals				
Microsoft Windows 3.1 (on 5 3.5” disks) in orig box with User's guide, etc				
Microsoft Windows 95 (on 13 3.5” disks plus upgrade CD) in orig box with setup guide, etc				
MS DOS 5 (on 3 3.5” floppy disks) with User's Guide, etc in orig box				
MS Word 6 on 3.5” floppies (2 sets) in orig box with User's Guide etc.				
Non PC\Mac variants				
DEC VT125 Terminal	Yes	Yes	Screen might need some adjustment, otherwise appears to work well	
NeXT Workstation	No	No	Needs a lot of work: missing hard drive, keyboard, mouse, fan, circuitry for driving mouse, keyboard and speakers, monitor, cables	
Radio Shack MC-10 TRS-80			Missing power supply & monitor cables	
Sun Ultra1 Workstation c. 1995	Yes	No	Boots into BIOS but not O/S. Configured to network boot but obviously not connected to network. Tried running disk boot command but could not find bootable disk. Possibly missing hard drive?	Purchased VGA adapter so modern monitor can be used. Located Sun keyboard but still need Sun mouse
Commodore 64	Yes	No	Black screen of death. Could be caused by many things, most likely a fried chip	
Miscellaneous				
2 Flip chip, stamped 1974 Modem test conn H3I5A			For DEC PDP11	
Atari mouse, SM 124 monitor owners manual, Atari ST Basic Quick reference, Computer to VHF TV connector				
Misc CalComp, Kurta drawing tablets (6 in box)				
1 box 10 DS DD 5.25” diskettes (J.K.Gill) Unopened(!)				
Floppy drive for Commodore 64				
DEC Terminal keyboards (3)			Some missing keys	