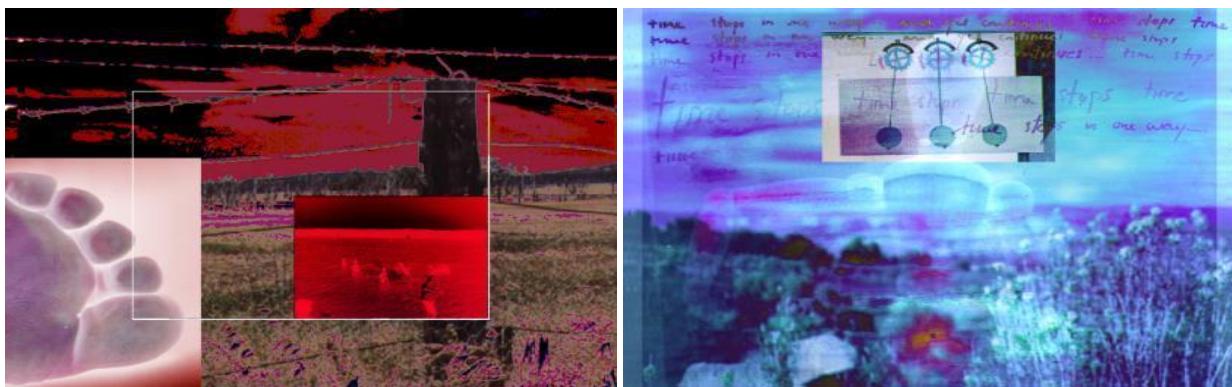


# Preserving and Emulating Digital Art Objects

November 2015

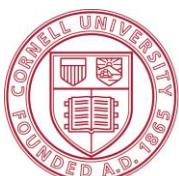


Norie Neumark; Shock in the Ear: <https://contactzones.cit.cornell.edu/artists/neumark.html/>

White Paper Report Submitted to the National Endowment for the Humanities

Oya Y. Rieger and Tim Murray, CoPIs  
Madeleine Casad, Project Manager  
Desiree Alexander, Collections Analysis Assistant  
Dianne Dietrich, Technical Lead  
Jason Kovari, Metadata Lead  
Danielle Mericle, Director, Digital Media Group  
Liz Muller, Digital Archives Advisor  
Michelle Paolillo, Repository Manager

Project Website: <https://confluence.cornell.edu/display/pafdao/Home>



Cornell University  
Library

## Table of Contents

<b>INTRODUCTION .....</b>	1
<b>MEDIA ART AND CULTURAL HISTORY .....</b>	2
<b>About the Goldsen Archive of New Media Art.....</b>	2
<b>Imminent Risks for New Media Art .....</b>	3
<b>Related Studies in Preservation Practice.....</b>	5
<b>Initial Project Goals .....</b>	6
<b>PROJECT ACCOMPLISHMENTS .....</b>	7
<b>Initial Advisory Group Meeting and Collection Review .....</b>	7
<b>Workstation and Disc Imaging .....</b>	8
<b>Metadata .....</b>	9
<b>Emulation Investigations.....</b>	10
<b>Emulation – Preservation Issues .....</b>	11
<b>Emulation – Rendering Issues .....</b>	12
<b>Understanding User Needs and Determining Significant Properties .....</b>	12
<b>Artist Questionnaire .....</b>	14
<b>CONCLUDING REMARKS.....</b>	15
<b>PROJECT TEAM .....</b>	18
<b>PROJECT CONSULTANTS AND ADVISORS .....</b>	20
<b>APPENDICES</b>	

## INTRODUCTION

In 2013, Cornell University Library received a research and development grant from the National Endowment for the Humanities to design a framework for preserving access to digital art objects. The Preservation and Access Frameworks for Digital Art Objects project (PAFDAO) was undertaken in collaboration with Cornell University's Society for the Humanities and the Rose Goldsen Archive of New Media Art, a collection of media artworks housed in the Library's Division of Rare and Manuscript Collections. This collection of complex interactive born-digital artworks is used by students, faculty, and artists from various disciplines.

Despite its "new" label, new media art has a rich 40-year history, making obsolescence and loss of cultural history an imminent risk. As a range of new media are integrated in art works, these creative objects are becoming increasingly complex and vulnerable due to dependence on many technical and contextual factors. The phrase "new media art" denotes a range of creative works that are influenced or enabled by technological affordances. The term also signifies a departure from traditional visual arts (e.g., paintings, drawings, sculpture, etc.). Another characteristic of new media art that adds further complications to the preservation process is its interactive nature. Works in this genre often entail, and indeed rely on, interactions between artists and viewers/observers.

Interactive digital assets are far more complex to preserve and manage than single uniform digital media files. The project aimed to develop scalable technical frameworks and associated tools to facilitate enduring access to complex, born-digital media objects, working primarily with a test bed of nearly 100 optical discs from the holdings of the Goldsen Archive. The preservation model developed will apply not merely to new media artworks, but to other rich digital media environments. Many of the issues we have been addressing within the framework of this project pertain to other rich digital contents, not limited to artistic productions.

This white paper describes the project's findings, discoveries, and challenges. The ultimate goal of the project team has been the creation of a preservation and access practice grounded in thorough and practical understanding of the characteristics of digital objects and their access requirements, seen from the perspectives of collection curators and users alike. Equally important has been the establishment of service frameworks and policies that are sustainable, realistic, and cost-efficient. So all through the project, one of our principles has been moving the experience gained through research into practice. This white paper aims to contribute to better and more practical understanding, management, and curation of digital assets. Although the initiative focused on new media art, we hope that our methodologies and findings will inform other types of complex born-digital collections as well.

## MEDIA ART AND CULTURAL HISTORY

### About the Goldsen Archive of New Media Art

The ultimate aim of the PAFDAO project was to create generalizable new media preservation and access practices that will be applicable for different media environments and institutional types. The nature of the project's test collection, a set of CD-ROM artworks from Cornell's Rose Goldsen Archive of New Media Art<sup>1</sup>, has meant that the project provides a case study in new media preservation that may be informative to library and museum contexts alike.

Rose Kohn Goldsen (1917-1985) was a professor of Sociology at Cornell University and an early critic of commercial mass media's impact on social and ethical imagination. Named in her honor, the Rose Goldsen Archive of New Media Art was founded in 2002 by Professor Timothy Murray (Director, Society for the Humanities, Cornell University) in the Cornell Library Division of Rare and Manuscript Collections as an international research collection for scholars of new media and media art history.<sup>2</sup> Since its founding, the Goldsen Archive has grown to achieve global recognition as a prominent research collection that documents more than 60 years of the history of aesthetic experimentation with electronic communications media. These collections span the two most crucial decades in the emergence of digital media art, from 1991 to the present, tracing the historical shift in emphasis within media culture from disc-based to networked and Web-based applications. They also mark the early stirrings of a networked, interactive digital culture that has subsequently become the global norm. The Goldsen Archive constitutes a vital record of our cultural and aesthetic history as a *digital* society.

A wide range of audiences explore the Goldsen, which supports on-site access to all of its holdings in keeping with CUL's archival mission. The Goldsen Archive is cataloged and accessible via the library's online catalog and the WorldCat global catalog. The archive's website includes guides and collection descriptions to facilitate discovery of materials by potential users. The Division staff are actively involved in Cornell courses, including Goldsen materials. The Goldsen Archive is regularly visited by academic researchers, media and technology scholars, and artists. The Goldsen Archive has also conducted a number of public programming initiatives and there are plans for a 2016 exhibit at Cornell to highlight the collection.

---

<sup>1</sup> The Goldsen Archive's holdings range to include media formats such as reel-to-reel videotape, floppy disk, database artworks housed on external hard drives, and works of net.art. All of these formats pose unique and significant preservation challenges. For more information, see the Goldsen Archive website:

<http://goldsen.library.cornell.edu/>

<sup>2</sup> Murray, T. (2008). Thinking Electronic Art Via Cornell's Goldsen Archive of New Media Art. NeMe: The Archival Event. <http://www.neme.org/843/the-archival-event>

## **Imminent Risks for New Media Art**

The Goldsen's interactive digital holdings enhance the world's understanding of a significant set of influential artists and their oeuvre. These collections chart the transformation of artistic practices across the two most crucial decades of the digital revolution, an historical shift in emphasis within media culture from analog to disc-based to networked and Web-based applications. The vast majority of the Goldsen's digital artworks engage or interrogate the terms of this transformation. In addition to providing an important collection of national and international art history, the Goldsen Archive constitutes a vital record of our cultural and aesthetic history as a *digital* society. In the coming years, historical collections of interactive, born-digital assets like those in the Goldsen Archive will become increasingly valuable for study, appreciation, and understanding of digital cultural history. The Goldsen is the only American *onsite-accessible* research collection of this work that continues to collect and preserve on an active basis.

A CD-ROM artwork may consist of varied, complex, and overlapping aesthetic experiences: sound recordings of music and dramatic monologues, digital paintings, short video clips, densely layered audiovisual essays that the user navigates and explores with the clicks and movements of a computer mouse. Expansive and complex, the artwork may include many sections, each with its own distinct aesthetic, expressed through rich sound and video quality and intuitive but non-standard modes of interactivity. Because so much of new media works' cultural meaning derives from their responsiveness to spontaneous input — and the user's direct, unrepeatable experience of this interaction — these technological threats pose serious challenges to the Goldsen's collections.

The PAFDAO project focused on a subset of born-digital media artworks on CD-ROM. These artworks were created for small-screen, single-user experience, and dated back as far as the early 1990s. To begin with, no archival best practices yet exist for preserving such assets. Many are stored on fragile storage media like optical discs, meaning that physical damage as well as data degradation or "bit rot" pose serious dangers to the integrity of the information. In the case of the PAFDAO project's test collection, many of these discs were artist-produced and irreplaceable.

Interactive digital assets are, furthermore, far more complex to preserve and manage than single, uniform digital media files. A single interactive work can comprise an entire range of digital objects and dependencies, including media files in different types and formats, applications to coordinate the files, and operating systems to run the applications. If any part of this complex system fails, the entire asset can become unreadable. This danger is especially acute in the case of artworks. In most cases, interactive digital artworks are designed to create unique, multimedia experiences for users. An even relatively minor problem with an artwork's

rendering—for example, an obsolete media player that no longer operates as expected—has the potential to significantly compromise an artwork's "meaning." Simply migrating information files to another storage medium is not enough to preserve their most important cultural content. When the PAFDAO project began, approximately 70 percent of the artworks in the test collection could not be accessed at all without using legacy hardware—a specialized computer terminal that runs obsolete software and operating systems.



- Media fragility
- Complex technical composition
- Interrelated dependencies
  - Third-party apps and plugins
  - Obsolete Hardware
  - Operating Systems
- Original documentation possibly missing or incorrect
- Lack of established best practices for archival preservation

### *Cultural Heritage at Risk: Preservation Challenges*

The project's objective was to provide "best-feasible" access to artworks, and document the distance between "feasible" and "ideal," as well as we could understand it. Very soon after beginning PAFDAO, the project team realized that, contrary to our initial assumptions, operating system emulation would be a viable access strategy at scale for our complex digital media holdings.<sup>3</sup> Embracing emulation as an access strategy meant that the team could provide better access more easily to more artworks in the collection. Though increasingly feasible, however, emulation is not always an ideal access strategy: emulation platforms can introduce rendering problems of their own, and emulation usually means that users will experience technologically out-of-date artworks with up-to-date hardware. This made it all the more important for the team to survey media art researchers, curators, and artists, in order to gain a better sense of the relative importance of the artworks' most important characteristics for different kinds of media archives patrons.

<sup>3</sup> For information about emulation, see Lange, A. (2012). KEEP Strategy Paper at:  
[http://bw-fla.uni-freiburg.de/wp-uploads/bw-fla.uni-freiburg.de/2012/03/strategy\\_paper\\_KEEP\\_expert\\_workshop\\_Joining\\_Forces\\_Berlin\\_final.pdf](http://bw-fla.uni-freiburg.de/wp-uploads/bw-fla.uni-freiburg.de/2012/03/strategy_paper_KEEP_expert_workshop_Joining_Forces_Berlin_final.pdf)

## **Related Studies in Preservation Practice**

Libraries, museums, and archives have been collecting and caring for complex born-digital materials for several decades. These institutions have been struggling to find ways to provide long-term access to content that is dependent on a number of technologies, including processors, operating systems, software, file systems, storage media, network interfaces, and file formats, some or all of which may have become obsolete. Recent research and development projects are investigating viable, sustainable strategies for preserving complex born-digital content. Strategies range from capturing digital collections using digital forensic tools, to supporting scholarly research by preserving the user experience of interactive works through video documentation.

Some examples of influential studies that have informed our study include:

- “Digital Forensics and Born-Digital Content in Cultural Heritage Collections,” published by CLIR in 2010, analyzes in detail the tools of the digital forensics community, and offers step-by-step guidance for archivists who want to use these same tools to clone or make a forensic image of the device.<sup>4</sup>
- Since 1999, arts archiving institutions have worked to develop archival frameworks that address the complexity and particular preservation challenges of complex, interactive artworks. Some of the most successful frameworks have come from Variable Media Initiative (VMI), a project undertaken by the Guggenheim Museum and the Daniel Langlois Foundation.<sup>5</sup> VMI collaboratively authored the Variable Media Questionnaire, an archival tool that emphasizes behavioral and ephemeral aspects of interactive artworks.
- Documentation and Conservation of the Media Arts Heritage (DOCAM), an international research alliance, has produced cataloging and preservation guides for complex media art objects.<sup>6</sup>
- Another important archival model has been that of “scoring” interactive media experiences as they unfold in time.<sup>7</sup>
- In Europe, the Keeping Emulation Environments Portable (KEEP 2009-2012) project developed “flexible tools for accessing, manipulating and storing a wide range of digital

---

<sup>4</sup> Kirschenbaum, Matthew G., et al, “Digital Forensics and Born-Digital Content in Cultural Heritage Collections,” available at: <http://www.clir.org/pubs/reports/pub149>

<sup>5</sup> Variable Media Initiative: <http://www.guggenheim.org/new-york/collections/conservation/conservation-projects/variable-media>

<sup>6</sup> See project description and publications at: <http://www.docam.ca/>

<sup>7</sup> See Rinehart, Richard, “A System of Formal Notation for Scoring Works of Digital and Variable Media Art,” available at <http://www.bampfa.berkeley.edu/about/formalnotation.pdf>

objects using emulation tools.” KEEP also released a metadata model about the technical environment required to access and manipulate complex digital objects.<sup>8</sup>

- The net.artdatabase initiative documents user interaction with obsolete computing environments through a simultaneous display of a video of the user interacting with a work on a desktop computer and a screen capture of the interaction.
- The Preserving Virtual Worlds Project has developed and tested imaging technologies, emulation strategies, metadata models to document complex dependencies, and methods for packaging complex objects and associated metadata.<sup>9</sup>

## Initial Project Goals

Our project is informed by and builds on the findings of the excellent R&D efforts described in the previous section. Within this context, the PAFDAO project has developed new preservation and access models at scale, using a real and sizeable test collection, and focusing on the following necessary advances and refinements as the project goals:

1. Identifying significant properties needed to provide for the long-term preservation and access of new media objects.
2. Defining a metadata framework to support capture of technical and descriptive information to support long-term preservation and potential reuse.
3. Creating Submission Information Packages (SIPs) that document dependencies through comprehensive representation information, and define and capture significant properties of interactive works, so that packages can be ingested into a preservation repository.
4. Exploring resource requirements, including staff skills, special equipment needs, and associated costs.
5. Understanding and assessing preservation viability for interactive digital assets to contribute to a more refined understanding of “preservation viability” for complex digital assets into the near future.

---

<sup>8</sup> For further information, including reports from the first and second years of the project, see the KEEP website at: <http://keep-project.eu>

<sup>9</sup> McDonough, Jerome P., et al, “Preserving Virtual Worlds Final Report” (2010), available at <https://www.ideals.illinois.edu/handle/2142/17097> pg. 6

## PROJECT ACCOMPLISHMENTS

### Initial Advisory Group Meeting and Collection Review

We began with an initial meeting of the project advisory board, and elicited our advisors' recommendations for workstation setup and preservation workflow. Advisors strongly recommended that we explore operating system emulation as an access strategy – a suggestion that had a significant impact on our plans for the remainder of the project. After this meeting, we began a broad analysis of CD-ROM artworks in the test collection. This work included reviewing existing MARC catalog records for the artworks, verifying system requirement information and unique identifiers for each work, and updating this information wherever necessary. We also ran each work in legacy and contemporary hardware and operating systems (using a Macintosh computer from 1998, and a PC workstation running Windows 7, for example) to get a general sense of how each work functioned in either a “native” or a current compatible computing environment. This analysis provided a baseline or control value against which we would test the quality of subsequent access experiments. Also, we photographed the physical discs, as well as any accompanying materials, such as booklets, pamphlets, or decorative jewel cases, in order to preserve and simplify future access to these contextual materials as well as the artworks themselves.



*Many of the artworks in our test collection were created for computer operating systems that are now obsolete. This vintage iMac is an important component of the project's digital workstation.*

At the center of our initial efforts was analyzing CD-ROM and Internet Art to determine classes and groupings, and selecting subsets of classes of material to test based on broad impact, feasibility, and scholarly value. Complementary to this effort was the identification of

appropriate data models for documentation of classes and representation information. These processes have overlapped and informed each other throughout the first project year. Thus far our analysis has focused on CD-ROM artworks already cataloged in the Goldsen core collection (100 discs representing approximately 200 artworks). We began with metadata derived from artworks' existing MARC catalog records and compared published and observed system requirements, updating MARC records as necessary. We ran the artworks on both contemporary and legacy hardware, documenting any problems, and took notes on the artworks' rendering in legacy environments to use as a baseline for emulation tests. We then created disk images—byte-perfect preservation copies—of all artworks in the test collection, noting any problems.<sup>10</sup> This overview led to the identification of classes based on operating systems, common software dependencies or plug-ins, the presence of audio-only formatting on a disc (so-called “hybrid” discs), or the need for an active Internet connection.

### **Workstation and Disc Imaging**

Next, we established a digital forensics workstation. We used a Dell computer partitioned to run both Windows 7 and BitCurator, which is based on Ubuntu 14.04. We provisioned a 2TB secondary storage drive for all working files and code development. We kept operating systems cleanly separated from storage of working files, and found this separation essential; on more than one occasion, we were forced to wipe and reload the operating systems, and we were able to do so without losing any of our data. All memory was backed up daily using Cornell University's EZ-Backup service. We used the AVPreserve Fixity tool to monitor the stability of directories whose assets were destined for ingest in the Cornell University Library Archival Repository (CULAR).

---

<sup>10</sup> We have used two main utilities for the disc imaging process: ISO Buster and Guymager. Guymager provides more thorough output files for technical metadata, which can be imported directly into our metadata framework. Of the two, ISO Buster is more adaptable and therefore more appropriate for hybrid discs or discs with initial imaging problems. Also, we have identified a number of digital forensics utilities that automatically capture file-level technical metadata from disc images: fiwalk, hls, disktype, FITS, and fido. We are exploring the Digital Forensics XML (DFXML) standard, which appears able to accommodate disparate and overlapping outputs.

```

Acquisition
-----
linux device : /dev/sr0
Device size : 671913984 (671.9MB)
Format : Linux split dd raw image - file extension
Image path and file name: /media/sdb1/datadrive/emulation/mac_system
Temp path and file name: /media/sdb1/datadrive/emulation/mac_system
Hash Calculation : MD5, SHA-1 and SHA-256
Source verification : on
Image verification : on

No bad sectors encountered during acquisition,
No bad sectors encountered during verification,
Status: successfull

MD5 hash:
MD5 hash verified source : dab4234c71f7efdf90478b5124ec5c57
MD5 hash verified image : dab4234c71f7efdf90478b5124ec5c57
SHA1 hash:
SHA1 hash verified source : 1e0011204cc800bc6ff3465bbad07572efbe81
SHA1 hash verified image : 1e0011204cc800bc6ff3465bbad07572efbe81
SHA256 hash:
SHA256 hash verified source : c18b0c54ca5617d8aa1a007d45e307871a545e1
SHA256 hash verified image : c18b0c54ca5617d8aa1a007d45e307871a545e1
Source verification ok. The device delivered the same data during as
image verification OK. The image contains exactly the data that was

Acquisition started: 2013-12-03 12:21:17 (ISO format YYYY-MM-DD HH:
Verification started: 2013-12-03 12:21:45
Ended
Duration: 0 hours, 0 minutes and 28 seconds
Acquisition speed: 4.33 MB/s (0 hours, 2 minutes and 28 seconds)
Verification speed: 4.36 MB/s (0 hours, 2 minutes and 27 seconds)

```

- Sector-by-sector copy of disc; retain structure and contents
- Separate data from fragile media
- Provide evidence we have made a trustworthy copy

### *Risk Imaging and Visual Documentation*

We customized the digital forensics computer with an internal Plextor CD drive from the early 2000s; however, internal adapters were needed to place this drive in a contemporary machine, and the drive never functioned to our specifications. We soon replaced it with a usb-connected external drive. We began making disc images of the artworks: bit-perfect copies that preserve the file structure of the original discs. We investigated several different imaging utilities.<sup>11</sup>

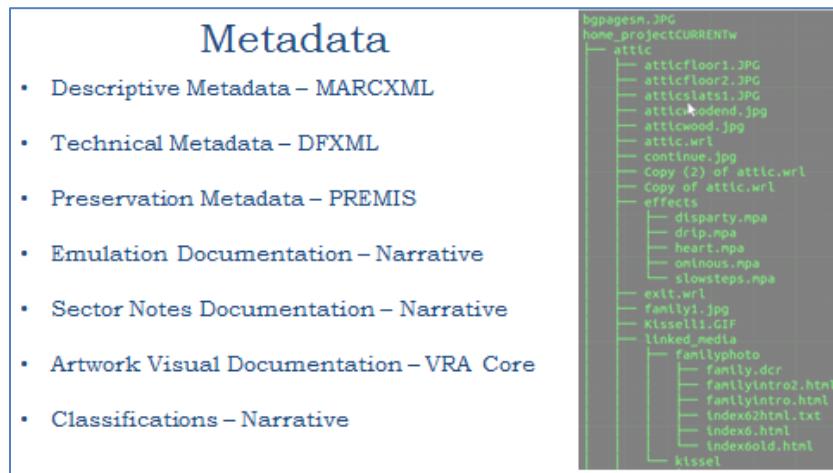
### **Metadata**

Creating a metadata framework for the project was an ongoing, iterative process guided by a range of goals and objectives. For example, the framework addressed the need to avoid over-description and preferred processes that could be automated. Further, it incorporated feedback from our survey of media art researchers addressing preservation and access concerns of artists themselves while working within the architecture of the Cornell University Library Archival Repository (CULAR). The final metadata framework drew on a combination of MARCXML descriptive metadata, Digital Forensics XML (DFXML) technical metadata, PREMIS XML preservation metadata and unstructured descriptive files in order to meet all of these needs. See APPENDIX A for a more exhaustive description of the project's metadata framework. We

---

<sup>11</sup> IsoBuster had a helpful and usable interface, and appears to be more flexible for use with hybrid discs or discs with imaging problems. However, we found that the level of documentation and error reports delivered with the imaging utility Guymager were a better match for our needs. Guymager provides a sector-by-sector analysis of the disc and produces reports that help the technician locate any imaging errors. Guymager also provides three different hashes or checksums to ensure validity. After a period of comparative exploration, we elected to adopt Guymager as our utility of choice, and re-imaged discs that had previously been imaged with ISOBuster to ensure consistency across the test collection.

also identified and tested a number of digital forensics utilities for automatic capture of file-level technical metadata during the disk imaging process: fiwalk, hls, disktype, FITS, and fido. Ultimately the team relied on The Sleuth Kit (TSK) for much technical metadata capture, using additional utilities and locally developed scripts for HFS filesystem details not handled by TSK. Custom scripts aggregated the output of these various utilities in order to generate DFXML metadata for disk image contents.<sup>12</sup>



### *Relevant Metadata Standards*

## **Emulation Investigations**

On the recommendation of our Advisory Board, we began investigating operating system emulation as an access strategy almost immediately after beginning the project. It was quickly apparent that we could use emulation at scale to provide access to the interactive digital artworks in our test collection. This had a dramatic impact on the rest of our work plan, as we were able to focus on emulation and spend fewer staff resources investigating alternatives that would be less effective and more costly to pursue. Serious consideration of emulation as an archival access strategy required us to address a number of problems, however, which will be elaborated below. We tested a range of emulators, and found success with several.<sup>13</sup> Many works created for Windows in the 1990s can still be accessed using contemporary machines, even without emulation. This is the result of market decisions by Microsoft to support

<sup>12</sup> Analysis tools for early Macintosh filesystems (i.e., HFS) are limited, and we relied heavily on custom code to fill out technical metadata for HFS data. We have shared code developed from our original DFXML scripts on GitHub. <https://github.com/cul-it/hfs2dfxml>

<sup>13</sup> Basilisk emulates older Macintosh systems (Motorola processor; OS 7.5.5-8.1); SheepShaver emulates slightly newer ones (PowerPC processor; OS 8.0-9.0.4). For Windows works, we used QEMU. Another tool, WineHQ, can be used to run PC executable files without emulating a complete Windows operating system, but we found that additional third-party software and plug-ins did not work as well, and our tests of WineHQ led to frequent crashes.

backwards compatibility. We note that one effect of this policy is to simplify the effort required of cultural heritage organizations seeking to provide access to such works, but note as well that this policy could change at any time; consequently, we found it prudent to explore emulation for legacy Windows environments as well.



*Project team at work, from left to right,  
Dianne Dietrich, Mickey Casad, and Desiree Alexander*

### **Emulation – Preservation Issues**

Emulation seems an excellent and flexible approach to providing fully interactive access to obsolete artworks at very reasonable quality. However, adopting emulation as an archival access strategy required that we consider a number of issues. First of all, we recognized that relying on emulation as a strategy would require us to preserve emulators as well as artworks. Establishing stable archival identities for emulators is not a simple matter, as most emulators are the product of open, enthusiast-driven development, and important forms of documentation, such as version histories or development notes, tend to be inconsistent. Moreover, emulators, too, are condemned to eventual obsolescence; as new operating systems emerge, the distance between “current” and “historical” operating systems must be recalculated, and new emulators created to bridge this distance anew. We attempted to establish archival practices that would mitigate these instabilities. For example, we collected preservation metadata specific to emulators that included documentation of versions used, rights information about firmware, date and source of download, and all steps taken in compiling them, including information about the compiling environment. We were also careful to keep metadata for artworks emulator-agnostic, in order to avoid future anachronism in our records. This need partially shaped our metadata framework and classification system for

artworks, as described below. [See APPENDIX A for more thorough elaboration of our classification and documentation frameworks.]

### **Emulation – Rendering Issues**

An additional concern with adopting emulation as an access strategy lay in the fact that no emulator can provide a fully “authentic” rendering of a software-based artwork. Emulators exist to allow software to run in a hardware and operating system environment other than the one it was created for. Accessing historical software with current hardware can subtly alter aspects of the work’s rendering. For example, a mouse with a scroll wheel may permit forms of user interactivity that were not technologically possible when a software-based artwork was created. Changes in display monitor hardware (for example, the industry shift from CRT to LED display) brings about color shifts that are difficult to calibrate or compensate for. The extreme disparity between the speed of current and historical processors can lead to problems with rendering speed, a problem that is unfortunately not trivial to solve.

Noting these patterns of rendering shortcomings in emulation helped us refine our criteria for determining the relationship between an artwork’s significant properties and “best feasible” access version. If any of the shortcomings associated with emulation would interfere with communication of an artwork’s most important aesthetic characteristics, we would need to elaborate our access version of the artwork in order to compensate.

### **Understanding User Needs and Determining Significant Properties**

In order to ensure that our access versions and our sense of the works’ most significant aesthetic properties reflected the needs of our user community, we devised and distributed a questionnaire of media art researchers’ practices and preferences. The questionnaire was designed to address a variety of potential disciplinary interests and perspectives and aimed to elicit open-ended, qualitative responses. We distributed the web-based questionnaire through email lists serving media art, digital culture, library and archives interest communities, and had an excellent return of 183 responses.<sup>14</sup> [See APPENDIX B for our survey instrument].

Survey responses did not, as we had initially hoped, lead to the identification of discernible researcher “personas” with distinct profiles of interests and access requirements. Responses did, however, deepen our understanding of researchers’ needs, and of the Goldsen Archive’s unique position in the cultural heritage community. Overall, we found that respondents

---

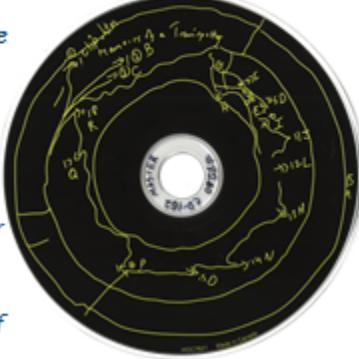
<sup>14</sup> A detailed analysis of the user survey was posted on the DSPS Press, the blog of Cornell University Library’s Department Digital Scholarship and Preservation Services, on July 30, 2014, by Oya Y. Rieger. <https://blogs.cornell.edu/dsps/2014/07/30/interactive-digital-media-art-survey-keyfindings-and-observations>

strongly emphasized the importance of contexts for understanding historical digital artifacts. As a research archive with extensive paper and print collections in addition to media works, the Goldsen Archive is in an exemplary position to provide access to such contexts for understanding alongside access to the artworks themselves.

## User Community & Access

*"I am worried about context and artistic intent – how do we retain authenticity in the long term?"*

*"Access to past works are incredibly valuable to me – understanding works not just for their message but also for their technical [aspects that] help new media artist[s] evolve [this] area of practice."*



### *User-Based Approach*

Survey respondents also noted a number of serious impediments they had encountered practicing research with interactive born-digital artworks. Key among these were lack of resources at the institutional level to support access to interactive digital works, and lack of understanding on the part of institutions about what such support might involve. For these issues to be identified by the media arts community as key impediments validated the objectives of our project, which is structured to address exactly these shortcomings in institutional understanding. We plan to distribute detailed accounts of our findings in order that institutions with comparable collections might more easily embark on preservation and access initiatives of their own.

A matter of greater concern for us was the fact that survey respondents were notably divided in their opinions about emulation. Emulation was controversial for many, in large part for its propensity to mask the material historical contexts (for example, the hardware environments) in which and for which digital artworks had been created. This part of the artwork's history was seen as an element of its authenticity, which the archiving institution must preserve to the best of its ability, or lose credibility in the eyes of patrons. We determined that cultural authenticity, as distinct from forensic or archival authenticity, derived from a number of factors in the eyes of the museum or archive visitor. Among our survey respondents, a few key factors stood out: acknowledgement of the work's own historical contexts, preservation of the work's most

significant properties, and fidelity to the artist's intentions, which is perhaps better understood as respect for the artist's authority to define the work's most significant properties.

The Goldsen Archive, as a research collection, is uniquely situated to provide extensive access to the various historical contexts that gave rise to media artworks. In reviewing survey responses, however, we recognized that we had not adequately incorporated the direct input of artists into our acquisitions process, and that this could have a negative effect on patrons' assessment of the works' cultural authenticity. The need to address this was acute, as we were already quite committed to emulation as an access strategy, which would also put authenticity into question for some users. Museums and galleries regularly conduct semi-structured interviews with artists as an essential element of preservation and conservation practice; it was clear that we should adapt this practice to the collections and the needs of the Goldsen Archive. We would need to work with artists systematically and directly, wherever possible, in order to determine works' most significant properties. If emulation proved problematic or inadequate, we would then be able to devise alternative access strategies in partnership with artists.

### **Artist Questionnaire**

Toward this end we created an artist questionnaire and interview tool. We reviewed several existing models, such as the Variable Media Questionnaire and the questionnaire used by Turbulence.org, but found that the requirements of our situation were unique enough to warrant a new and specific questionnaire tool. [See APPENDIX C for our artist questionnaire.]

The artist questionnaire is designed to be web-based and scalable; it is a first step toward a more complete process that may also involve follow-up interviews and collaboration on access versions and documents to include with the presentation of artworks to archive visitors. Some of the important goals of the questionnaire include:

- Inquiring about basic technical and development information about the artwork; for example, what system was used to create the work, and whether the artist still has working files related to the artwork, and whether they would consider depositing those working files with the Goldsen Archive.
- Asking the artist to describe the ways in which the work reflected the technology of its moment, and how they used that technology to communicate the most important meanings of the work.
- Disclosing our intention of using emulation to provide access to artworks, disclose some common rendering problems we have encountered with emulators, request permission to pursue this strategy for the artwork in question, invite the artist to author a

statement about how emulation may alter the work's intended meaning or impact, and invite the artist to discuss alternative access strategies with us.

## CONCLUDING REMARKS

Throughout our project, a reoccurring theme in our findings involved the difficulties associated with capturing sufficient information about a digital art object to enable an authentic user experience.<sup>15</sup> This challenge cannot and should not be reduced to the goal of ensuring bit-level fixity checks or even providing technically accurate renderings of an artwork's contents as understood on the level of individual files. As Rinehart and Ippolito argue, the key to digital media preservation is variability, not fixity.<sup>16</sup> The trick is finding ways to capture the experience—or a modest proxy of it—so that future generations will get a glimpse of how early digital artworks were created, experienced, and interpreted. So much of new media works' cultural meaning derives from users' spontaneous and contextual interactions with the art objects. Espenschied, *et al.* point out that digital artworks relay digital culture and "history is comprehended as the understanding of how and in which contexts a certain artifact was created and manipulated and how it affected its users and surrounding objects."<sup>17</sup> For a work to be understood and appreciated, it is essential for the archiving institution to communicate a cultural and technological framework for interpretation. As one user survey respondent noted, some works that come across as mundane now may have been among the highly innovative trailblazers of yesterday. Given the speed of technological advances, it will be essential to capture these historical moments to help future users understand and appreciate such creative works.

The PAFDAO survey of users of media archives affirmed the importance of institutions like the Rose Goldsen Archive, which is able to provide a breadth of media technological, historical, and cultural contexts to researchers and educators through its extensive and accessible collections.<sup>5</sup> It also underscored the need for archiving institutions to be in contact with one another, and to be conscious of the need for greater integration of discovery and access frameworks across multiple institutions as they move forward in developing new preservation plans and access strategies for their collections. Providing appropriate cultural and historical contexts for

---

<sup>15</sup> Madeleine Casad, Oya Y. Rieger, Desiree Alexander. "Enduring Access to Rich Media Content: Understanding Use and Usability Requirements." D-Lib Magazine. September-October 2015.

<http://www.dlib.org/dlib/september15/casad/09casad.html>

<sup>16</sup> Rinehart, Richard, and Jon Ippolito, (2014) *Re-Collection: Art, New Media, and Social Memory*. Leonardo. Cambridge, Massachusetts: The MIT Press.

<sup>17</sup> Espenschied, D., Rechert, K., Valizada, I., Von Suchodoletz, D., Russler, N. (2013). "Large-Scale Curation and Presentation of CD-ROM Art", iPRES 2013. [http://purl.pt/24107/1/iPres2013\\_PDF/Large-Scale%20Curation%20and%20Presentation%20of%20CD-ROM%20Art.pdf](http://purl.pt/24107/1/iPres2013_PDF/Large-Scale%20Curation%20and%20Presentation%20of%20CD-ROM%20Art.pdf)

understanding and interpreting new media art is part of each institution's individual mission, but also a matter of collective importance, given the rarity of such collections, the numerous challenges of establishing preservation protocols, and the overall scarcity of resources.

One of the initial project goals was to understand the costs involved in preserving and providing access to new media art. Unfortunately this turned out to be very difficult task due to the experimental and research nature of the project. As much as possible, the project team has considered how the tools and methodologies used can be streamlined beyond the project's duration. We made an effort to factor in the library's broader preservation technology and service framework in order to ensure that the workflows and practices we developed can be mainstreamed.

Throughout our project, we have witnessed several trends and advancements, for instance, the increasing prominence of video and web art. The objective of a 2013 study by the New York Art Resources Consortium (NYARC) was to identify the organizational, economic, and technological challenges posed by the rapidly increasing number of web-based resources that document art history and the art market.<sup>18</sup> One of the conclusions of the study was that regardless of the progress made, "it often feels that the more we learn about the ever-evolving nature of web publishing, the larger the questions and obstacles loom." Although there are relevant standards and technologies, web archiving solutions remain to be costly, and harvesting technologies as of yet lack maturity to completely capture the more complex cases. The study concluded that there needs to be organized efforts to collect and provide access to art resources published on the web. So as we develop strategies for CD-ROM-based content, we are mindful that there are more significant challenges ahead.

We aimed to document use cases and associated costs but realized that this is a challenging task because use cases are profoundly shaped by what is possible, not only what is required by users. As described in the report, although emulation was not included in the original project plan, it emerged as a viable strategy. We would like to underscore one of our project advisor's remarks about emulation, "first it looks great as the process makes it possible to experience art work again – after the initial awe fades, you start seeing glitches." Institutional experiences and perspectives on emulation will not scale unless there are communities of emulation involving archivists and curators. We need to explore how cultural institutions can interface with groups involved in emulation, which currently is driven by games communities and hobbyists.

---

<sup>18</sup> New York Art Resources Consortium (2013). Reframing Collections for a Digital Age: A Preparatory Study for Collecting and Preserving Web-based Art Research Materials.  
[http://www.nyarc.org/sites/default/files/reports/reframing\\_final\\_report2013.pdf](http://www.nyarc.org/sites/default/files/reports/reframing_final_report2013.pdf)

As we conclude, we must emphasize that, as artists have increasing access to ubiquitous tools and methodologies for creating complex art exhibits and objects, we should expect to see an increasing flow of such creative works to archives, museums, and libraries. It is nearly impossible to preserve these works through generations of technology and context changes. Therefore, diligent curation practices are going to be more essential than ever in order to identify unique or exemplary works, project future use scenarios, assess obsolesce and loss risks, and implement cost-efficient strategies. Also, we would like to emphasize that access is the keystone of preservation. The preservation of digital art objects needs to be conceptualized, motivated, informed, and energized by present and future use.

## PROJECT TEAM

**Desiree Alexander** is the PAFDAO Collections Analysis Assistant and has worked with the Goldsen Archive since 2012, assisting with the Goldsen's experimental video and digital media preservation projects. She is also co-lead in surveying Cornell's A/V assets to locate at risk materials campus-wide in an effort to develop preservation and access strategies. She holds a MS in Information Studies and an MA in Public History from SUNY Albany, and an undergraduate degree in Art History from Ithaca College.

**Madeleine Casad** is Curator for Digital Scholarship at Cornell University Library. As Associate Curator of the Rose Goldsen Archive of New Media Art, she manages an exciting collection of media objects that present a wide range of preservation and access challenges. She coordinates many of the Library's Digital Humanities initiatives, and plays a leading role in education and outreach programs to promote the innovative use of digital collections in humanities scholarship. She holds a PhD in Comparative Literature from Cornell University.

**Dianne Dietrich** is Physics and Astronomy Librarian in the Cornell University Library, where her responsibilities include scholarly communication, collection development, instruction, outreach, and reference. From 2013-2015, she was a Fellow in Digital Scholarship and Preservation Services serving as Digital Forensic Analyst for PAFDAO. She received a MS in Information from the University of Michigan and a BA in Mathematics from Wesleyan University.

**Jason Kovari** is Head of Metadata Services and Web Archivist at Cornell University Library (CUL). In addition to providing direction for the Metadata Services unit, Kovari participates on a variety of digital collection and digital preservation efforts across Cornell, including his service on the Cornell University Library Archival Repository (CULAR) steering group and his efforts to capture and preserve web content. He holds an MLS from SUNY Buffalo.

**Danielle Mericle** is the Director for the Digital Media Group at Cornell University Library. In this capacity, she coordinates the work of Digital Consulting and Productions Services, as well as repository development for CUL, including ARTstor, Hydra, and Kaltura. Danielle Mericle holds an MFA from Syracuse University, and has worked in digital library development for over 15 years.

**Liz Muller** is Assistant Director for Technical Services and Curator of Digital and Media Collections in the Division of Rare and Manuscript Collections. She coordinates digital initiatives, media preservation, and the description of archival materials within the Division. She also curates archival collections of historical and digital media, as well as those related to

architecture and urban planning. Muller holds an MS in Library and Information Science from the University of Illinois, an MA in the History of Architecture and Urban Development from Cornell University, and an AB in History from Princeton University.

**Timothy Murray** is a Professor of Comparative Literature and English, and the Taylor Family Director of the Society for the Humanities, Curator of the Rose Goldsen Archive of New Media Art, and co-moderator of the -empyre- new media listserv. A curator of new media art, and theorist of the digital humanities and arts, he sits on the Executive Board of HASTAC, the International Advisory Board of the Consortium of Humanities Centers and Institutes, and the Board of Directors of the National Humanities Alliance. He is currently completing a book, *Archival Events: Curatorial Instabilities @ New Media Art*, which is a sequel to *Digital Baroque: New Media Art and Cinematic Folds* (Minnesota, 2008). His research and teaching cross the boundaries of new media, film and video, visual studies, twentieth-century Continental philosophy, psychoanalysis, critical theory, performance, and English and French early modern studies.

**Michelle Paolillo** is Digital Curation Services Lead at Cornell University Library. She is responsible to strengthen and develop Cornell University Library's (CUL) digital preservation program, and advice as to good practice in lifecycle management. She is the service manager for the Cornell University Library Archival Repository (CULAR) overseeing its development and deposited assets. Michelle holds a BA from Cornell University and 15 years of experience in Information Technologies.

**Oya Y. Rieger** is Associate University Librarian for Scholarly Resources and Preservation Services at Cornell University Library. She provides leadership for full lifecycle management of scholarly content, including selection, creation, design, maintenance, preservation, and conservation. She is interested in current trends in scholarly communication with a focus on needs assessment, requirements analysis, business modeling, and information policy development. She holds a PhD in Human-Computer Interaction (HCI) from Cornell University.

## **PROJECT CONSULTANTS AND ADVISORS**

### **Project Consultants**

AVPreserve: Chris Lacinak, Kara Van Malssen  
Alex Duryee, Optical Media Consultant

### **Advisory Group**

Ben Fino-Radin (MoMA, New York)  
Jean Gagnon (Montreal Cinemateque)  
Rebecca Guenther (Independent Metadata Consultant)  
Matthew Kirschenbaum (Maryland Institute for Technology in the Humanities)  
Jon Ippolito (University of Maine)  
Norie Neumark (Centre for Creative Arts and Media Studies Program, La Trobe University, Melbourne, Australia)  
Christiane Paul (Whitney Museum of American Art)  
Richard Rinehart (Samek Art Gallery, Bucknell University)  
Simeon Warner (Cornell University Library)

**Preserving and Emulating Digital Art Objects**  
**Cornell University Library**

**APPENDICES**

**A: SAMPLE PROJECT DOCUMENTS**

- A1: Artwork Classifications Document
- A2: Emulation Documentation Document
- A3: Pre-Ingest Work Plan for Cornell University Library Archival Repository (CULAR)

**B: SURVEY INSTRUMENT**

- B1: Sample Message Distributed to Mailing Lists and Individuals to Elicit Responses to the Survey of Media Art Researchers
- B2: Survey Distributed to Media Art Researchers

**C: ARTIST QUESTIONNAIRE INSTRUMENT**

- C1: Sample Message Distributed to Artists to Elicit Responses to the Initial Artist Questionnaire
- C2: Initial Interview Questionnaire Distributed to Artists

# Artwork Classifications—PAFDKO Project

Dianne Dietrich

# Contents

A Structure of the classifications	2
B Browser-Based Works	3
C Virtual Reality Components	7
D Executables in Works	10
E Macromedia and Related Executables	15
F HFS File System	18

# A — Structure of the classifications

One sentence describing the purpose of the following classification, including the identifier used in PREMIS to assign this classification to an artwork.

## Short description

A few sentences describing the properties that define a work that have the stated classification.

## Implications for access

This section includes a description of the issues the project team encountered when trying to access a work in this classification. When applicable, there will be a description of possible strategies to consider when providing access to such a work.

## Restoration potential

In this section, the project team considers the process by which one might get such a work running on a current system, without the use of an emulator or virtual machine. It is important to note here that the project team did not attempt to restore every work in the collection, and some descriptions in this section are based on an understanding of the properties of the work and external research, rather than first-hand restoration experience.

## References

Sources consulted and other recommended reading.

## **B — Browser-Based Works**

This classification defines browser-based works. The PREMIS identifier used is “BROWSER.”

### **Short Description**

Browser-based works are defined as those artworks where the user interacts with the work by accessing files through a web browser. While browser-based works use file formats commonly associated with the Internet, they may or may not require Internet connectivity to function properly.

### **Implications for access**

Access to browser-based works can range from simple to complex to access, depending on a number of variables.

In the simplest case, a browser-based work is on a disc that has the standard ISO9660 filesystem and consists solely of files that do not require third-party plugins or specialized auxiliary software (beyond a web browser) to be rendered properly. (Note that it is not required in this case that the only filesystem present be ISO9660, but that at least one of the filesystems present on the disc is in that standard format.) It should be possible to view the work on a current system. It is important to note several complicating factors, even for this simple case.

First, web standards have evolved considerably over time. To take one example, HTML has changed since its first version. Various HTML elements have been deprecated with subsequent versions, meaning that their use is discouraged now.<sup>a</sup> There are also some elements from prior versions that are no longer included in subsequent versions and are no longer considered valid HTML. If these elements exist in HTML files in browser-based works, they may not be displayed [1].

If the browser-based work requires third-party plugins or other auxiliary software to render properly, access can get considerably more complicated. It may not be possible to install the preferred plugin or software on a current system, because the installation files are no longer available.<sup>b</sup> Even if it is possible to obtain the specific installer file for the required plugin or software—if, for instance, the artist included it with the work—it may not be possible to install these executable files on a current system. The available newer versions of the required plugins or software may or may not be compatible with the files in the work.

In both of these cases, viewing the work in an emulator or virtual machine that closely matches the stated system requirements of the work may be the optimal way to interact with the work. If no system requirements are explicitly stated, it is recommended to consider which operating systems and web browsers (and versions) were contemporary with the work, and configuring an emulator or virtual machine to closely match that environment.

A further complication arises when the browser-based work is on a disc that only has an HFS filesystem. In this case, the above description of issues still does apply—with respect to plugins and browser versions—but it is also recommended to read the considerations for “HFS File System” (Section F) for additional information.

Finally, there are aesthetic concerns with browser-based works. Styling within browsers has evolved dramatically over the years [2, 3], and the look and feel of a browser-based work may no longer match the artist’s vision if viewed on a current system.

## Restoration potential

As with access, restoration potential of browser-based works is variable. The process of reworking HTML files or other auxiliary files, such as scripts, so that they no longer contain deprecated or invalid elements may be straightforward or complicated, depending on how much and what kind of revision is needed.

---

<sup>a</sup>It is worth mentioning here that artists might not have adhered to contemporaneous HTML standards, so that invalid or deprecated elements might have always existed in the work, rather than being a product of aging.

<sup>b</sup>It also might be possible that only a specific version is compatible and others are not.

Browser-based works that include files that require specific third-party plugins or auxiliary software may be more difficult to restore. The reasons for this follow from the difficulties in providing access to works that contain executable files (See Section D). Again, as before, while restoration in this manner may be feasible, the process of restoring a work may alter the aesthetics of the work significantly and this may not be an optimal outcome.

## References

- [1] *HTML element - Wikipedia, the free encyclopedia.* URL: [http://en.wikipedia.org/w/index.php?title=HTML\\_element&oldid=628440350#Element\\_status](http://en.wikipedia.org/w/index.php?title=HTML_element&oldid=628440350#Element_status).
- [2] *The Evolution of the Web.* URL: <http://evolutionofweb.appspot.com/>.
- [3] *browsers.evolt.org.* URL: <http://browsers.evolt.org/>.

# C — Virtual Reality Components (Parent: Browser-Based Works)

This classification addresses browser-based works that have virtual reality components. The PREMIS identifier used is “VRML.”

## Short description

Virtual reality components in browser-based works were often built using Virtual Reality Markup Language (VRML), which is now obsolete and has been superseded by X3D. VRML is a file format specifying how to render “3-dimensional interactive vector graphics” [1] with various properties that correspond to real-world attributes of objects. Even though VRML files are plain text, they do require specific software to render properly within a work. The most common file extension for these files is .wrl.

## Implications for access

Browser-based works with VRML often require appropriate plugins to run properly. Technically, this means that, as long as one can install the appropriate plugin, it should be theoretically possible to view a browser-based work with VRML components on a current system.

The project team found several complications in running browser-based works with VRML. First, many of the browser plugins that render VRML are no longer supported, have ceased active development, or are no longer available for download on the web. Second, some browser-based works may also require additional, sometimes proprietary and/or also obsolete, plugins that may, in combination, come into conflict with one another on a current system [2].

Given these complications, it may often be easiest to run a browser-based work with VRML on an emulator or virtual machine that is set up to closely match the stated system requirements of the work. Many of these artworks included preferred software and plugins on the disc itself, obviating the need to track down now-obsolete or unavailable software. Additionally, if an artist developed a work with a particular piece of rendering software in mind, it may only render properly in that particular software or there may also be preferred software-specific settings to optimize viewing of the work.

## Restoration potential

VRML files are plain text, so it is technically possible to read the code. This does mean that it may be possible for one to re-create the work in another programming language, though this will require a considerable amount of effort and programming expertise.

VRML was superseded by X3D in 2004 [3]. As of this writing (2014), while the project team was able to find some tools that purported to convert VRML files into X3D [4, 5, 6], their efforts were not successful. For instance, when trying the online converter in [5], it was difficult to copy longer .wrl files into the site's web form. Second, the resulting .x3d file did not have any of the color information that the original file had. Since a file-by-file translation proved unsuccessful at this time using one tool, the level of difficulty in restoring a VRML-based work—with multiple interrelated files—is not known.<sup>a</sup>

---

<sup>a</sup>The project team was unable to try conversion using the other two software mentioned.

## References

- [1] VRML - Wikipedia, the free encyclopedia. URL: <http://en.wikipedia.org/w/index.php?title=VRML&oldid=605021649>.
- [2] Dependency hell - Wikipedia, the free encyclopedia. URL: [http://en.wikipedia.org/w/index.php?title=Dependency\\_hell&oldid=625304085](http://en.wikipedia.org/w/index.php?title=Dependency_hell&oldid=625304085).
- [3] Visualizing the Semantic Web: XML-based Internet and Information Visualization. URL: <http://books.google.com/books?id=YbWxDNZkiaUC&lpg=PR2&pg=PA127#v=onepage&q&f=false>.
- [4] VRML97 to X3D Translation. URL: [http://ovrt.nist.gov/v2\\_x3d.html](http://ovrt.nist.gov/v2_x3d.html).
- [5] instantreality 1.0 - tools - X3D encoding converter. URL: [http://doc.instantreality.org/tools/x3d\\_encoding\\_converter/](http://doc.instantreality.org/tools/x3d_encoding_converter/).
- [6] MeshLab. URL: <http://meshlab.sourceforge.net/>.

## D — Executables in Works

This classification outlines considerations when executable files are present in works. The PREMIS identifier used is “EXE.”

### Short description

Executable files are files that can be readily run by a computer [1]. These files are often only compatible with specific hardware and/or operating systems. An artwork may consist of one or more executable files, or may consist of executable file(s) that installs software to a specific operating system. Further, executable files may be present on a work because they are for auxiliary software (a browser plugin, for example) for the work. It should be noted that auxiliary software may not be required for the work to function properly.

### Implications for access

The difficulty of accessing artworks with executables can range from relatively straightforward to complex, depending on the hardware and software environment required to run the work. The work may contain PC-compatible executable files, Macintosh-compatible executables, or both, depending on the file systems present on the disc. As noted above, executables files contained within a work may be auxiliary and not required for the work to function properly, so it is important to take that into consideration when evaluating access strategies for such works.

Macintosh-compatible executable files will most likely require an emulator or virtual machine in order to access. Works designed for the Motorola 68k processors [2] and the PowerPC processors [3] could run in Macintosh OS versions up to OS 9. Early versions of OS X did include a compatibility environment, called “Classic Environment” [4, 5] that would allow a user to run programs originally designated for the pre-OS X operating systems. Support for this environment ended with OS 10.4, and the newer Intel-based Macintosh computers (manufactured after mid-2006) will not support the “Classic Environment.” Programs de-

signed for OS X for a PowerPC-based Macintosh could be run using the Rosetta [6] software; however, this software only ran on OS 10.4 through 10.6. As of OS 10.7, Rosetta is not supported, and there is no way to run PowerPC programs on a machine running 10.7 or above without using an emulator or virtual machine.

PC-compatible executable files may or may not require an emulator or virtual machine in order to access. Works that have 16-bit executable files designed to run in Windows 3.1 can run in Windows 95, Windows 98, and Windows ME natively. A compatibility layer, “Windows on Windows” allows 16-bit executables for Windows 3.1 to run on 32-bit versions of Windows NT, 2000, XP, Server 2003, Vista, Server 2008, 7, and 8 [7]. Windows operating systems that are 64-bit, including Windows XP Professional x64 and Server 2012, and the 64-bit versions of Server 2003, Vista, Server 2008, 7, and 8 do not have the capability to run 16-bit applications without the use of an emulator or virtual machine. The 64-bit Windows operating systems include a subsystem, “WoW64” (Windows 32-bit on Windows 64-bit) [8] that will allow some 32-bit applications to be run on 64-bit systems. Additionally, the Windows operating system has also supported a feature called “compatibility mode” intended to allow a user to run executable files originally built for earlier versions of Windows [9], going back as far as offering a compatibility mode setting for 32-bit executables built for Windows 95. While this feature has been included in various Windows operating systems for the last decade [10, 11, 12], there is no guarantee that Microsoft will continue to include it in future versions of the operating system. Similarly, there is no guarantee that future Windows operating systems will include the compatibility layer necessary to allow users to run legacy code and older executable files. The project team found that in some cases, even if some of the executable files ran on a current system, the work performed and rendered better when using an emulator or virtual machine.

It is worthwhile to note that PC-compatible executable files may also run under Wine [13] on a Linux environment. Here it is preferable to configure the Wine environment so that it matches the Windows version referenced in the work’s stated system requirements or was contemporary with the work’s release year. Within Wine, there is the option to “install” one or more executables to simulate a Windows environment (with a specific organization of files consistent with a particular Windows installation). The project team did find that Wine was most reliable at running executable files that needed few (or no) additional required programs to render properly.

On either platform, if a work has required programs it needs to run, for example, QuickTime or a browser plugin, access can get considerably more complicated. As the number of required programs increases, so does the likelihood that shared packages and libraries needed by required programs may be in conflict [14, 15, 16], a condition known informally as “dependency hell.” It is important to note that these complexities may very well have also existed for the work as it ran in its intended software and hardware environments.

## **Restoration potential**

The restoration potential for works with executable files is variable. Here, it can be helpful to distinguish between the executable files that are created by the artist and files that are auxiliary to the work, such as third-party plugins or other software.

When considering strategies for restoring an artist-created executable file, if source code is available for the work, it might be possible to recompile the source code on a current system. If documentation is available, this can provide additional help in this process. Re-compiling on a current system may require revisions to the source code. If none of the source code is available, this will not be possible.

For auxiliary files, such as third-party plugins or other software, there are some additional considerations. It may not be possible to install the preferred software (and version) on a current system, because the installation software is no longer available. Even if it is possible to obtain the specific installer file for the required software—if, for instance, the artist included it with the work—it may not be possible to install these executable files on a current system. The newer versions of the required software may or may not be compatible with the files in the work.

# References

- [1] *Executable definition by The Linux Information Project (LINFO)*. URL: <http://www.linfo.org/executable.html>.
- [2] *Motorola 68000 series - Wikipedia, the free encyclopedia*. URL: [http://en.wikipedia.org/w/index.php?title=Motorola\\_68000\\_series&oldid=606048541](http://en.wikipedia.org/w/index.php?title=Motorola_68000_series&oldid=606048541).
- [3] *PowerPC - Wikipedia, the free encyclopedia*. URL: <http://en.wikipedia.org/w/index.php?title=PowerPC&oldid=625091139>.
- [4] *Classic Environment - Wikipedia, the free encyclopedia*. URL: [http://en.wikipedia.org/w/index.php?title=Classic\\_Environment&oldid=627118741](http://en.wikipedia.org/w/index.php?title=Classic_Environment&oldid=627118741).
- [5] *Mac OS X: What is a Classic Application?* URL: [https://support.apple.com/kb/TA20902?locale=en\\_US](https://support.apple.com/kb/TA20902?locale=en_US).
- [6] *Rosetta (software) - Wikipedia, the free encyclopedia*. URL: [http://en.wikipedia.org/w/index.php?title=Rosetta\\_%28software%29&oldid=615449686](http://en.wikipedia.org/w/index.php?title=Rosetta_%28software%29&oldid=615449686).
- [7] *Windows on Windows - Wikipedia, the free encyclopedia*. URL: [http://en.wikipedia.org/w/index.php?title=Windows\\_on\\_Windows&oldid=625220312](http://en.wikipedia.org/w/index.php?title=Windows_on_Windows&oldid=625220312).
- [8] *WoW64 - Wikipedia, the free encyclopedia*. URL: <http://en.wikipedia.org/w/index.php?title=WoW64&oldid=627942315>.
- [9] *Make older programs compatible with this version of Windows - Windows Help*. URL: <http://windows.microsoft.com/en-us/windows-8/older-programs-compatible-version-windows>.
- [10] *How to use Windows Program Compatibility mode in Windows XP*. URL: <http://support.microsoft.com/en-us/kb/292533>.
- [11] *How To Enable Application Compatibility-Mode Technology in Windows 2000 SP2 and SP3*. URL: <http://support.microsoft.com/en-us/kb/279792>.
- [12] *What is program compatibility? - Windows Help*. URL: <http://windows.microsoft.com/en-us/windows/what-is-program-compatibility#1TC=windows-7>.
- [13] *WineHQ - Run Windows applications on Linux, BSD, Solaris and Mac OS X*. URL: <https://www.winehq.org/>.
- [14] *Dependency hell - Wikipedia, the free encyclopedia*. URL: [http://en.wikipedia.org/w/index.php?title=Dependency\\_hell&oldid=625304085](http://en.wikipedia.org/w/index.php?title=Dependency_hell&oldid=625304085).

- [15] *DLL Hell* - Wikipedia, the free encyclopedia. URL: [http://en.wikipedia.org/w/index.php?title= DLL\\_Hell&oldid=614150870](http://en.wikipedia.org/w/index.php?title= DLL_Hell&oldid=614150870).
- [16] *Extension conflict* - Wikipedia, the free encyclopedia. URL: [http://en.wikipedia.org/w/index.php?title=Extension\\_conflict&oldid=501463436](http://en.wikipedia.org/w/index.php?title=Extension_conflict&oldid=501463436).

## **E — Macromedia and Related Executables Parent: Executables in Works**

This classification will outline considerations for works that consist of executable and auxiliary files made using the Macromedia Director software. The PREMIS identifier used is “MACROMEDIA.”

### **Short description**

Macromedia Director was multimedia authoring software that could be used to develop graphical user interfaces and interactive games with embedded graphics and video [1]. With this software, it was possible to create ready-to-distribute CD-ROMs that contained all of the necessary files for users to run such works.<sup>a</sup> With Macromedia Director, it was possible for a user to create a CD-ROM with files that could be run on either a Macintosh or PC computer, or create a single CD-ROM that could be accessed and run on both platforms [2].

### **Implications for access**

Works built using Macromedia Director often consist of executable files. As such, access depends largely on the considerations outlined for Executables in Works (See Section D).

Additionally, Macromedia-based works may contain embedded audiovisual files. It may be necessary to install additional third-party applications or programs for these embedded files to render properly within a work. Works that depend on a specific version of an application or program may have been included that on the CD-ROM, but this may not always be the case. Some works’ documentation includes a list of all necessary additional third-party applications or programs, while others may not.

---

<sup>a</sup>In newer works, artists also used Macromedia Flash to create Flash and Shockwave files. The properties of those works are often similar to those created by Macromedia Director, with standalone executable files and additional auxiliary files.

It is helpful to note that the project team found that, often, a Macromedia-based work required some version of QuickTime [3] to be installed on the system. Some works' system requirements specified a particular version of QuickTime, while other works did not. Further, some works' system requirements may not have explicitly specified that QuickTime was a requirement, but some components of the work—such as sound or video—may not render properly without QuickTime installed on the system.

## Restoration potential

The restoration potential for Macromedia works is variable and depends on how the artist originally created the CD for the work. Works that contain only the executable and no other auxiliary files are most likely impossible to restore or migrate to another platform. Works that contain protected Director files, which typically have an x in the extension—such as .dxr, .ext—are likely also impossible to restore.

If the artist included any unprotected Director files on the CD—typically with an extension of either .dir or .cst—it may be possible to restore the work, though it will most likely be an extremely challenging process. It is important to note here that Macromedia Director gave artists multiple ways to protect the underlying component files that comprise the work. The difficulty in restoring a work will also depend on how motivated an artist was to keep the components of the work obscured from the user [2].

From the project team's research, restoring or migrating the work will most likely require consultation with the artist (to ensure all of the necessary files are present), a copy of Macromedia Director contemporaneous with the one used to create the original artwork (and a system—virtual or legacy hardware—to run it on), knowledge of how to use the version of Macromedia, and a working understanding of the artwork's underlying structure. The project team did find that there were some cases where an artist had already migrated a Macromedia Director-built work to another format, such as re-building the executable file for a newer platform, or generating a Flash or Shockwave version that could be run in a web browser, rather than using a system-dependent executable file. It will be worthwhile to do a preliminary investigation to determine whether a work has been migrated already by the artist when considering restoration of Macromedia-built works.

## References

- [1] *Macromedia Director* - Wikipedia, the free encyclopedia. URL: [http://en.wikipedia.org/w/index.php?title=Macromedia\\_Director&oldid=29951992](http://en.wikipedia.org/w/index.php?title=Macromedia_Director&oldid=29951992).
- [2] *Mastering Macromedia Director 6* Chuck Henderson. CUL Catalog Record Brief View. URL: <https://catalog.library.cornell.edu/cgi-bin/Pwebrecon.cgi?BBID=6172854&DB=local>.
- [3] *QuickTime* - Wikipedia, the free encyclopedia. URL: [http://en.wikipedia.org/w/index.php?title=QuickTime&oldid=629403146#QuickTime\\_3.x](http://en.wikipedia.org/w/index.php?title=QuickTime&oldid=629403146#QuickTime_3.x).

# F — HFS File System

This classification will outline considerations when a work contains an HFS file system. The PREMIS identifier used is “HFS.”

## Short description

Hierarchical File System (HFS) is a proprietary file system developed by Apple designed for use on Macintosh computers. It was originally developed in 1985 and was eventually superseded by HFS+, which was introduced in 1998 when Apple released the Mac OS 8.1 operating system. As of 2009, with the release of Mac OS X Snow Leopard (10.6), there is no longer support for writing to an HFS-formatted volume [1]. While HFS is a proprietary format, the specification is still online [2] as of this writing.

One feature of the HFS filesystem is the existence of a data fork and resource fork for files. Data forks store “unstructured data” and resource forks store “structured data... information in a specific form, containing details such as icon bitmaps, the shapes of windows, definitions of menus and their contents, and application code” [3]. All files on an HFS-formatted volume may have a resource fork, but it is not required.

## Implications for access

While HFS is technically a deprecated filesystem, it is still possible to mount an HFS-formatted disk image on certain current systems. If the work contains executable files, it will only be possible to run these executable files on their supported platform(s), however. As indicated in the classification document Executables in Works (Section D),<sup>a</sup> this often means access to the work must be through an emulator or virtual machine.

---

<sup>a</sup>In particular, the section describing Macintosh-compatible executables.

If the work does not contain executable files, access *may* be possible on a current system. The information in a file's resource fork may or may not be critical to access the file: therefore, there may be loss of detail or data when accessing a file that originated in an HFS filesystem on a non-HFS filesystem. Viewing the work in an appropriate emulator or virtual machine may be the best way of determining whether there is loss of significant detail when viewing the work on a current system.

## Restoration potential

The restoration potential for HFS-formatted volumes is largely dependent on whether there are platform-specific files (i.e., executable files) as part of the work.

One of the most common issues with transferring files from an HFS volume to another system is ensuring that both forks are preserved in the file transfer. The `hfsutils` tools suite includes `hcopy`, which will properly transfer both forks for further inspection and evaluation.

HFS-formatted volumes can be converted into HFS+ volumes, but if it is not possible to run the actual files on the target platform (for example, a PowerPC executable will not run on a contemporary Intel-based Apple hardware), transfer to a different file system will not allow one to access the work on a current system.

Some system-independent files may have no (or non-essential) data in their resource forks. These can be transferred to an HFS+ or non-Apple file system for possible restoration and/or access. If the files can run and/or are viewable in a different system, the project team strongly suggests viewing the work in an emulator, virtual machine, or legacy hardware to identify what attributes of the work might have been affected in the transfer.

It is worth noting that artworks that only have HFS file systems were designed for specific Apple computers. Conversion to a newer system may negatively alter the aesthetic experience of the work, including the aesthetic experience of viewing the contents of the disc on a Macintosh computer.

## References

- [1] *Hierarchical File System - Wikipedia, the free encyclopedia*. URL: [http://en.wikipedia.org/w/index.php?title=Hierarchical\\_File\\_System&oldid=594424332](http://en.wikipedia.org/w/index.php?title=Hierarchical_File_System&oldid=594424332).
- [2] *Data Organization on Volumes (IM: F)*. URL: <http://web.archive.org/web/20130815052517/http://developer.apple.com/legacy/library/documentation/mac/Files/Files-99.html>.
- [3] *Resource fork - Wikipedia, the free encyclopedia*. URL: [http://en.wikipedia.org/w/index.php?title=Resource\\_fork&oldid=615966339](http://en.wikipedia.org/w/index.php?title=Resource_fork&oldid=615966339).

**A2: Emulation Documentation Document**

# Emulation Documentation—PAFDAO Project

Dianne Dietrich

# Contents

<b>1 Overview</b>	<b>4</b>
<b>2 Emulator Selection</b>	<b>11</b>
2.1 Examples of determining compatible emulators . . . . .	14
<b>3 General Emulator Configuration and Setup</b>	<b>18</b>
<b>4 Emulating Apple/Macintosh systems with Basilisk II and SheepShaver</b>	<b>19</b>
4.1 Settings . . . . .	20
4.2 ROMs for Apple/Macintosh Emulation . . . . .	21
4.3 Disk Image File with installed Operating System . . . . .	22
4.4 Running the emulators and loading an artwork . . . . .	23
<b>5 Emulating PC-compatible systems with QEMU</b>	<b>46</b>
5.1 Settings . . . . .	47
5.2 Disk Image File with installed Operating System . . . . .	48
5.3 Running the emulator and loading an artwork . . . . .	49
5.4 Process-level emulation—WineHQ and QEMU user mode . . . . .	56
<b>Appendix A Compiling notes for Basilisk II and Sheepshaver executables</b>	<b>58</b>

Appendix B Creating a new disk image file for Basilisk II and SheepShaver and installing an operating system	62
Appendix C Installation of QEMU	94
Appendix D Creating a disk image file for the hard disk for QEMU from scratch	95

# List of Tables

1.1	Recommended emulators for access. . . . .	5
1.2	Emulators tested by project team. . . . .	6
2.1	Recommended starting points for emulators based on system requirements and creation year of work. . . . .	12
4.1	Apple ROMs used by project team for testing and access. . . . .	21
4.2	Disk image files and corresponding installed Macintosh operating systems for Basilisk II and Sheepshaver. . . . .	22
5.1	Disk image files and corresponding installed Windows operating systems for QEMU. . . . .	48

# 1 — Overview

The PAFDAO project team explored emulation as a strategy to provide “best feasible access” to works whose system requirements specify now-obsolete hardware and software. Various emulators were tested to determine their suitability for providing access to the materials in the Goldsen collection on contemporary hardware. Emulation requirements were drawn from an analysis of the stated system requirements of individual items in the collection; the project team found that the selected emulators and virtual machines supported the majority of the system requirements indicated by individual artworks in the testbed collection.

This document will describe the emulators chosen by the project team to provide the “best feasible access” to the majority of artworks in the collection.

The specific emulators the project recommends for access are listed in Table 1.1. For each emulator listed in the table, this document will provide a description of any auxiliary files needed to access an artwork using the respective emulator with along with step-by-step instructions on their use. Additional notes and caveats drawn from the project team’s experiences will also be included in this documentation. Additional details, such as installation/setup or compiling notes will be included in Appendix A.

<b>Emulator</b>	<b>Guest System(s) Emulated</b>
Basilisk II [1]	Macintosh systems with a 68K processor (generally until 1994 <sup>a</sup> ) (includes IIci, Quadra); Macintosh System 7.5 through OS 8.1
SheepShaver [2]	Macintosh systems with a PowerPC [3] (1994-2006) processor (includes Power Macintosh); Macintosh System 7.5 through OS 9.0.4
QEMU [4, 5] <sup>b</sup>	PC systems with x86 processors <sup>c</sup>

Table 1.1: Recommended emulators for access.

It is important to note here additional emulator software that the project team investigated or tested throughout the course of the project. The list of software below includes some projects that appeared, in 2013-2014, to be abandoned or no longer in use. Some software is actively developed, and may be considered as part of a future access strategy, given appropriate resources. Project team notes are included in Table 1.2. Emulators are listed in alphabetical order.

In Computer Science, there is a distinction between “virtualization” and “emulation” software; the project team investigated software that fell under both of these categories and did not limit by one or the other. The project team uses the term “emulation” and “emulator” to connote any strategy of accessing an artwork on a contemporary system using a program that allows one to run a guest operating system, including guest operating systems that may be originally compatible with another hardware platform than the host machine.

---

<sup>a</sup>A review of consumer Apple computers (on Wikipedia) suggests that Macintoshes produced up until 1993/1994 would have 68K processors.

<sup>b</sup>While QEMU has the capability to emulate multiple systems, the project team used it primarily for emulating PC systems.

<sup>c</sup>The project team also tested WineHQ for running individual processes without emulating an entire guest operating system; details are provided in section 5.4.

Table 1.2: Emulators tested by project team.

<b>Emulator</b>	<b>Guest System(s) Emulated</b>	<b>Project Team Notes</b>
bochs [6]	x86 PC emulator (and can be compiled to include x86-64).	The project team did test out this software, but found it difficult to run; therefore, the project team did not test it with any of the Goldsen artworks. As it is currently actively developed [7], it may be explored as a future option for access.
DOSBox [8]	Intel x86 PC.	This is mostly geared towards running DOS-based programs, as it emulates a x86 PC with a “DOS-like command prompt.” [9] Its main goal is to get DOS-based games running on current systems. Since the Goldsen artworks require some version of Microsoft Windows, it may not be the ideal method for access.
JMESS (Javascript MESS) [10]	Same as MESS; below.	This is a browser-based port of MESS (see entry for MESS). It appears to be a standalone system built to interact with a suite of pre-selected games on specific consoles. The project team could not test this with the Goldsen artworks.

(Table 1.2 – Continues on next page.)

<b>Emulator</b>	<b>Guest System(s) Emulated</b>	<b>Project Team Notes</b>
MESS (Multi Emulator Super System) [11]	Many guest systems; see Supported Systems [12] in documentation. Part of MAME project, and as of 2014, under active development. [13, 14]	The project team tested this software out in 2013 and found it difficult to get a guest system running (and thus, could not test it with a Goldsen artwork). It may be worth future investigation, as it comes highly recommended by an advisor on the grant and it may include CRT emulation. At the time of testing, MESS may not support the Macintosh PowerPC architecture; additional possible limitations were found on a Macintosh emulation site. [15]
Mini vMac [16]	Early Macintosh systems (from 1984-1996) running 68K processors.	While this seems to still be under active development as of 2014 [17], the project team noted that its focus appeared to be much earlier Macintosh computers than those indicated in the system requirements of the Goldsen collection. This software was not tested by the project team.
PearPC [18]	PowerPC architecture.	Development appears to have stalled since 2011. [19] While this does emulate a PowerPC Macintosh, it only supports OS 10.1 through 10.4. The majority of the Goldsen artworks require older Macintosh systems than this. The project team did not test PearPC for this reason, but notes it could be useful for future work in this area.

(Table 1.2 – Continues on next page.)

<b>Emulator</b>	<b>Guest System(s) Emulated</b>	<b>Project Team Notes</b>
VirtualBox [20]	Virtualizing x86 and AMD64/Intel64.	The project team did use and test this software for accessing Windows-based works. Specifically, a guest system running Windows2000 was used as a demo (with VirtualBox) because it was rather straightforward to run on the department's current Macintosh laptop. It can be considered a viable alternative to using QEMU to access Windows-based artworks. It is important to note that VirtualBox isn't technically "emulation" as other software is. [21]

# References

- [1] *The Official Basilisk II Home Page*. URL: <http://basilisk.cebix.net/>.
- [2] *The Official SheepShaver Home Page*. URL: <http://sheepshaver.cebix.net/>.
- [3] *PowerPC - Wikipedia, the free encyclopedia*. URL: <http://en.wikipedia.org/w/index.php?title=PowerPC&oldid=637595599>.
- [4] *QEMU*. URL: [http://wiki.qemu.org/Main\\_Page](http://wiki.qemu.org/Main_Page).
- [5] *QEMU Emulator User Documentation*. URL: <http://wiki.qemu.org/download/qemu-doc.html>.
- [6] *bochs: The Open Source IA-32 Emulation Project (Home Page)*. URL: <http://bochs.sourceforge.net/>.
- [7] *bochs 2.6.7 Release notes*. bochs 2.6.7 Release notes. URL: [http://svn.code.sf.net/p/bochs/code/tags/REL\\_2\\_6\\_7\\_FINAL/bochs/CHANGES](http://svn.code.sf.net/p/bochs/code/tags/REL_2_6_7_FINAL/bochs/CHANGES).
- [8] *DOSBox, an x86 emulator with DOS*. URL: <http://www.dosbox.com/>.
- [9] *DOSBoxWiki*. URL: [http://www.dosbox.com/wiki/Main\\_Page](http://www.dosbox.com/wiki/Main_Page).
- [10] *The Javascript MESS*. URL: <http://jsmess.textfiles.com>.
- [11] *Welcome to the MESS Wiki! [MESS]*. URL: <http://www.mess.org/>.
- [12] *Supported systems*. URL: <http://www.progettoemma.net/mess/sysset.php>.
- [13] *Previous MAME Releases*. URL: <http://www.mamedev.org/oldrel.html> (visited on 03/23/2013).
- [14] *Previous MAME Versions - MAMEDEV Wiki*. URL: [http://wiki.mamedev.org/index.php/Previous\\_MAME\\_Versions](http://wiki.mamedev.org/index.php/Previous_MAME_Versions).
- [15] *MESS [E-Maculation wiki]*. URL: <http://www.emaculation.com/doku.php/mess>.
- [16] *Mini vMac - early Macintosh emulator*. URL: <http://www.gryphel.com/c/minivmac/>.
- [17] *Gryphel Project News*. URL: <http://www.gryphel.com/c/news/index.html>.
- [18] *PearPC - PowerPC Architecture Emulator*. URL: <http://pearpc.sourceforge.net/>.
- [19] *PearPC - PowerPC Architecture Emulator - Downloads*. URL: <http://pearpc.sourceforge.net/downloads>.
- [20] *Oracle VM VirtualBox*. URL: <https://www.virtualbox.org/>.

- [21] *Virtualization – Oracle VM VirtualBox*. URL: <https://www.virtualbox.org/wiki/Virtualization>.

## **2 — Emulator Selection**

The following should be taken into consideration when selecting an appropriate emulator for an artwork: the work's original stated system requirements, including any included readme files and onscreen help and the year of publication of the work. It may also be necessary to consult the system requirements of the target operating system for additional guidance in setting up the guest system. In cases where system requirements information is unclear or insufficient, the project team offers some guidance drawn from its experience analyzing the files contained on the discs that make up the Goldsen artwork. This is located in Section 2.1. Table 2.1 [1, 2, 3, 4] provides an overview of the process of choosing an emulator.

<b>Compatible Platform<sup>a</sup></b>	<b>Processor Identified</b>	<b>Emulator Needed</b>	<b>Year of work</b>	<b>Artwork</b>	<b>Contemporaneous Operating Systems<sup>b</sup></b>
Apple/Macintosh	Motorola 68K <sup>c</sup>	Basilisk II	1996 and before 1997–1999	1996 and before	System 7, OS 7.6 <sup>d</sup>
		SheepShaver		1996 and before 1997–1999 1999–2001	OS 8 (8.0-8.6) System 7, OS 7.6 OS 8 (8.0-8.6) OS 9 (9.0-9.2)
PC/Windows	n/a <sup>f</sup>	QEMU	1994 and before 1995–1999	1994 and before	Windows 3.1 <sup>g</sup>
				1995–1999	Windows 95, Windows NT
			2000–2001	4.0, Windows 98, Windows 98SE	Windows 98, Windows ME,
				Windows 2000	Windows XP, Windows 2000

Table 2.1: Recommended starting points for emulators based on system requirements and creation year of work.

It is important to note that there are cases where stated system requirements are either missing, incomplete, or ambiguous, and it is difficult to infer from publication year to determine a suitable emulator. In these cases, the project team found the following guidelines helpful:

1. If no information is given, it is possible to look at the filesystems present on the disk to determine whether the work might be compatible with a Macintosh system, a Windows system, or both. The presence of an ISO9660 filesystem indicates that the work might be compatible with PC-based systems, and the presence of an HFS (or HFS+) filesystem suggests Macintosh compatibility).

<sup>a</sup>Taken from System Requirements or inferred by the filesystem present on the disc (i.e., HFS partition suggests Macintosh compatibility).  
<sup>b</sup>This chart lists the operating systems the project team researched to be contemporaneous with the years listed and may or may not be the optimal operating system for rendering the work. For Macintosh works, the “Classic Environment” in OS 10.0-10.3 may also be an alternative environment to run a work, though this is not guaranteed to work in all cases.

<sup>c</sup>Can be listed as Mac 68030, Mac 68040, 680x0, Quadra, Mac IIci in system requirements.

<sup>d</sup>The project team did not obtain retail media for every version of the Macintosh operating systems in the ranges specified. Please see Table 4.2 for the full list of operating systems on disk image files generated by the project team.

<sup>e</sup>Can be listed as PowerPC, G3, G4, Power Macintosh, PowerMac in system requirements.

<sup>f</sup>It generally doesn’t apply here for emulator selection, but PC processors can be listed as Pentium, Pentium 2 or Pentium II, Intel 80486, 486 in system requirements

<sup>g</sup>The project team did not obtain retail media for every version of the Windows operating systems in the ranges specified. Please see Table 5.1 for the full list of operating systems on disk image files generated by the project team.

tem indicates the work might be compatible with Apple/Macintosh systems. The project team found several examples where the presence or absence of a filesystem was misleading. Ultimately, running the work in an appropriate system (which might be an emulated/virtual system) may be required to make a final determination.

2. Additional documentation may be available within the files contained on the disk image that will help determine additional settings or minimum system requirements.
3. In some cases, it may be necessary to pick a starting point (an emulator and an operating system) and test various configurations until something works.
  - (a) On Macintosh systems, executables made for the 68K processor will likely work on SheepShaver, but not vice versa (a PowerPC executable will not work on Basilisk II). Running an executable on an incompatible system may or may not result in a helpful error message (there were cases where it simply failed silently) so if “nothing happens” when attempting to run a work on Basilisk II, it is advisable to try switching to SheepShaver first, rather than trying another operating system within Basilisk II.
  - (b) On Windows systems, given the history of supporting legacy code and executables, it is less likely to encounter an error message indicating incompatibility. An error message might indicate that what version of Windows is compatible with the work.

The artwork’s original stated system requirements for hardware and software and any additional relevant documentation should provide the user the best starting point for determining which emulator is appropriate to access the work, and how it should be configured. The following examples demonstrate how to use a work’s system requirements to determine a starting point for an appropriate emulator (and settings). It is important to note here that the work’s stated system requirements might suggest multiple compatible system configurations, even on the same hardware platform.

## 2.1 Examples of determining compatible emulators

The following section will reproduce actual system requirements in their entirety from works in the Goldsen collection. After each set of system requirements there will be guidance from the project team on how to provide access through an appropriate emulator.

*System requirements: Macintosh with 68040 33MHz processor or Power Macintosh; 8MB RAM; System 7.0.1 or greater; 5MB free disk space (12MB recommended); 640 x 480 or larger monitor with thousands of colors; minimum double speed CD-ROM drive. (Beyond, 1996)*

The system requirements for this work indicate only Macintosh emulators will be compatible, and specifically mention that the work runs on both 68K and PowerPC processors, so SheepShaver and Basilisk II will work. The earliest Macintosh system (prepared by the project team) that will work is System 7.5.5. The system requirements also include guest system configuration (i.e., screen resolution, amount of free space needed).

*System requirements for accompanying computer disc: Windows 3.1 - 8MB RAM; SoundBlaster compatible sound card; MAC Version - Mac 68040 (or higher), System 7.1 (or higher). (The body and the object, 1996)*

The system requirements for this work indicate Basilisk II or QEMU will work. (It is also likely here that SheepShaver will work, because PowerPC Macintosh systems can run 68K executable files). While Windows 3.1x is stated as a compatible emulator, given Microsoft's support in subsequent operating systems for legacy executables, it is likely that Windows 95/98/NT/2000 will work as well.

*System requirements: This CD-ROM is compatible with Windows (mpc2 with 486 or Pentium chip) and Macintosh (68040 or power PC), it needs 256 colors and a minimum 5 mb of free memory (RAM), although more is better. (1996, Book of Shadows)*

The system requirements for this work suggest that Basilisk II, SheepShaver, and QEMU can work, since it is compatible with the two Macintosh processors, 68040 and PowerPC, and a PC-compatible processor, a 486 or Pentium. For the Macintosh, System 7.5.3 was released in 1996,[5] and it would most likely be the earliest Macintosh operating system that will work.<sup>h</sup> To access the Windows version of the work, it is likely that Windows 95 is the earliest Windows operating system that will be compatible with the work, since it was released in 1995.<sup>i</sup>

---

<sup>h</sup>System 7.5.3 is also the earliest Macintosh operating system tested by the project team.

<sup>i</sup>It is possible that Windows 3.1x might also work here, but only if the executable file for the artwork was 16-bit.

## **Examples where a publication year can help determine a suitable emulator**

In cases where there is ambiguity, or in cases where the stated system requirements are not written with sufficient detail, it can be helpful to consider the publication year of the artwork, and select an environment contemporaneous with that.

*System requirements: System requirements for Macintosh: Macintosh; 8 megabytes of RAM; system 7 or later; 16 bit color and 13 inch monitor; CD-ROM drive. (1995, Cyberflesh Girlmonster)*

The system requirements for this work indicate that only Macintosh emulators will be compatible. Even though no specific processors are mentioned (i.e., 68K or PowerPC), it is possible to use the publication date to determine which specific Macintosh emulators might work. Since the publication date was in 1995 it is possible that either Basilisk II (68K) or SheepShaver (PowerPC) could work, though SheepShaver is more likely, as in 1995, the 68K processor had largely been replaced (except for a few Macintosh models) by the PowerPC architecture.

*System Requirements: Macintosh with at least 8 megabytes RAM, a color video monitor and a CD-ROM drive. (Culture vs. the Martians, 1998)*

Since the specific Macintosh processor isn't indicated, it will be necessary to consider the year of publication to determine a suitable emulator. In 1998, Apple was manufacturing computers using the PowerPC processor, so SheepShaver is likely to be the optimal Macintosh emulator for this work.

*System requirements: Macintosh. (Let's tell lies, 1999)*

While this work's system requirements does not provide much detail, it is possible to determine, from the year of publication, that the work functions in a PowerPC Macintosh. SheepShaver would be the preferred emulator. It is also important in cases where little detail is given at the outset, to read any documentation in Readme files (or similar) within the work itself.

## **Examples of determining guest system settings from system requirements**

System requirements can also indicate various guest system settings. It is important to note that these may need to be cross-checked against the system requirements for the operating system on the guest system to ensure proper functioning of the artwork within emulation.

*System requirements: Pentium processor with 32 MB RAM, 8 CD drive, 16 bit color display, sound card, Windows NT 4.0 or better or Windows 2000. (Divine Comedy Digital, 1998)*

These requirements indicate the guest system should be set up with at least 32MB of RAM; it should also have 16-bit color and have sound enabled. It is important to note that Windows NT has a minimum RAM requirement of 12 MB,[6] and Windows 2000 has a minimum ram requirement of 32 MB,[7] which should also be taken into consideration when setting up the guest system.

*System requirements for IBM PC: Windows 95 or NT, 16 MB RAM, monitor set at 640 x 480 or higher, monitor set to 32 bit millions of colors, Quicktime 3.0 or higher, 4x CD-ROM drive or faster, sound card. System requirements for Mac: Power PC Macintosh, 16 MB RAM, monitor set at 640 x 480 or higher, monitor set to 32 bit millions of colors, Quicktime 3.0 or higher, 4x CD-ROM drive or faster. (As american as apple pie, 1999)*

These requirements indicate the guest system should be set up with at least 16 MB of RAM; the preferred screen resolution is 640x480<sup>j</sup> and it should have 32-bit color. Additionally, Quicktime 3.0 should be installed on the guest system.

---

<sup>j</sup>Note that on the emulators, setting the screen resolution may resize the window on the host machine.

## References

- [1] *List of Microsoft Windows versions - Wikipedia, the free encyclopedia.* URL: [http://en.wikipedia.org/w/index.php?title=List\\_of\\_Microsoft\\_Windows\\_versions&oldid=641680709](http://en.wikipedia.org/w/index.php?title=List_of_Microsoft_Windows_versions&oldid=641680709).
- [2] *Mac OS 8 - Wikipedia, the free encyclopedia.* URL: [http://en.wikipedia.org/w/index.php?title=Mac\\_OS\\_8&oldid=637807563](http://en.wikipedia.org/w/index.php?title=Mac_OS_8&oldid=637807563).
- [3] *Mac OS 9 - Wikipedia, the free encyclopedia.* URL: [http://en.wikipedia.org/w/index.php?title=Mac\\_OS\\_9&oldid=642160131](http://en.wikipedia.org/w/index.php?title=Mac_OS_9&oldid=642160131) (visited on 03/19/2015).
- [4] *System 7 - Wikipedia, the free encyclopedia.* URL: [http://en.wikipedia.org/w/index.php?title=System\\_7&oldid=638152696](http://en.wikipedia.org/w/index.php?title=System_7&oldid=638152696).
- [5] *System 7, OS 7.6 - Wikipedia, the free encyclopedia.* URL: [http://en.wikipedia.org/w/index.php?title=System\\_7&oldid=638152696#Version\\_history](http://en.wikipedia.org/w/index.php?title=System_7&oldid=638152696#Version_history).
- [6] *INFO: Windows NT 4.0 Setup Troubleshooting Guide.* URL: <https://support.microsoft.com/en-us/kb/126690>.
- [7] *System requirements for Microsoft Windows 2000 operating systems.* URL: <https://support.microsoft.com/en-us/kb/304297/en-us>.

## **3 — General Emulator Configuration and Setup**

In general, emulators—including those selected by the project team for the Goldsen artworks—include various settings and configurations. The project team noted that it is helpful to consider them in the following three categories:

1. Settings that pertain to the program’s functioning within the host system (and how events within the host system are interpreted in the guest system);
2. Options to configure the guest system within the emulator; and
3. Guest environment settings configured within the guest operating system such as display resolution (within the guest system), color depth, and internet/network configuration.

In addition to the program itself, each emulator requires additional files to run properly. All emulators need a disk image file that has a functional operating system installed to it and a disk image of the artwork. (Appendix B and D include detailed instructions on how to create, format, and prepare a blank volume (disk image file) to install an operating system for each emulator.) Basilisk II and SheepShaver also require a ROM file that represents Macintosh firmware in order to run. More detail on creating and using these additional files, including ROM files for the Macintosh emulators, is provided in subsequent sections of this document and the appendices.

The project team found that for each of these emulators, while there are options to specify a disk image should be loaded as a CD-ROM in the guest system, it is possible to load a disc image of an artwork as if it were an additional hard drive, rather than read-only media. A user should pay special attention to this, as loading a disk image as “read-write” media in the guest system means that the disk image can be altered and, consequently, will no longer be a faithful copy of the original CD-ROM. Changes to a disk image may have unintended consequences for interacting with the work. A user can further safeguard against these effects by setting artwork disk images as read-only within the host system or verifying the disk image file’s checksum periodically, making a new clean copy if a change has been detected.

## 4 — Emulating Apple/Macintosh systems with Basilisk II and SheepShaver

The project team used the following two Apple Macintosh emulators in their tests to provide access to works in the Goldsen whose system requirements indicated an Apple computer. Basilisk II [1] emulates a 680x0 Motorola-based Apple Macintosh, and can run up to Mac OS 8.1. SheepShaver [2] emulates a PowerPC Apple Macintosh and can run from System 7.5.2 through 9.0.4. Both emulators are open-source projects and can run on a variety of platforms: the project team did the majority of its testing of these programs on a Linux machine running Ubuntu.

In order to run either Basilisk II and SheepShaver, the following components are needed:

1. Basilisk II or SheepShaver executables.<sup>a</sup>
2. Mac OS ROM File
3. Disk image File with installed Operating System, or disk image of retail installation media.
4. Disk image of the CD-ROM with the original artwork.

The following is an overview of general settings for the emulators, a brief description of the ROM files needed, how to load a disk image and access an artwork, and other important information to consider, drawn from the project team's notes, observations, and testing.

---

<sup>a</sup>The project team compiled these from source; the details will be outlined in Appendix A.

## 4.1 Settings

There are various settings and configuration options for Basilisk II and SheepShaver. Some settings pertain to the operation of the emulator software, while settings determine settings for the guest system (i.e., the hardware/software environment being emulated).

In Basilisk II and SheepShaver, the following program settings can be adjusted:

- Shared directory between guest and host machine
- Window Refresh Rate, and QuickDraw Acceleration (to optimize rendering)
- Host devices for Output and Mixer; Modem and Printer; Ethernet
- Width and Height of the guest display
- Behavior of the mouse wheel in guest system
- Ignore illegal memory accesses (may reduce the number of times the program crashes unexpectedly)
- Enable JIT compiler (can improve performance of program)

Basilisk II and SheepShaver offer the following guest system settings:

- Primary (and secondary) hard disk volumes to mount
- CD-ROM images to mount
- Device to boot from
- RAM (Random access memory) size
- ROM file (firmware)

ROM	Source (Original URL)	Compatible Emulator
PERFORMA.ROM	<a href="http://www.redundantrobot.com/macemulator/1mbMacrom.zip">http://www.redundantrobot.com/macemulator/1mbMacrom.zip</a>	Basilisk II
QUAD650.ROM	<a href="http://www.redundantrobot.com/macemulator/Quad650.zip">http://www.redundantrobot.com/macemulator/Quad650.zip</a>	Basilisk II
Quadra.rom	<a href="http://www.redundantrobot.com/macemulator/Quadra.zip">http://www.redundantrobot.com/macemulator/Quadra.zip</a>	Basilisk II
mac_oldworld_rom4mb.rom	<a href="http://www.redundantrobot.com/macemulator/mac_oldworld_rom4mb.rom.zip">http://www.redundantrobot.com/macemulator/mac_oldworld_rom4mb.rom.zip</a>	SheepShaver <sup>e</sup>
newworld86.rom	<a href="http://www.redundantrobot.com/macemulator/newworld86.rom.zip">http://www.redundantrobot.com/macemulator/newworld86.rom.zip</a>	SheepShaver

Table 4.1: Apple ROMs used by project team for testing and access.

## 4.2 ROMs for Apple/Macintosh Emulation

In order to emulate Apple hardware using Basilisk II and SheepShaver, it is necessary to obtain a ROM file. A ROM file “contains the instructions that tell the computer what to do when the power button is pressed.”<sup>b</sup> Some older tutorials suggested that it was necessary to obtain a ROM from an already-owned Macintosh, but the project team found some ROM files freely available on the web and used these for testing.<sup>c</sup> ROM files are not interchangeable; some combinations of ROMs, emulators, and operating systems are non-functional. Additionally, some ROMs work better than others. The ROM files listed in Table 4.1 included in the CULAR deposit, and have been tested extensively by the project team. Full provenance information and checksums will be in the file ingested into the Cornell University Library Archival Repository (CULAR).

Additionally, the Internet Archive hosts a collection of ROM files from older Macintosh models.<sup>d</sup> While these may be helpful for future access, the project team did not test these out extensively.

<sup>b</sup>Source: Ben Fino-Radin, personal correspondence.

<sup>c</sup>The CULAR deposit will include all of the ROMs that the project team tested; other ROMs, like those already available at the Internet Archive, have not been included.

<sup>d</sup>[https://archive.org/details/mac\\_rom\\_archive\\_-\\_as\\_of\\_8-19-2011](https://archive.org/details/mac_rom_archive_-_as_of_8-19-2011)

<sup>e</sup>SheepShaver can only run pre-OS 8.1 with this ROM.

Name of disk image file	Size (in MB)	Operating System In- stalled	Compatible emula- tor
basilisk_ii_os761_500mb	500	Mac OS 7.6.1	Basilisk II
basilisk_ii_os80_500mb	500	Mac OS 8.0	Basilisk II
basilisk_ii_os81_500mb	500	Mac OS 8.1	Basilisk II
sheepshaver_os761_500mb	500	Mac OS 7.6.1	SheepShaver
sheepshaver_os80_500mb	500	Mac OS 8.0	SheepShaver
sheepshaver_os81_500mb	500	Mac OS 8.1	SheepShaver
sheepshaver_os85_1000mb	1000	Mac OS 8.5	SheepShaver
sheepshaver_os90_1000mb	1000	Mac OS 9.0	SheepShaver

Table 4.2: Disk image files and corresponding installed Macintosh operating systems for Basilisk II and Sheepshaver.

### 4.3 Disk Image File with installed Operating System

In order to use Basilisk II and SheepShaver to emulate a Macintosh system, it is necessary to have a disk image file with an installed operating system. The project team created multiple disk image files and installed various Macintosh operating systems for use with both emulators.

Table 4.2 shows the various files created for access, identifying the name of the disk image file, the installed version of the operating system, and the compatible emulator.

## 4.4 Running the emulators and loading an artwork

The project team compiled both emulators to include a graphical user interface to easily select configuration options and load disk images of artworks. Note that it might be necessary to manually edit a configuration file for optimal results.<sup>f</sup>

Once all of the required components have been assembled, and a suitable ROM has been chosen, the process for accessing an artwork is very similar in Basilisk II and SheepShaver. The following sections will include screenshots of the configuration options for both emulators and an overview of the process. Note that in both cases, while it is possible to add disk image of the CD-ROM of the artwork in the graphical interface, the project team recommends that it be added by editing the configuration file to ensure that the disk image will be loaded in the guest system as a CD-ROM, that is, as a read-only volume.

---

<sup>f</sup>Note that it is also possible to compile SheepShaver and Basilisk II so that the graphical interface is bypassed: users must manually edit a preferences file and run the emulator from the command line or from a desktop shortcut. This configuration, along with “helper” scripts might be one approach for a production system that allows users access to the artworks while minimizing the risk of unwanted changes to the emulation layer.

## Basilisk II

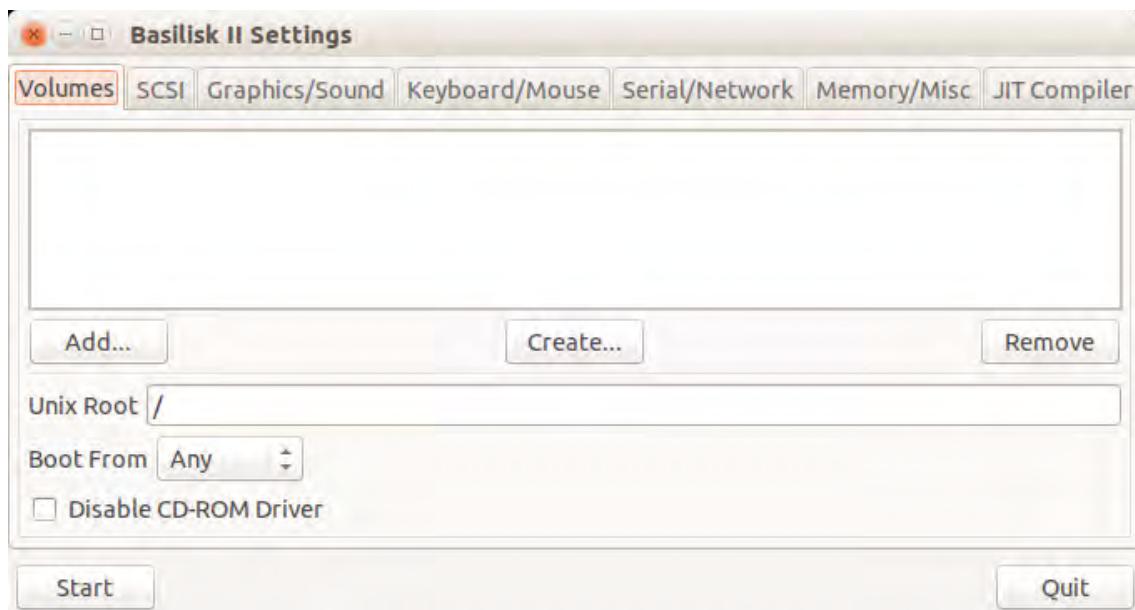


Figure 4.1: Volumes tab in Basilisk II Settings.

By default, the Basilisk II graphical interface displays the **Volumes** tab. Once the required operating system has been identified, it can be added to the list of volumes by clicking the **Add...** button and selecting the appropriate disk image file from the file selection dialog. (Figure 4.1)

Once the appropriate operating system is added (Figure 4.2), it will be included in the list (Figure 4.3).

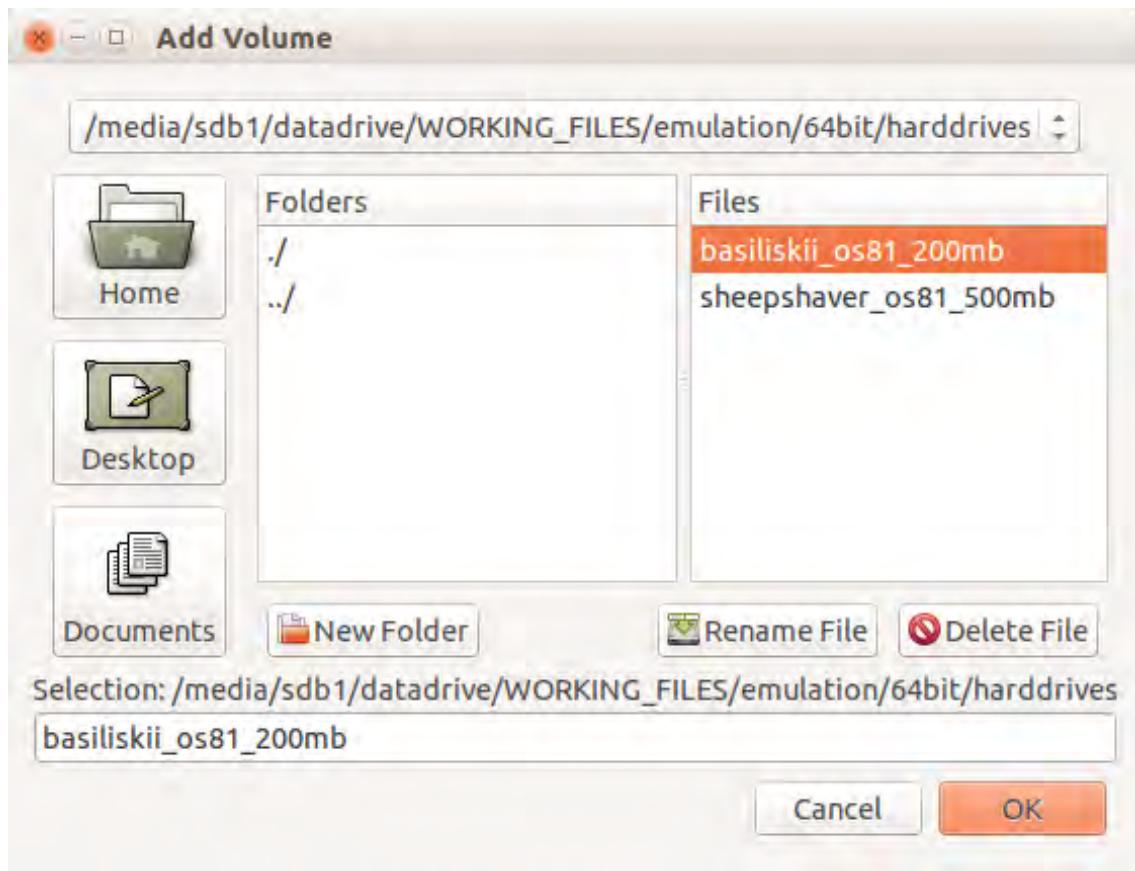


Figure 4.2: Add Volume dialog in Basilisk II.

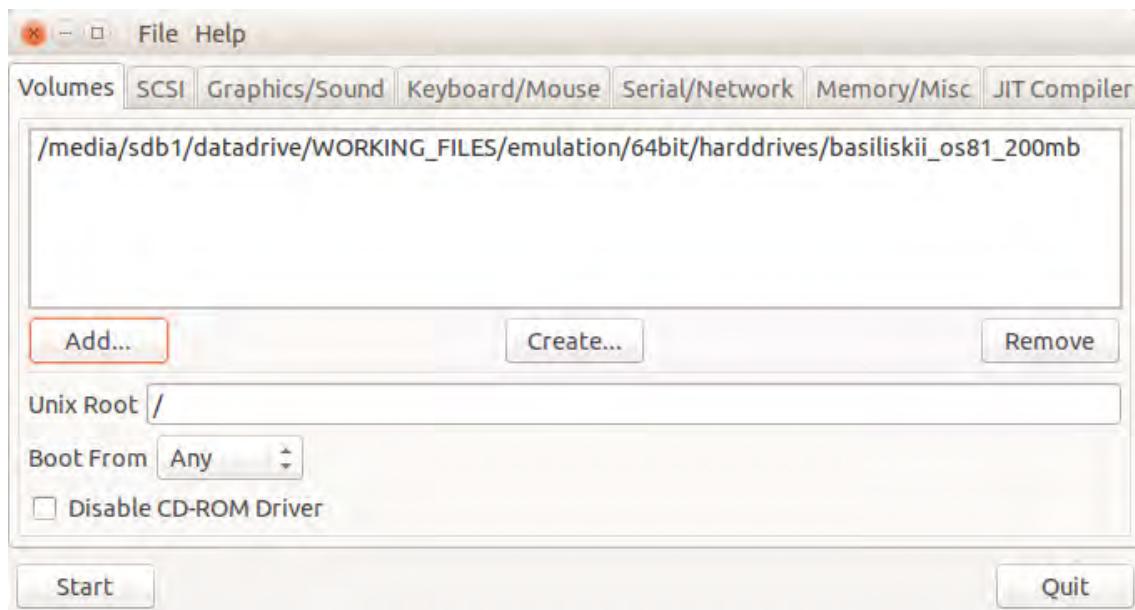


Figure 4.3: Volumes tab in Basilisk II Settings with new disk image file selected.

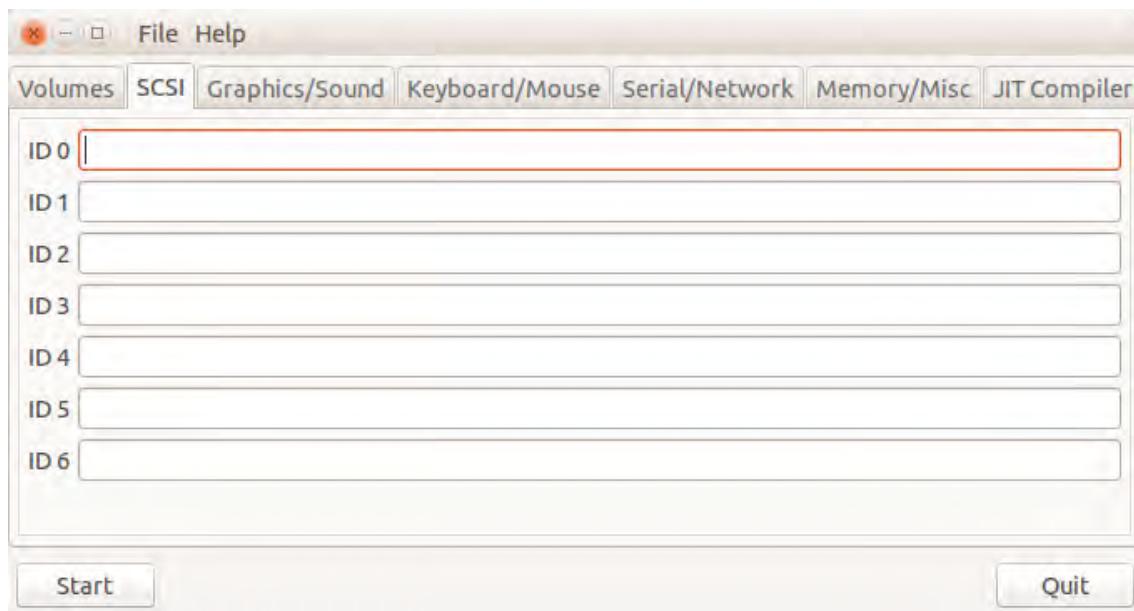


Figure 4.4: SCSI tab in Basilisk II Settings.

The project team did not make use of SCSI hard drive images when testing out the Goldsen artworks (Figure 4.4).

The project team found that keeping the Window Refresh Rate (in the **Graphics/Sound** tab) set to Dynamic produced the best rendering output (Figure 4.5).

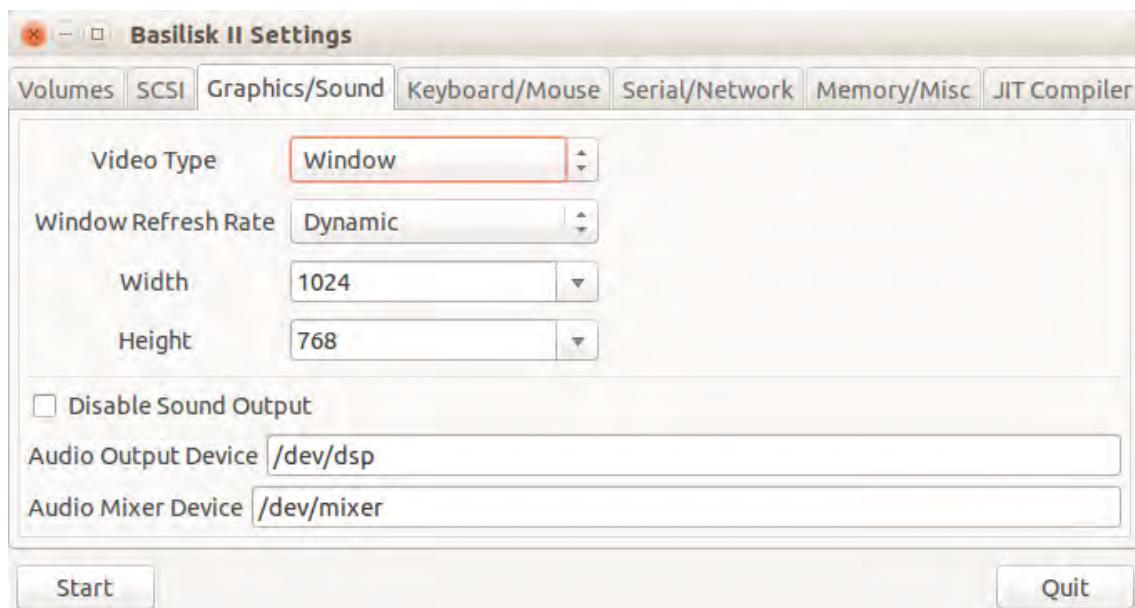


Figure 4.5: Graphics/Sound tab in Basilisk II Settings.

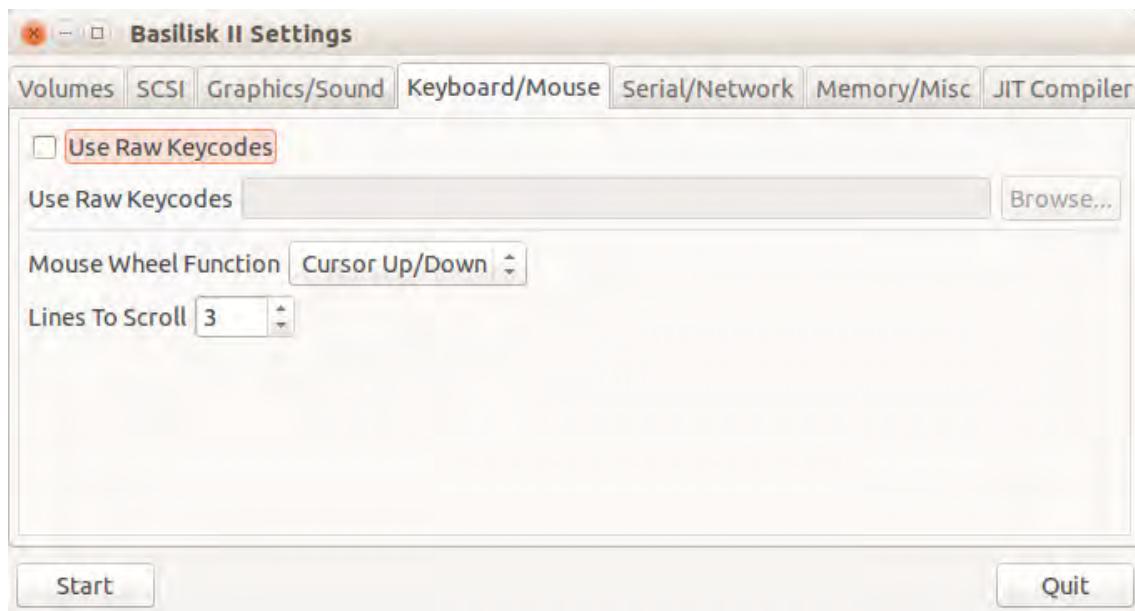


Figure 4.6: Keyboard/Mouse tab in Basilisk II Settings.

The project team did not adjust the default settings on the **Keyboard/Mouse** tab (Figure 4.6).

Enter in **slirp** into the **Ethernet Interface** field if networking/internet access is needed for an artwork (Figure 4.7).

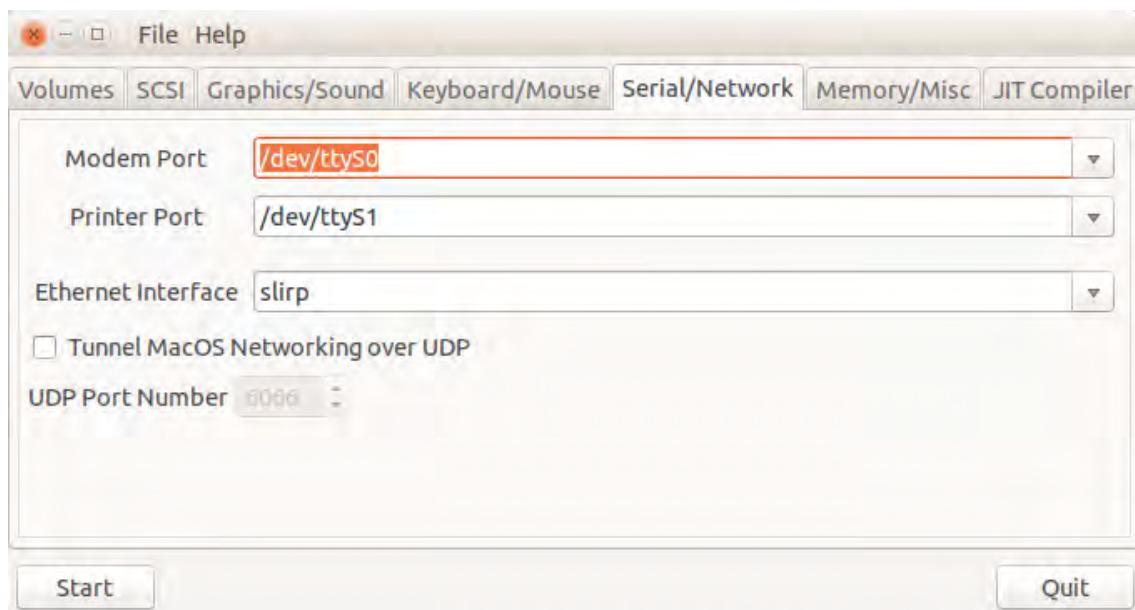


Figure 4.7: Serial/Network tab in Basilisk II Settings.

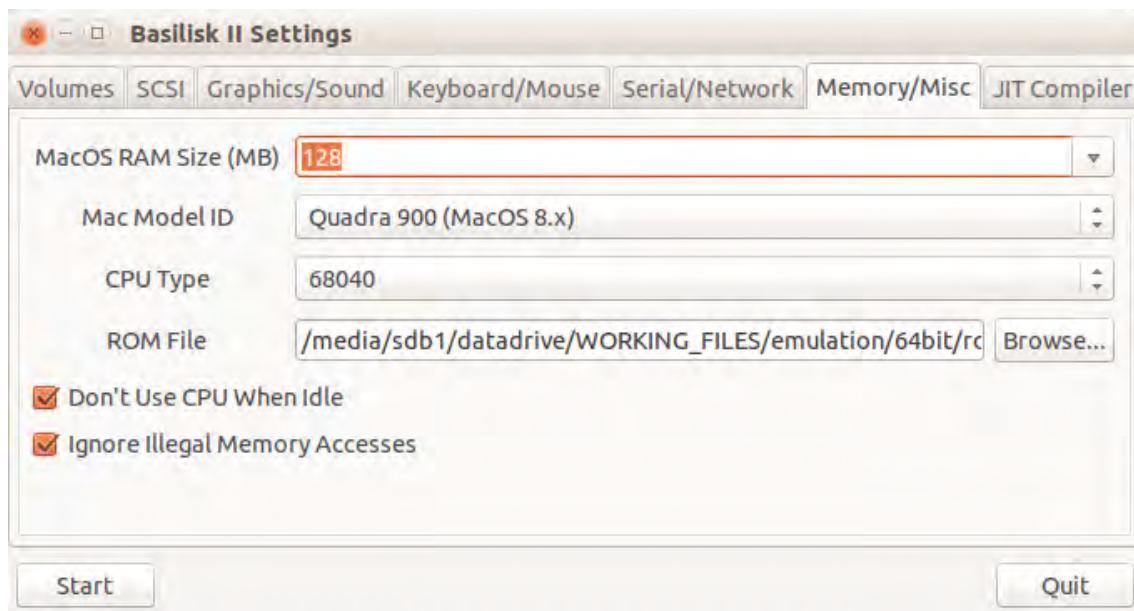


Figure 4.8: Serial/Network tab in Basilisk II Settings.

If available, use the work's stated system requirements to determine the **MacOS RAM Size**. The project team found that either 64MB or 128MB was a good starting point when no other information was available.

The project team tended to use **Quadra 900 (Mac OS 8.x) Mac Model ID** when working with OS 8.0 and above<sup>g</sup> and **Mac IIci (MacOS 7.x)** when using OS 7.6 or System 7.5.x.

The project team suggests leaving the **CPU Type** set at the default value (68040) because other settings did not noticeably change the rendering of a work; however, there may be works where this setting is important to adjust.

Consult Table 4.1, which lists ROMs for Apple/Macintosh Emulation to determine which ROM File is appropriate. Note that BasiliskII can only use a “512K or 1MB MacII ROM.” Even if the origin (i.e., MacII) isn’t known, it can be helpful to check the size of a ROM file to ensure it is exactly 512K or 1MB in size to confirm its potential to work in Basilisk II.

The project team also found that selecting the checkboxes next to **Don't Use CPU When Idle** and **Ignore Illegal Memory Accesses** improved performance of the emulator (Figure 4.8).

<sup>g</sup>Note that the last operating system Basilisk II is compatible with is Mac OS 8.1.

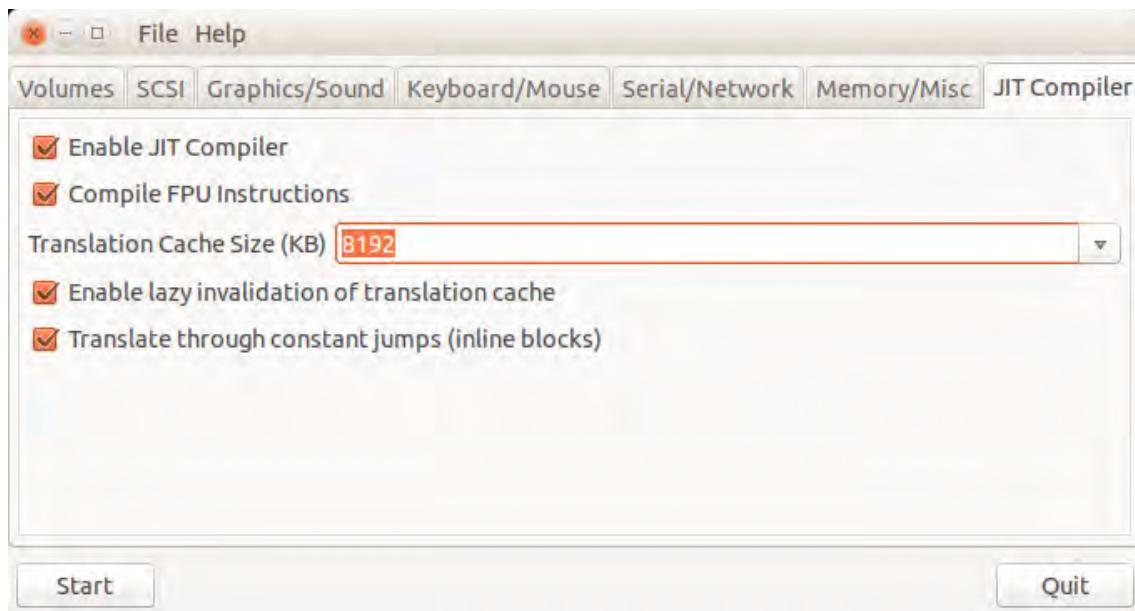


Figure 4.9: JIT Compiler tab in Basilisk II Settings.

The project team found that checking **Enable JIT Compiler** improved performance of the emulator. The rest of the above settings are default and were found to be adequate for testing the artworks (Figure 4.9).

## Starting Basilisk II

The project team recommends starting the emulator once before the artwork is loaded. This saves the selected settings in the preferences file and gives the user the opportunity to preview the operating system environment to ensure that system requirements within the guest system are appropriate for the selected artwork. For instance, if the system requirements specify “requires QuickTime extensions,” the user can check to see that Quicktime is in the **System:Extensions** folder, and QuickTime is installed. It can also be helpful here to verify that sound is working properly. (Figures 4.10, 4.11, 4.12)

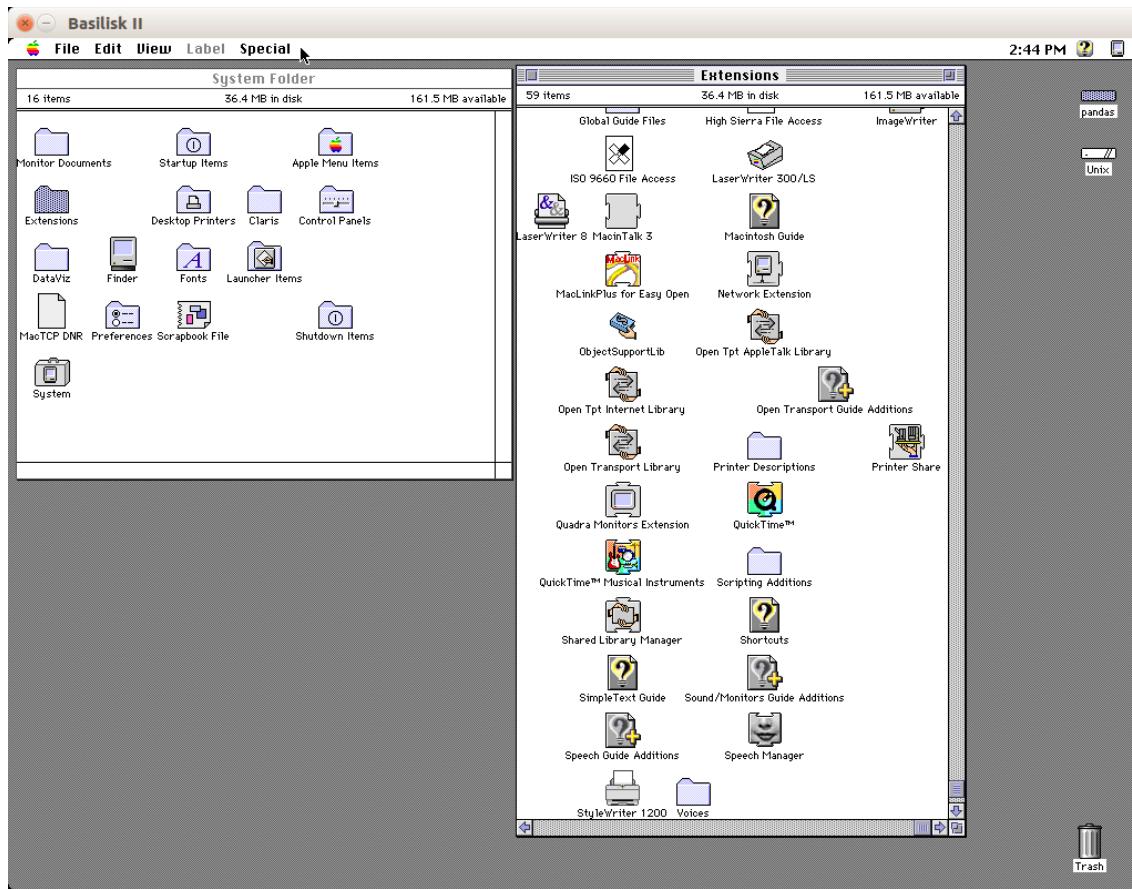


Figure 4.10: System Folder:Extensions is the general location for the “classic” Mac OS systems.

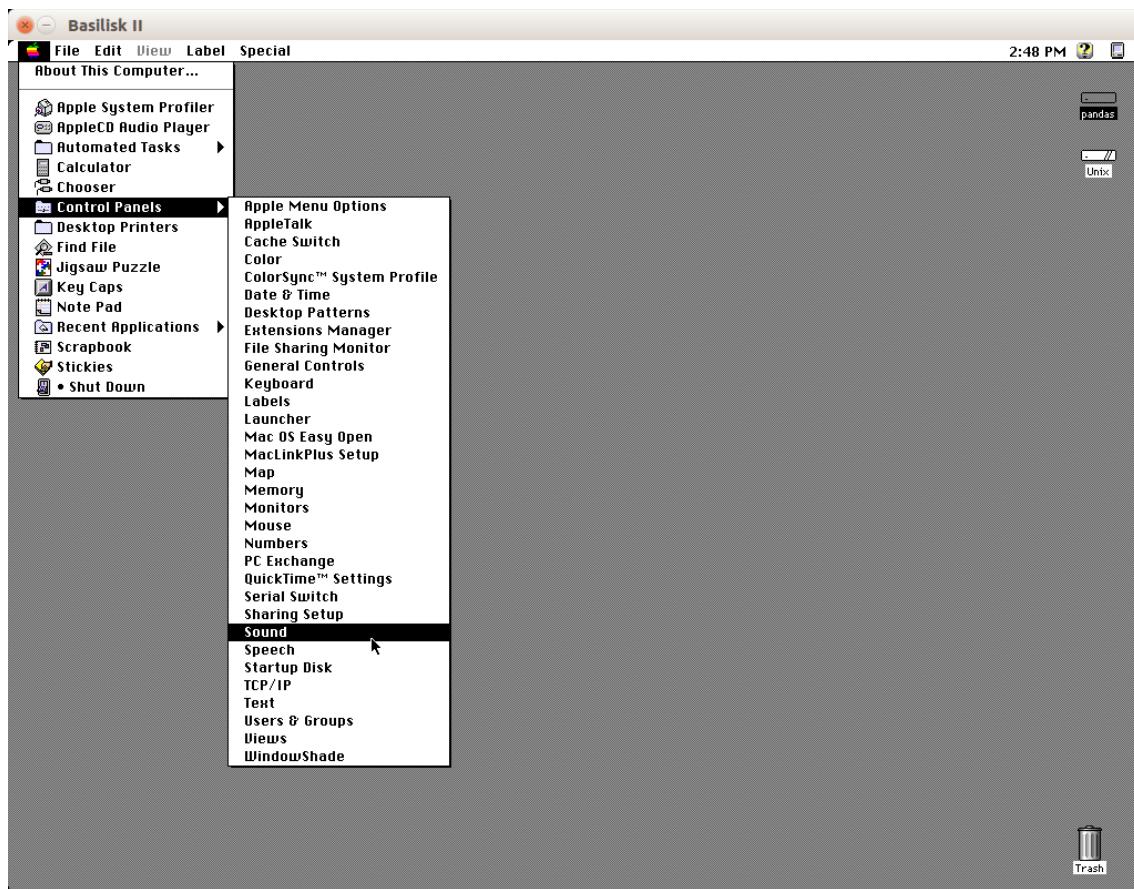


Figure 4.11: Selecting the Control Panel for Sound in OS 7.6.1.

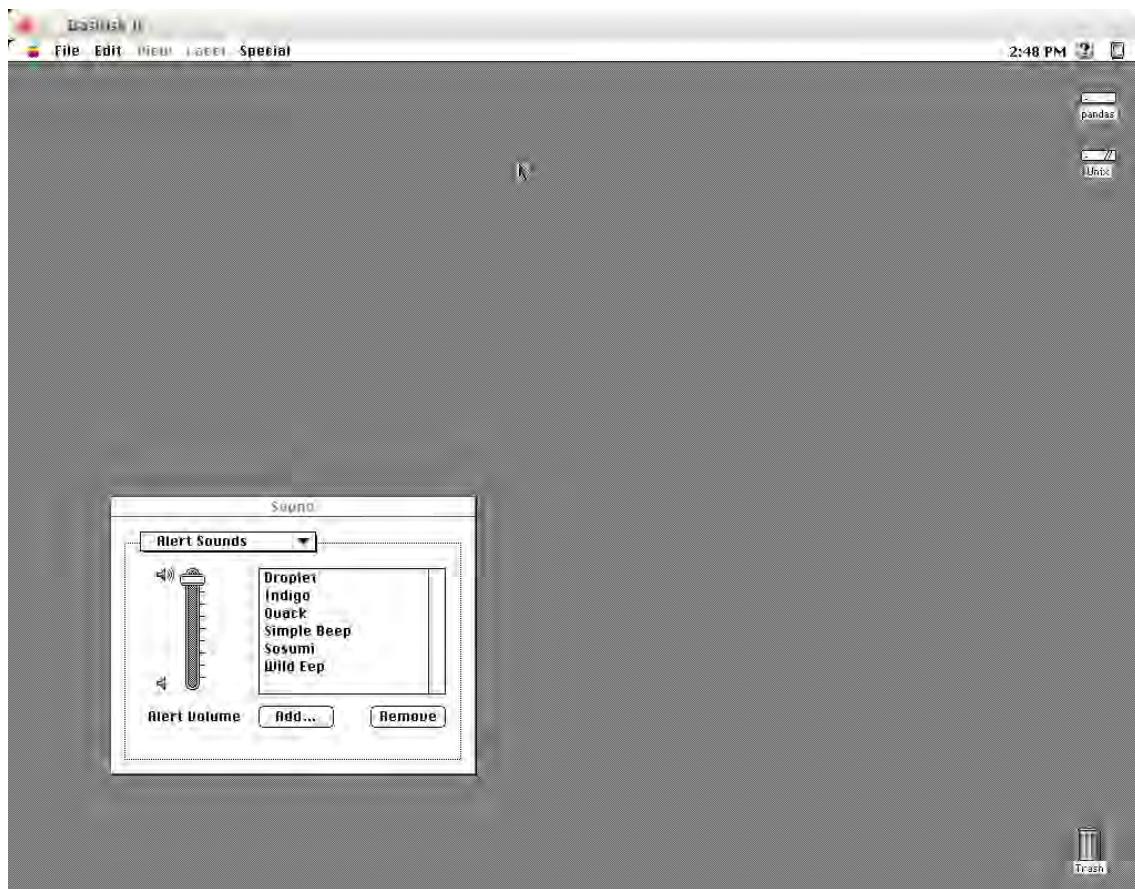
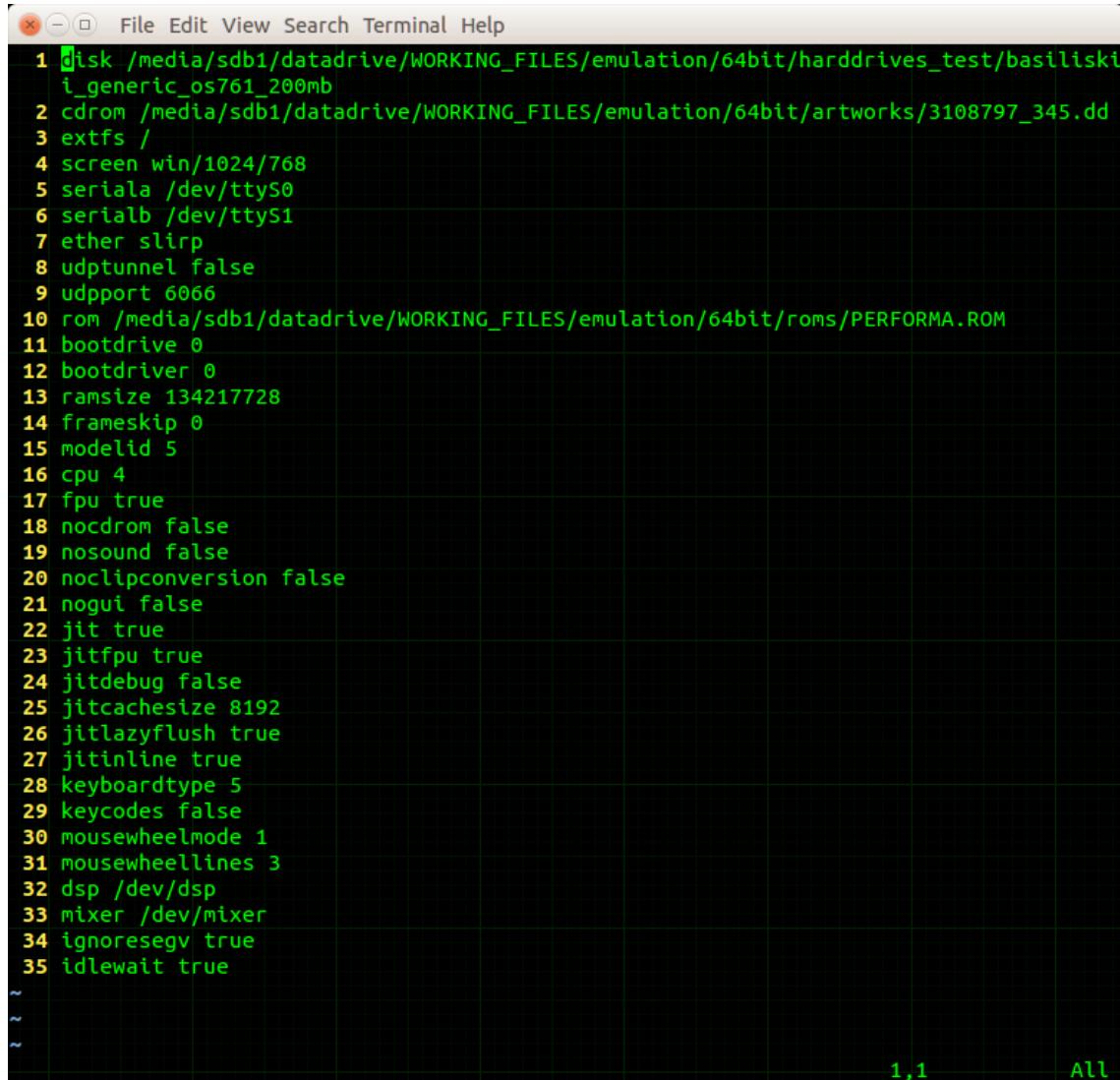


Figure 4.12: Using the Sound dialog to confirm sound is working properly in OS 7.6.1.

Once system requirements have been confirmed (as best as possible), the user can shut down the guest system and prepare the preferences file to load the artwork. Instead of opening up the graphical user interface, edit the `7.basilisk_ii_prefs` file and add the following line:

```
cdrom /full_path/artwork.isoh
```

Where `full_path` is the exact path to the disk image of the artwork. Figure 4.13 is an example of a properly edited preferences file.



The screenshot shows a terminal window with a dark background and light-colored text. The window title is "File Edit View Search Terminal Help". The content of the terminal is a configuration file with numbered lines from 1 to 35. Lines 1 through 34 contain configuration parameters and their values, such as "disk /media/sdb1/datadrive/WORKING\_FILES/emulation/64bit/harddrives\_test/basiliski\_generic\_os761\_200mb" and "rom /media/sdb1/datadrive/WORKING\_FILES/emulation/64bit/roms/PERFORMA.ROM". Line 35 contains the command "idlewait true". Below the terminal window, there are three small icons: a red square, a green triangle, and a blue circle. In the bottom right corner of the terminal window, there is a status bar with the text "1,1 All".

```
1 disk /media/sdb1/datadrive/WORKING_FILES/emulation/64bit/harddrives_test/basiliski_generic_os761_200mb
2 cdrom /media/sdb1/datadrive/WORKING_FILES/emulation/64bit/artworks/3108797_345.dd
3 extfs /
4 screen win/1024/768
5 seriala /dev/ttys0
6 serialb /dev/ttys1
7 ether slirp
8 udptunnel false
9 udpport 6066
10 rom /media/sdb1/datadrive/WORKING_FILES/emulation/64bit/roms/PERFORMA.ROM
11 bootdrive 0
12 bootdriver 0
13 ramsize 134217728
14 frameskip 0
15 modelid 5
16 cpu 4
17 fpu true
18 nocdrom false
19 nosound false
20 noclipconversion false
21 nogui false
22 jit true
23 jitfpus true
24 jitdebug false
25 jitcachesize 8192
26 jitlazyflush true
27 jitinline true
28 keyboardtype 5
29 keycodes false
30 mousewheelmode 1
31 mousewheellines 3
32 dsp /dev/dsp
33 mixer /dev/mixer
34 ignoresegv true
35 idlewait true
~
~
~
```

Figure 4.13: Properly configured configuration file to load an artwork as read-only media in Basilisk II.

---

<sup>h</sup>Extension might be .iso, .dd, or .bin; any of these will work as long as it is a disk image.

After the file has been edited and saved, the user can open up the Basilisk II graphical interface and click **Start** to start the emulator with the artwork loaded as a CD-ROM, that is, read-only. It is important to note here that the artwork will *not* be listed in the Volumes tab of the graphical interface after it has been added to the preferences file as cdrom.

## SheepShaver

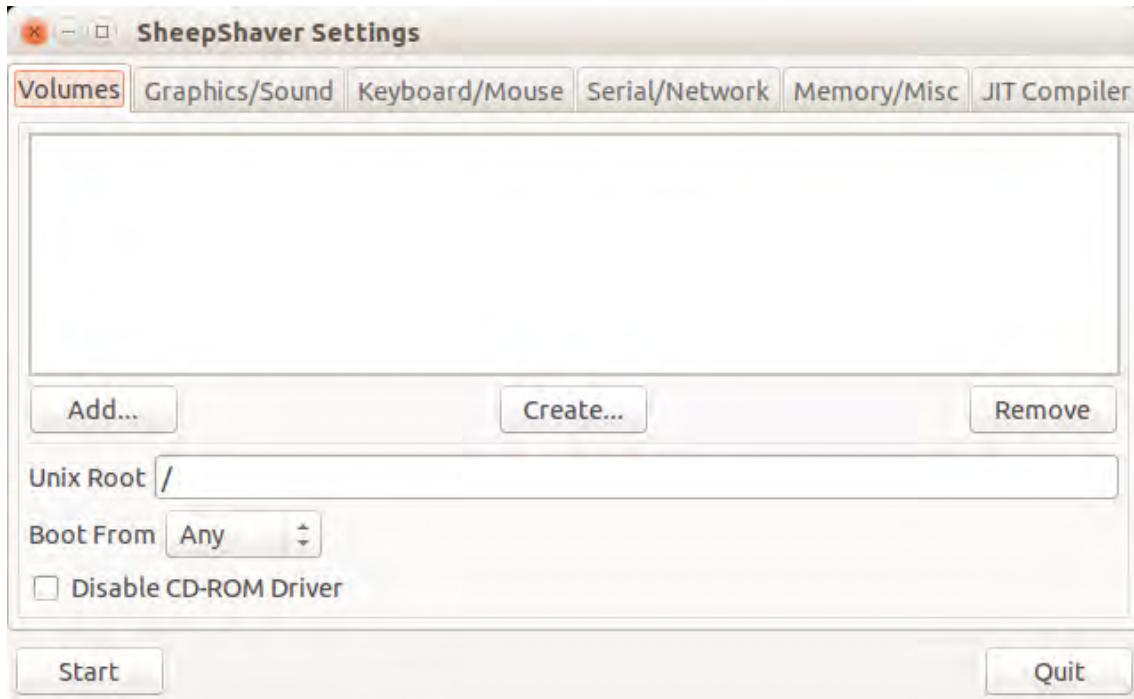


Figure 4.14: Volumes tab in SheepShaver Settings.

By default, the SheepShaver GUI displays the **Volumes** tab. Once the required operating system has been identified, it can be added to the list of volumes by clicking the **Add...** button and selecting the appropriate disk image file from the file selection dialog (Figures 4.14, 4.15, 4.16).

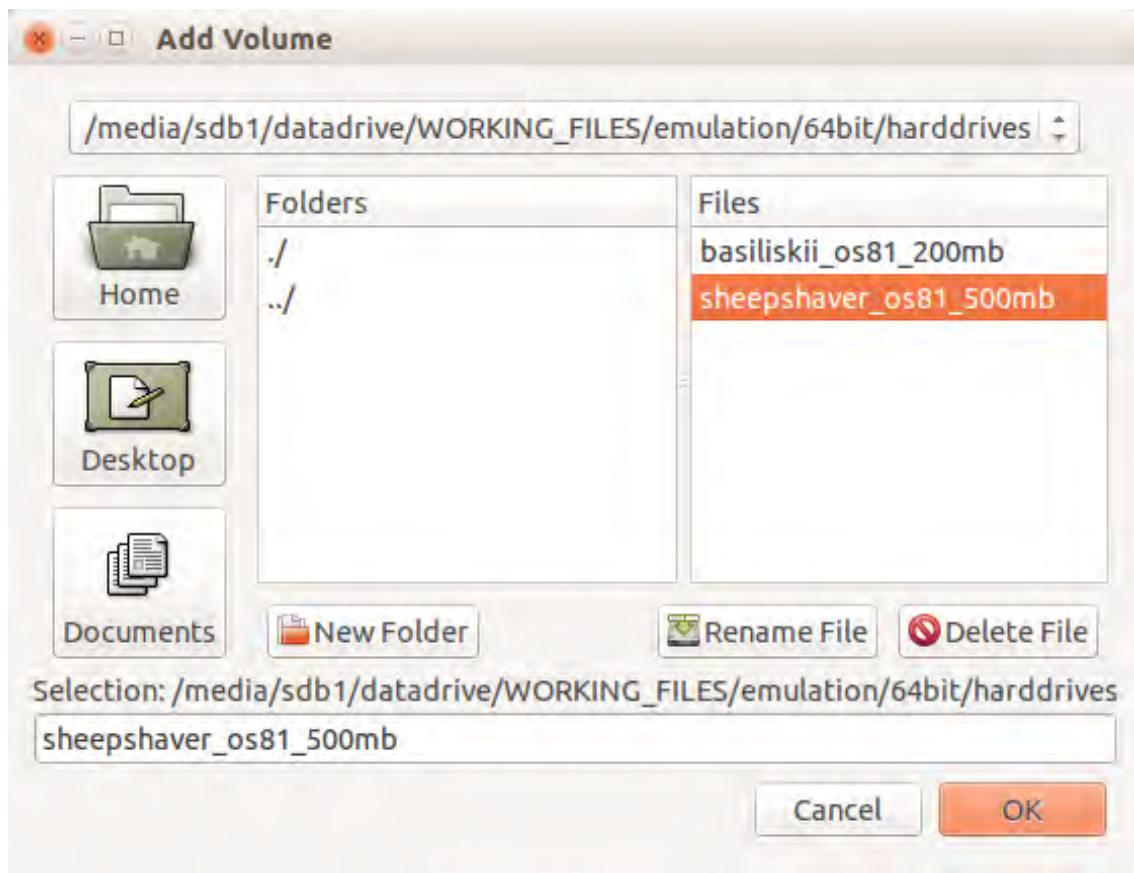


Figure 4.15: Add Volume dialog in SheepShaver.

The project team found that setting the **Window Refresh Rate** (on the **Graphics/Sound** tab) at the default of 7.5 Hz worked well for testing (Figure 4.17).

The project team found that the default settings on the **Keyboard/Mouse** tab worked well for testing (Figure 4.18).

To enable networking/internet access, enter **slirp** for the **Ethernet Interface** field in the **Serial/Network** tab (Figure 4.19).

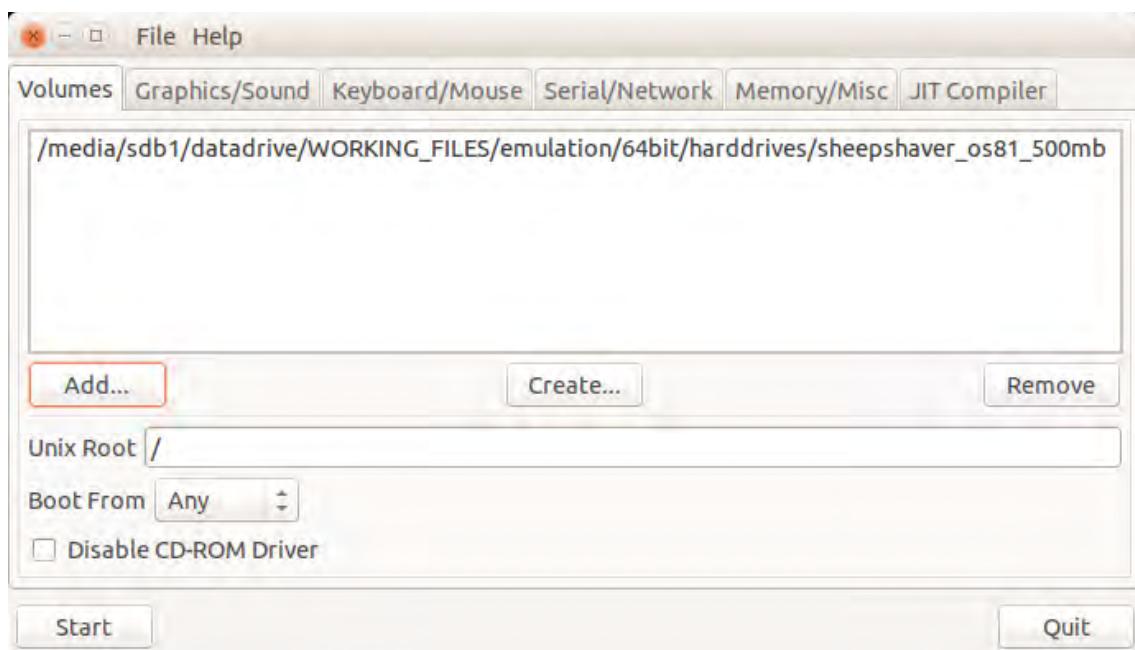


Figure 4.16: Volumes tab in SheepShaver Settings with new disk image file selected.

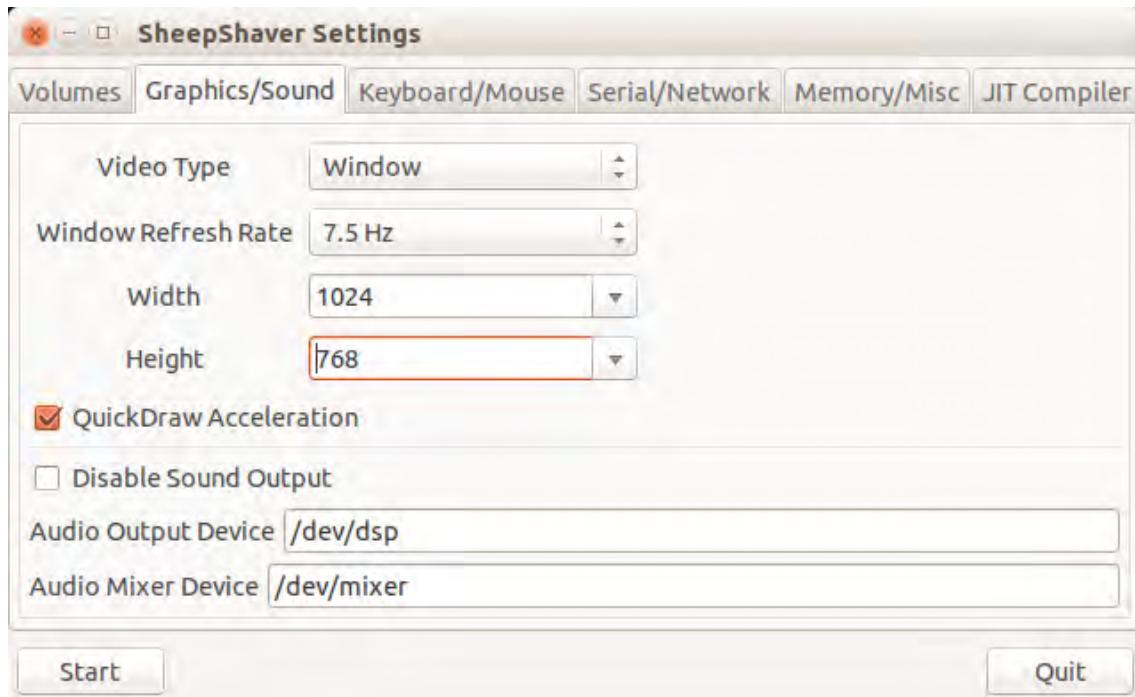


Figure 4.17: Graphics/Sound tab in SheepShaver Settings.

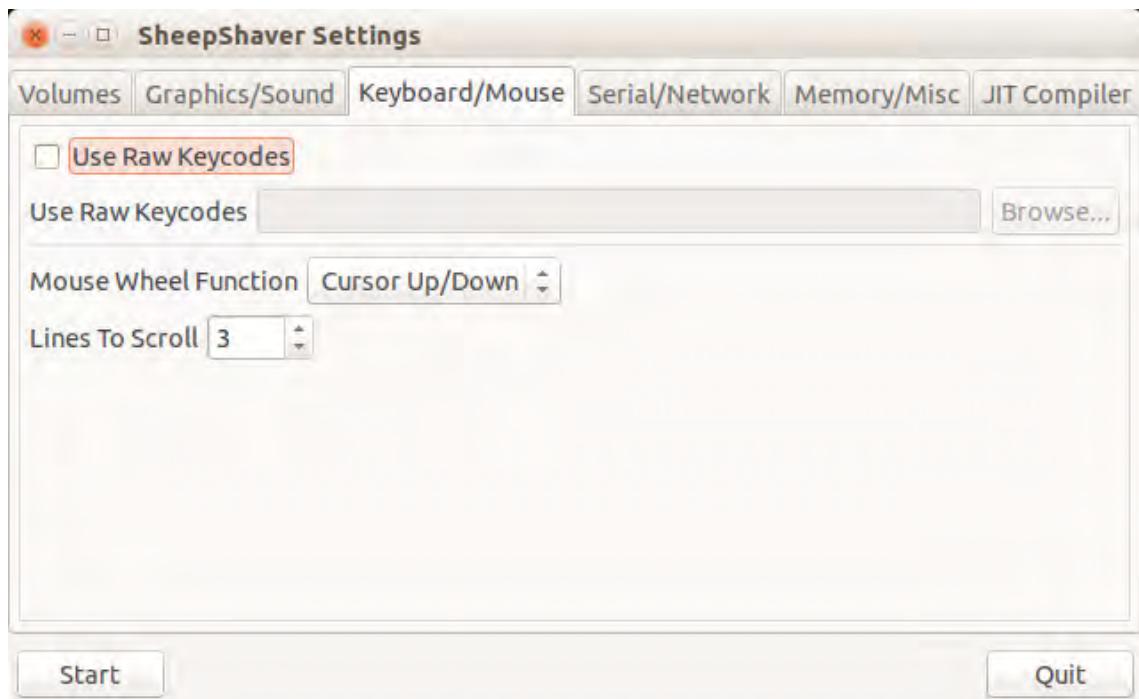


Figure 4.18: Keyboard/Mouse tab in SheepShaver Settings.

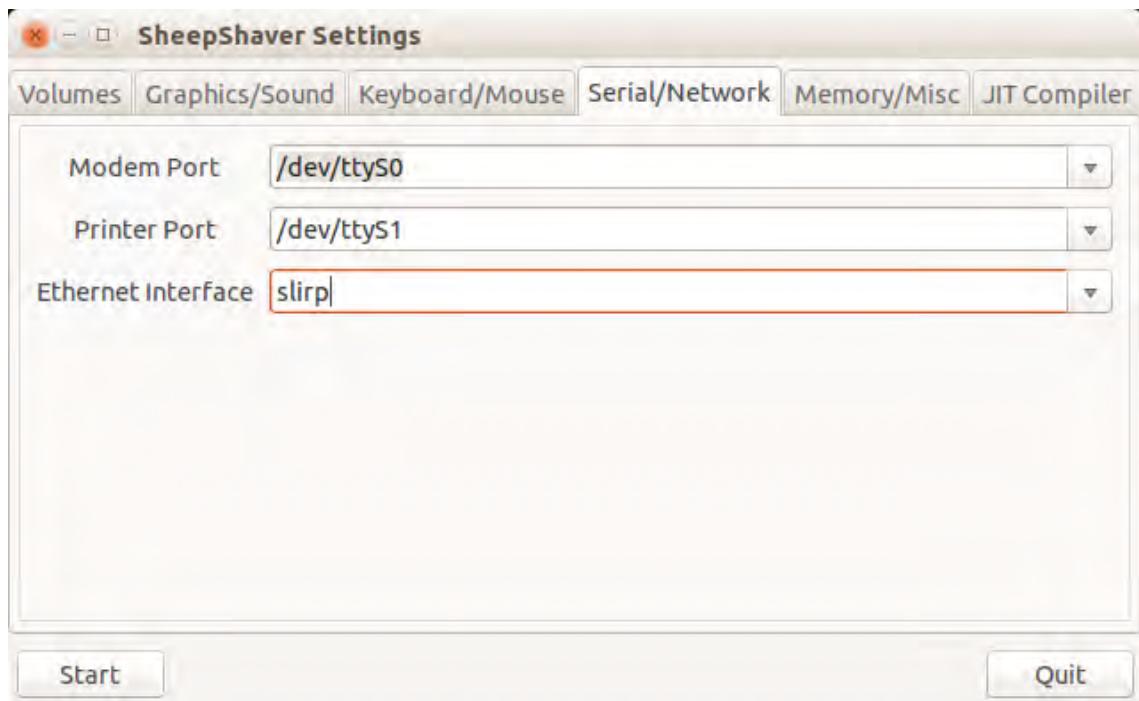


Figure 4.19: Serial/Network tab in SheepShaver Settings.

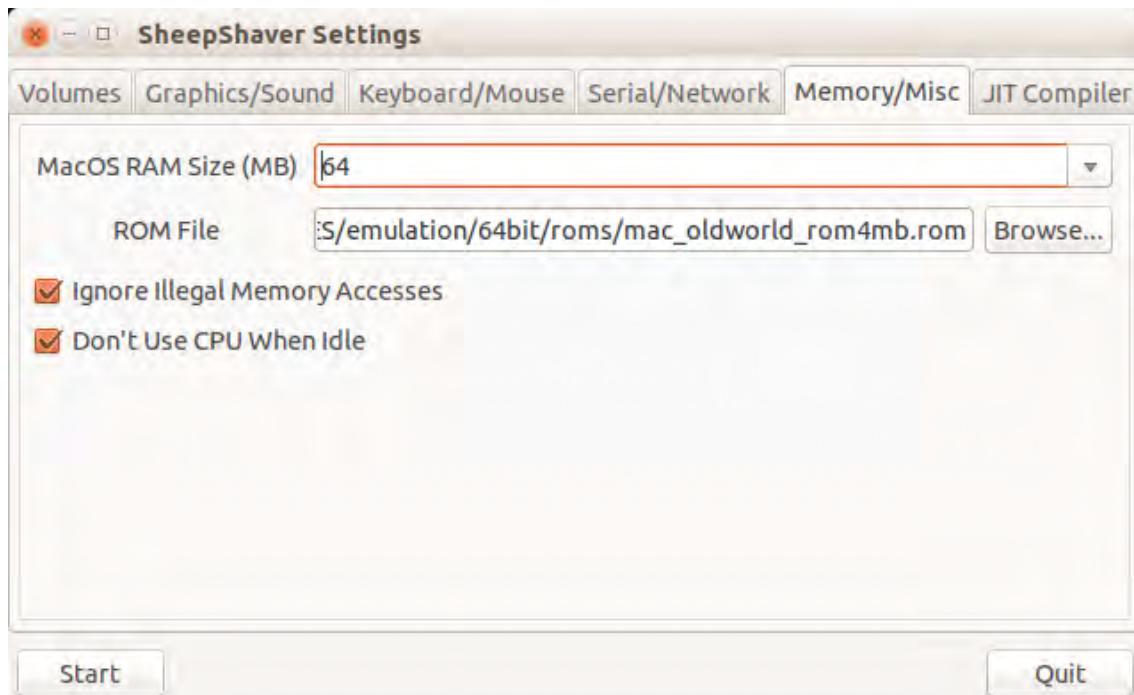


Figure 4.20: Memory/Misc tab in SheepShaver Settings.

When on the **Memory/Misc** tab, use the work's stated system requirements to determine the **MacOS RAM Size**. The project team found that either 64MB or 128MB was a good starting point when no other information was available.

Consult Table 4.1, which lists ROMs for Apple/Macintosh Emulation to determine which ROM File is appropriate. The project team found that SheepShaver worked best with an “old world” ROM for emulating systems prior to OS 9, and a “new world” ROM worked best with OS 9.

The project team also found that selecting the checkboxes next to **Don't Use CPU When Idle** and **Ignore Illegal Memory Accesses** improved performance of the emulator (Figure 4.20).

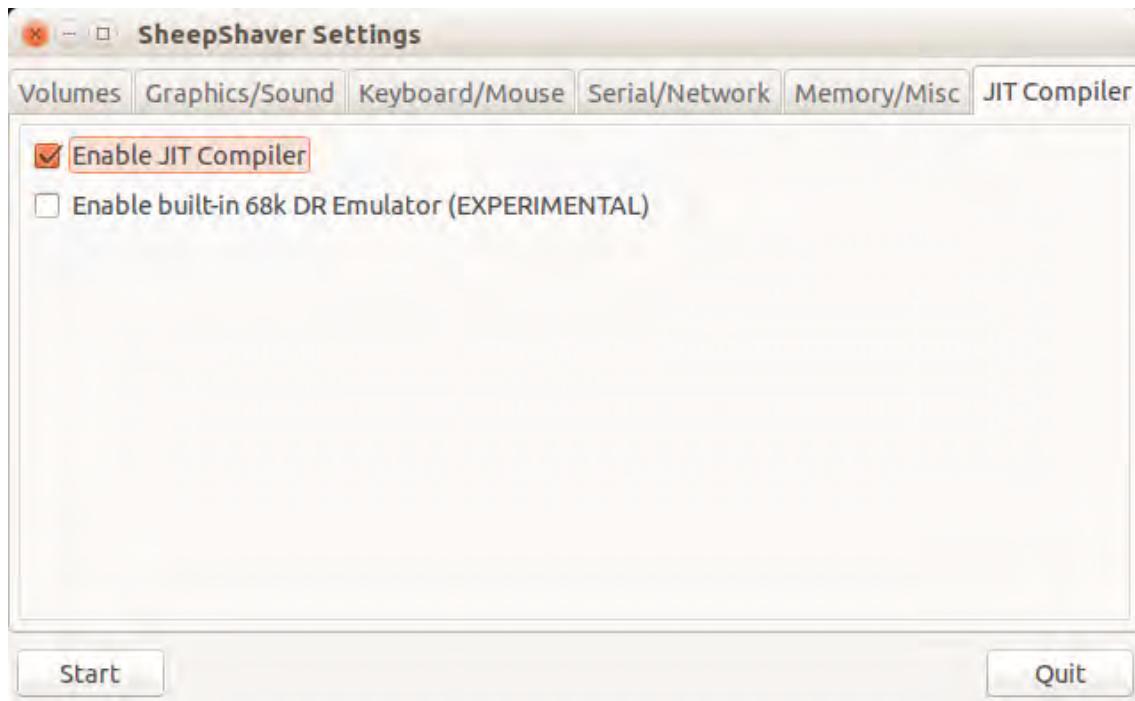


Figure 4.21: JIT Compiler tab in SheepShaver Settings.

The project team found that checking **Enable JIT Compiler** on the **JIT Compiler** tab improved performance. The project team did not test out the feature **Enable built-in 68k DR Emulator (EXPERIMENTAL)** (Figure 4.21).

## Starting SheepShaver

The project team encountered an issue running SheepShaver on an Ubuntu-based system. The issue has a straightforward permanent fix, but a user may encounter it when running SheepShaver on a new or different system that has not yet implemented the fix. If the user receives the error message “Cannot map Low Memory Globals: Operation not permitted” it is necessary to do the following to correct the problem.

As root, edit the file in `/etc/sysctl.conf` and add the following line to the end of the file:

```
vm.mmap_min_addr = 0
```

Next, run the following command (as root or using sudo):

```
service procps start
```

It is important to note that an alternate solution to this problem is to run SheepShaver as root, but this is not recommended.<sup>i</sup>

The project team recommends starting the emulator once before the artwork is loaded. This saves the selected settings in the preferences file and gives the user the opportunity to preview the operating system environment to ensure that system requirements within the guest system are appropriate for the selected artwork. For instance, if the system requirements specify “requires QuickTime extensions,” the user can check to see that Quicktime is in the System:Extensions folder, and QuickTime is installed. It can also be helpful here to verify that sound is working properly. The previous section on Starting Basilisk II (4.4) includes several screenshots (Figures 4.11, 4.12, 4.10) to demonstrate how to examine the system’s extensions and test for sound.

Once system requirements have been confirmed (as best as possible), the user can shut down the guest system and prepare the preferences file to load the artwork. Instead of opening up the graphical user interface, edit the `.sheepshaver_prefs` file and add the following line:

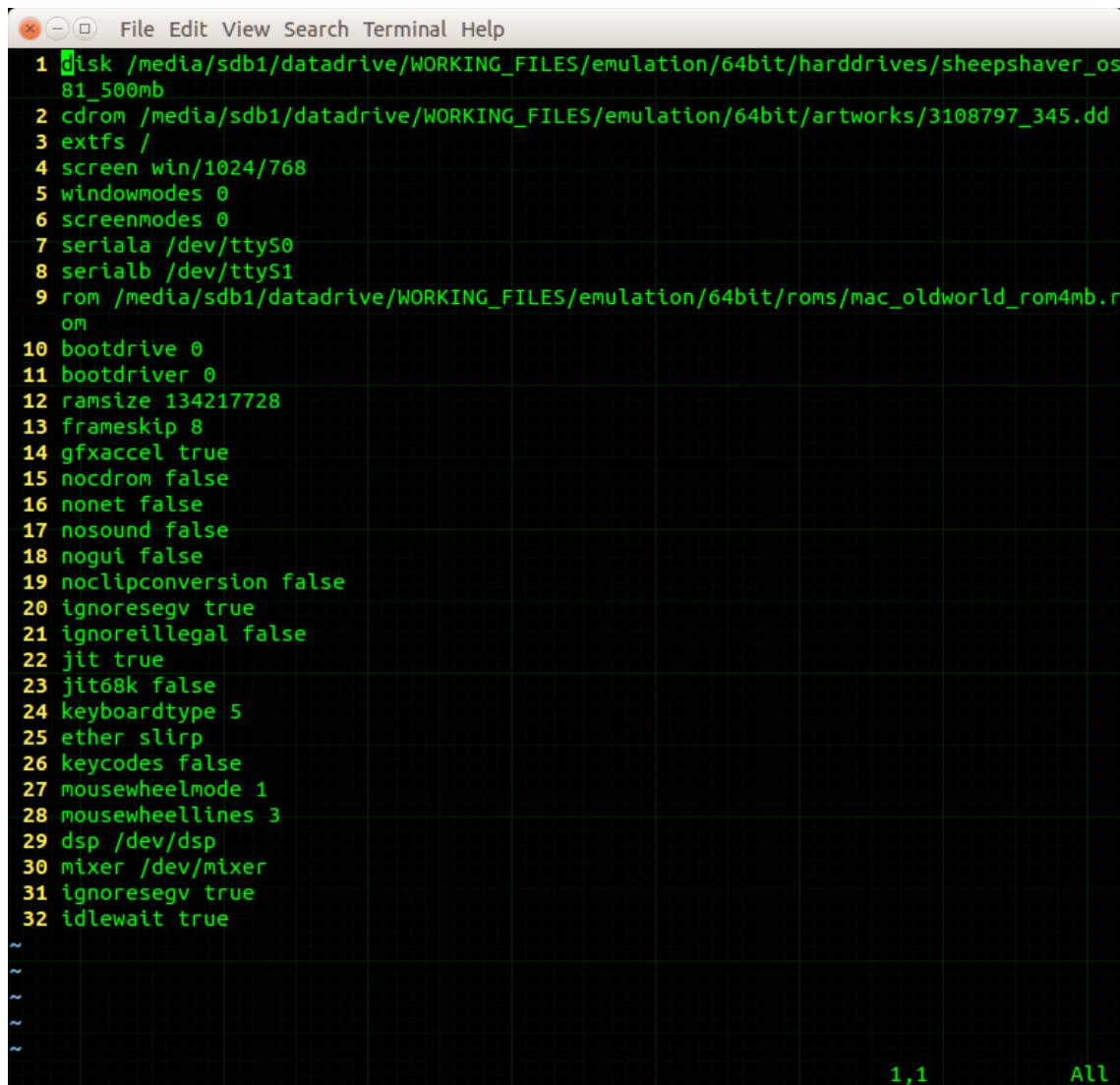
```
cdrom /full_path/artwork.isoj
```

Where `full_path` is the exact path to the disk image of the artwork. Figure 4.22 is an example of a properly edited preferences file.

---

<sup>i</sup>By default, SheepShaver will share a folder with the host machine, and this is typically set to the root directory if not changed. Running SheepShaver as root means that the process has full permissions to that shared folder.

<sup>j</sup>Extension might be .iso, .dd, or .bin; any of these will work as long as it is a disk image.



A screenshot of a terminal window titled "File Edit View Search Terminal Help". The window contains a configuration file for SheepShaver. The file includes settings for disk, cdrom, extfs, screen, windowmodes, screenmodes, seriala, serialb, rom, bootdrive, bootdriver, ramsize, frameskip, gfxaccel, nodrom, nonet, nosound, nogui, noclipconversion, ignoresegv, ignoreillegal, jit, jit68k, keyboardtype, ether, keycodes, mousewheelmode, mousewheellines, dsp, mixer, ignoresegv, and idlewait. The file ends with several tilde characters (~).

```
1 disk /media/sdb1/datadrive/WORKING_FILES/emulation/64bit/harddrives/sheepshaver_os
 81_500mb
2 cdrom /media/sdb1/datadrive/WORKING_FILES/emulation/64bit/artworks/3108797_345.dd
3 extfs /
4 screen win/1024/768
5 windowmodes 0
6 screenmodes 0
7 seriala /dev/ttys0
8 serialb /dev/ttys1
9 rom /media/sdb1/datadrive/WORKING_FILES/emulation/64bit/roms/mac_oldworld_rom4mb.r
om
10 bootdrive 0
11 bootdriver 0
12 ramsize 134217728
13 frameskip 8
14 gfxaccel true
15 nodrom false
16 nonet false
17 nosound false
18 nogui false
19 noclipconversion false
20 ignoresegv true
21 ignoreillegal false
22 jit true
23 jit68k false
24 keyboardtype 5
25 ether slirp
26 keycodes false
27 mousewheelmode 1
28 mousewheellines 3
29 dsp /dev/dsp
30 mixer /dev/mixer
31 ignoresegv true
32 idlewait true
~
~
~
~
~
```

Figure 4.22: Properly configured configuration file to load an artwork as read-only media in SheepShaver.

After the file has been edited and saved, the user can open up the SheepShaver graphical interface and click **Start** to start the emulator with the artwork loaded as a “CD-ROM” and therefore read-only. It is important to note here that the artwork will not be listed in the Volumes tab of the graphical interface after it has been added to the preferences file as cdrom.

## References

- [1] *The Official Basilisk II Home Page.* URL: <http://basilisk.cebix.net/>.
- [2] *The Official SheepShaver Home Page.* URL: <http://sheepshaver.cebix.net/>.

## 5 — Emulating PC-compatible systems with QEMU

The project team used QEMU to provide access to works in the Goldsen that required PC-compatible systems with x86 processors only, even though QEMU is capable of emulating a wide variety of systems and architectures. The project team tested QEMU<sup>a</sup> with Windows 3.1, Windows 95, and Windows 2000. This emulator is open-source and can be compiled for a variety of architectures. The project team used version 2.0.0 provided in the Ubuntu 14.04 repository; this seemed to work well enough that it did not seem necessary to compile the application from source.

In order to run QEMU, the following components are needed:

1. qemu and qemuctl, which are installable through Ubuntu's software repository.<sup>b</sup>
2. Disk image file with installed operating system, or disk image of retail installation media.
3. Disk image of the CD-ROM with the original artwork.

The following is an overview of general settings for the emulators, how to load a disk image and access an artwork, and other important information to consider, drawn from the project team's notes, observations, and testing.

---

<sup>a</sup>QEMU emulator version 2.0.0 (Debian 2.0.0+dfsg-2ubuntu1.9), Copyright (c) 2003-2008 Fabrice Bellard

<sup>b</sup>qemuctl provides Qemu Launcher, a graphical interface that is technically not required to run the qemu binary.

## 5.1 Settings

There are various settings and configuration options for QEMU. Some settings pertain to the operation of the emulator software, while settings determine settings for the guest system (i.e., the hardware/software environment being emulated).

In QEMU, the following program settings can be adjusted:

- Shared directory between guest and host machine (SMB)
- Configuration for host handling of Network interface
- Toggle full screen for guest display
- Serial port device and keyboard layout to use for guest machine
- Toggle control panel and logging (and what program behavior to log)
- Enable acceleration, process priority for QEMU on host system
- Monitor device (disabled when the control panel is on)
- Paths for data directory (typically home directory), QEMU, QEMU-img, QEMU-ctl and pre-launch command (blank by default)

QEMU offers the following guest system settings:

- Toggle on CD-ROM
- Boot device (can be CD-ROM, Floppy device, or Hard disk)
- Floppy device to mount (maximum 2)
- CD-ROM image to mount (maximum 1)
- Hard disk volumes to mount (maximum: 4; maximum 3 if CD-ROM is enabled)
- RAM (Random access memory) size
- System type to emulate<sup>c</sup>
- Toggle on audio, selection for sound card
- Toggle off display, selection for video card
- Clock setting (toggle synchronization with host machine)

---

<sup>c</sup>PC, 32-bit (x86) will emulate a PC-compatible system suitable for the images in the Goldsen collection.

Name of disk image file	Size (in MB)	Operating System Installed
qemu_windows95_1000mb	1000	Windows 95
qemu_windows2000_2000mb	2000	Windows 2000 Professional

Table 5.1: Disk image files and corresponding installed Windows operating systems for QEMU.

## 5.2 Disk Image File with installed Operating System

In order to use QEMU to emulate a PC-compatible system, it is necessary to have a disk image file with an installed operating system. The project team created multiple disk image files and installed various Windows operating systems for use with QEMU.

Table 5.1 shows the various files created for access, identifying the name of the disk image file and its installed version of the operating system.

## 5.3 Running the emulator and loading an artwork

The Qemu Launcher provides access to nearly all of the settings required for running the emulator and loading a disk image of a CD-ROM as a CD-ROM device. It is also possible to specify all configuration options for QEMU on the command line, and in some cases, such as specifying an alternate sound card or enabling networking, it may be necessary to launch qemu from the command line.

Once all settings are configured and a disk image file with an appropriate operating system has been selected, the process for accessing an artwork with QEMU is relatively straightforward. The following sections will include screenshots of the configuration options for QEMU and an overview of the process. While it is possible to add a disk image of the CD-ROM of the artwork as a “Hard Disk,” the project teams recommends that it be added as a CD-ROM device so that the disk image will be loaded in the guest system as a CD-ROM, that is, as a read-only volume.

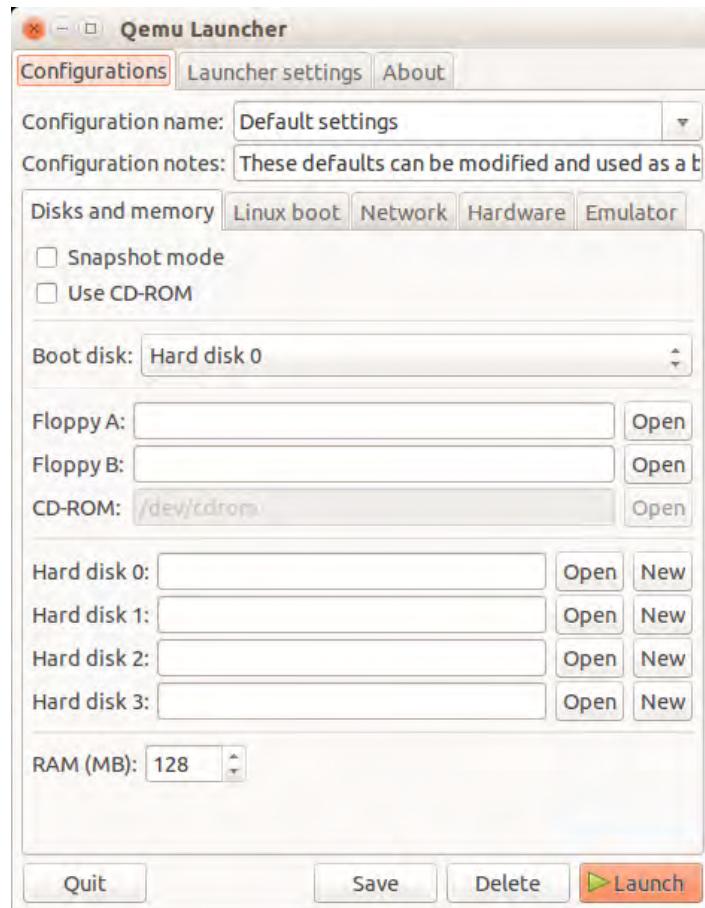


Figure 5.1: Configurations tab/Disks and Memory subtab in Qemu Launcher.

**Disks and Memory** is the default subtab within the **Configurations** tab; click **Use CD-ROM** to active the selector to load a CD-ROM image file (Figure 5.1).

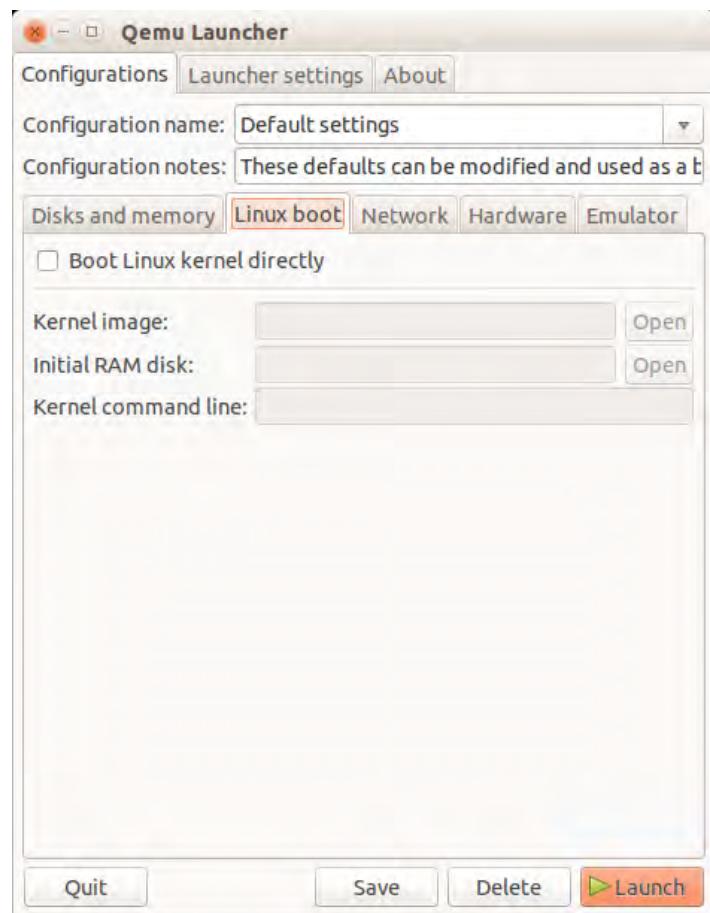


Figure 5.2: Configurations tab/Linux boot subtab in Qemu Launcher.

The defaults for the **Linux boot** subtab can be left as they are (Figure 5.2).

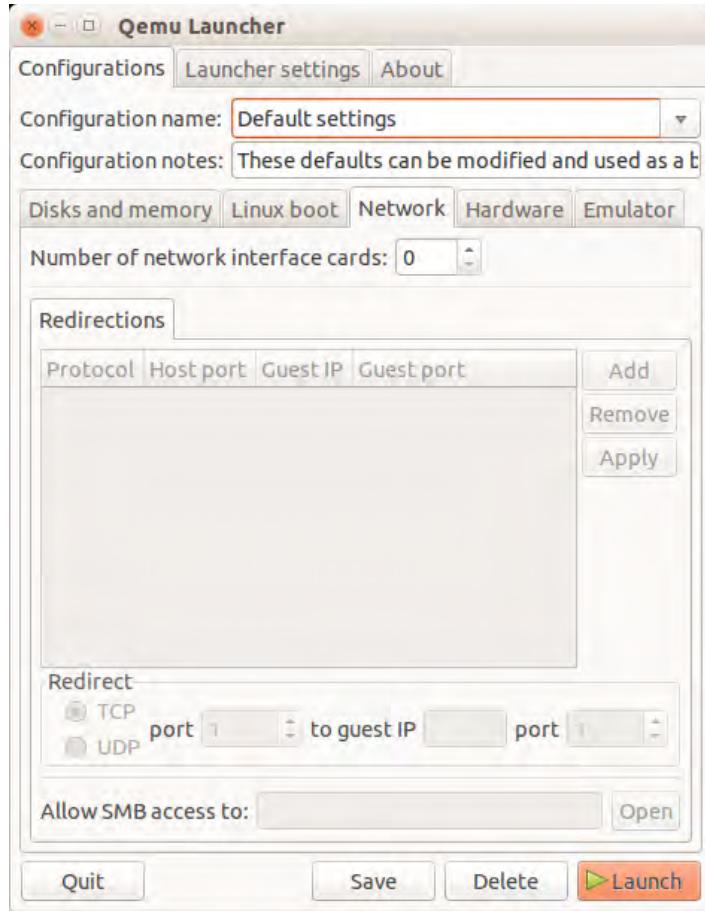


Figure 5.3: Configurations tab/Network subtab in Qemu Launcher.

If the work does not need an active network connection, set the **Number of network interface cards** to **0** (Figure 5.3). Since older versions of Windows are no longer actively maintained by Microsoft and these operating systems had a history of vulnerabilities, the project team suggests not enabling the networking features unless absolutely required for a work.

If an active internet/network connection is needed, the project team found it easiest to launch qemu using the command line, as the default configuration settings used by the graphical interface to specify network devices do not reliably work.

When launching qemu from the command line, the following additional flags seemed to work to enable networking.

```
-net nic,vlan=0,model=pcnet
```

The project team notes that this is a deprecated method of enabling networking within QEMU, and a future revised strategy may be needed.

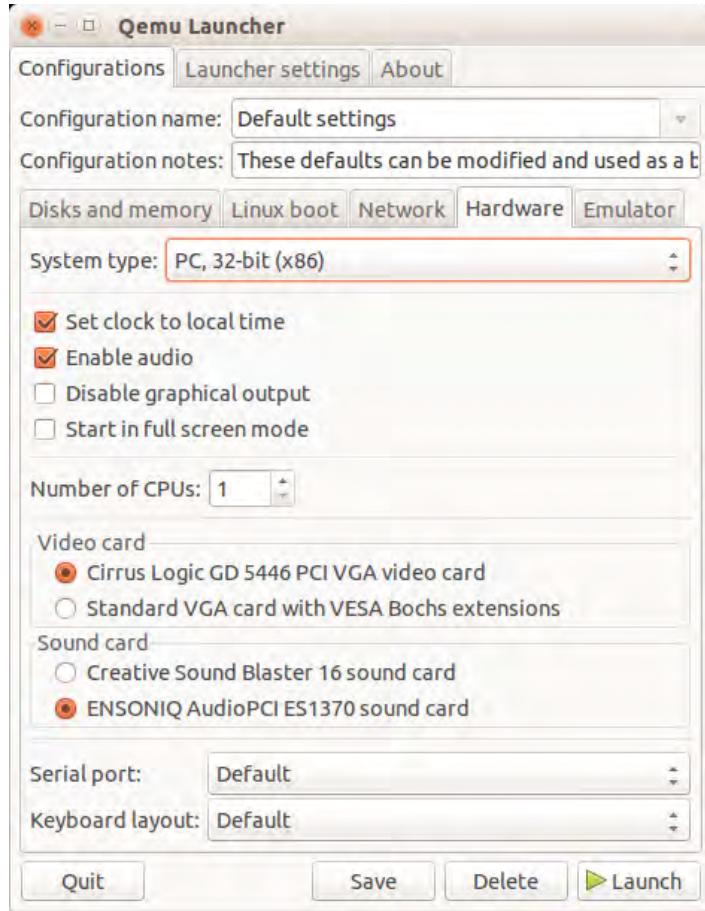


Figure 5.4: Configurations tab/Hardware subtab in Qemu Launcher.

The defaults on the Hardware subtab work well (Figure 5.4); it is important to select “Enable audio” if the work requires audio. The project team noted that when the **Creative Sound Blaster 16 sound card** was set, the guest operating system often froze and was not functional. The **ENSONIQ AudioPCI ES1370 sound card** did fare better in this respect in the guest operating system, but the sound could sometimes be choppy. The project team found that additional sound card options (that worked better than the two available in the graphical interface) were available when launching qemu on the command line.<sup>d</sup> It would be advisable to test several sound options in QEMU when accessing a work to determine what produces optimal sound, if sound is critical for rendering the artwork properly.

---

<sup>d</sup>ac97 performed quite well, though it is not an available option in Qemu Launcher. It is available when launching qemu from the command line.

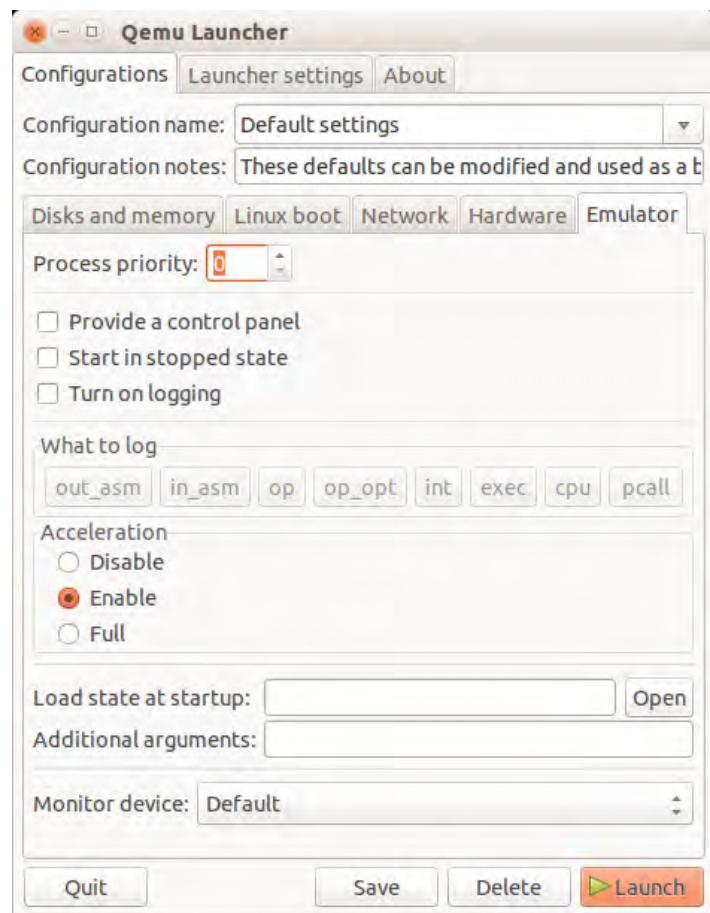


Figure 5.5: Configurations tab/Emulator subtab in Qemu Launcher.

On the **Emulator** subtab, the project team found that the defaults work well. If cursor movement within the guest system is lagging, it might be helpful to set **Acceleration** to **Full**. (Figure 5.5)

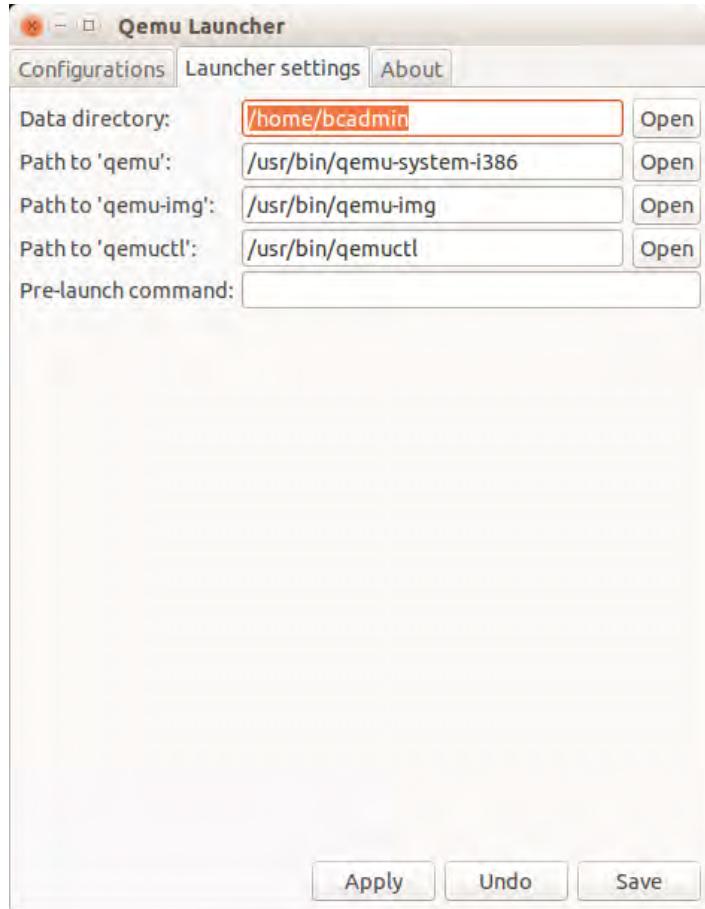


Figure 5.6: Launcher settings tab in Qemu Launcher.

On the **Launcher settings** tab, the project team found it necessary to specify **qemu-system-i386** as the Path to ‘qemu,’ as selecting the **System type** under the **Hardware** subtab did not auto-select the appropriate qemu binary on the testing system (Figure 5.6).

Within Qemu Launcher, settings can be saved if a particular system configuration is helpful to have as a preset.

Finally, within qemu, it is possible to “swap” out devices while the guest machine was running. This can be helpful when looking at multiple works on the “same” guest system. In order to “eject” a CD-ROM from a system and “load” a new one, it is necessary to interact with the QEMU monitor [1]. To enter into the monitor console, while the guest system is running, press **Ctrl-Alt-Shift-2**. This will change the screen from the guest operating system to a console window. To swap out a CD-ROM, use the **info block** command to determine the name of the CD-ROM device, and then the **change** command to specify a new ISO file to load. A screenshot of the default interface is shown below. Directions are found by typing **help** at the prompt or by consulting the Wikibooks manual for QEMU/Monitor [2] (Figure 5.7).

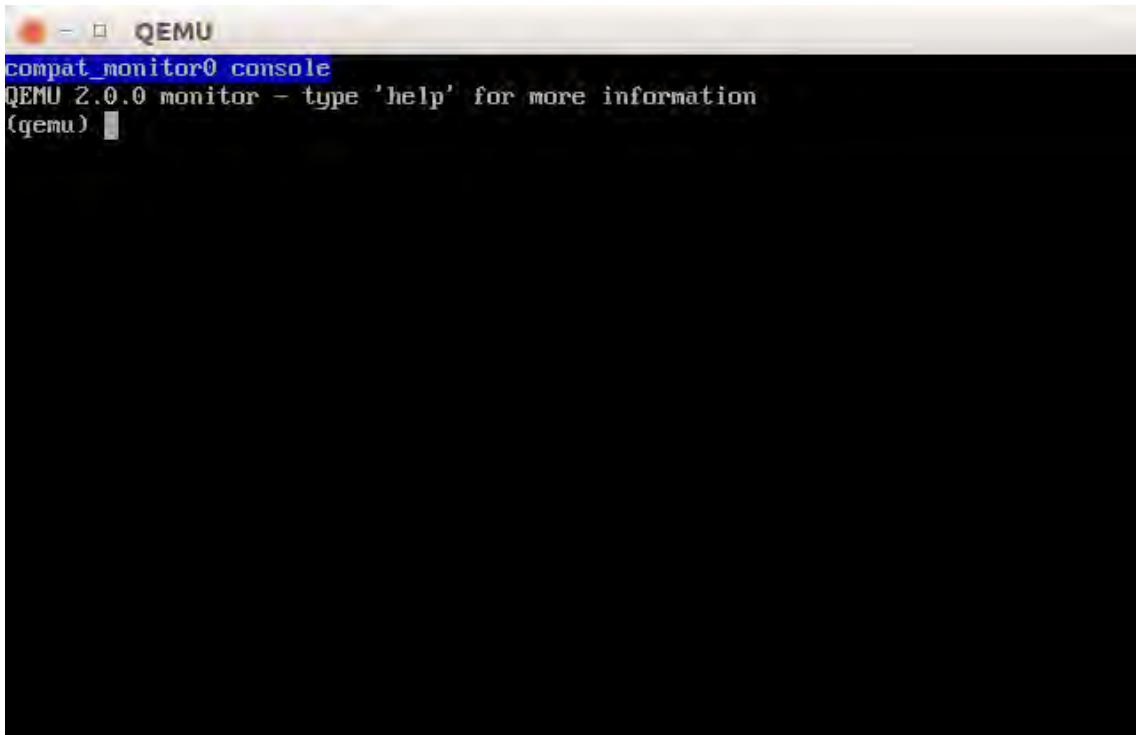


Figure 5.7: QEMU Monitor console.

Alternatively, if qemu is run on the command line, it is possible to specify a monitor to run on the Terminal. This is shown in Figure 5.8.

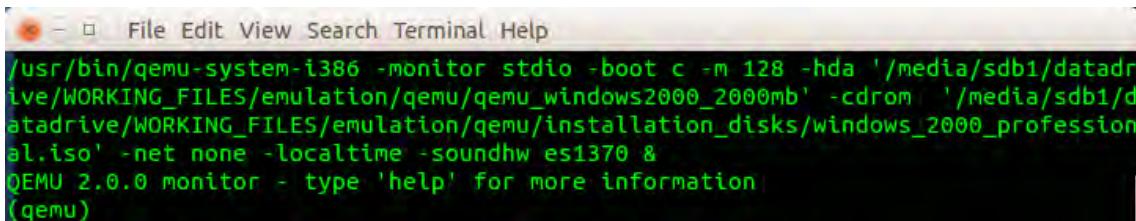


Figure 5.8: QEMU Monitor in terminal window.

## **5.4 Process-level emulation—WineHQ and QEMU user mode**

The project team experimented briefly with WineHQ [3], a utility that would allow PC-compatible executable files to be run without emulating a complete Windows operating system. While some artworks did function smoothly under Wine, the project team noted that artworks that required the user to install additional third-party plugins (e.g., QuickTime) did not fare as well under WineHQ and often crashed or failed to start. It may be a viable option for some artworks, however. Further, there may be potential in using the QEMU user mode to provide emulation for standalone executable files [4, 5] but the project team did not test this out. Based on the preliminary findings from testing with WineHQ, the project team can speculate that works that require the user to install additional third-party software may be difficult to run with this method.

## References

- [1] *QEMU/Monitor: eject* - Wikibooks, open books for an open world. URL: <http://en.wikibooks.org/wiki/QEMU/Monitor#eject>.
- [2] *QEMU/Monitor* - Wikibooks, open books for an open world. URL: <http://en.wikibooks.org/wiki/QEMU/Monitor>.
- [3] *WineHQ* - Run Windows applications on Linux, BSD, Solaris and Mac OS X. URL: <https://www.winehq.org/>.
- [4] *How to use qemu* - FedoraProject. URL: [http://fedoraproject.org/wiki/How\\_to\\_use\\_qemu](http://fedoraproject.org/wiki/How_to_use_qemu).
- [5] *QemuUserEmulation* - Debian Wiki. URL: <https://wiki.debian.org/QemuUserEmulation>.

## A — Compiling notes for Basilisk II and Sheepshaver executables

A Basilisk II or SheepShaver executable file is one of the components needed to run the emulator. The project team found it best to compile these emulators from source so that they were compatible with our preferred platform and that we could control the configuration options.

While neither project is under active development as of this writing (October 2014), a small group of volunteers continually add additional fixes to the source code, which is hosted on GitHub.com. Therefore, over the two years for the grant, the project team has compiled the code twice, to ensure compatibility with the updates to the project's official workstation. The project team anticipates that this section will have future revisions when the code needs to be recompiled and offers the following structure for future curators to capture all of the pertinent information for the compiled code.

### **Current Basilisk and SheepShaver Executables: Revision 2, August 2014**

These emulators consisted of the following:

- 64-bit executables for Basilisk II and SheepShaver

The files are named BasiliskII and SheepShaver and are executable files.

The major limitations of the emulators at the time were as follows:

- The code still needed to be compiled against an earlier version of gcc (4.4) in order to run properly. (Using a higher version of gcc would cause the SheepShaver executable to immediately segfault on launch.)

- The code for the PPP emulation (slirp) was still not 64-bit safe; however, a user offered some modifications to that code (that drew from a more recent version of slirp) which were available as a branch on GitHub. Since this code allowed us to compile Basilisk II and SheepShaver to have functional networking capability, we opted to use it instead of the main code branch.

## Technical details

GitHub URL:

<https://github.com/krechert/macemu/tree/e8284b544f3c2608c8d6e869bdc8ee3983cdbccb>

Compiled on: BitCurator 0.9.13 (Ubuntu 14.04LTS) Kernel name, revision, version: Linux 3.13.0-30-generic #54-Ubuntu SMP Mon Jun 9 22:45:01 UTC 2014

Packages/libraries installed:

- gcc-4.4, g++-4.4
- libsdl1.2-dev

Note: Everything else needed appeared to be already installed on the machine.

Additional configuration notes:

- The configuration flags used for Basilisk II:  
`./configure --enable-sdl-video --enable-sdl-audio  
--disable-vosf --enable-jit-compiler --with-gtk --with-bincue`
- The configuration flags used for SheepShaver:  
`./configure --enable-sdl-video --enable-sdl-audio  
--disable-vosf --with-gtk --with-bincue`

## Revision 1, October 2013

These emulators consisted of the following:

- 32-bit executables for Basilisk II and SheepShaver.
- A folder with the necessary 32-bit libraries to run both executables (this list was obtained by running `ldd` on each).
- A shell script that added the 32-bit libraries to `LD_LIBRARY_PATH` and then ran the respective executable.

Even though our working environment (BitCurator 0.3.0) was 64-bit, we opted to compile these as 32-bit executables in order to address one of the key limitations of the emulators at the time.

The major limitations of the emulators at the time were as follows:

- Compiling against the version of gcc on the BitCurator machine (gcc-4.7) created an executable for SheepShaver that immediately segfaulted when run. So it was necessary to deliberately use gcc-4.4 so that this would not happen.
- The code for both emulators that handled PPP emulation (slirp) was not 64-bit safe. Compiling either emulator on a 64-bit machine resulted in an executable that, when set to use slirp for networking, immediately segfaulted when attempting to access the internet.

### Technical details

GitHub URL:

<https://github.com/cebix/macemu/tree/43779be7750fea2f96bcc99136c802ddd014d70b>

Compiled on: 32-bit virtual machine running Debian 7 Kernel name, release, version: Linux 3.2.0-4-686-pae #1 SMP Debian 3.2.46-1+deb7u1

Packages/libraries installed: (Note: those in parentheses might not have been required, but were installed anyway.)

- `gcc-4.4`, `g++-4.4`, (`gcc-4.4-base`, `gcc-4.4-locales`, `gcc-4.4-multilib`, `gcc-4.4-source`)
- `libsdl1.2-dev`, (`libsdl1.2-dbg`, `libsdl1.2-debian`)
- `libgtk2.0-dev`, `libgnome2-dev`

- **hdparm**

Additional configuration notes:

- Edit `/etc/hdparm.conf` to set `dma=on`; for the `/dev/mapper` device. (Note: This is necessary for the `--enable-jit-compiler` option, but probably only for compiling on a virtual machine.)

- The configuration flags used for Basilisk II:

```
./configure --enable-sdl-video --enable-sdl-audio  
--disable-vosf --enable-jit-compiler --with-gtk
```

- The configuration flags used for SheepShaver:

```
./configure --enable-sdl-video --enable-sdl-audio  
--disable-vosf --with-gtk
```

## B — Creating a new disk image file for Basilisk II and SheepShaver and installing an operating system

In the event that the disk image files with installed operating systems become unusable or the team acquires an additional retail copy of a Macintosh operating system supported by either Basilisk II [1] or SheepShaver [2], it is rather straightforward to create a new disk image file and install an operating system on it.

If not already available, make a disk image of the original retail installation CD for the desired operating system. Any disk imaging program should be adequate for this task; the project team used Guymager to create images of the retail CDs it acquired. Next, the project team recommends making a copy of the disk image and storing it in a separate folder, along with a record of the checksum for the disk image of the installation media. Installation can proceed using one of the copies of the disk image file. In the event that the installation media disk image file is corrupted, it is straightforward to make a new copy (rather than image the media again) and confirm that an exact copy of the original media is used for the installation.

The following screenshots show, in detail, the steps involved for installing Mac OS 8.1 using SheepShaver to provide one example of the process.

Start at the **Volumes** tab of the SheepShaver Settings screen. Click **Create...** to create a blank file that will represent the disk image file for the hard disk (Figure B.1).

The project team recommends a size of 500MB,<sup>a</sup> which exceeds the recommended hard disk size for OS 8 [3] and the following naming convention for the file itself: the name of the emulator, followed by an underscore; the name and version of the operating system; the size and units of the file, followed by an underscore. Click **OK** to proceed once the name is entered. (Figures B.2, B.3)

---

<sup>a</sup>The project team found that Mac OS 8 requires approximately 120 MB.

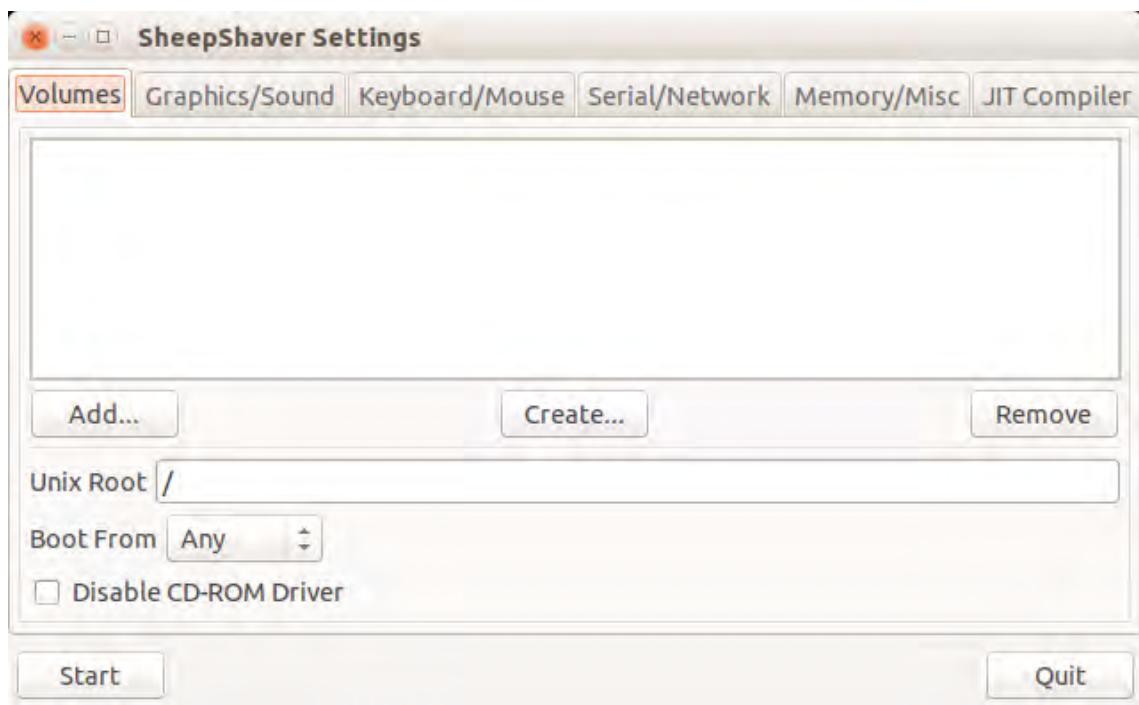


Figure B.1: SheepShaver Volumes tab in Settings.

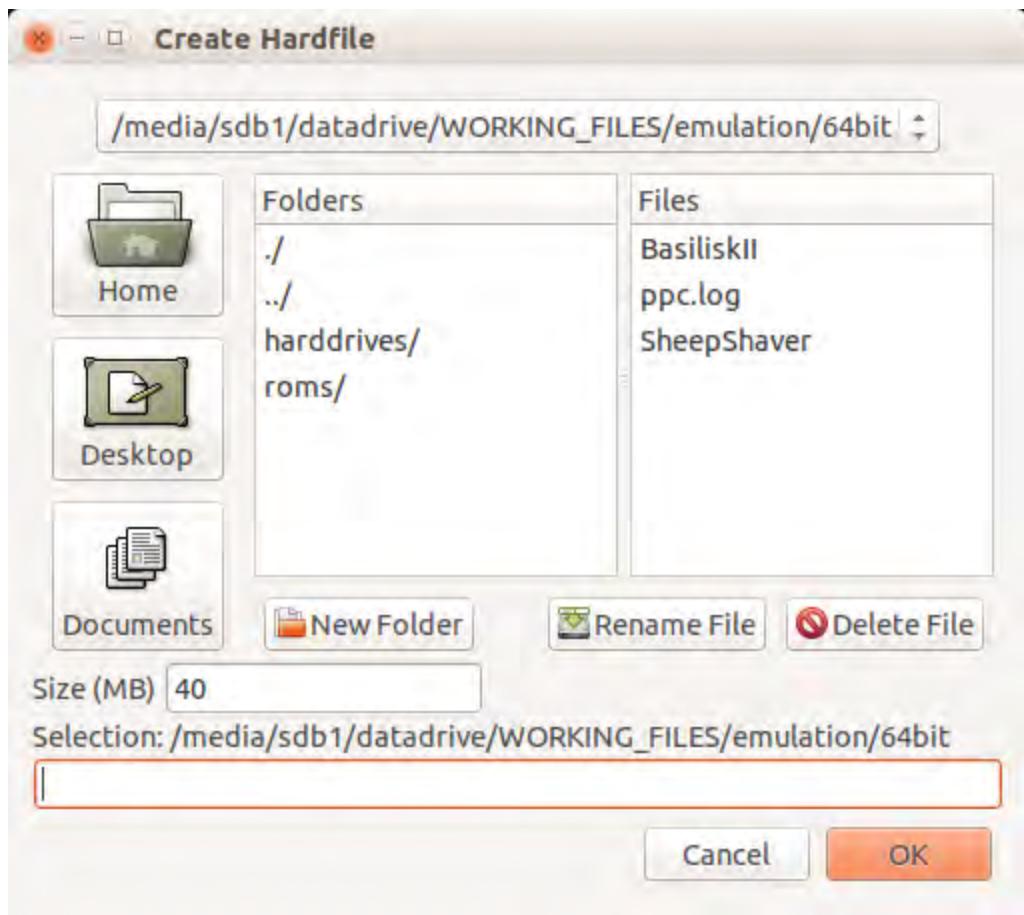


Figure B.2: Create Hardfile dialog box in SheepShaver.

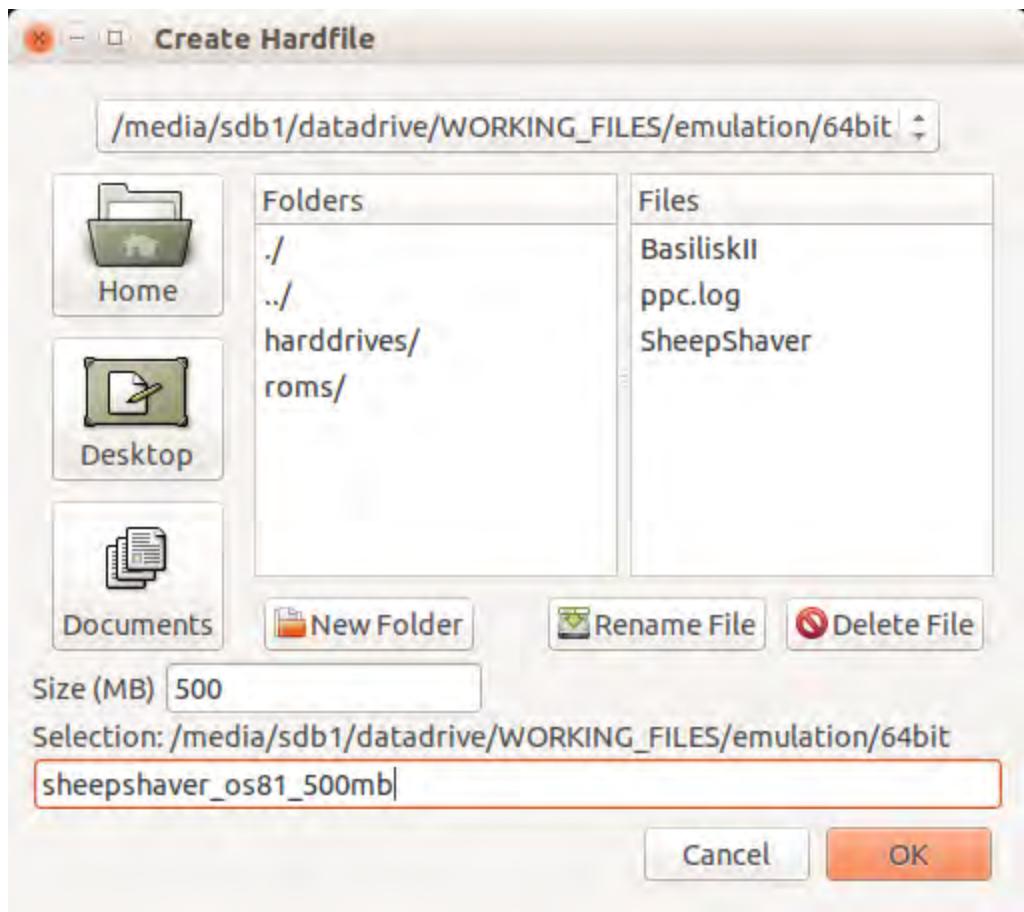


Figure B.3: Create Hardfile dialog box in SheepShaver with size and name entered.

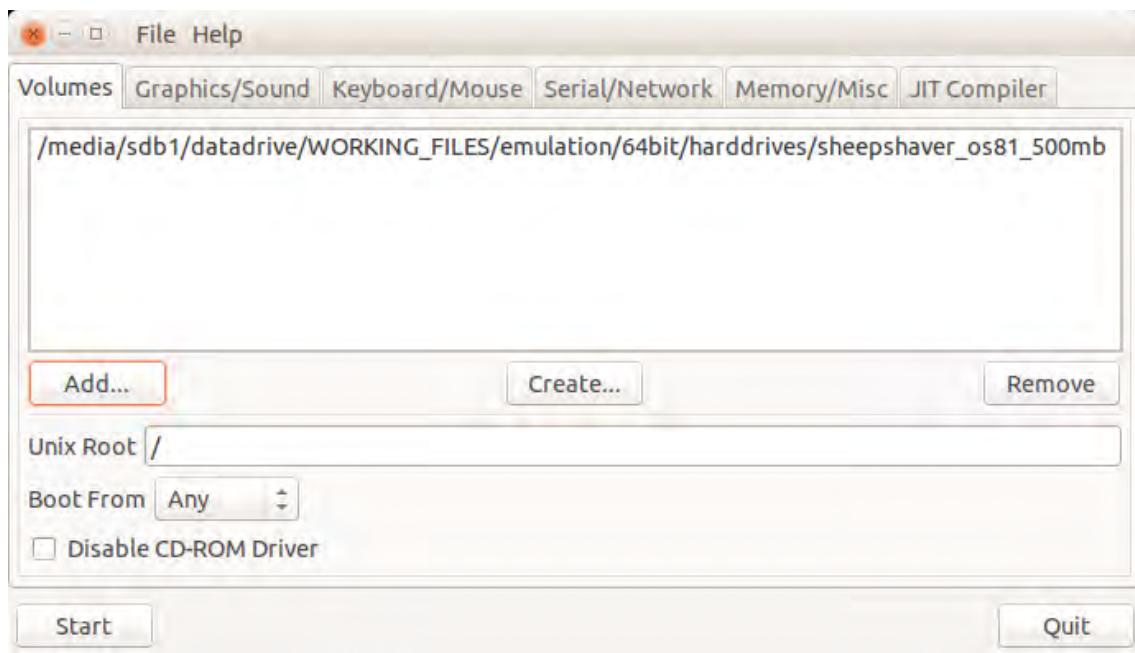


Figure B.4: Volumes tab in SheepShaver Settings with disk image loaded.

The newly created file will be added to the list in the Volumes tab (B.4). To add the installation media, click **Add** and follow the prompts in the dialog box. (Figure B.5)

The default settings in **Graphics/Sound** should be acceptable, though increasing the Width and Height of the window may be desired (Figure B.7).

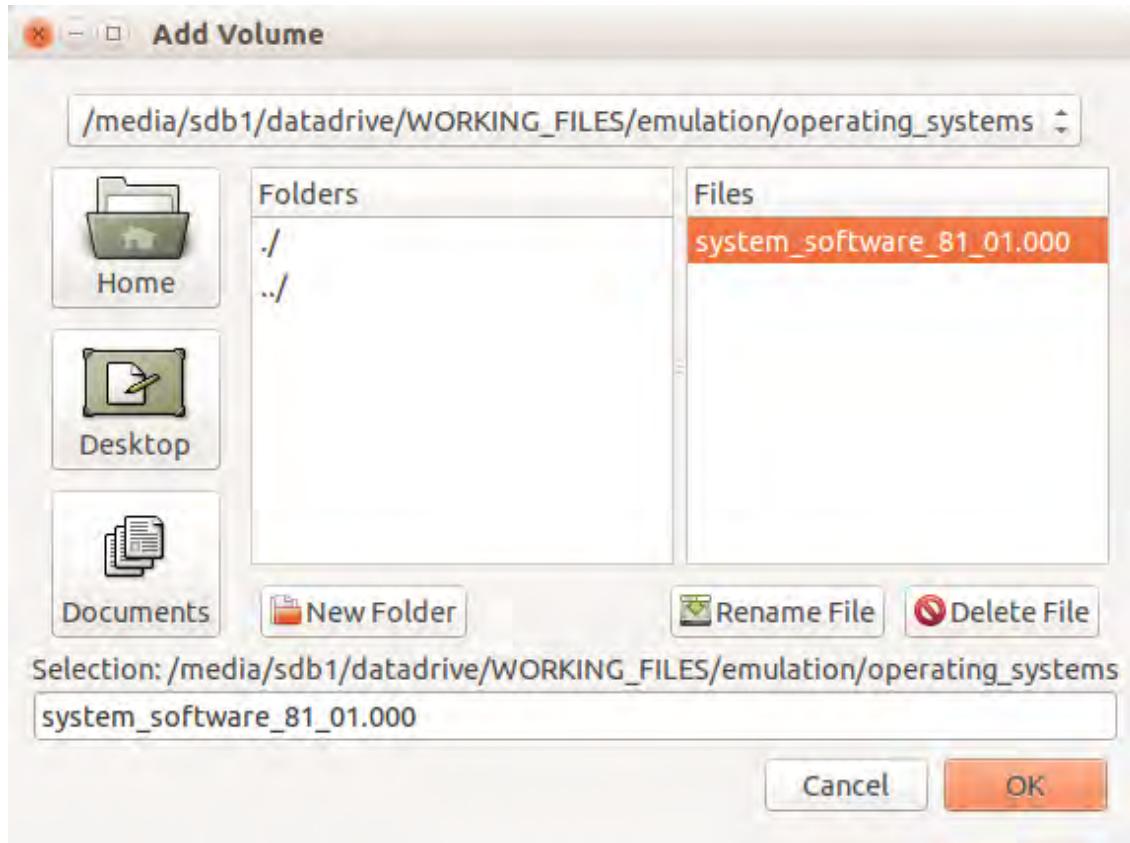


Figure B.5: Add Volume dialog in SheepShaver.

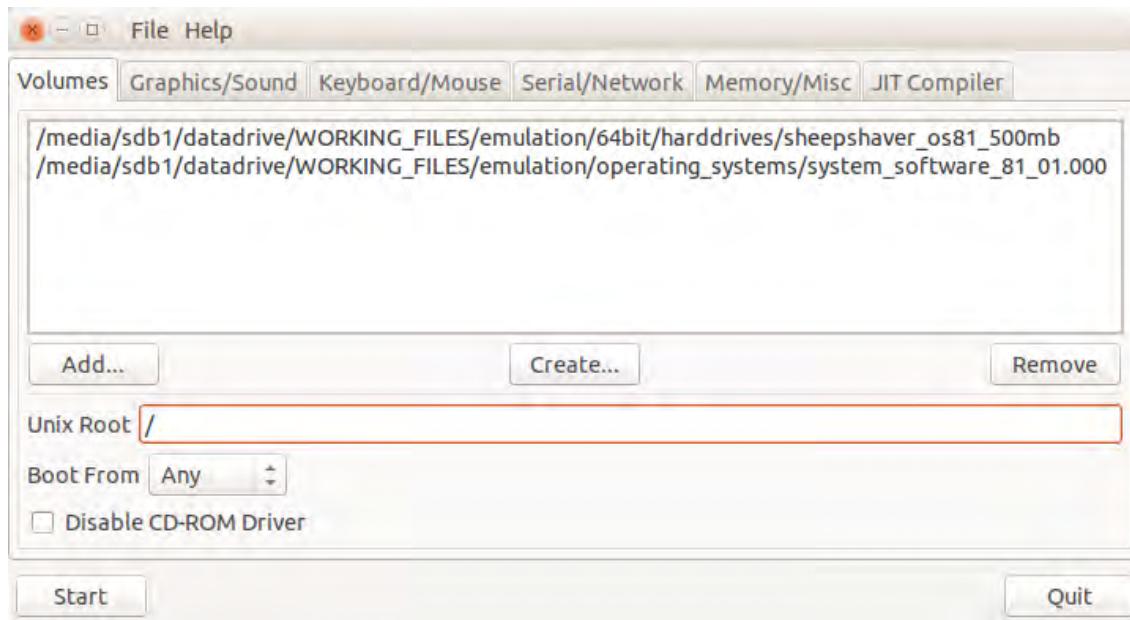


Figure B.6: Volumes tab in SheepShaver Settings with blank disk image file and installation media disk image file loaded.

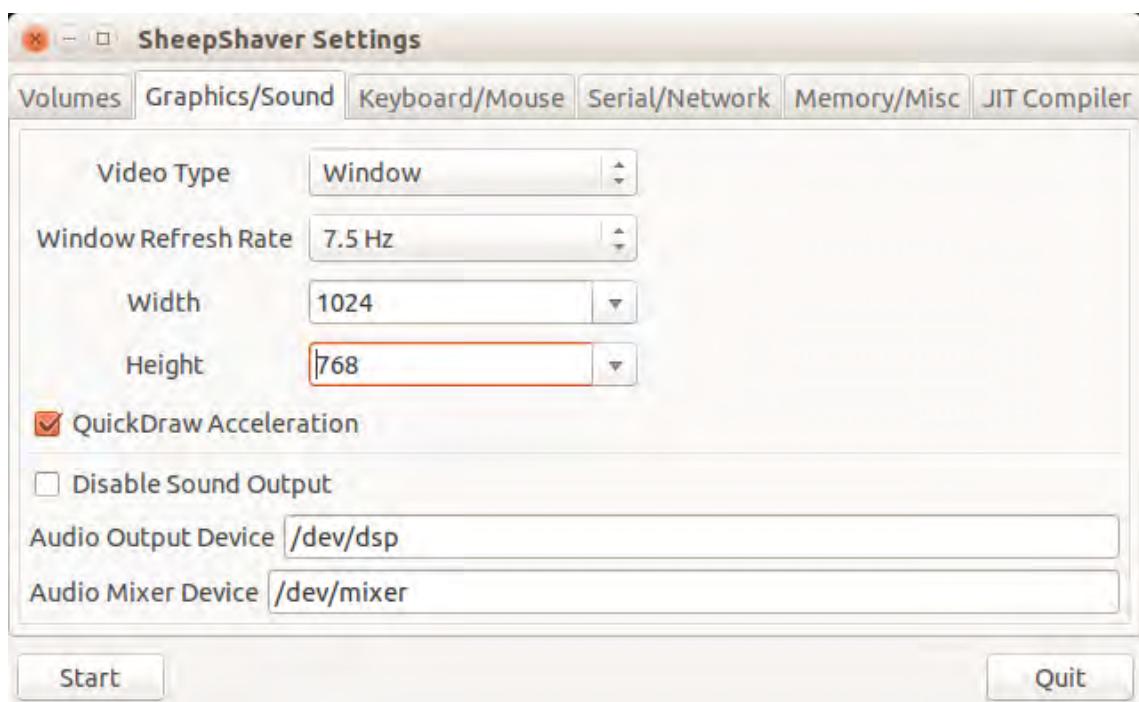


Figure B.7: Graphics/Sound tab in SheepShaver Settings.

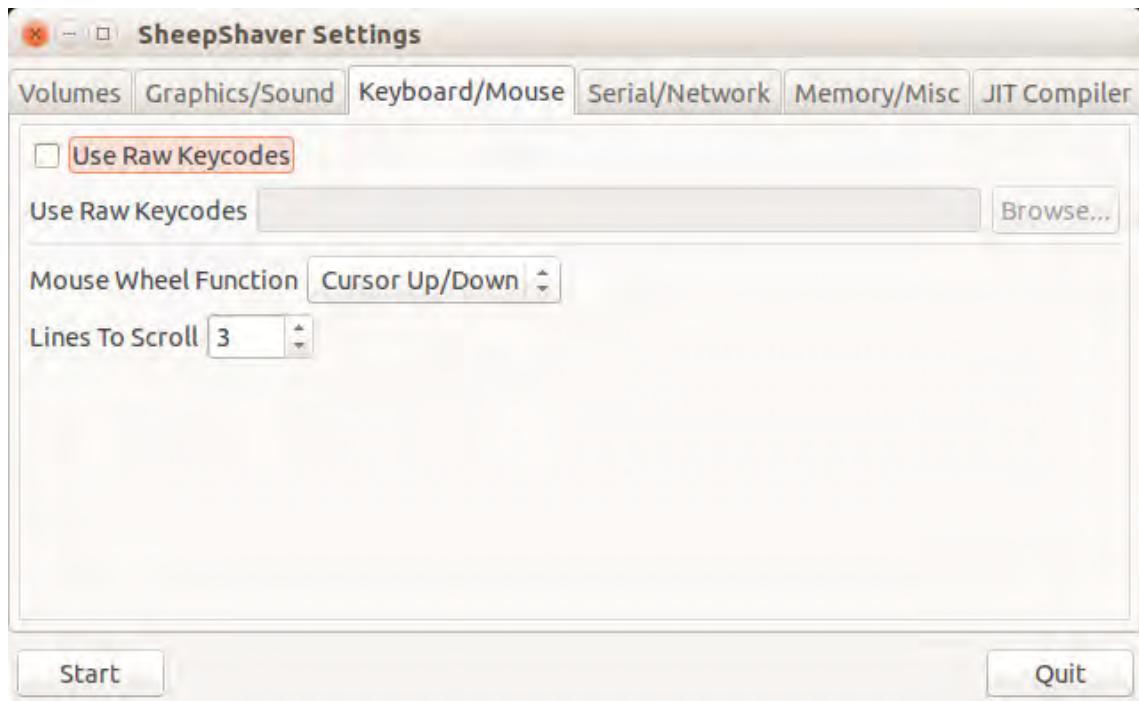


Figure B.8: Keyboard/Mouse tab in SheepShaver Settings.

The default settings in the **Keyboard/Mouse** tab should be acceptable (Figure B.8).

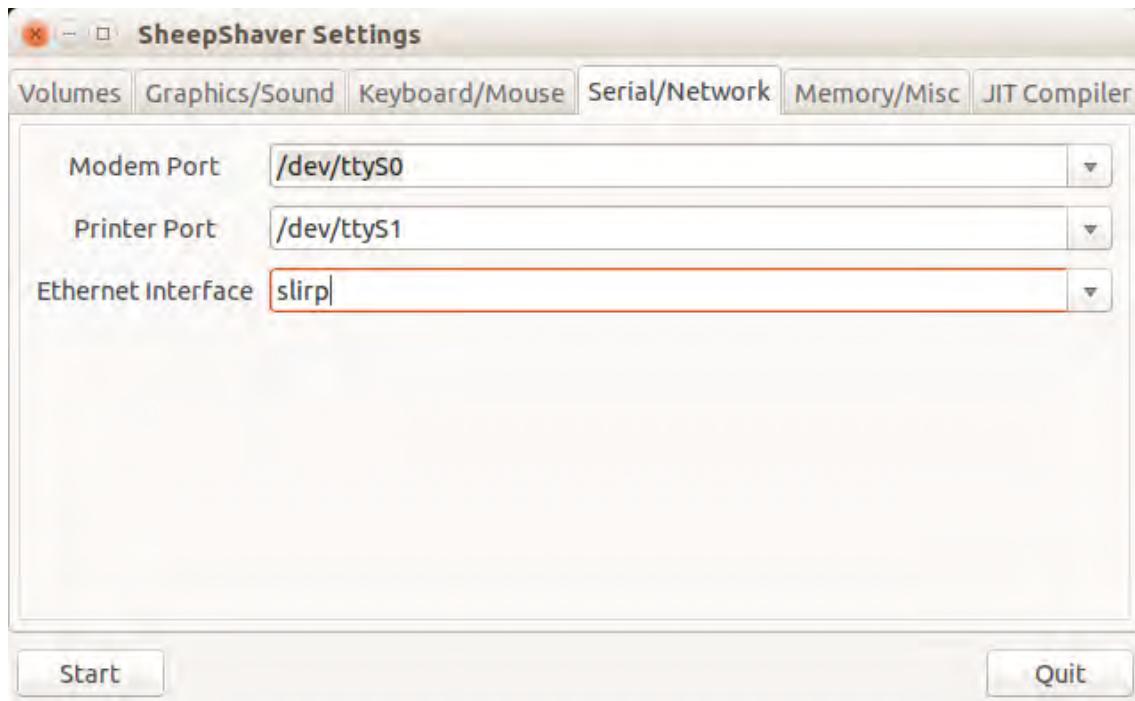


Figure B.9: Keyboard/Mouse tab in SheepShaver Settings.

If the work requires internet access, select **slirp** under **Ethernet Interface** to enable networking. (Note that this is the same process for Basilisk II.) (Figure B.9)

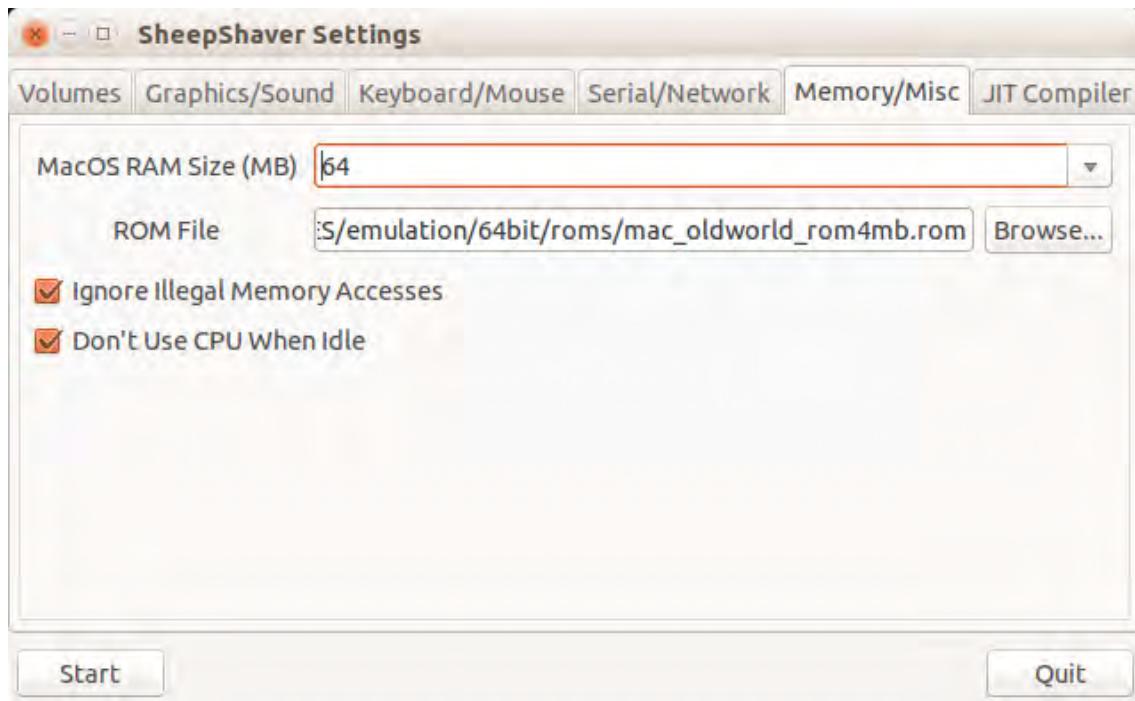


Figure B.10: Memory/Misc tab in SheepShaver Settings.

The **MacOS RAM Size** can be adjusted at any point depending on the system requirements of a particular work. The project team found for this installation 64 MB was sufficient to complete this install [3, 4]. Check the box for **Ignore Illegal Memory Accesses** and **Don't Use CPU When Idle**. (The project team found the software less prone to crashing when these boxes were selected.) (Figure B.10)

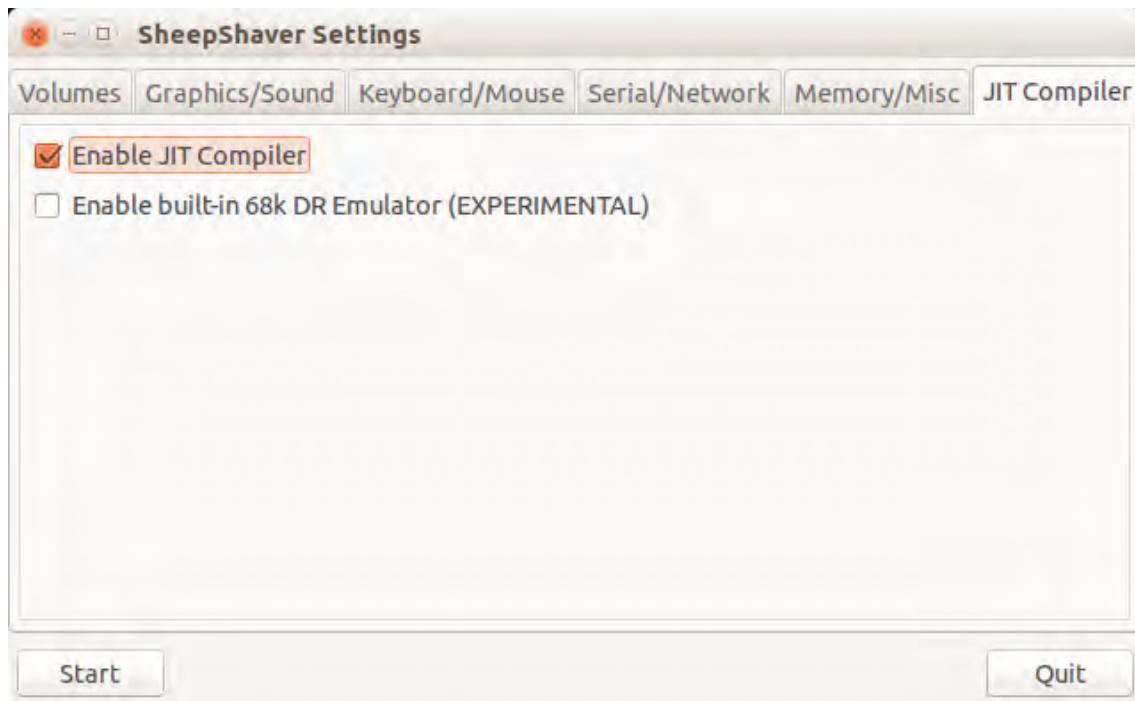


Figure B.11: JIT Compiler tab in SheepShaver Settings.

Ensure that **Enable JIT Compiler** is set on the JIT Compiler tab. (Figure B.11)

Select **Start** on the **Volumes** tab to start the emulator.

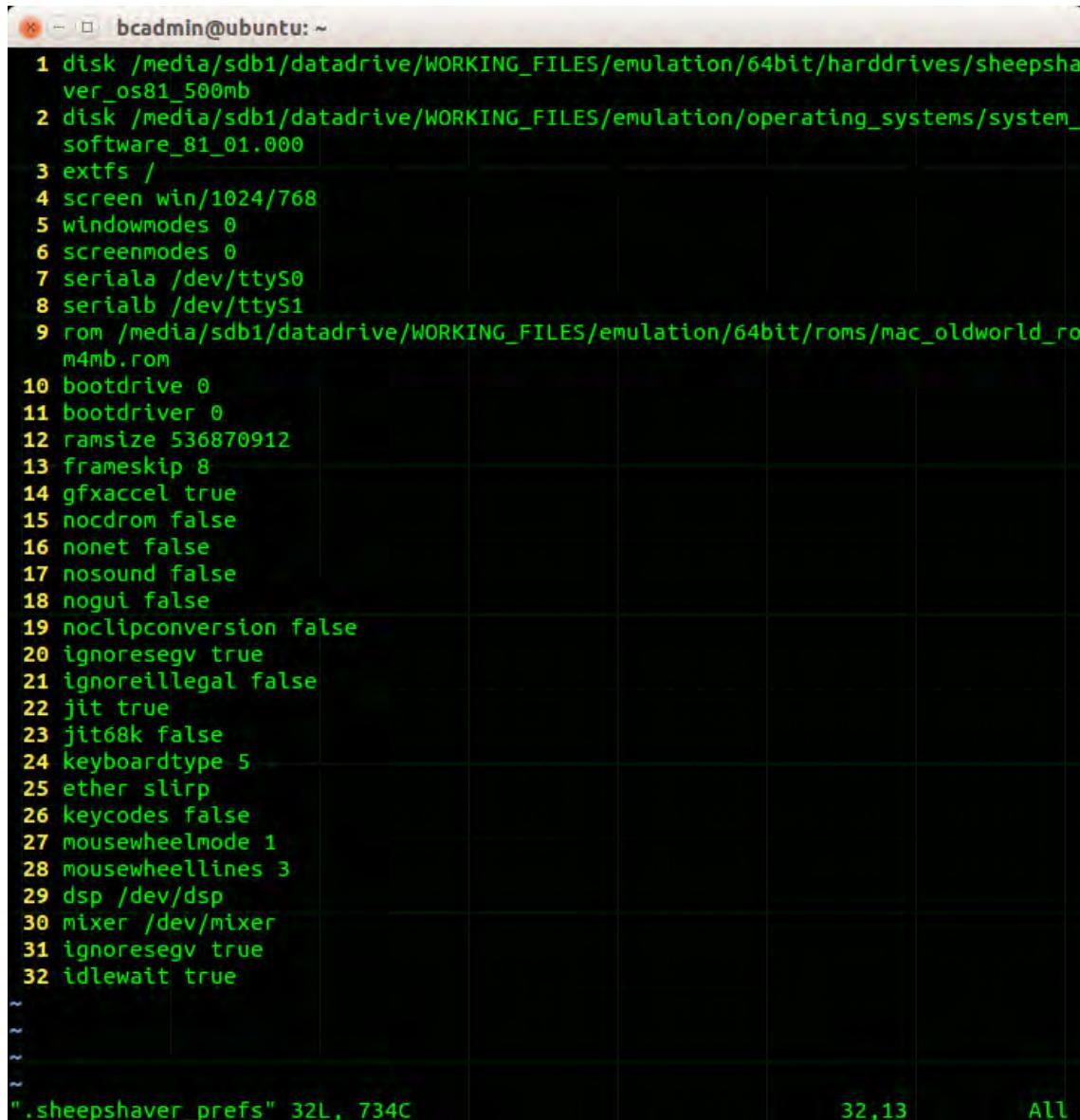
If no valid, readable, or bootable media is available, there will be a disk image with a question mark displayed within the guest system. (Figure B.12)



Figure B.12: No valid, readable, or bootable media in SheepShaver.

In order to load the disk image file for the installation media, it is necessary to load it as a cdrom, rather than disk. For this, it is necessary to manually edit the preference file. The file located in the currnet user's home directory. For SheepShaver, the file is `~/.sheepshaver_prefs`. (The file for Basilisk II is `~/.basilisk_ii_prefs`.) (Figure B.13)

In line two, the installation media is referenced by the word cdrom and then the full path of the disk image file. This setting ensures that the disk image is seen as media in the CD drive, rather than another hard disk on the system. Save the preferences file, and launch the SheepShaver settings dialog again and start the emulator. Note that the installation media will not be listed in the graphical interface in the list on the Volumes tab if it is referenced using the cdrom prefix.



A screenshot of a terminal window titled "bcadmin@ubuntu: ~". The window displays the contents of the `~/.sheepshaver_prefs` file. The file is a configuration script with numbered lines from 1 to 32. Lines 1 through 32 contain various configuration parameters such as disk paths, screen resolutions, ram sizes, and boot options. The terminal window has a dark background with green text. The status bar at the bottom shows the file name ".sheepshaver\_prefs", its size "32L, 734C", and the current line number "32,13".

```
1 disk /media/sdb1/datadrive/WORKING_FILES/emulation/64bit/harddrives/sheepsha
ver_os81_500mb
2 disk /media/sdb1/datadrive/WORKING_FILES/emulation/operating_systems/system_
software_81_01.000
3 extfs /
4 screen win/1024/768
5 windowmodes 0
6 screenmodes 0
7 seriala /dev/ttyS0
8 serialb /dev/ttyS1
9 rom /media/sdb1/datadrive/WORKING_FILES/emulation/64bit/roms/mac_oldworld_ro
m4mb.rom
10 bootdrive 0
11 bootdriver 0
12 ramsize 536870912
13 frameskip 8
14 gfxaccel true
15 nocdrom false
16 nonet false
17 nosound false
18 nogui false
19 noclipconversion false
20 ignoresegv true
21 ignoreillegal false
22 jit true
23 jit68k false
24 keyboardtype 5
25 ether slrp
26 keycodes false
27 mousewheelmode 1
28 mousewheellines 3
29 dsp /dev/dsp
30 mixer /dev/mixer
31 ignoresegv true
32 idlewait true
~
~
~
~
```

".sheepshaver\_prefs" 32L, 734C 32,13 All

Figure B.13: SheepShaver Preferences file in terminal window.



Figure B.14: Improper system software media format in SheepShaver.

Note that if the error message, “The system software on the startup disk only functions on the original media, not if copied to another drive” Figure B.14 appears, this means that the disk image file was added using the graphical interface. It is added as a “disk” and therefore seen as “read/write” and not as the “original” CD.

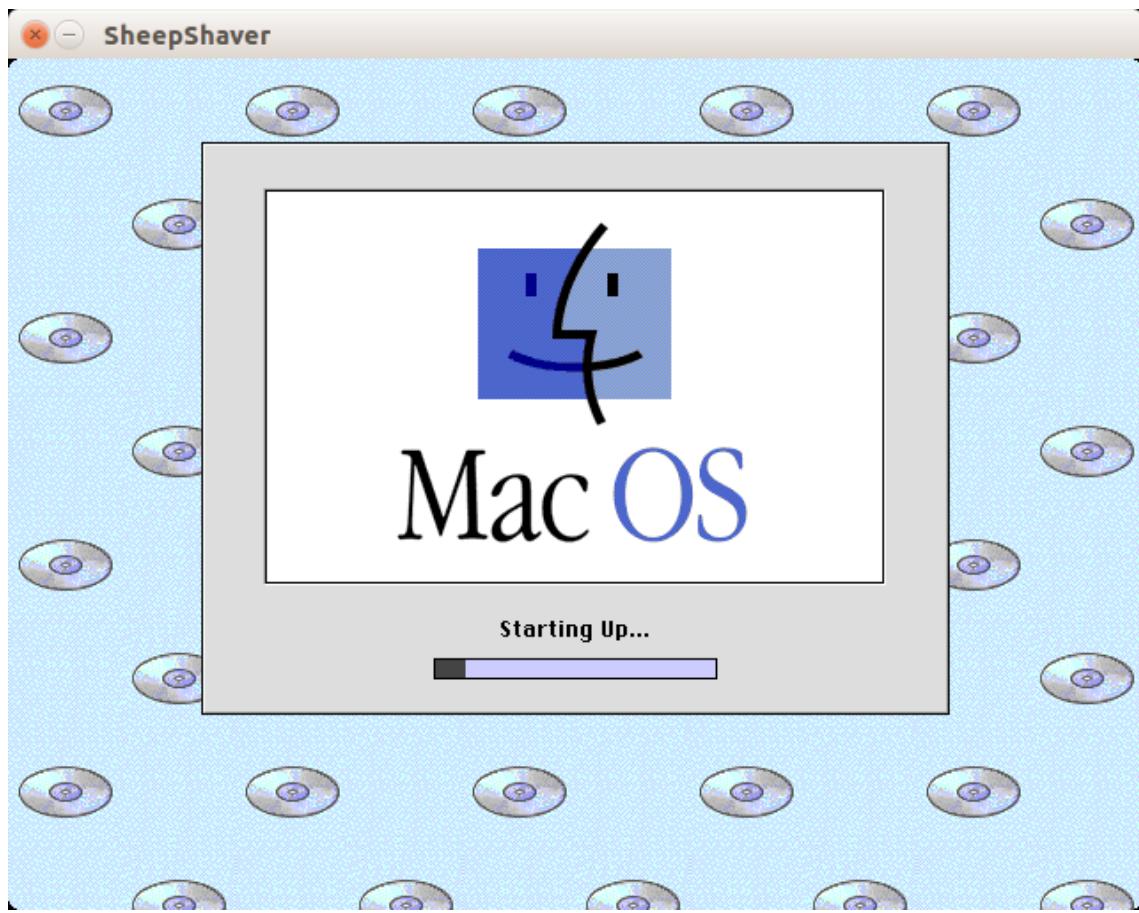


Figure B.15: Initial startup screen for installation of Mac OS 8.1 on SheepShaver.

If the emulator successfully recognizes the operating system as being on its “original media” the operating system initial screen will appear. (Figure B.15)

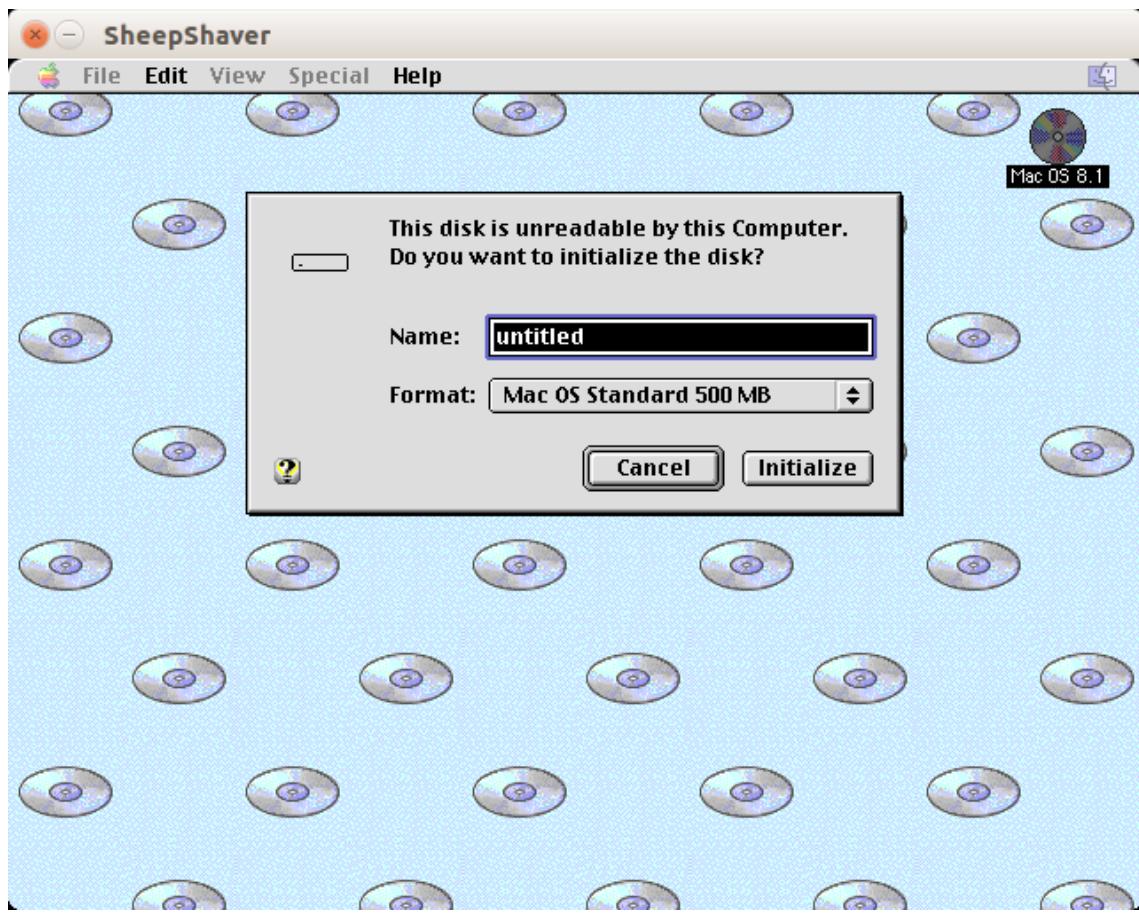


Figure B.16: Initializing disk image file for Mac OS 8.1.

The disk image file will initially appear as “unreadable” and the first dialog should ask if you want to initialize the disk. (Figure B.16)

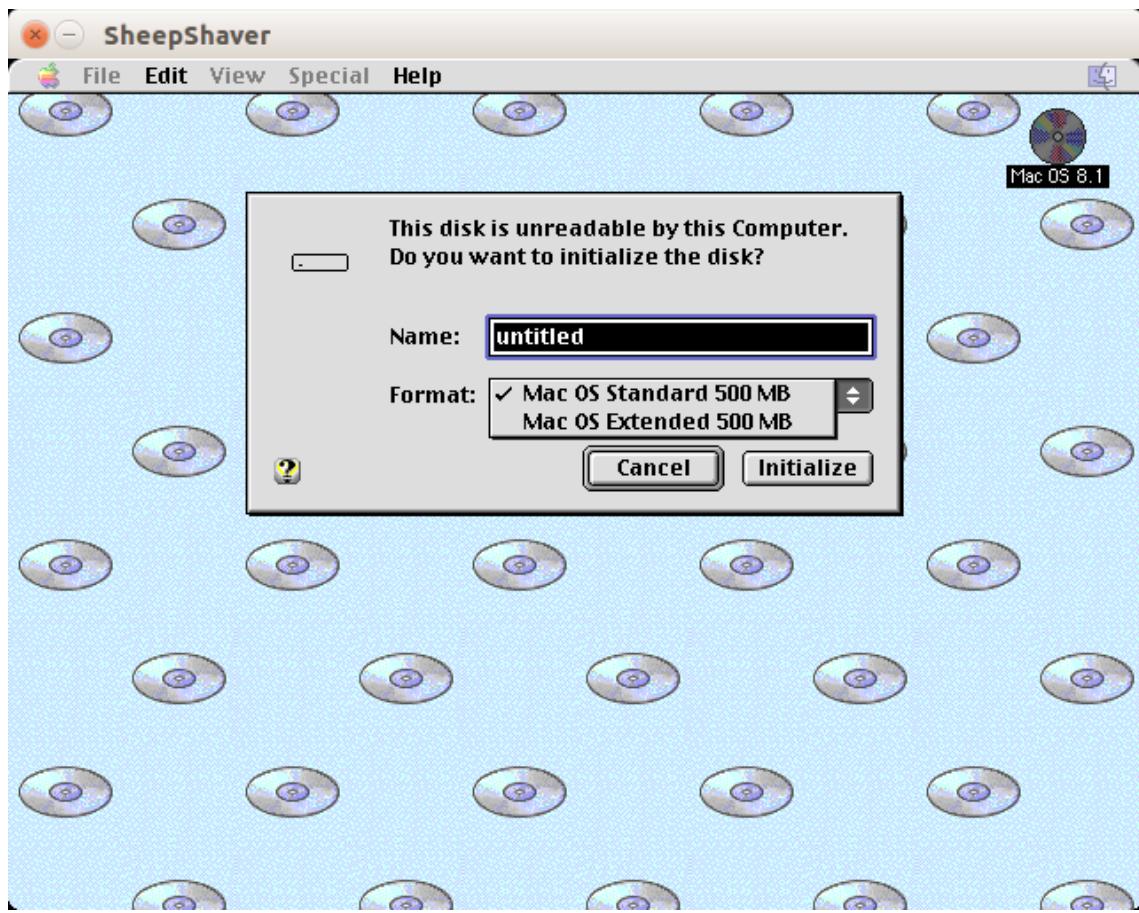


Figure B.17: Selection of files system for initializing disk on Mac OS 8.1.

Note that MacOS may provide multiple options here. **Mac OS Standard** refers to an HFS formatted disk, while **Mac OS Extended** refers to an HFS+ formatted disk [5, 6]. The project team found **Mac OS Standard** functioned for accessing artworks and is preferred here. Select a name for the volume and click **Initialize**. (Figure B.17)

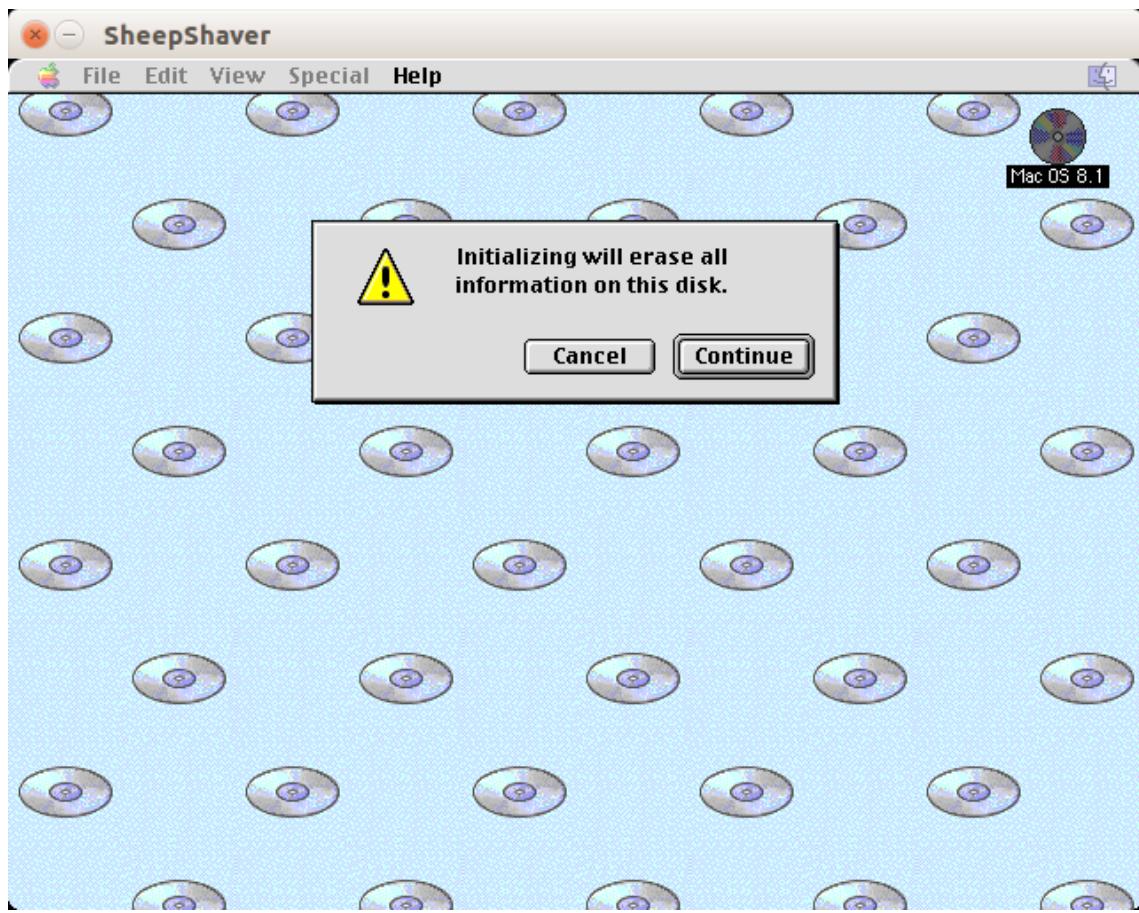


Figure B.18: Confirming initialization of disk on SheepShaver.

Click **Continue** to proceed initializing the disk. (Figure B.18)

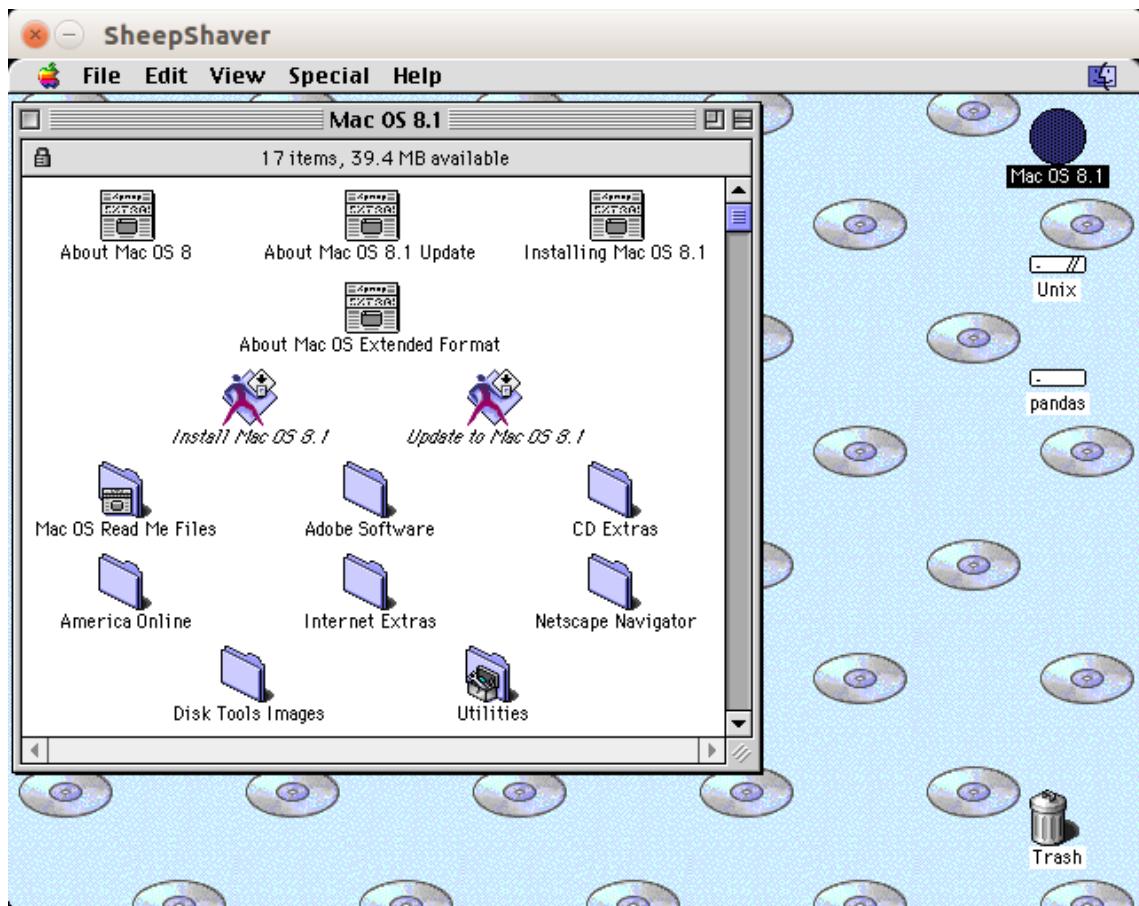


Figure B.19: Mac OS 8.1 disc installation contents.

After disk initialization, the contents of the installation media may be displayed. (It might be necessary to double-click on the icon on the desktop for this to appear.) Select **Install Mac OS 8.1** (or the appropriate icon for the installation). (Figure B.19)

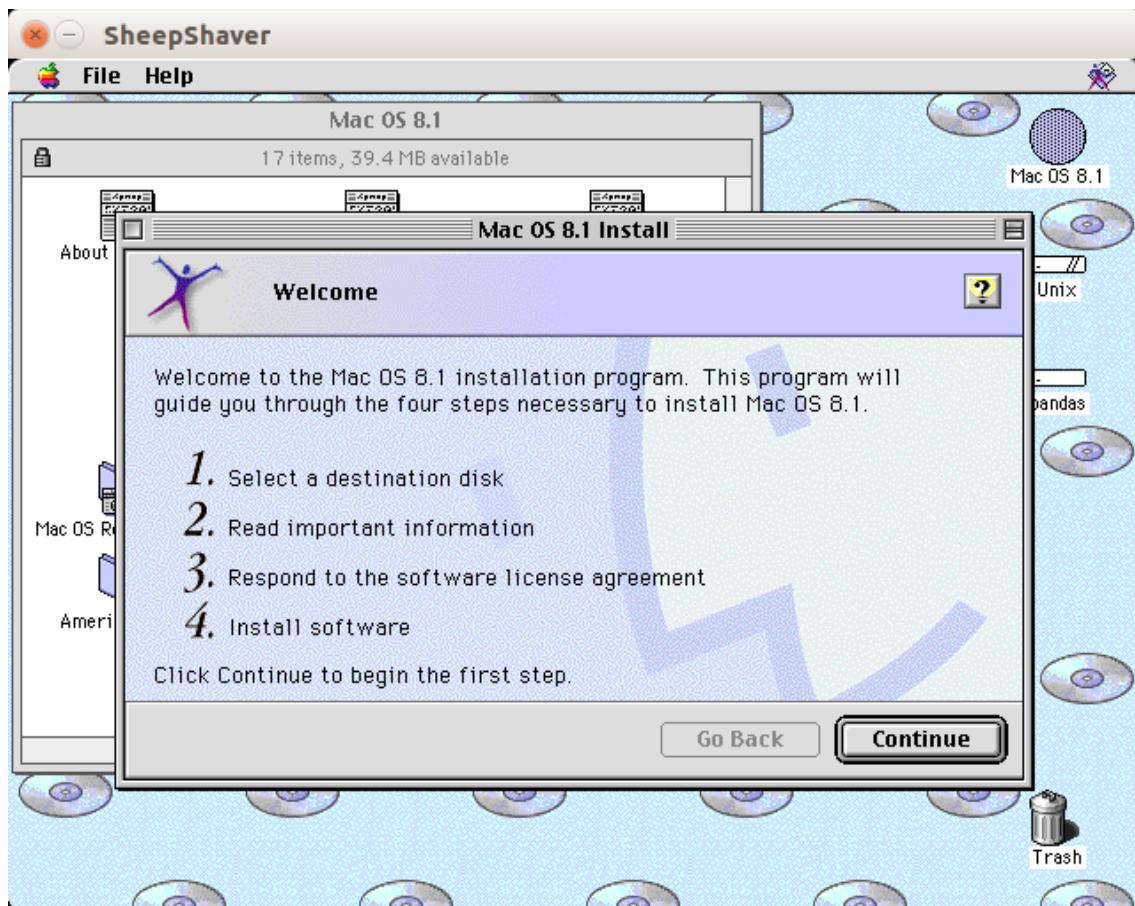


Figure B.20: Confirming install for Mac OS 8.1.

Click **Continue** to proceed with the installation. B.20 The **Destination Disk** should be selected to the disk you initialized in the previous steps. Click **Select** to continue. (Figure B.21)

Continue to click through the prompts during the installation process. (Figures B.22, B.23, B.24, B.25, B.26, B.27, B.28, B.29)

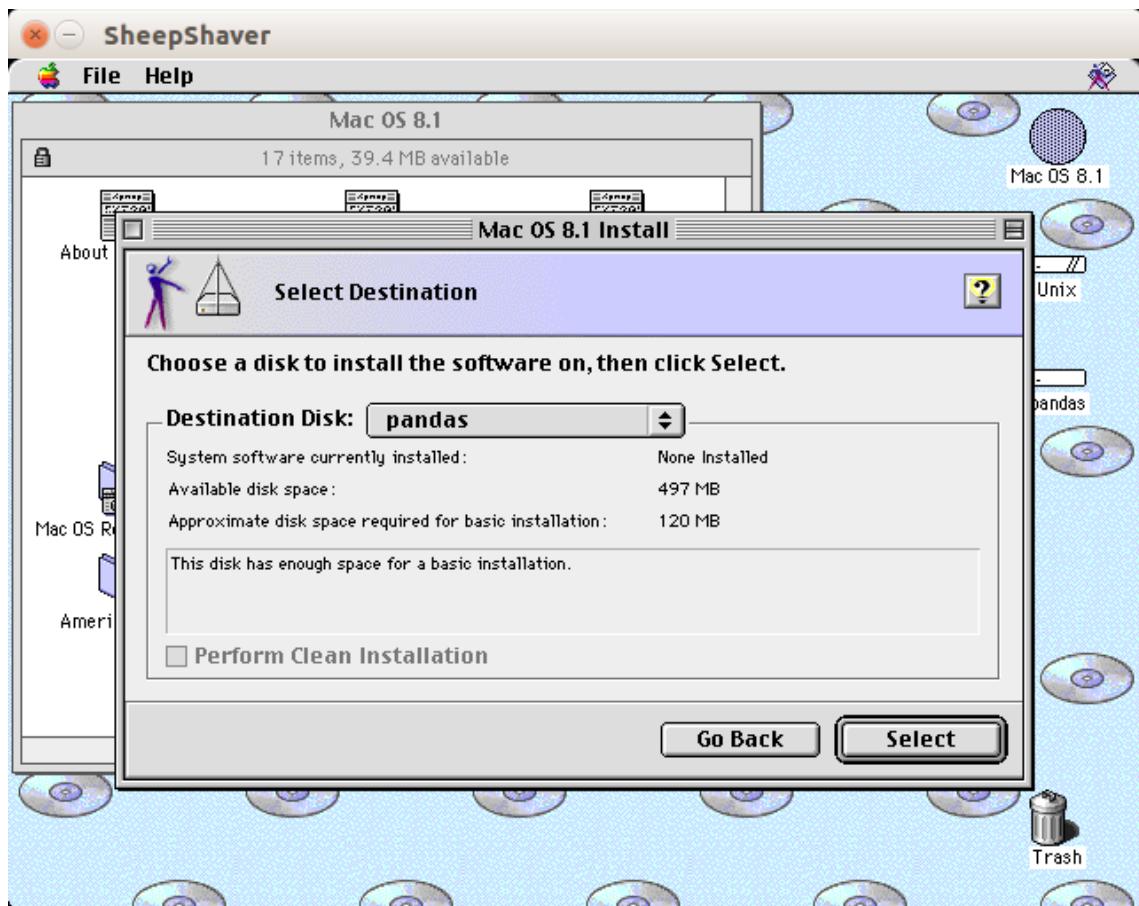


Figure B.21: Destination disk for installation of Mac OS 8.1.



Figure B.22: Important Information dialog in Mac OS 8.1 installation.

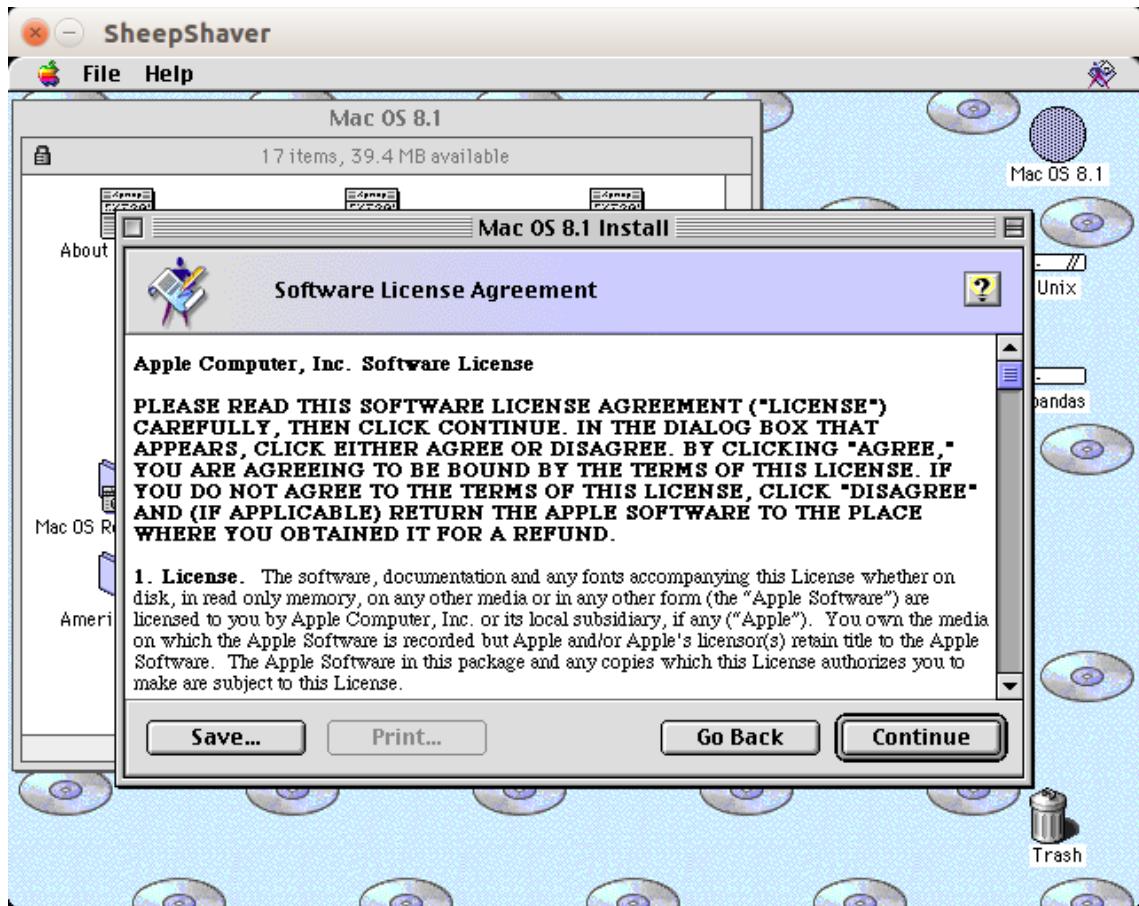


Figure B.23: Important Information dialog in Mac OS 8.1 installation.

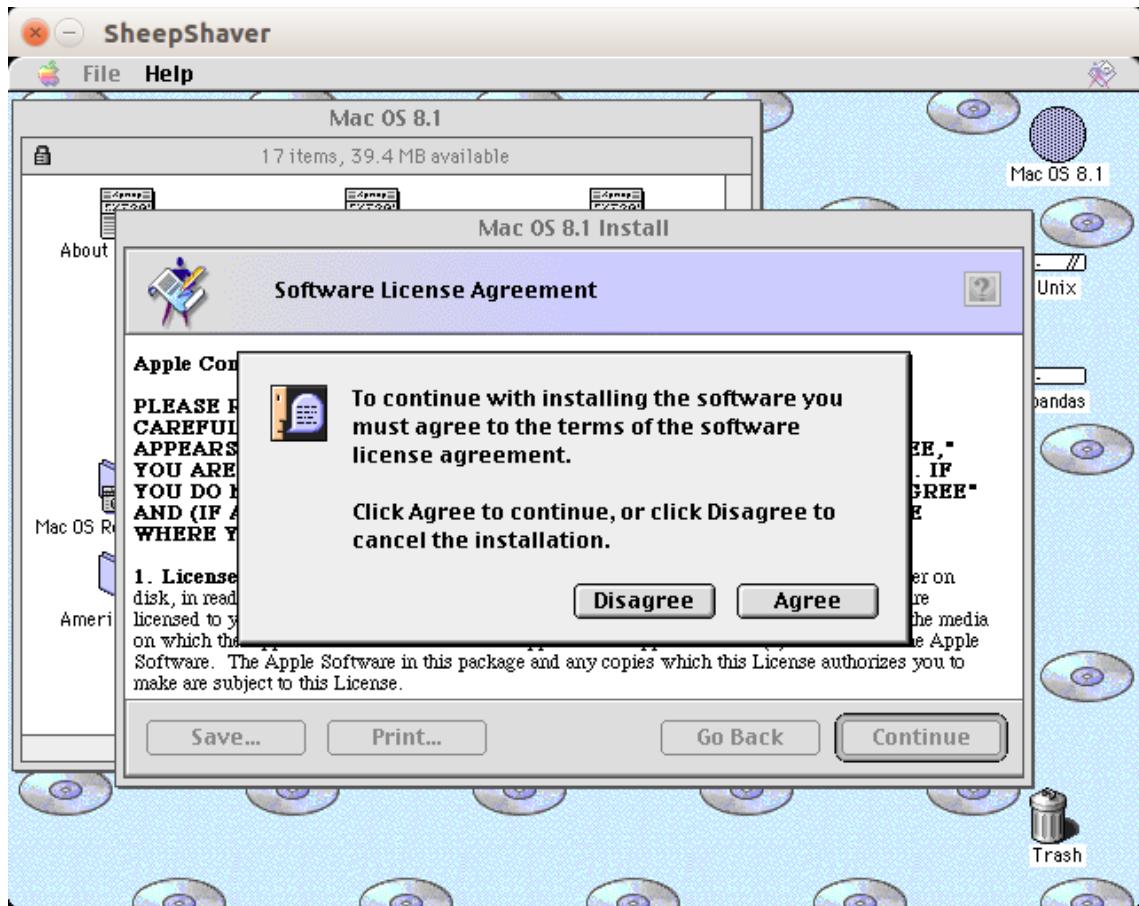


Figure B.24: Agreeing to Software License Agreement in Mac OS 8.1 installation.

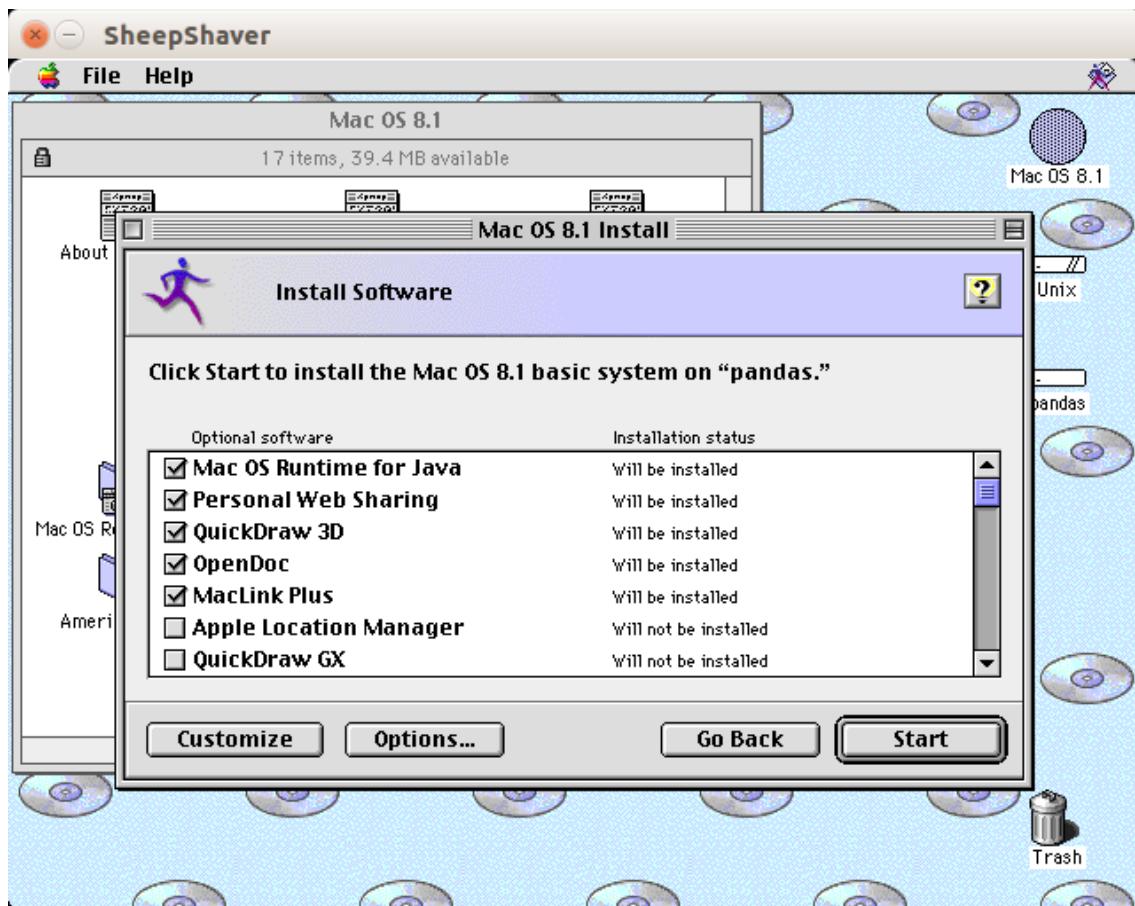


Figure B.25: Selection of components to install in Mac OS 8.1.

The default subcomponents were typically sufficient for the project team’s needs. (Figure B.25)

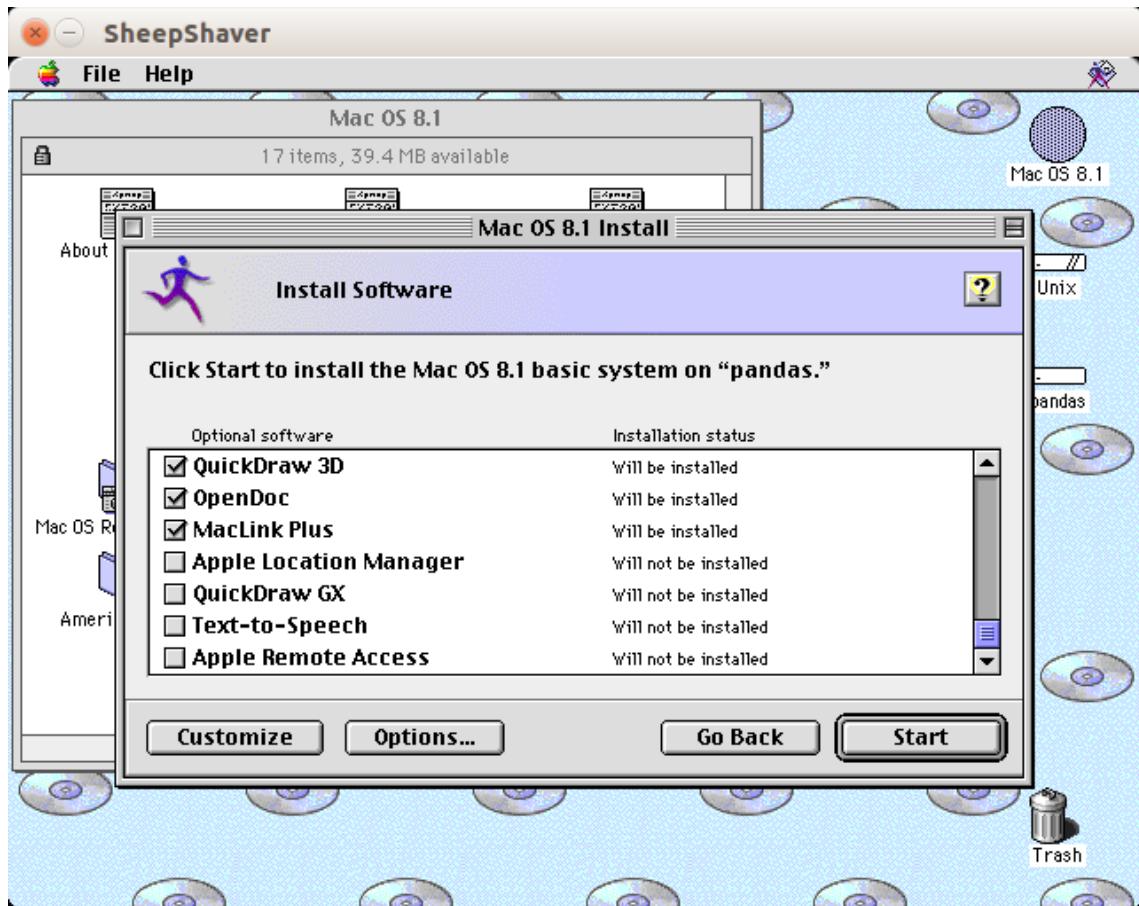


Figure B.26: Installation of basic system for Mac OS 8.1.

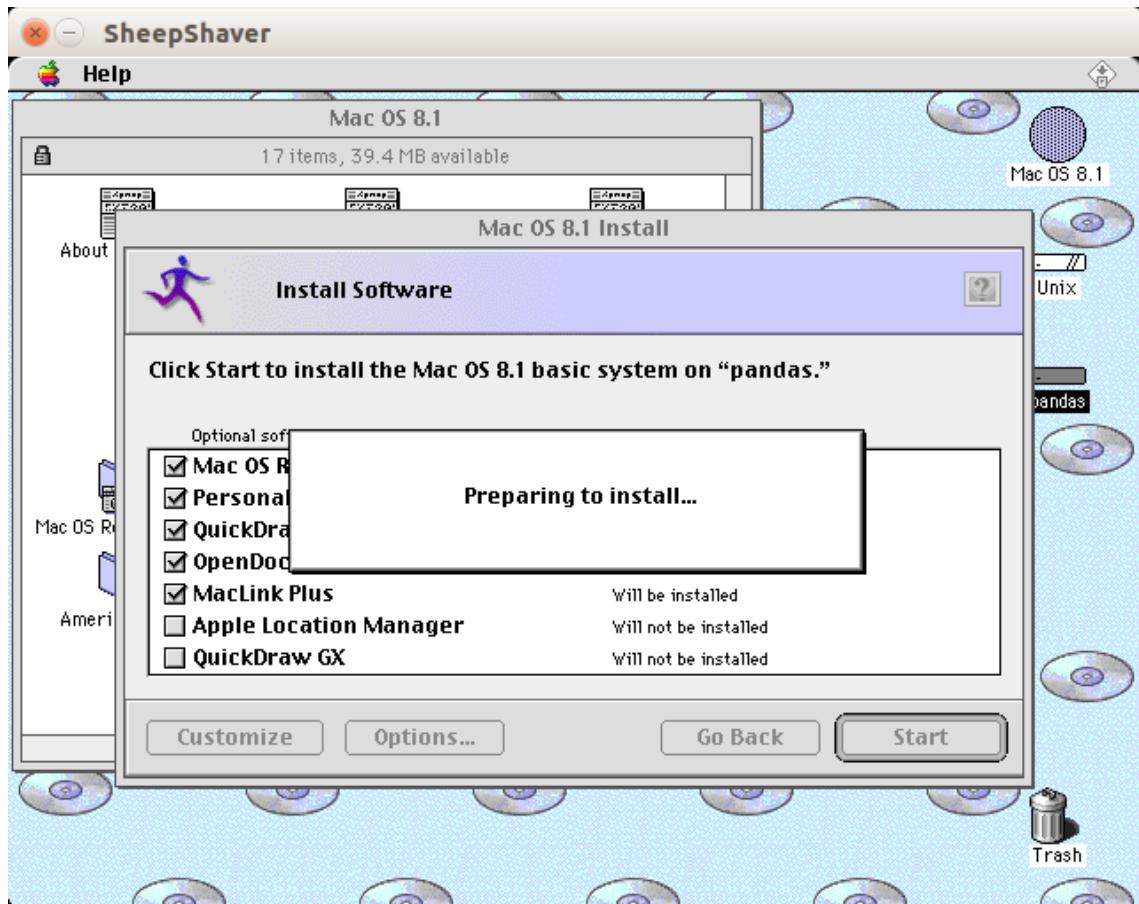


Figure B.27: Preparing to install Mac OS 8.1.

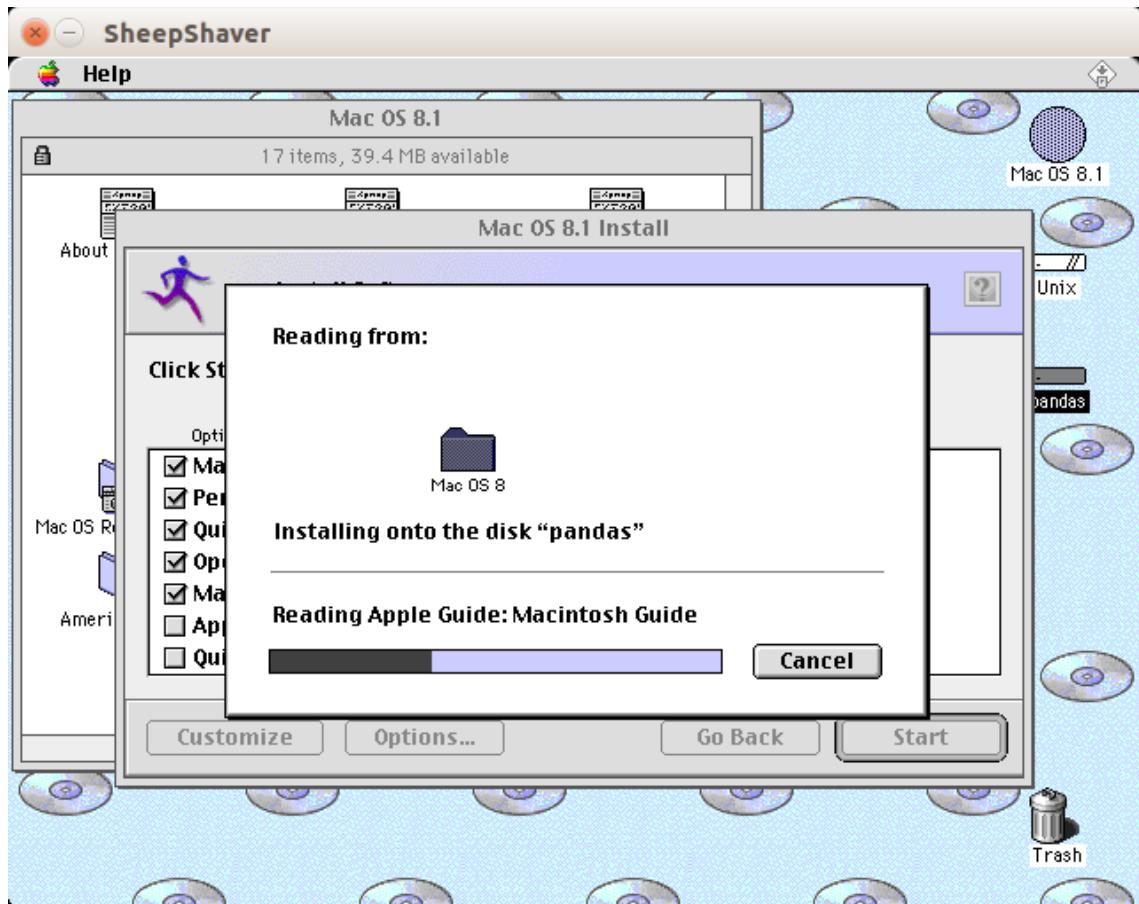


Figure B.28: Installation progress for Mac OS 8.1.

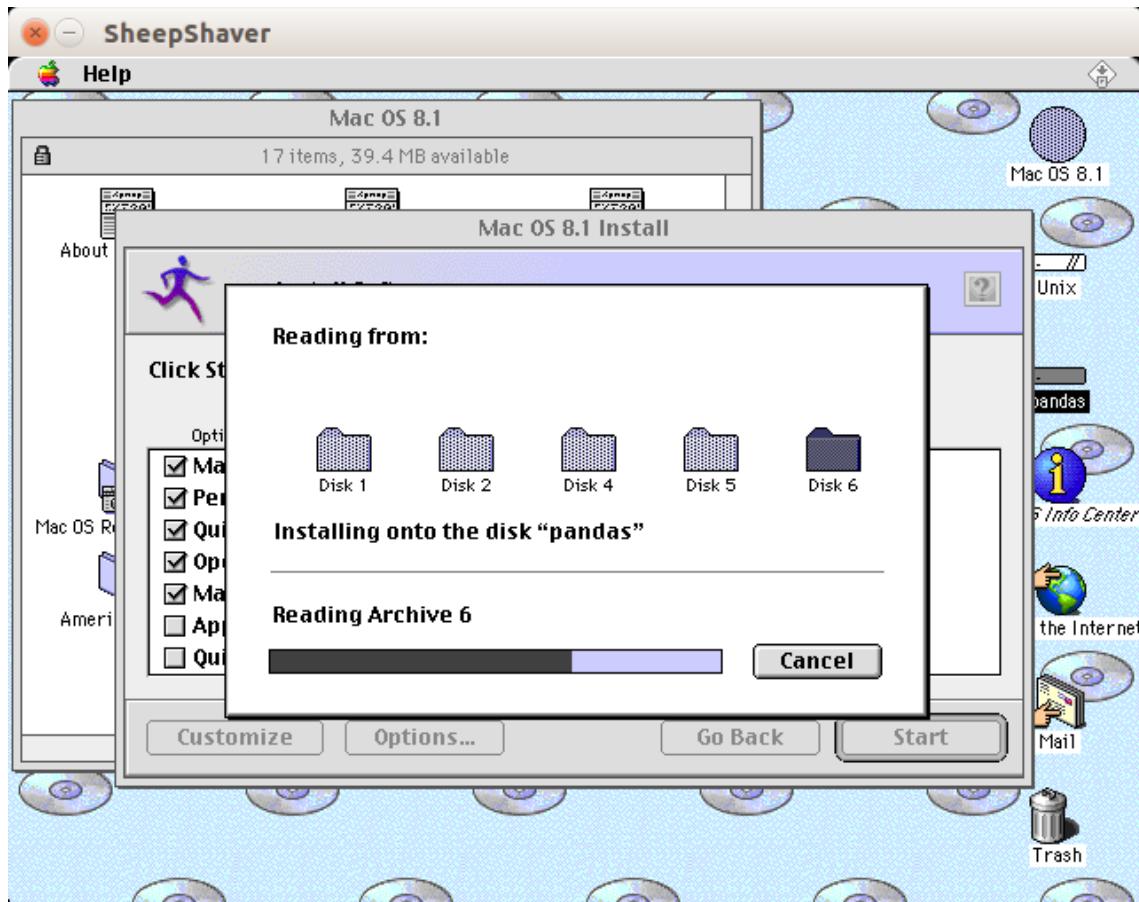


Figure B.29: Continued installation progress for Mac OS 8.1.

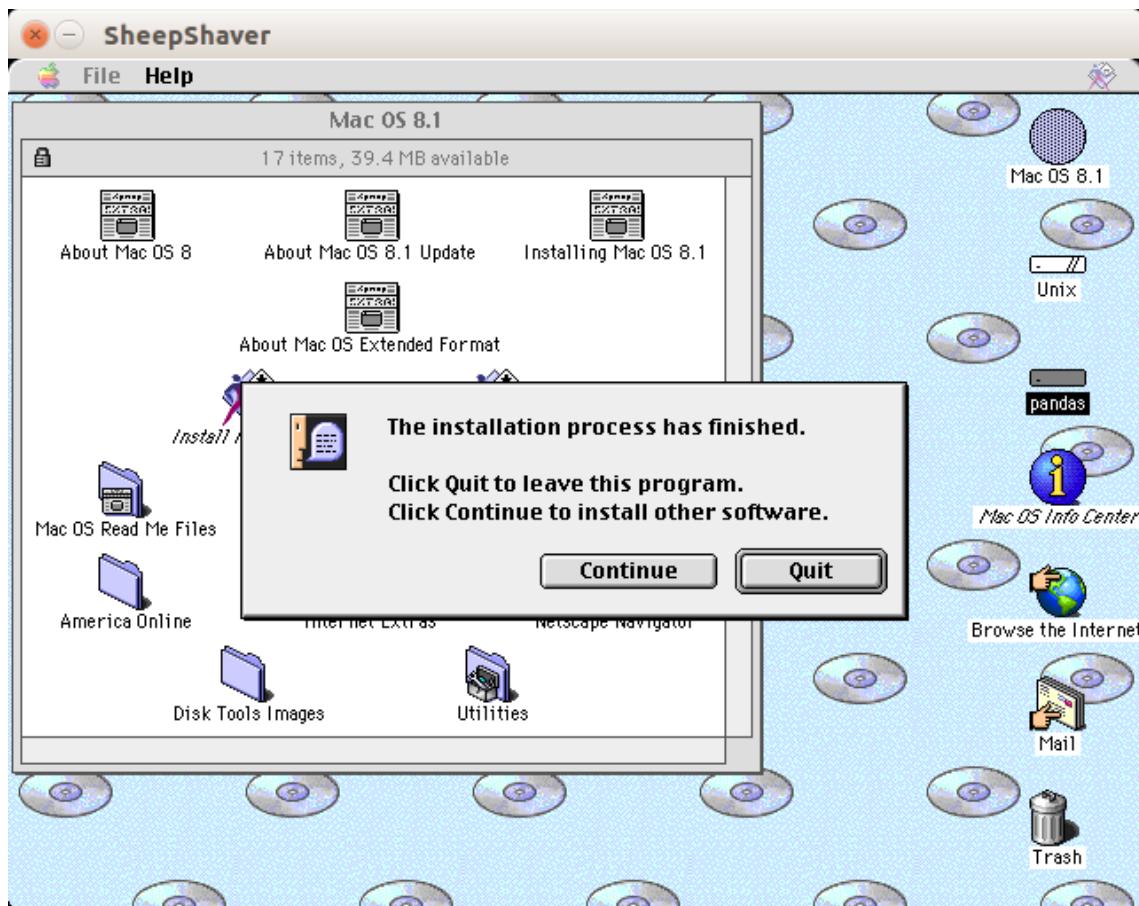


Figure B.30: Confirmation installation process has finished for Mac OS 8.1.

When the installation has finished, press **Quit** to finish. If there are other software components to be configured or installed here, press **Continue** to proceed. (Figure B.30)

Shut down the system by selecting **Special** from the menu and then **Shut Down** from the options listed. (Figure B.31)

To restart the emulator without the installation media, it is necessary to remove it from the list in **Volumes**. This may be done by editing the preferences file manually and removing the line that references the disk image for the installation CD.

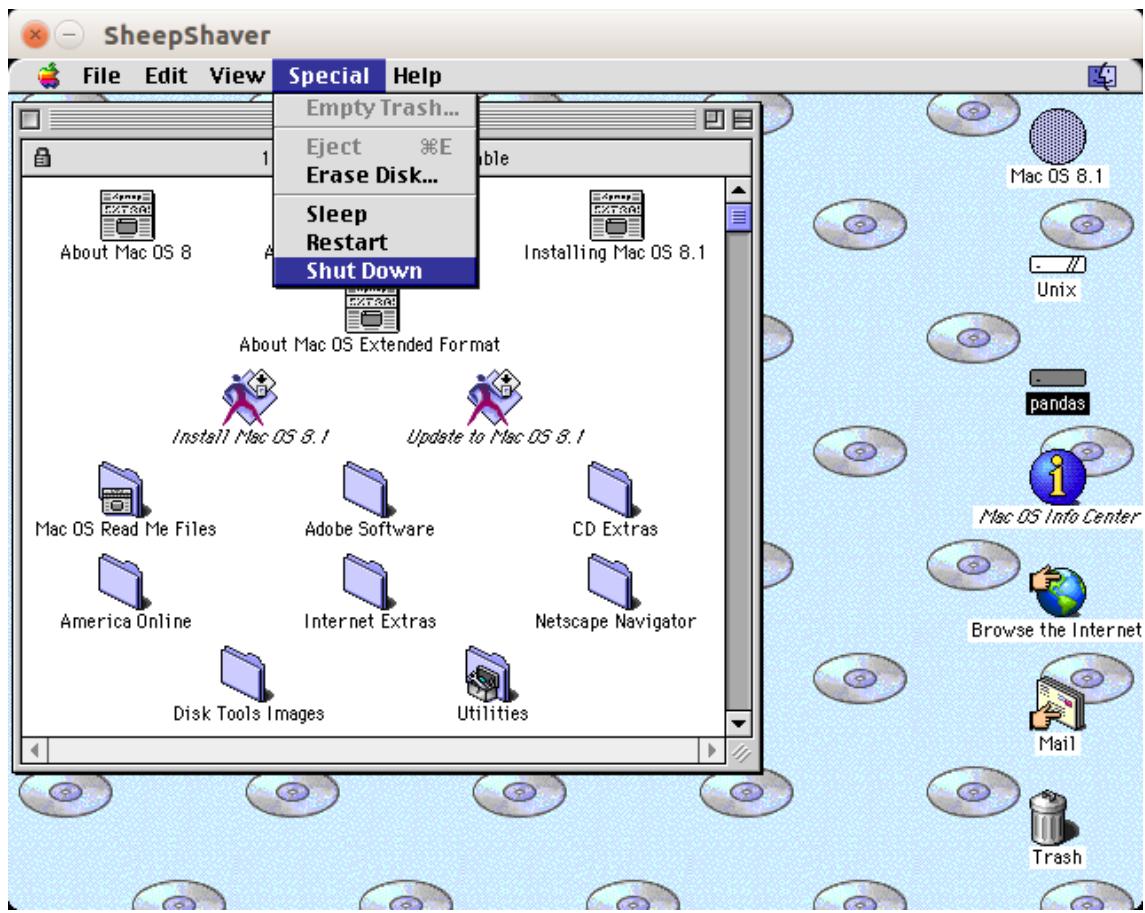


Figure B.31: Shutting down system after installation of Mac OS 8.1.

## References

- [1] *Basilisk II* - Wikipedia, the free encyclopedia. URL: [http://en.wikipedia.org/w/index.php?title=Basilisk\\_II&oldid=594815375](http://en.wikipedia.org/w/index.php?title=Basilisk_II&oldid=594815375).
- [2] *SheepShaver* - Wikipedia, the free encyclopedia. URL: <http://en.wikipedia.org/w/index.php?title=SheepShaver&oldid=612878953>.
- [3] *Mac OS 8 and 8.1: Maximum Size, Maximum Convenience — Low End Mac*. URL: <http://lowendmac.com/2014/mac-os-8-and-8-1-maximum-size-maximum-convenience/>.
- [4] *Mac OS 9: System Requirements - Apple Support*. URL: <http://web.archive.org/web/20141216215425/http://support.apple.com/en-us/HT2299>.
- [5] *Hierarchical File System* - Wikipedia, the free encyclopedia. URL: [http://en.wikipedia.org/w/index.php?title=Hierarchical\\_File\\_System&oldid=594424332](http://en.wikipedia.org/w/index.php?title=Hierarchical_File_System&oldid=594424332).
- [6] *Mac OS: About Mac OS Extended Volume Hard Drive Format or HFS+ - Apple Support*. URL: <http://web.archive.org/web/20141113142812/http://support.apple.com/en-us/HT1604>.

## C — Installation of QEMU

The QEMU emulator suite is open-source and can be compiled for a variety of architectures. The project team used version 2.0.0<sup>a</sup> provided in the Ubuntu 14.04 repository; this seemed to work well enough that it did not seem necessary to compile the application from source. The project team also installed qemuctl, which provides Qemu Launcher, a graphical interface for configuring options and running qemu.

---

<sup>a</sup>QEMU emulator version 2.0.0 (Debian 2.0.0+dfsg-2ubuntu1.9), Copyright (c) 2003-2008 Fabrice Bellard

## D — Creating a disk image file for the hard disk for QEMU from scratch

In the event that the disk image files with installed operating systems become unusable or the team acquires an additional retail copy of a Windows operating system, it is rather straightforward to create a new disk image file and install an operating system on it.

If not already available, make a disk image of the original retail installation CD for the desired operating system. Any disk imaging program should be adequate for this task; the project team used Guymager to create images of the retail CDs it acquired. Next, the project team recommends making a copy of the disk image and storing it in a separate folder, along with a record of the checksum for the disk image of the installation media. Installation can proceed using one of the copies of the disk image file. In the event that the installation media disk image file is corrupted, it is straightforward to make a new copy (rather than image the media again) and confirm that an exact copy of the original media is used for the installation.

The following screenshots show, in detail, the steps involved for installing Windows 2000 using QEMU to provide one example of the process.

Keep the default settings when launching Qemu Launcher for the first time. Click **New** next to **Hard disk 0**. (Figure D.1)

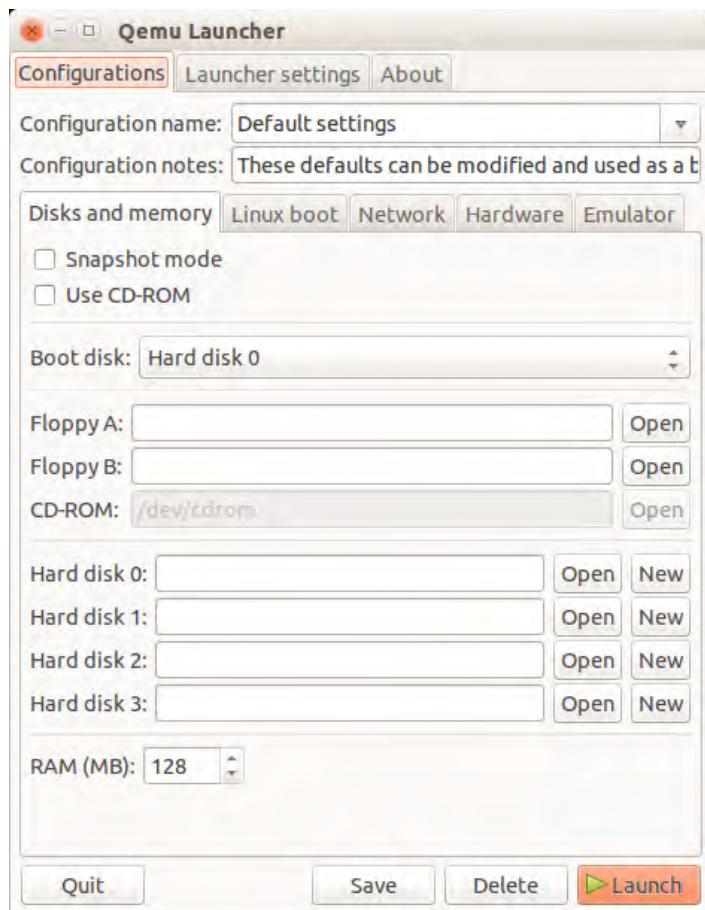


Figure D.1: Configurations tab and Disks and memory subtab in Qemu Launcher.

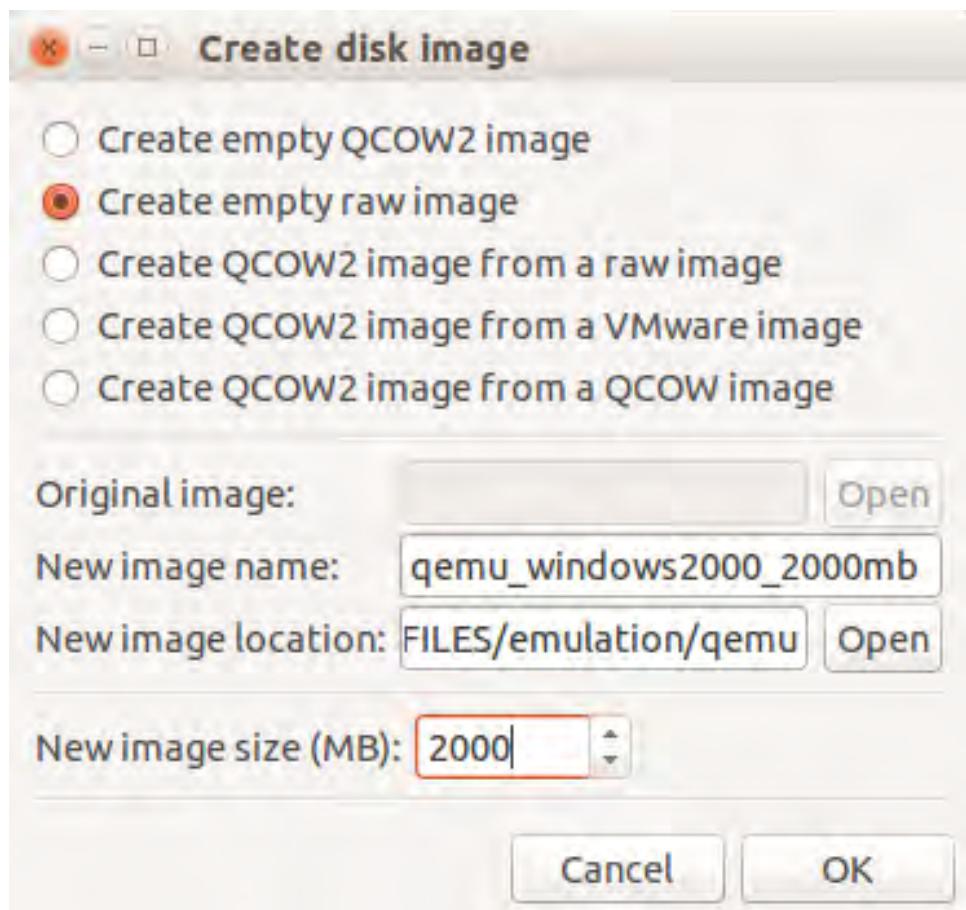


Figure D.2: Create disk image dialog in Qemu Launcher.

**Create empty raw image** will produce an image in a form that can be mounted in other emulators. Enter in the name – the project team recommends the following naming convention for the file itself: the name of the emulator, followed by an underscore; the name and version of the operating system; the size and units of the file, followed by an underscore and select an image size. The project team used 2GB (2000 MB), as it is specified in the system requirements for Windows 2000 [1]. (Figure D.2)

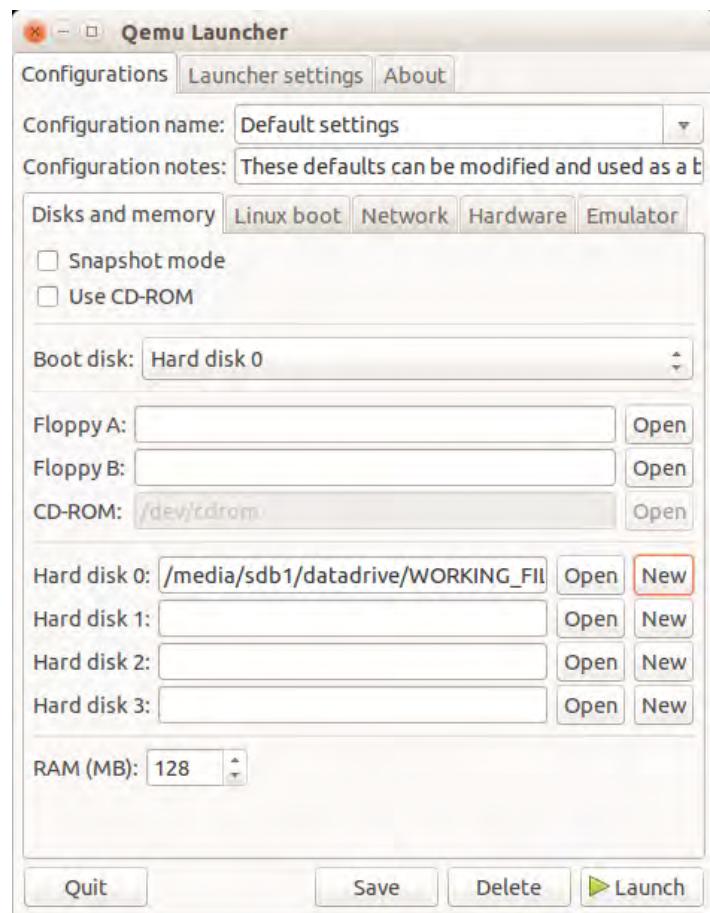


Figure D.3: Configurations tab and Disks and memory subtab in Qemu Launcher with new disk image file loaded.

Once created, the name of the new **Hard disk 0** will be entered into the corresponding field. (Figure D.3)

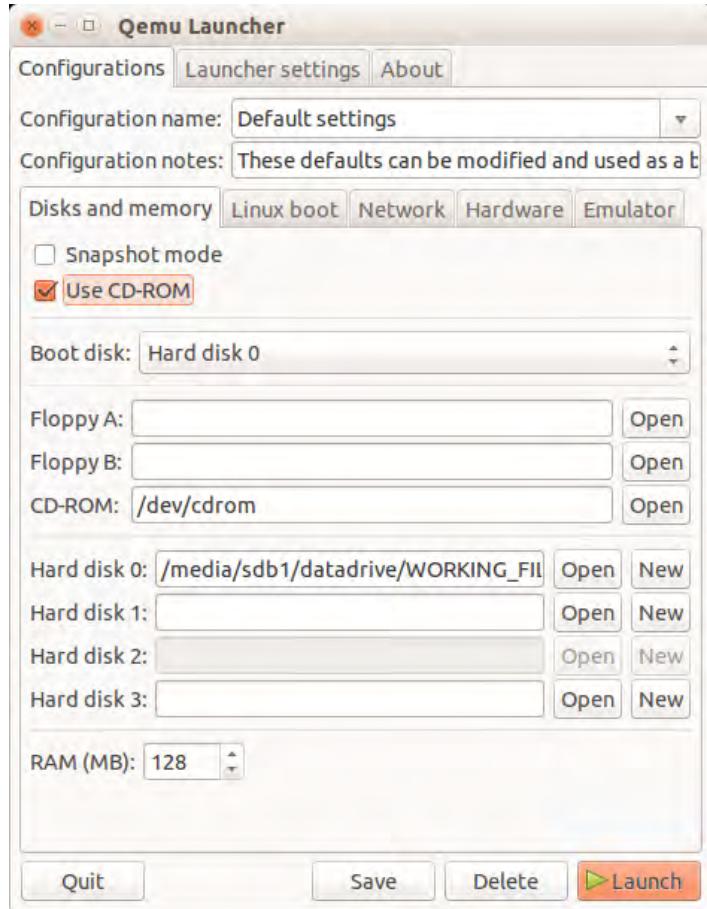


Figure D.4: Selecting Use CD-ROM in Qemu Launcher.

Select **Use CD-ROM**; the line for CD-ROM will no longer be greyed out. The default will be to use a physical CD-ROM in the host system, indicated by `/dev/cdrom`. Click on **Open** to select a disk image file for the operating system to load into the emulator. (Figures D.4, D.5)

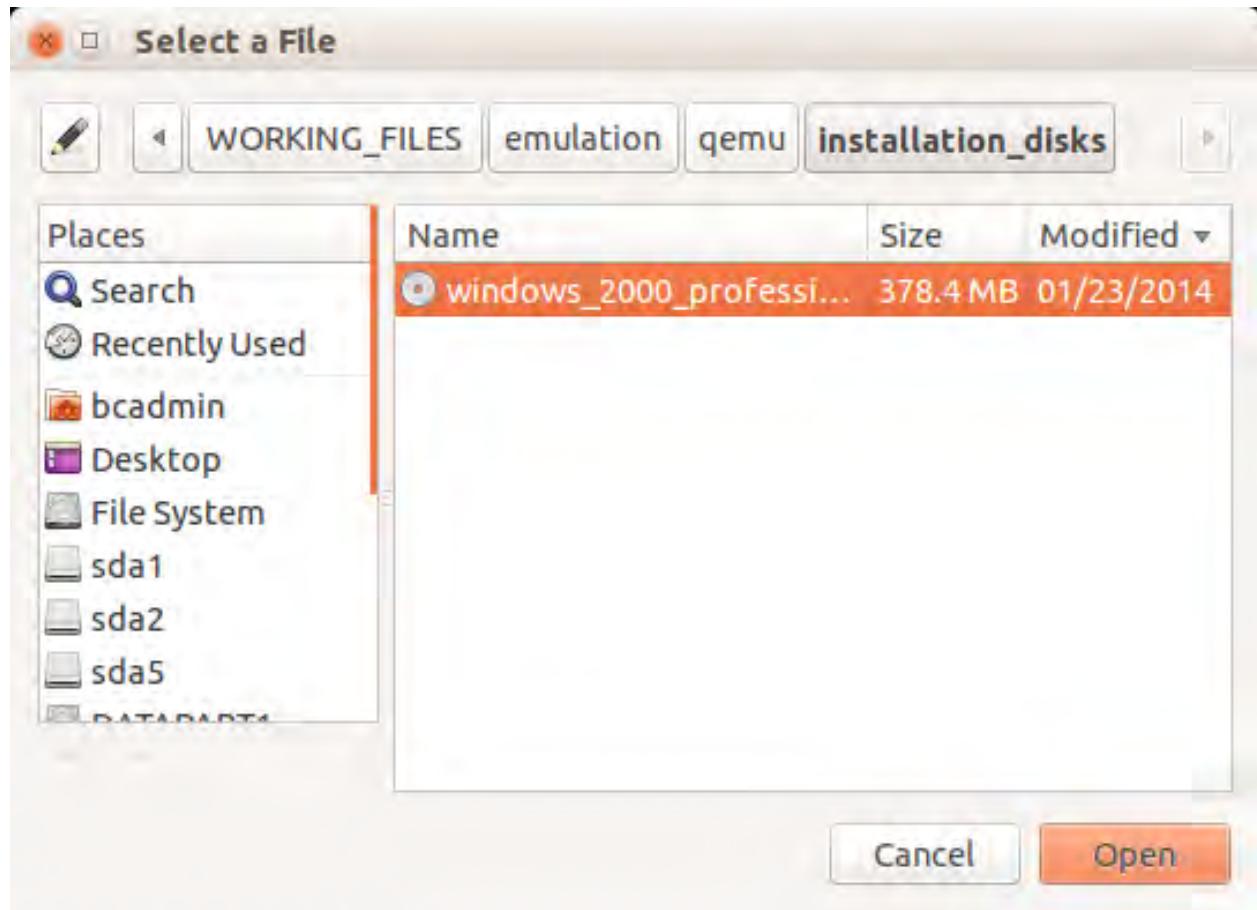


Figure D.5: Selecting a File dialog in Qemu Launcher.

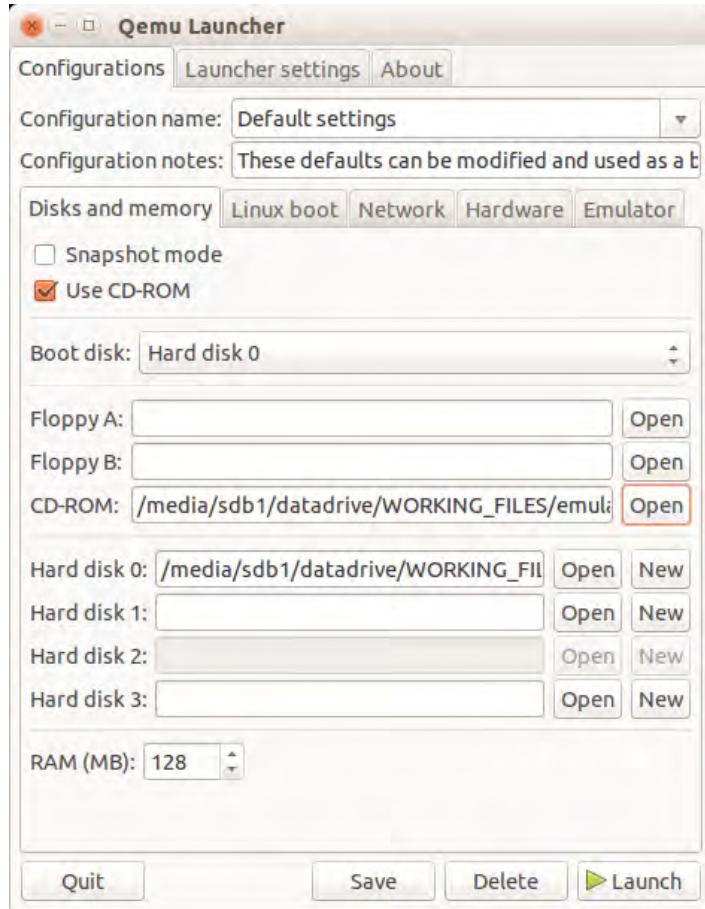


Figure D.6: Configurations tab and Disks and memory subtab with CD-ROM disk image file and hard disk image file loaded.

Once the the disk image for the installation media is set and the new hard disk file is created, the default settings in Qemu Launcher should be sufficient. The following screenshots will quickly review these settings. On the **Configurations** tab and **Disks and Memory** subtab, set the **RAM (MB)** to an appropriate size for the operating system. In this example, 128 MB exceeded the minimum requirements for Windows 2000 [1].

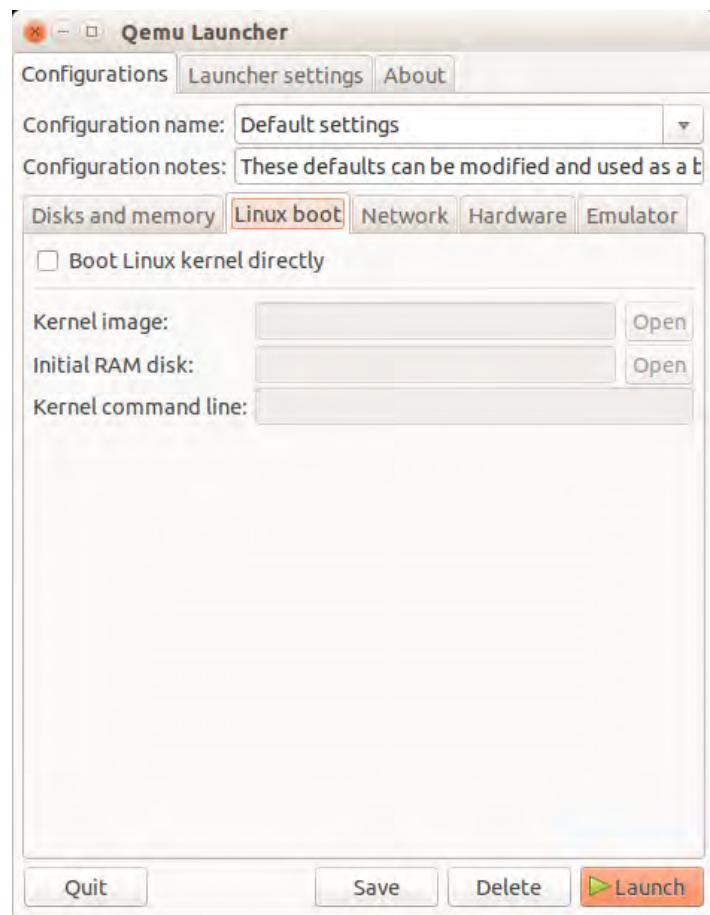


Figure D.7: Configurations tab and Linux boot subtab on Qemu Launcher.

The defaults on the **Linux boot** subtab do not need to be changed. (Figure D.7)

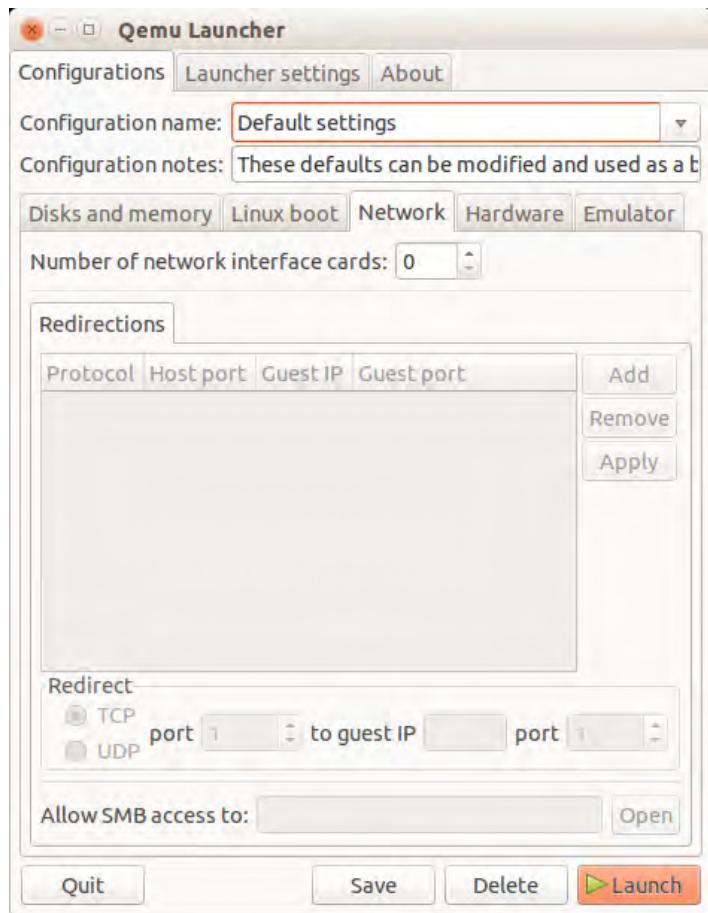


Figure D.8: Configurations tab and Network subtab on Qemu Launcher.

Unless the operating system specifically requires an active internet connection, **Number of network interface cards** can be set to zero here. Since older versions of Windows are no longer actively maintained by Microsoft and these operating systems had a history of vulnerabilities, the project team suggests not enabling the networking features unless absolutely required. (Figure D.8)

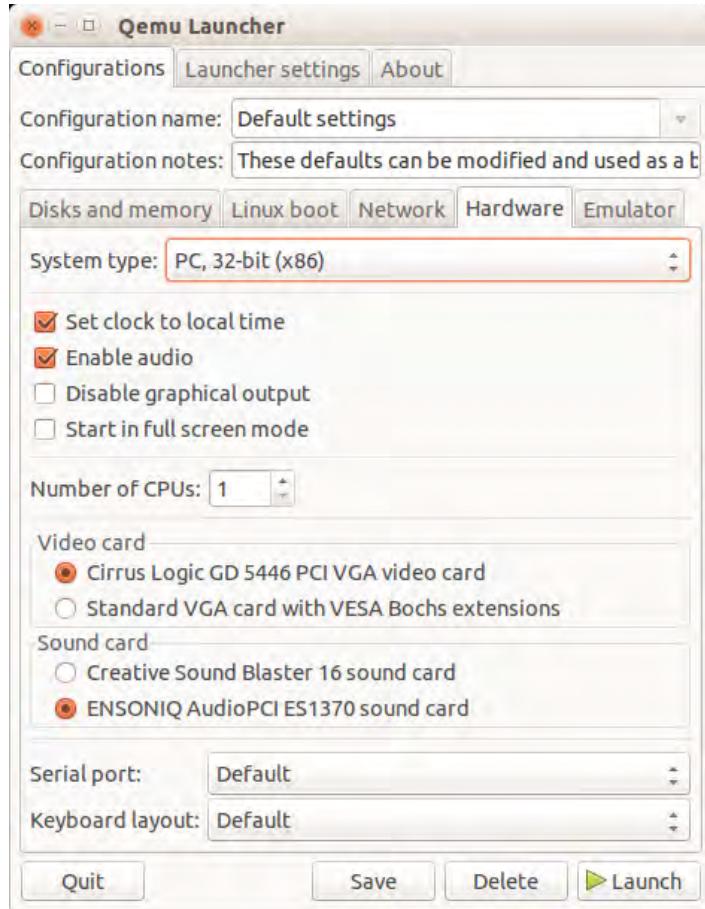


Figure D.9: Configurations tab and Hardware subtab on Qemu Launcher.

While it can be helpful to have audio enabled to test, it may not be required for installing the operating system. This setting selected on this screenshot should be sufficient for installation. The project team encountered some issues with the emulated Sound Blaster setting, and opted not to use it. (Figure D.9)

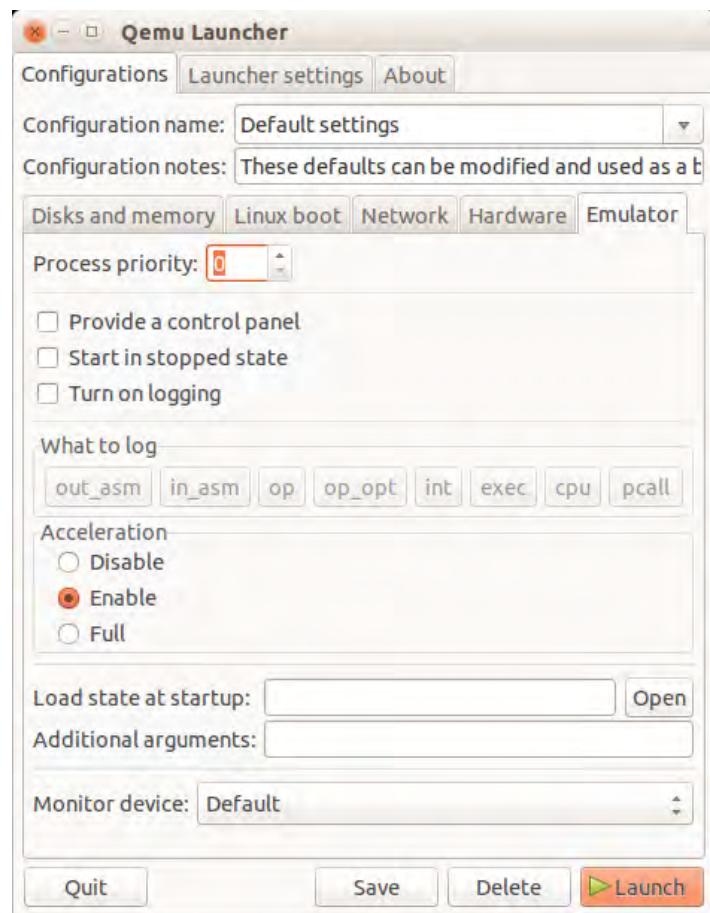


Figure D.10: Configurations tab and Hardware subtab on Qemu Launcher.

The defaults on the **Configurations** tab and **Emulator** subtab should be sufficient for installation. (Figure D.10)

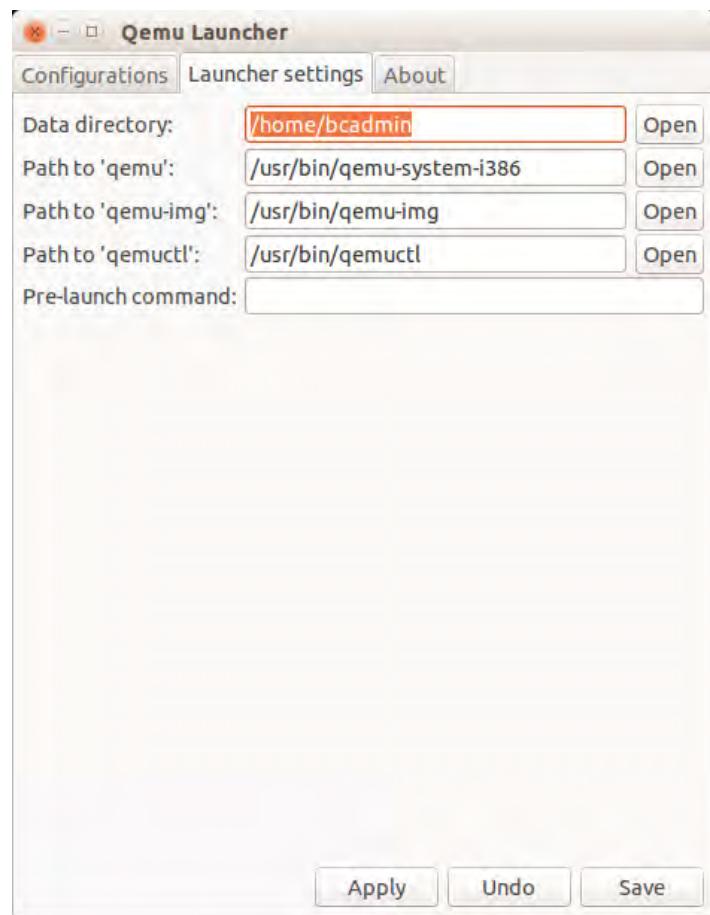


Figure D.11: Launcher settings tab on Qemu Launcher.

Confirm that the **Path to ‘qemu’** is set properly to **qemu-system-i386** before launching.  
(Figure D.11)

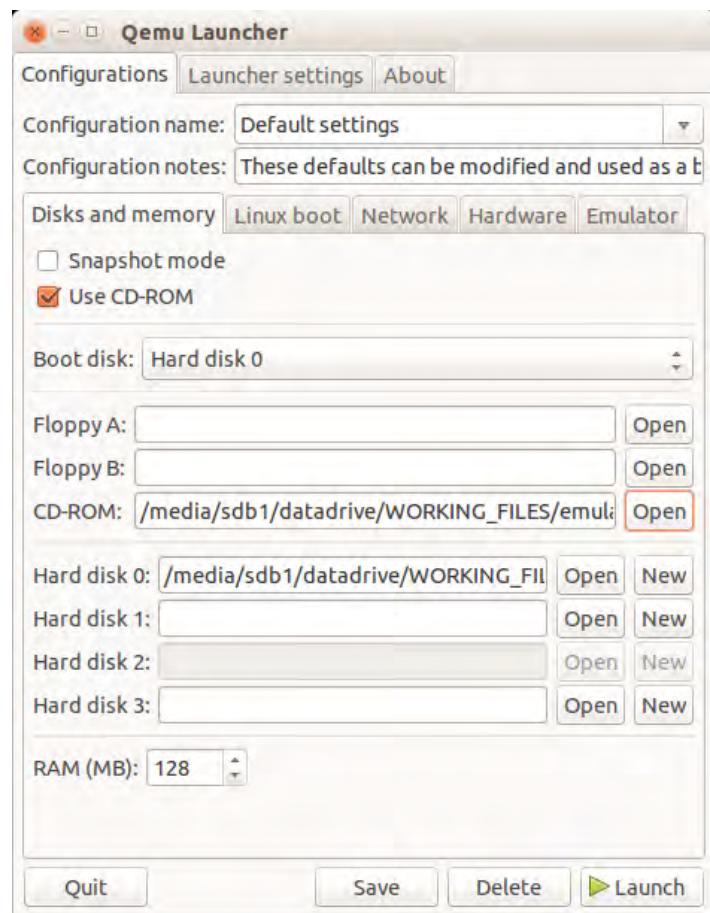


Figure D.12: Configurations tab and Disks and memory subtab with CD-ROM selected, disk image file for CD-ROM chosen, and disk image file for Hard Disk 0 chosen.

Return to the main tab for configurations. If saving these settings is desired, click Save. To launch the QEMU, click Launch. (Figure D.12)

This example so far is of installation of a Windows operating system where there is only disk image file for the installation media. In the event that there are multiple disk image files (representing multiple floppy disks or CD-ROMS), it will be necessary to use the Monitor [2] to swap out disk image files.

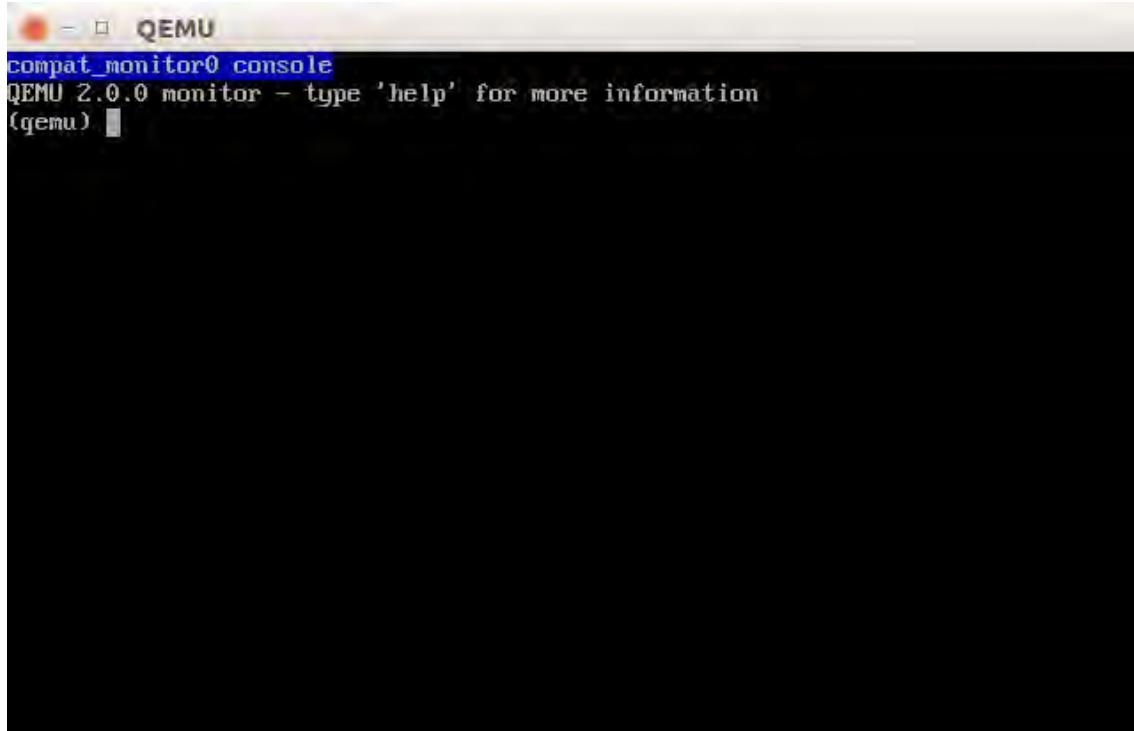
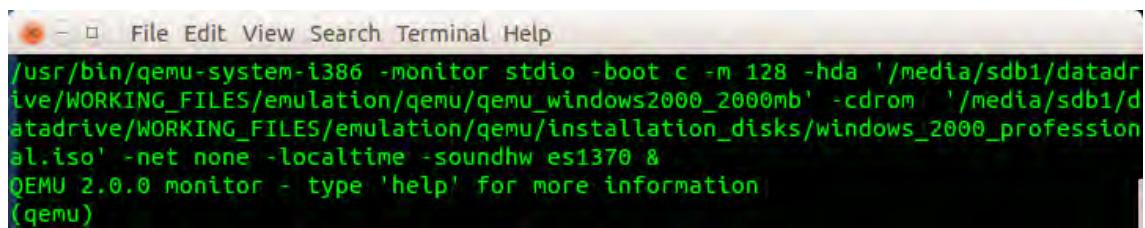


Figure D.13: QEMU Monitor on console.

To enter into the monitor console, while the guest system is running, press **Ctrl-Alt-Shift-2**. This will change the screen from the guest operating system to a console window. To swap out a CD-ROM, use the **info block** command to determine the name of the device corresponding to the installation media, and then the **change** command to specify a new file to load. A screenshot of the default interface is shown below. Directions are found by typing “help” at the prompt or by consulting the Wikibooks manual for QEMU/Monitor [3]. (Figure D.13)

Alternatively, if qemu is run on the command line, it is possible to specify a monitor to run on the Terminal. This is shown in Figure D.14.



A screenshot of a terminal window titled "Terminal". The window contains the following text:

```
/usr/bin/qemu-system-i386 -monitor stdio -boot c -m 128 -hda '/media/sdb1/datadrive/WORKING_FILES/emulation/qemu/qemu_windows2000_2000mb' -cdrom '/media/sdb1/datadrive/WORKING_FILES/emulation/qemu/installation_disks/windows_2000_professional.iso' -net none -localtime -soundhw es1370 &
QEMU 2.0.0 monitor - type 'help' for more information
(qemu)
```

Figure D.14: QEMU Monitor on console.

The following screenshots with onscreen documentation are of installing Windows 2000 on QEMU.

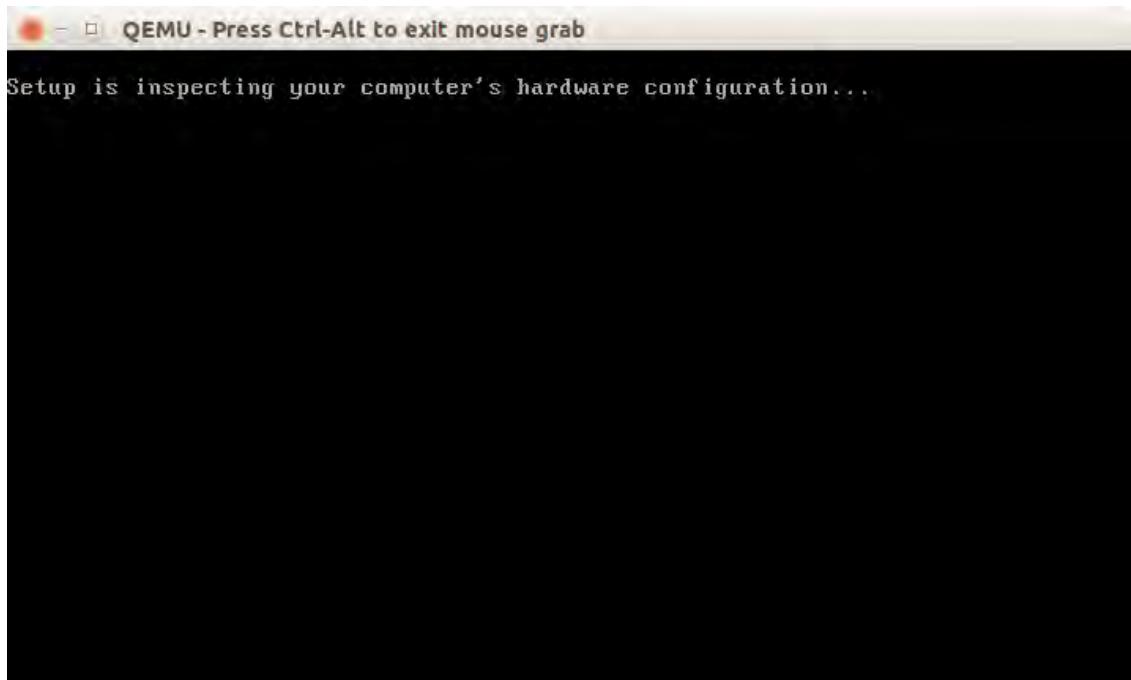


Figure D.15: Windows 2000 inspecting hardware configuration.

The first part of the installation process consists of formatting the target drive. (Figure D.16) Continue through the process by following the on screen directions. (Figures D.17, D.18, D.19)

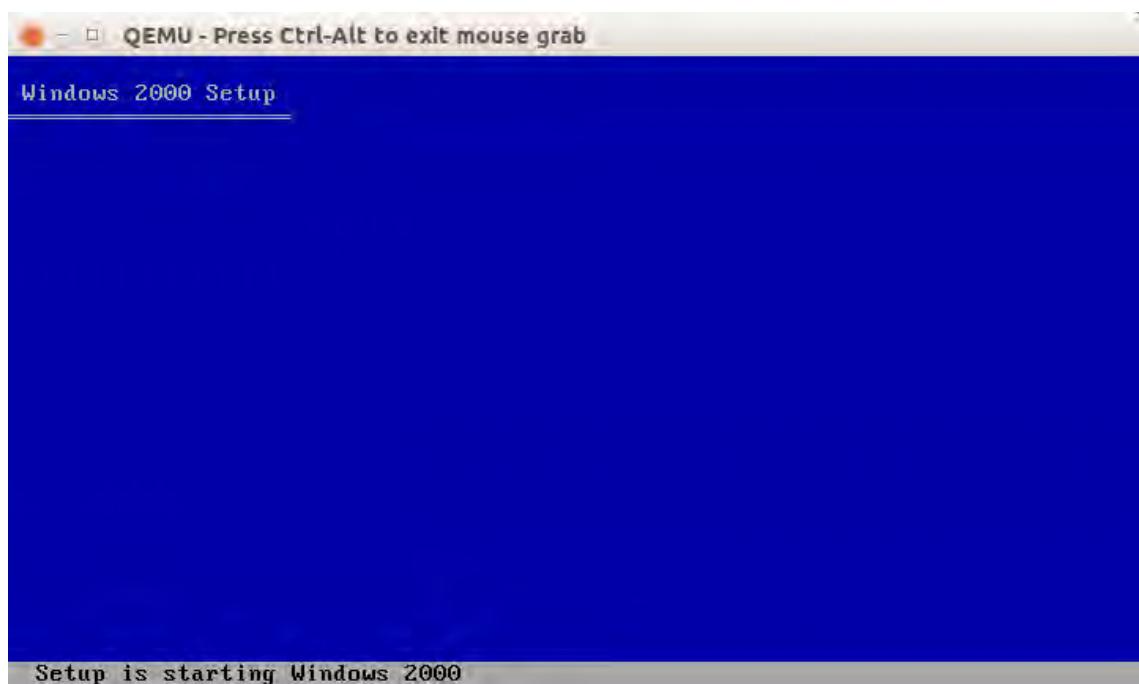


Figure D.16: Windows 2000 initial setup screen.

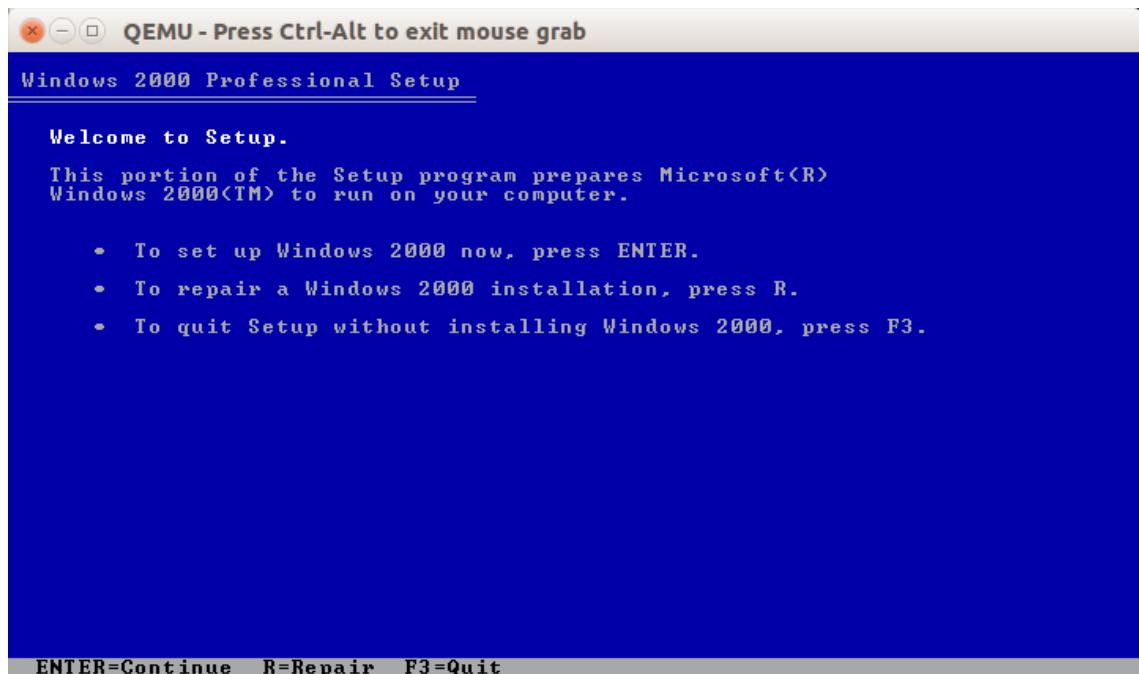


Figure D.17: Windows 2000 Welcome to Setup screen.

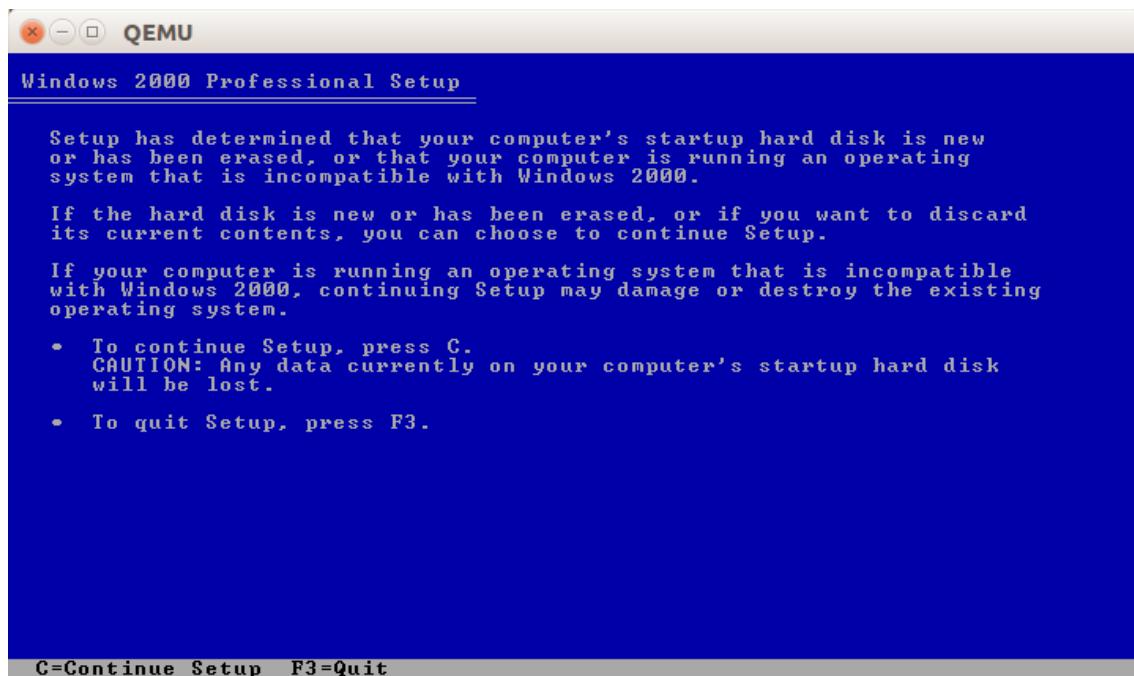


Figure D.18: Windows 2000 Professional Setup screen.

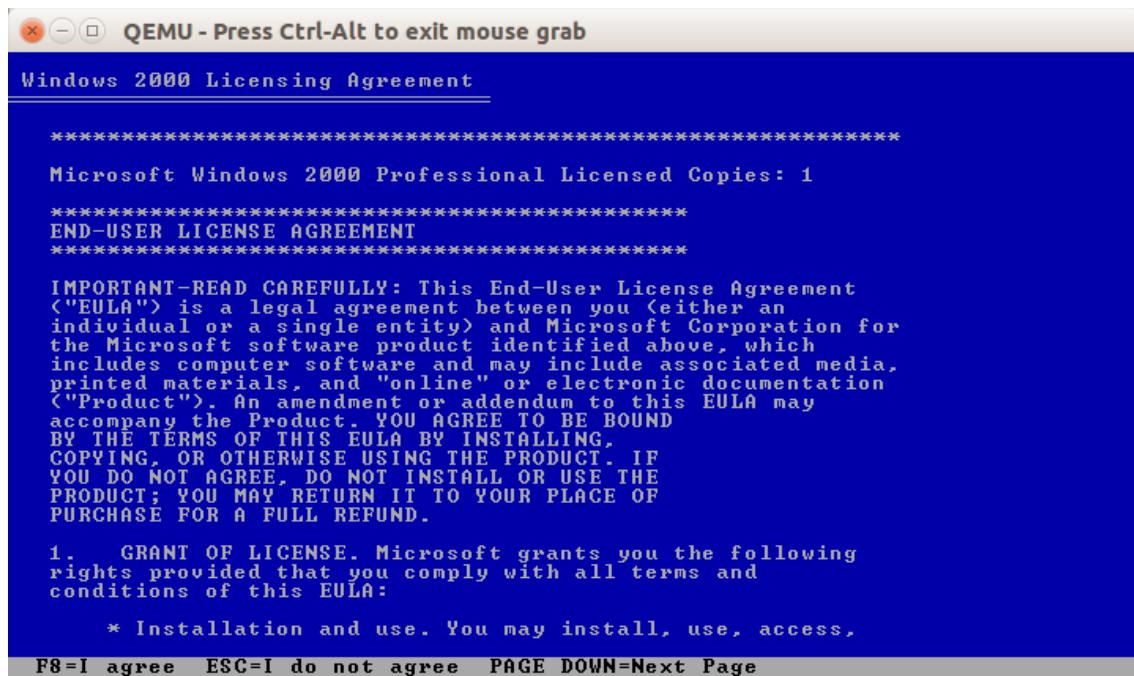


Figure D.19: Windows 2000 Licensing Agreement screen.

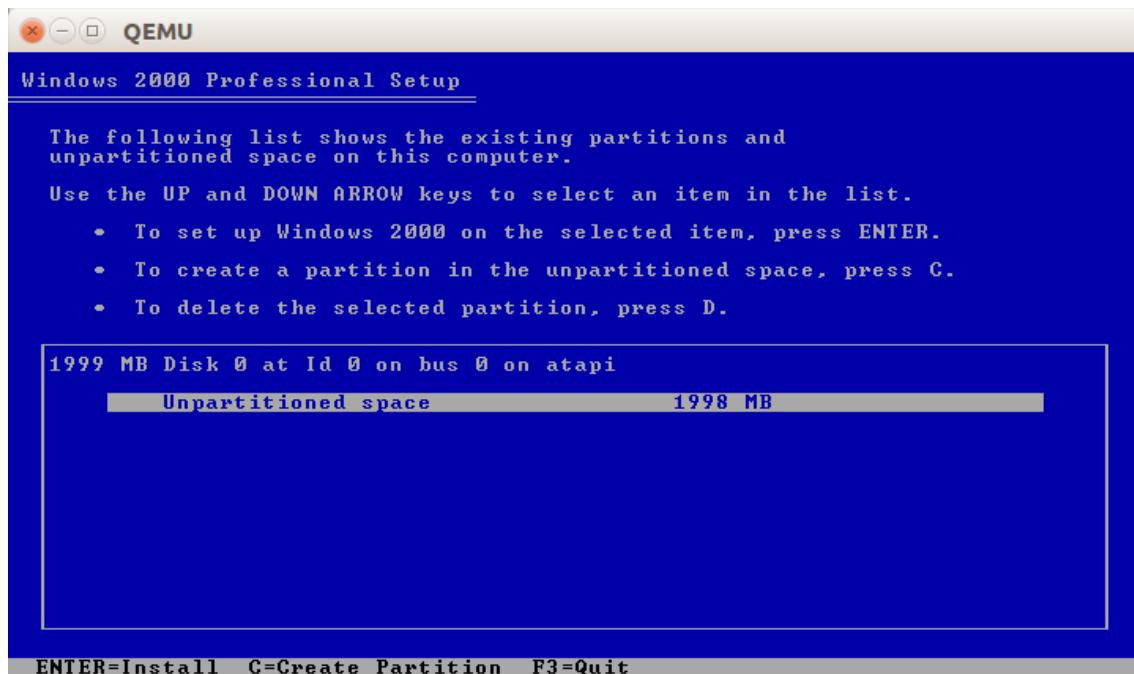


Figure D.20: Windows 2000 existing partitions list screen.

Press **C** to create a partition on the newly created disk image on the partitions list screen (Figure D.20). Note that its size will not be displayed as the exact value you specified in the earlier step. (It should be close, however.)



Figure D.21: Windows 2000 partition format type screen.

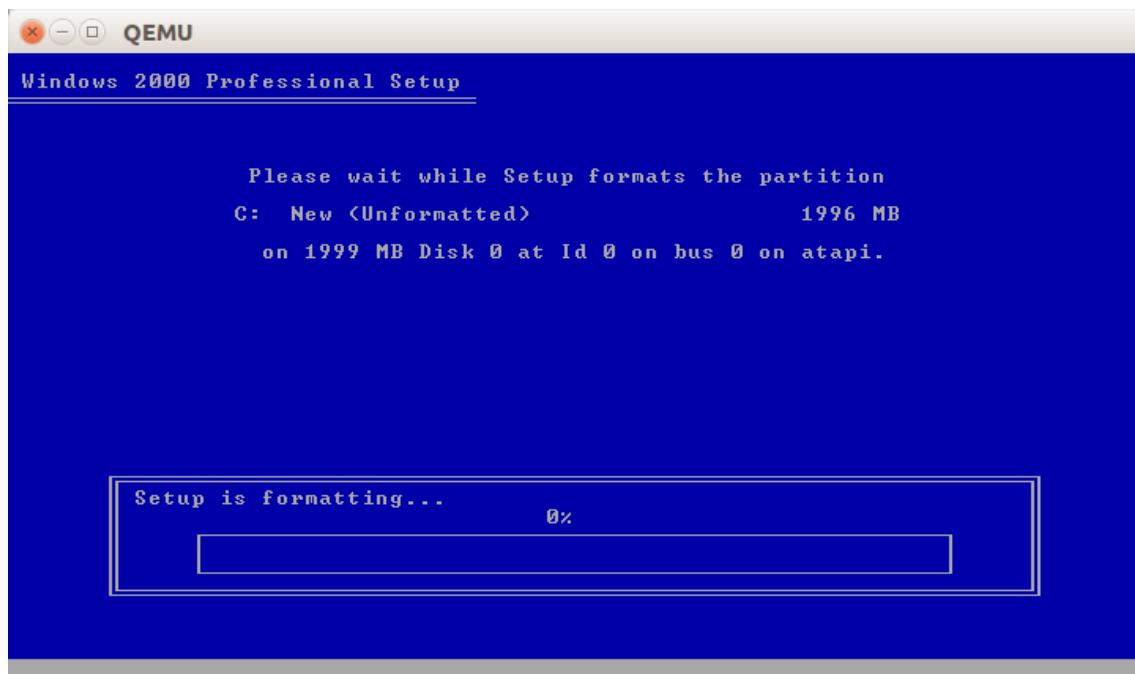


Figure D.22: Windows 2000 format partition progress bar.

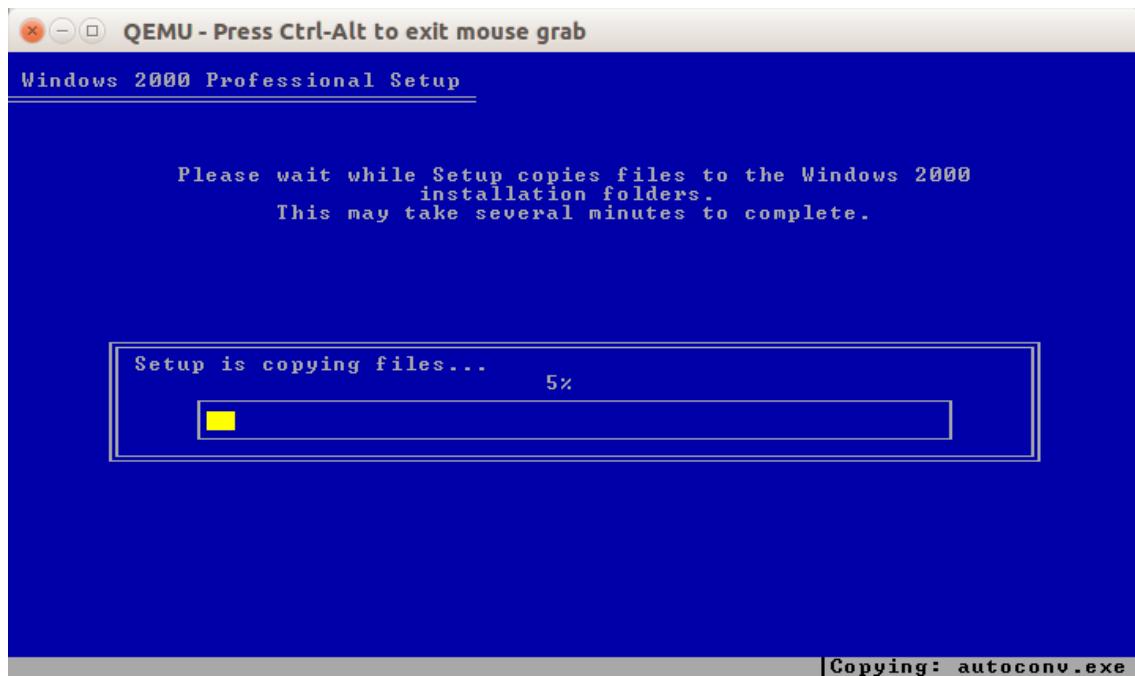


Figure D.23: Windows 2000 setup copying files progress bar.

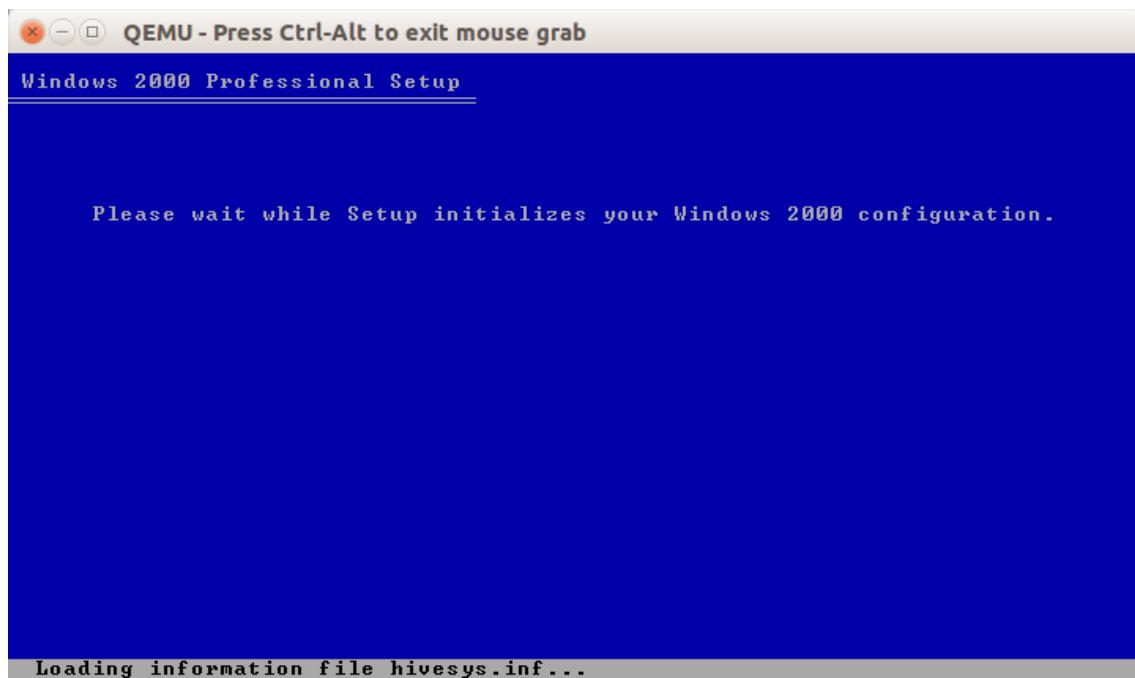


Figure D.24: Windows 2000 initializing configuration screen.

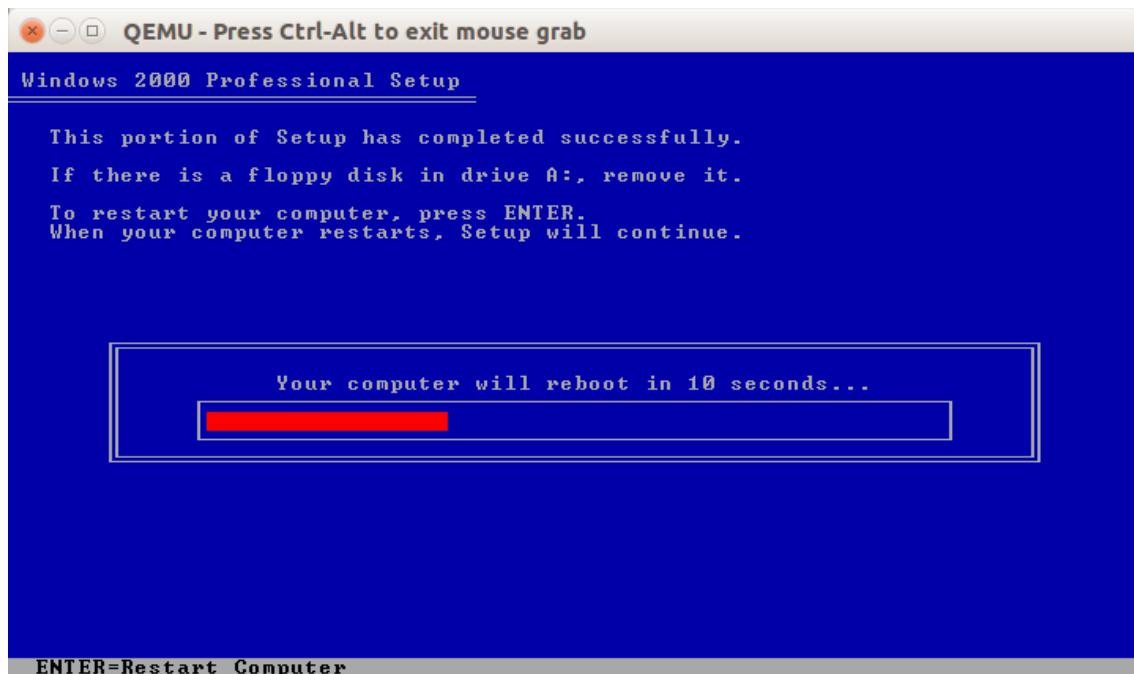


Figure D.25: Windows 2000 initializing configuration screen.

When the first portion of the setup has finished, press **Enter** to continue. Once the disk has been formatted, the system will reboot to begin Windows installation. (Figure D.25)

The graphical installer will then begin. (Figures D.26, D.27)



Figure D.26: Windows 2000 graphical installation splash screen.



Figure D.27: Windows 2000 graphical setup screen.



Figure D.28: Windows 2000 Setup Wizard start screen.

Click through the prompts to continue the second portion of the installation process. (Figure D.28) The progress bar will move to the right as installation progresses. (Figure D.29) Further screens in the Wizard may offer customization options and to proceed with the installation. (Figure D.30)

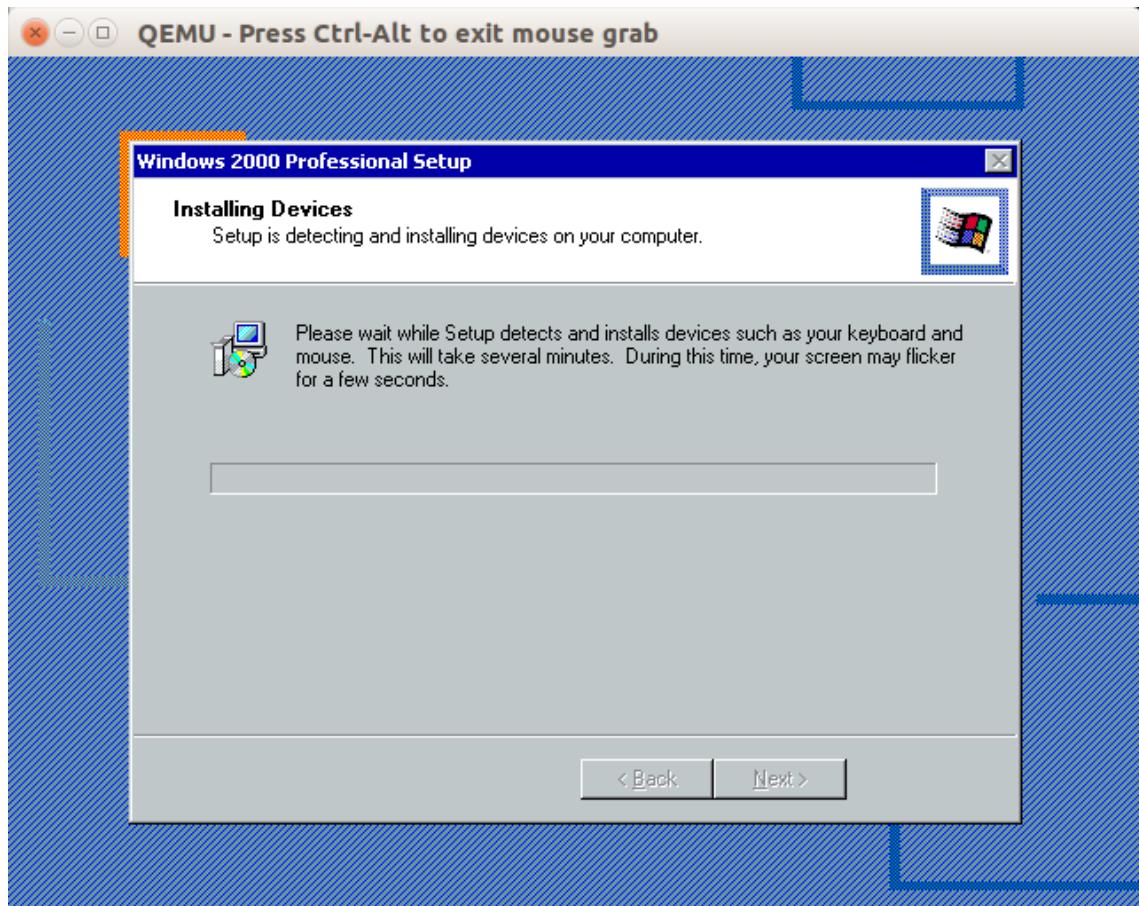


Figure D.29: Windows 2000 Installing Devices progress bar.

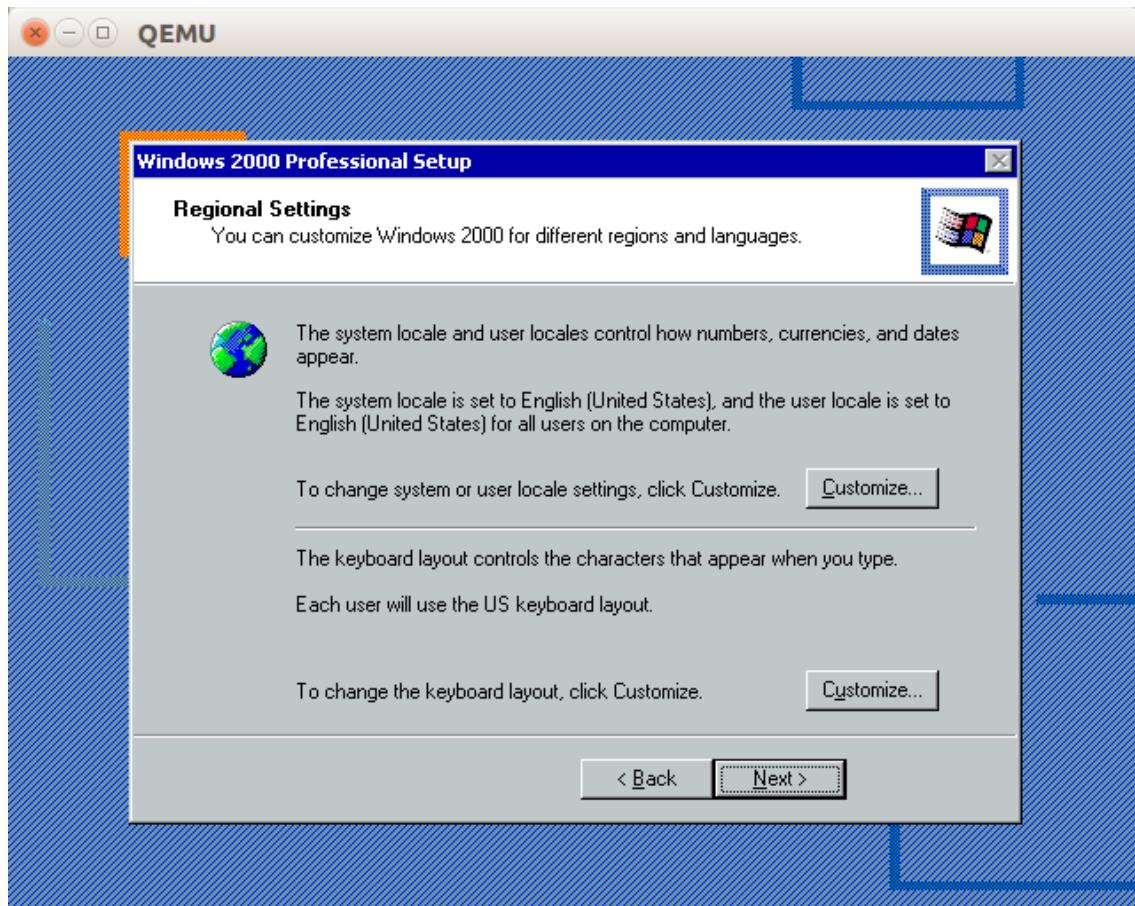


Figure D.30: Windows 2000 Regional Settings options.

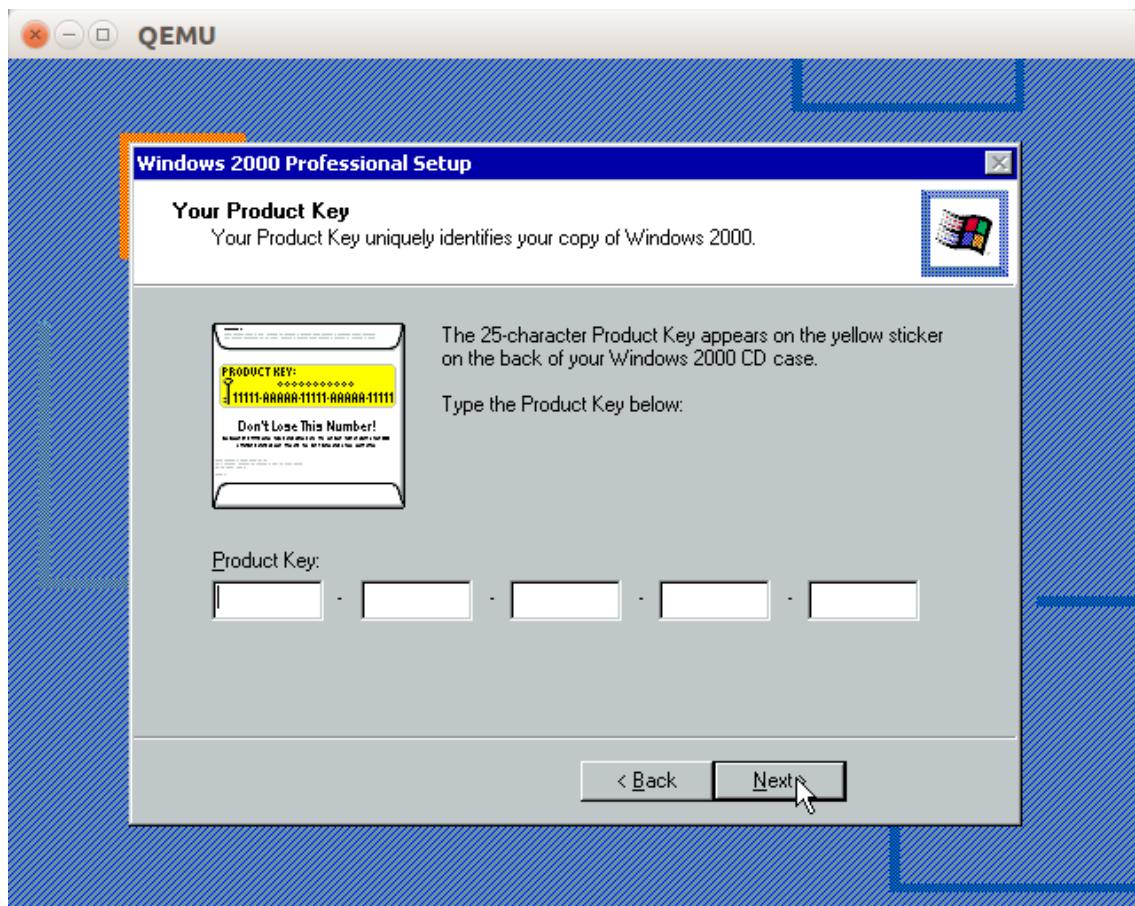


Figure D.31: Windows 2000 Product Key screen.

When prompted, enter the product key provided with the original installation media and click **Next**. (Figure D.31)



Figure D.32: Windows 2000 Computer Name and Administrator Password screen.

When prompted, select a name for the computer and set and confirm an Administrator password. (Figure D.32)

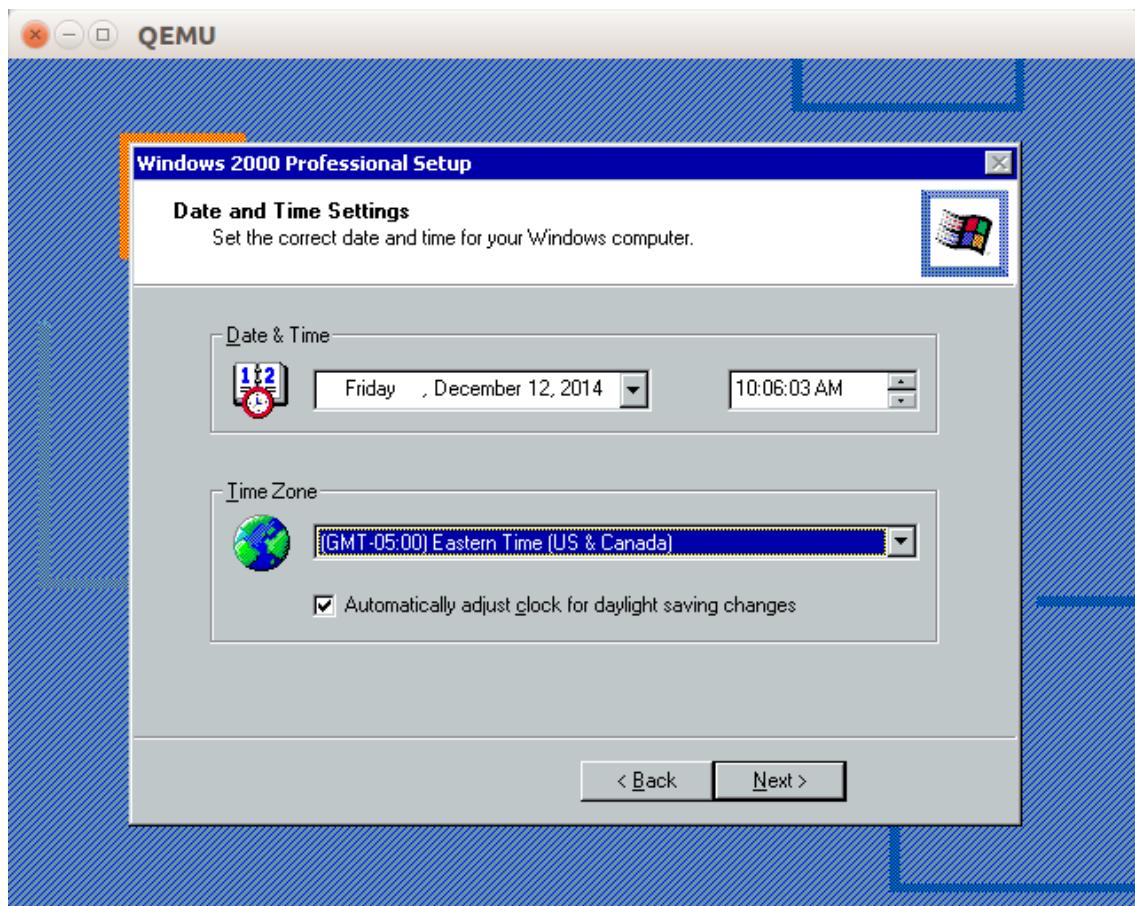


Figure D.33: Windows 2000 Date and Time settings screen.

When prompted, confirm the date, time, and time zone, and click **Next** to continue. (Figure D.33)

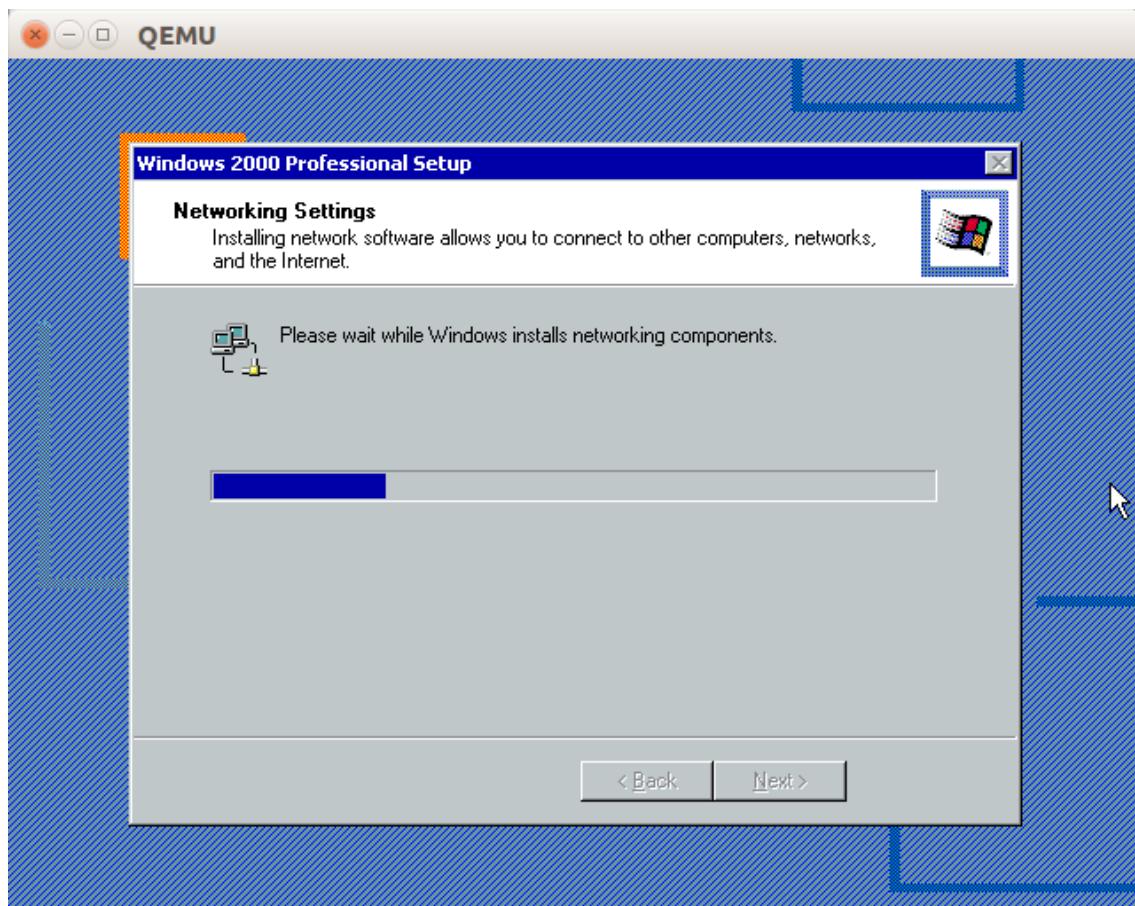


Figure D.34: Windows 2000 Networking Settings progress bar.

The progress bar will move to the right as installation progresses. (Figures D.34, D.35, D.36)

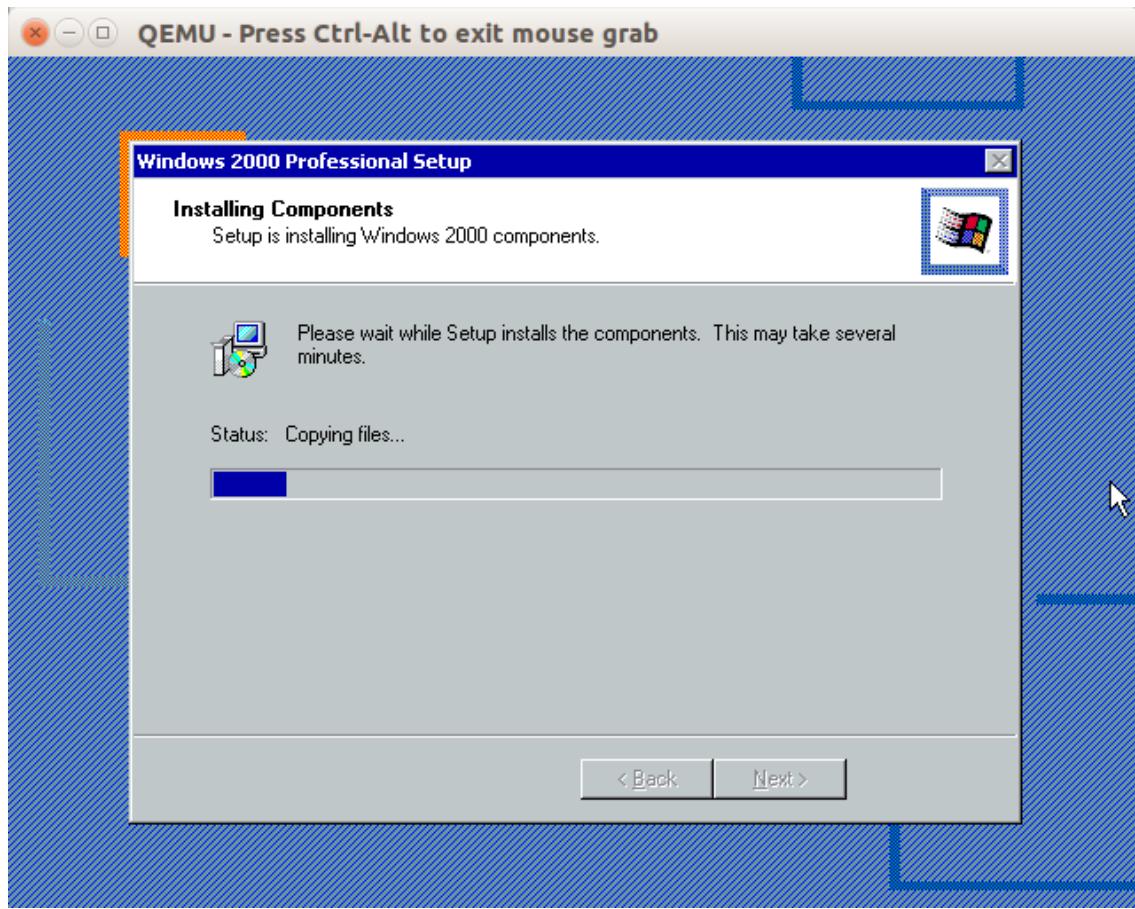


Figure D.35: Windows 2000 Installing Components progress bar.

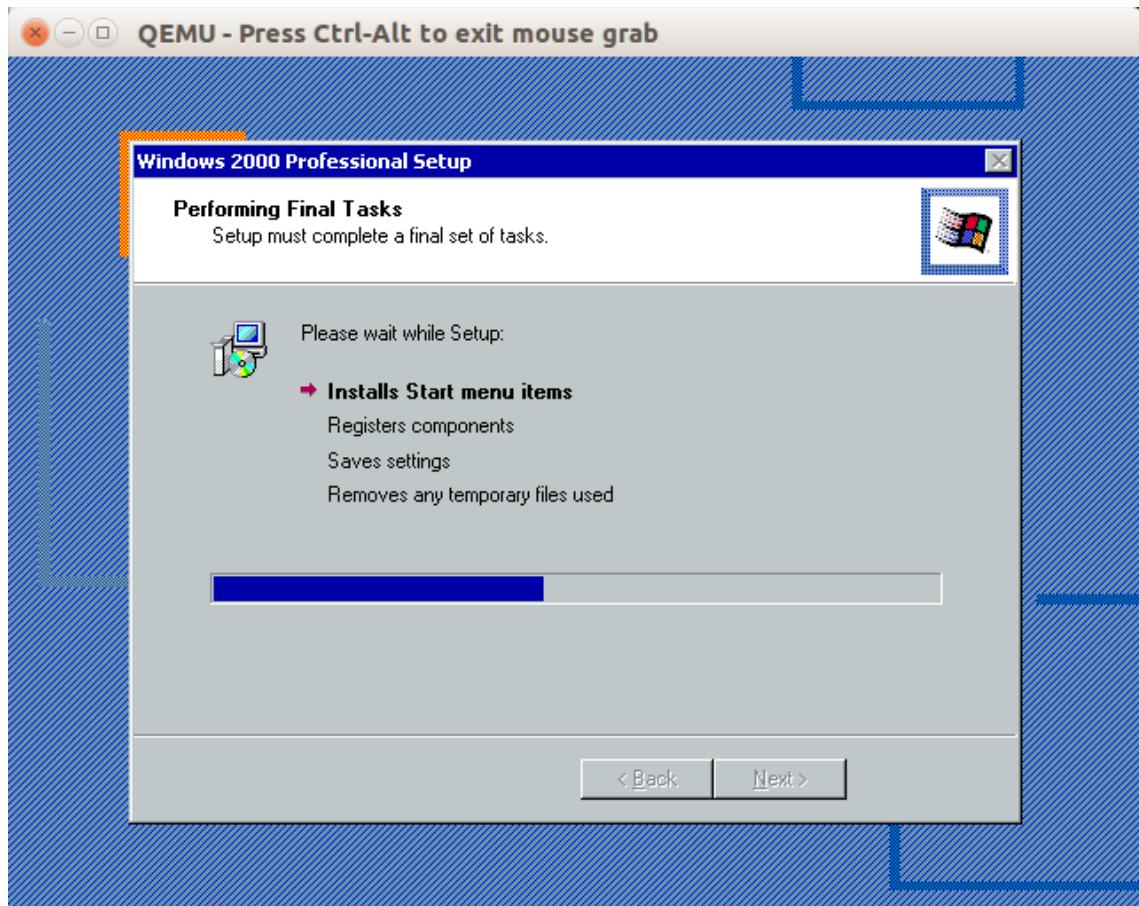


Figure D.36: Windows 2000 Performing Final Tasks progress bar.



Figure D.37: Completing the Windows 2000 Setup Wizard screen.

At the end of the installation from the CD-ROM, there is a **Finish** button. The system restarts once it is clicked. (Figure D.37)

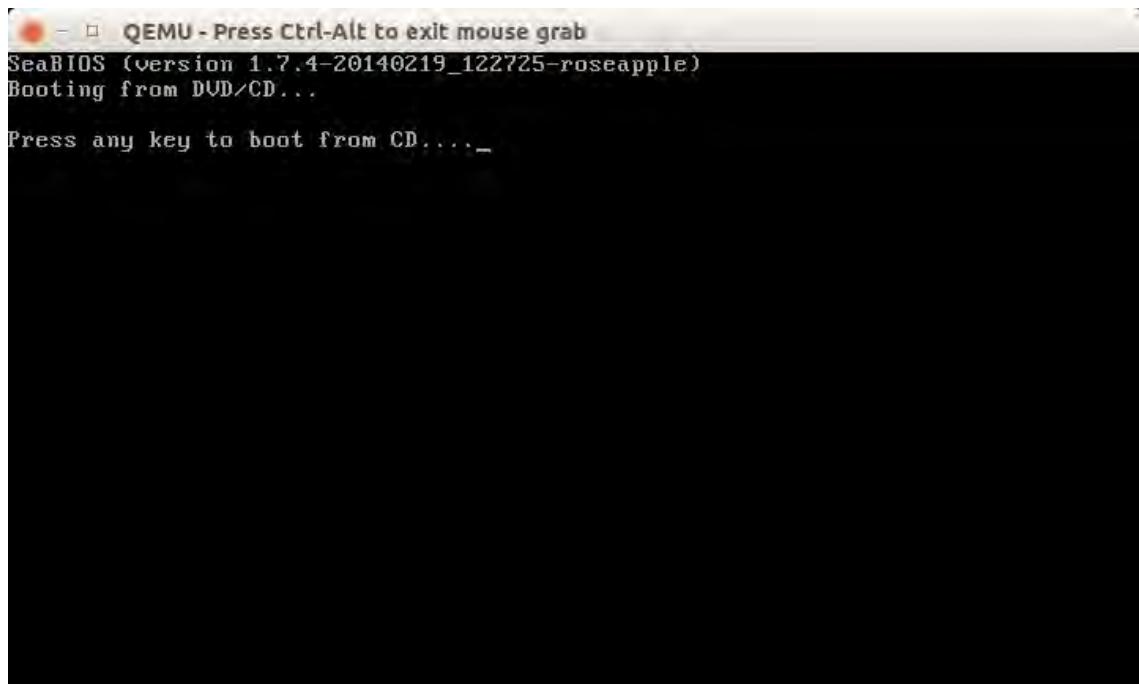


Figure D.38: QEMU rebooting system screen.

Once restarted, the system should boot from disk, rather than from the CD-ROM at this point. If not, it may be necessary to eject the CD-ROM from the guest system. (See D with screenshots of QEMU/Monitor for details.) (Figure D.38)

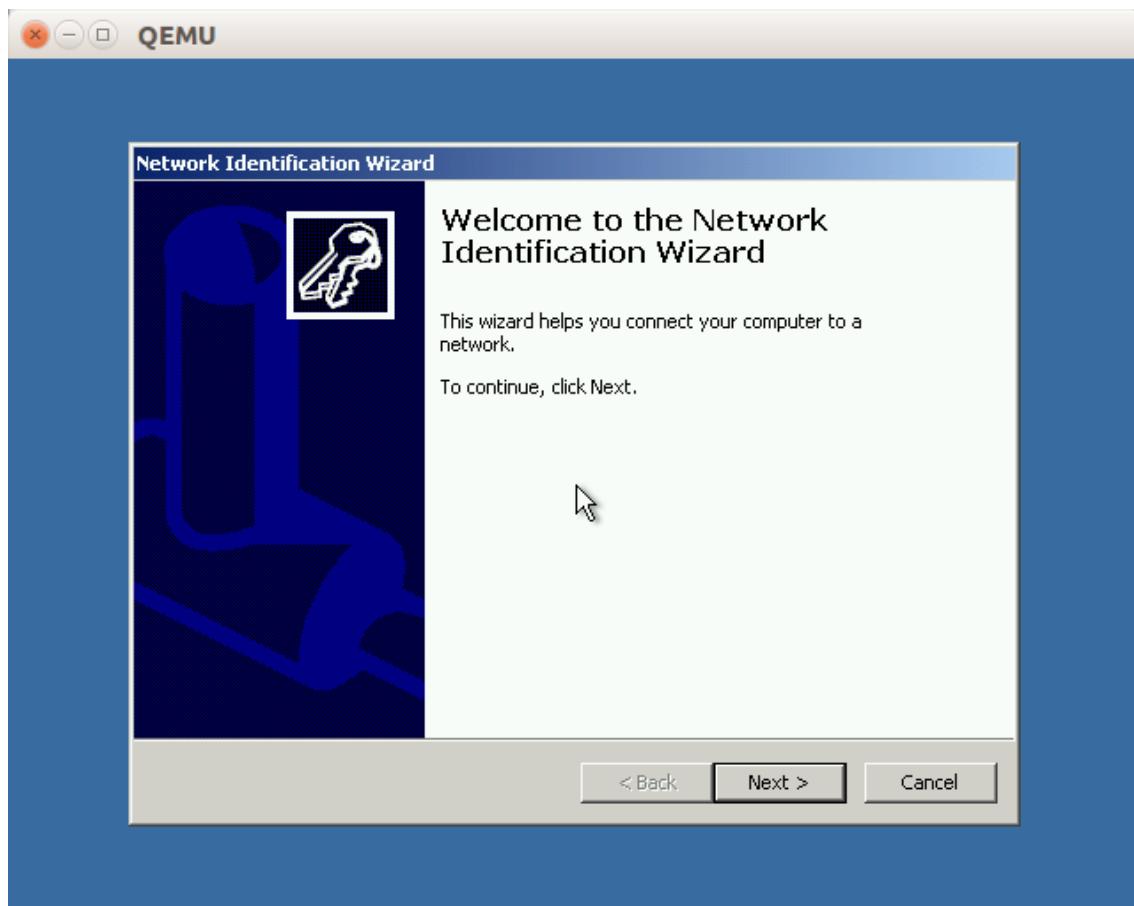


Figure D.39: Windows 2000 Network Identification Wizard screen.

Windows 2000 has additional configuration once the operating system is installed. Click **Next** to continue when this additional configuration starts. (Figure D.39) In the Network Identification Wizard, set a username and confirm a password, then click **Next**. (Note that this setting may not be present on other Windows operating systems.) (Figure D.40)

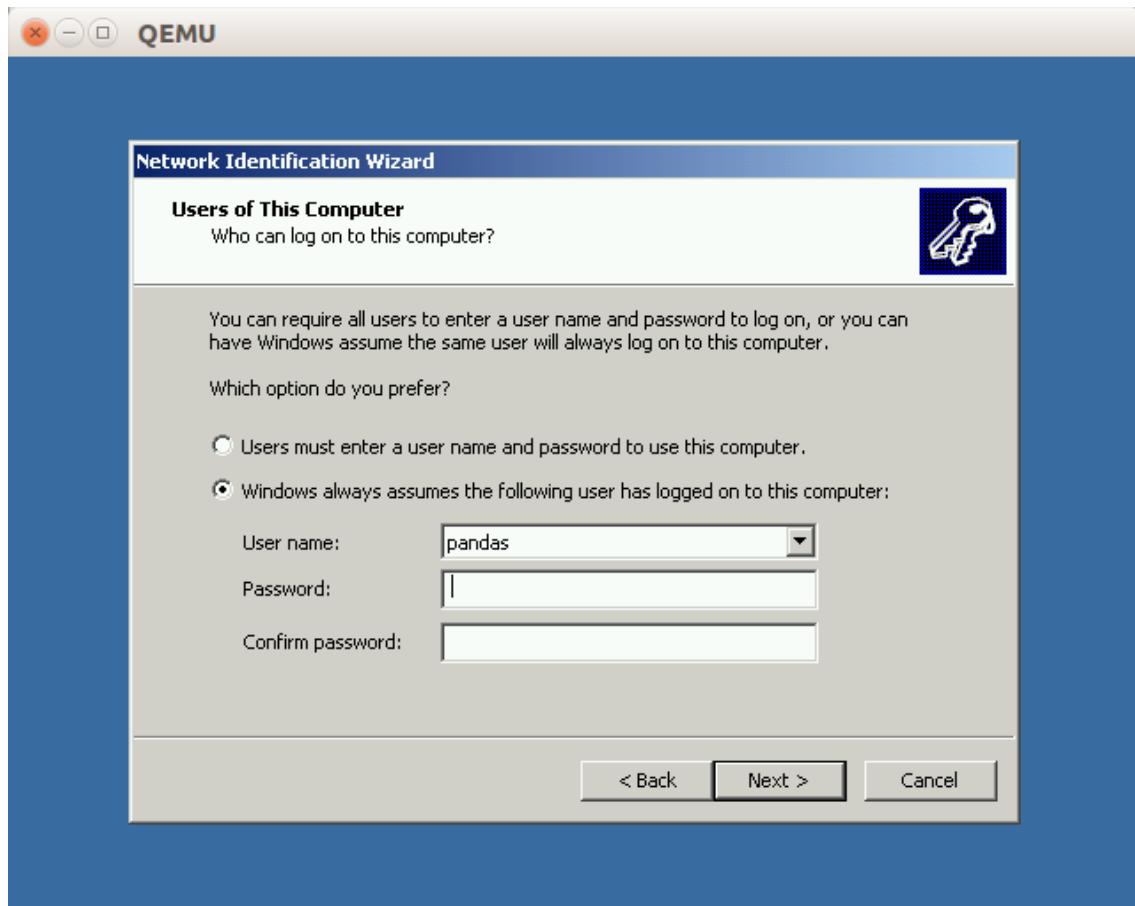


Figure D.40: Windows 2000 Users of This Computer dialog.



Figure D.41: Windows 2000 Getting Started screen.

Once this configuration has completed, the initial startup screen should appear.

## References

- [1] *System requirements for Microsoft Windows 2000 operating systems.* URL: <https://support.microsoft.com/en-us/kb/304297/en-us>.
- [2] *QEMU/Monitor: eject - Wikibooks, open books for an open world.* URL: <http://en.wikibooks.org/wiki/QEMU/Monitor#eject>.
- [3] *QEMU/Monitor - Wikibooks, open books for an open world.* URL: <http://en.wikibooks.org/wiki/QEMU/Monitor>.

## **A3: Pre-Ingest Work Plan for Cornell University Library Archival Repository (CULAR)**

### Partners

1. Tim Murray and Oya Rieger are co-PIs of the NEH grant that created these digital objects. They are informed of this deposit, but do not otherwise need to be involved.
2. Madeleine (Mickey) Casad is the associate curator of the Rose Goldsen Archive of New Media Art, a collection area of the division of Rare and Manuscript collections (RMC). She will inform the artists as to the deposit as necessary.
3. Liz Muller, Curator of Digital and Media Collections (RMC) will represent as the active steward for CULAR.

### Fitness for CULAR - PASS

1. General nature of the collection: The digital objects for this deposit are the output of a two year research grant from the National Endowment for the Humanities (NEH) awarded for the Cornell University Library proposal “Preservation & Access Framework for Digital Art Objects” in 2012. The proposal concerns itself with the research of the effort required to adequately preserve a selection of works from the Rose Goldsen Archive of New Media Art (<http://goldsen.library.cornell.edu>), an archive of complex born-digital materials that trace history of artistic experimentation in digital media. The work of the grant seeks to create contemporary emulation environments for artworks selected from the archive, to classify works according to type and document research discoveries regarding the preservation effort. All aspects of this work (disk images, compiled emulators, ROMs, Operating Systems, virtual disks, notes, reports, etc.) will need to be preserved and will be part of this CULAR project/collection.
2. Desire for deposit: CULAR is expected to assist in the following ways.
  - a. The works in question are currently located on CDROMs and DVD ROMs. CULAR will keep disk images stable and secure, addressing the threat of media rot and media obsolescence of the selected works. CULAR will also preserve related source code of operating systems and ROMS, compiled editors, and all documentation of the project.
  - b. Delivery of these new media art objects will be accomplished through the archive through other mechanisms. It is foreseen that all relevant images, emulators, operating systems, ROMS and virtual disks necessary for delivery will be separately provided, with no need to address the objects within CULAR directly to accomplish delivery.
  - c. CULAR is expected to perform the function of backup and preservation of master images and derivative use copies. Access would be limited to cases where a fresh copy of any of these is required in the delivery instance.
  - d. The active steward may desire to download XML MD of parts or the whole of the deposited assets, or descriptors for creating a local list of files within an image with checksums for ascertaining and/or remediating local use copies.

3. Grant output is currently backed up in daily increments using Cornell's EZBackup Service.
4. This collection is not currently preserved elsewhere.

#### Technical Notes

1. The deposit will be ingested as assets of RMC, and be added to CULAR according to the RMC schedule.
2. The overall deposit will consist of master and use copy images of approximately 80 born-digital works stored natively on CDs and DVDs, source code for compiled emulators, ROMS, relevant OS versions, virtual disks, notes and metadata. See detail at the end of this document for description of deposit structure and file types of deposit objects.
3. Initial deposit size will be approximately 65 GB in aggregate.
4. File system dates will not require preservation. Imaged disks will already package the dates of the original files internal to the disk images. Dates of images themselves will not require documentation, and basic timeframe of grant will adequately date the project output.
5. There is no sensitive data contained in this deposit, although some of the objects are covered by commercial copyright.
6. Ingest is envisioned as one initial deposit (perhaps chunked into multiple initial deposits at the convenience of programmers), although possibly there may be periodic ingestions to CULAR for this project:
  - Subsequent deposits may be added to this project if other similar works are preserved.
  - Subsequent documentation may be added after the bulk of the disk images, emulators, etc. are ingested (a sort of phased initial ingest.)
  - If emulation strategy changes there will be need to update emulators and use copies. We may rely on CULAR versioning to organize/mark versions and manage subsequent ingestions.
7. Structure of ingest is described below and the depositor will arrange data accordingly
  - Conceptual works (disk images and imaged documentation, etc.) will be deposited within the aggregate "Works" within the relevant aggregate that corresponds to the collection referenced in the MARC record for the work in the 710 Field. Here is the definitive list of collection aggregates for which PAFDAO created output.

RMM08200 - Rose Goldsen Archive of New Media Art

RMM08201 - Shilpa Gupta Collection of Media Art

RMM08202 - Calin Man Collection of Digital Art

RMM08205 - Nancy D. Nisbet Collection of Media Art

RMM08206 - Ana Maria Uribe Collection of Poetry and Media Art

RMM08207 - Roadmap to an Artscene: An introduction to artists from Flanders, Belgium

RMM08208 - Narvika Bovcon and Ales Vaupotic Collection of Media Art

RMM08210 - Jordan Detev Collection of Media Art  
RMM08211 - John (Craig) Freeman Collection of Media Art  
RMM08213 - Machine Life  
RMM08214 - Panos Kouris Collection of Media Art  
RMM08236\_Per\_Pegelow\_Gating\_Collection

- RMM8200 will be considered the main deposit for the project, and will contain project related documentation, all emulators, but only the collection assets relevant to the RMM08200 collection.
- References to general project documentation will be made in other collections to ...RMM08200/PAFDAO for relevant documentation, emulators, etc.

Key:

(A) – Aggregate object  
(R) – Resource object  
(M) – Metadata object (understood to be descriptive of Aggregate object immediately containing it and immediately preceding in this document)

### Expected Structure within CULAR

#### **Within “RMM08200\_Rose\_Goldsen”**

- **PAFDAO** (A) This functions as a wrapper that defines deposits to these collections that were derived from the PAFDAO project. There are currently other assets in these collections (DVDs, audio, etc.) that are not part of the project scope.
  - **\_Documentation** (A) (...RMM08200/PAFDAO only)
    - documentation of ingest plan (2 versions of this document) (R) ODT and PDF/A
    - Basic emulator and access notes (R) PDF/A document describing emulators used and notes regarding their compilation (part of grant output)
    - Classification Document PDF/A (R) a document describing classes of works and attributes (part of grant output)
    - Other documents as needed. (Note - Project should rely on separate deposit to eCommons for delivery solution if items are meant to be publicly accessible.) (R) TXT, PDF/A, other open formats
  - **CompiledEmulators** (A) (...RMM08200/PAFDAO only)
    - Hard\_Disc (A)
      - Hard disk images (R) ISO that represent an OS running on an emulator (200-300 mb apiece; 5-6 per emulator.)
      - ReadMe(s) (R) TXT regarding peculiarities of the OS
    - Executables (A)
      - Emulator as compiled (R) (binaries written in C++) (R) TAR.GZ
      - Readmes (R) – will be versioned if/as recompilations are made.

- ROMS (A)
  - Zip(s) of ROMS (R)BIN
  - Readmes (R)
- **Works** (A) (all RMM numbers)
  - 6086156 (Aggregates named after BibID, one for each BibID) (A) – individual conceptual work (iterate this folder and children for each work in the deposit); (M) MARCXML from library catalog record associated here
    - Disk Images (A) – might be multiple for conceptual work
      - Disk Images (R). At least one image for each physical disk, but may also have multiple image formats for a single physical disk. Naming to include BibID and file extension of format  
( 6086156 .ISO, 6086156 .BIN, 6086156 .DD, 6086156 .cu e ); Associate DFXML to each matching disk image; (M): Associate PREMIS to each matching disk image (M)
      - Info (A)
        - 6086156 .info - Output file for images created by Guymager (R) TXT
        - ImagingNotes.txt (R)
      - DerivativeDiskImage (Optional - provisional for cases of reconstructed works, or works upgraded to a new OS) (A)
        - Disk Images (R) bite perfect image. One image for each physical disk.
        - Guymager - Output file for each image (R) TXT
      - Coverscans (A)
        - Image files of cases, booklets, flyers, inserts and other ephemera associated with work (R) TIFF
        - Document containing all scans (R) PDF/A

### **Supplied metadata**

- A single file manifest of all resource files contained within “Works”, containing filename (full path) and MD5 hashes.
- One MARC XML file for each conceptual work.
  - All files will be contained in one non-hierarchical (“flat”) directory called PAFDAO\_MARCXML.
  - These should be derived out of the library catalog and named after the BibID of the work they describe.
  - For example: “6086156\_marc.xml” is the file that describes the work with the BibID “6086156”
  - There is a 1:1 correspondence between each MARCXML file and the conceptual work it describes. The intention is for these to be ingested and associated as metadata to the aggregate named for the same BibID.

- One DFXML file for every disk image.
  - All files will be contained in one non-hierarchical (“flat”) directory called PAFDAO\_DFXML.
  - Each file will be named after the filename of the dusk image, with XML as the file suffix. For example:
    - “6086156\_239\_copy\_dFXML.xml” is the file that describes “6086156\_239\_copy.dd”
    - “6086156\_239\_dFXML.xml” describes “6086156\_239.dd”
  - There is a 1:1 correspondence between a DFXML file and the image it describes. There *may be* a 1:many correspondence between a DFXML file and resource filenames. However, the files will be arranged such that the file contained in the “parent-most” path will be the resource file to which the DFXML file associates as metadata. The metadata tab in the administrative interface should be labelled “DFXML”.
- One PREMIS XML file for every disk image
  - All files will be contained in one non-hierarchical (“flat”) directory called PAFDAO\_PREMISXML.
  - Each file will be named after the filename of the dusk image, with XML as the file suffix. For example:
    - “6086156\_239\_copy\_premis.xml” is the file that describes “6086156\_239\_copy.dd”
    - “6086156\_239\_premis.xml” describes “6086156\_239.dd”
  - There is a 1:1 correspondence between a PREMIS XML file and the image it describes. There *may be* a 1:many correspondence between a PREMIS XML file and resource filenames. However, the files will be arranged such that the file contained in the “parent-most” path will be the resource file to which the PREMIS XML file associates as metadata. The metadata tab in the administrative interface should be labelled “PREMIS”.

## **APPENDIX B: SURVEY OF MEDIA ART RESEARCHERS**

### **B1: Sample Message Distributed to Mailing Lists and Individuals to Elicit Responses to the Survey of Media Art Researchers**

Dear Colleagues,

We would appreciate your assistance with an NEH-funded digital media art preservation project currently underway at Cornell University. This project aims to develop scalable preservation strategies for complex, interactive, born-digital media artworks, using the collections of Cornell's Rose Goldsen Archive of New Media Art as a test bed (<http://goldsen.library.cornell.edu/>).

In developing a preservation framework that will address the needs of the broadest range of archive users, we seek the input of artists, researchers, educators, curators, and others who work with interactive digital artworks and artifacts. Would you please take a few minutes to respond to this questionnaire about your practices?

[https://cornell.qualtrics.com/SE/?SID=SV\\_6mPEBGQWr2K4nmR](https://cornell.qualtrics.com/SE/?SID=SV_6mPEBGQWr2K4nmR)

Depending on your responses, we estimate that this questionnaire will take 10-25 minutes to complete.

Information about questionnaire results will be published and made available to the broader media archives community. Information about this preservation initiative is available at:

<http://news.cornell.edu/stories/2013/02/humanities-grant-helps-library-preserve-digital-art>

Feel free to contact Mickey Casad, project manager, at [mir9@cornell.edu](mailto:mir9@cornell.edu) for more information.

Many thanks for your help with this investigation, and apologies for any cross-postings.

Yours on behalf of the project team,

Madeleine Casad  
Associate Curator, The Rose Goldsen Archive of New Media Art  
Digital Scholarship & Preservation Services  
Cornell University Library

## **B2: Survey Distributed to Media Art Researchers**

The following is a PDF version of the online survey we distributed to media art researchers.

## Introduction

Thank you for responding to this questionnaire about your media art research and preservation practices.

Your responses will help guide an NEH-funded project to create Preservation and Access Frameworks for Digital Art Objects in the holdings of Cornell University Library's Rose Goldsen Archive of New Media Art. If you would like more information about this initiative, or about the Goldsen Archive, please share your contact information at the end of the questionnaire.

The questionnaire begins by asking you to decide whether to respond as an individual researcher or as a representative of an archiving institution. You will have an opportunity to return to this decision and take both versions of the questionnaire, if you choose.

We estimate that it will take 15-30 minutes to complete these questions, depending on your responses.

- Respond as an individual researcher / practitioner
- Respond on behalf of an archive / museum / cultural heritage institution

The respondent's selection determines which set of questions appear next--"Individual Researcher" (I.), or "Museums/Archives" (II.). In either case, the respondent will have an opportunity to answer the second set of questions.

### INDIVIDUAL RESEARCHER

What is your name?

Which of the following best describe your engagement with media art? (Select any that apply)

- Researcher / Writer
- Educator
- Curator
- Artist
- Other (please describe below)

The respondent's selections here trigger questions targeted to educators, curators, artists, etc., later in the questionnaire.

When you visit collections of digital media artworks and artifacts, do you have a specific research agenda or question in mind?

- Always
- Often
- Rarely
- Never

Questions marked in green are targeted to researchers, but appeared to all respondents.

Please describe some of the questions that have guided your research with digital media artworks and artifacts.

What kinds of digital artifacts do you work with most often?

What is your ideal scenario for interacting with archive-owned versions of such materials? (For example, your response could reference spatial surroundings, hardware and peripherals, control over settings, availability of physical media, remote or networked access, or anything that seems especially important to you.)

What are the most serious impediments you have encountered in researching such materials?

Which of the following genres or content types do you emphasize in your media art research? (Check any that apply.)

- Interactive artist portfolios or compendia of works
- Installation / performance / media sculpture
- Video / cinema
- Code
- Hypertext
- Games
- Interactive narrative
- Expanded cinema
- Virtual reality / augmented reality
- Other (please describe below)

The respondent's selections here appear again to be ranked later in the questionnaire.

Use this space to elaborate on any of your responses from the previous question, if desired.

Which of the following platforms, forms, formats, or physical media do you emphasize in your media art research? (Check any that apply.)

- CD-ROM or DVD-ROM
- Floppy discs
- Hardware / peripherals
- Personal computers or devices
- Web-based artworks
- Locative media
- Installation / sculpture / performance
- Participatory artworks
- Specific software or filetypes (please describe below)
- Other (please describe below)

The respondent's selections here appear again to be ranked later in the questionnaire.

Use this space to elaborate on any of your responses from the previous question, if desired.

Which of the following research frameworks do you emphasize in your media art research? (Check any that apply.)

- Cultural or art-historical contexts (includes exhibition and reception histories, artists' writings, and other work)
- Social contexts and histories
- Technological contexts and histories
- Production histories
- Media theory
- Aesthetic theory
- Other (please elaborate below)

The respondent's selections here appear again to be ranked later in the questionnaire.

Use this space to elaborate on any of your responses from the previous question, if desired.

Please rank your selected genres or content types in order of importance. (1 = most important)

- » Interactive artist portfolios or compendia of works
- » Installation / performance / media sculpture
- » Video / cinema
- » Code
- » Hypertext
- » Games
- » Interactive narrative
- » Expanded cinema
- » Virtual reality / augmented reality
- » Other (please describe below)

A respondent would see only those genres he or she had selected as relevant earlier in the questionnaire.

1	2	3	4	5	6	7	8	9	10
<input type="radio"/>									
<input type="radio"/>									
<input type="radio"/>									
<input type="radio"/>									
<input type="radio"/>									
<input type="radio"/>									
<input type="radio"/>									
<input type="radio"/>									
<input type="radio"/>									
<input type="radio"/>									
<input type="radio"/>									

Please rank your selected platforms, forms, formats, or physical media in order of importance. (1 = most important)

- » CD-ROM or DVD-ROM
- » Floppy discs
- » Hardware / peripherals
- » Personal computers or devices
- » Web-based artworks
- » Locative media
- » Installation / sculpture / performance
- » Participatory artworks
- » Specific software or filetypes (please describe below)
- » Other (please describe below)

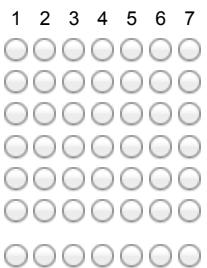
A respondent would see only those platforms he or she had selected as relevant earlier in the questionnaire.

1	2	3	4	5	6	7	8	9	10
<input type="radio"/>									
<input type="radio"/>									
<input type="radio"/>									
<input type="radio"/>									
<input type="radio"/>									
<input type="radio"/>									
<input type="radio"/>									
<input type="radio"/>									
<input type="radio"/>									
<input type="radio"/>									

Please rank your selected research frameworks in order of importance. (1 = most important)

- » Cultural or art-historical contexts (includes exhibition and reception histories, artists' writings, and other work)
- » Social contexts and histories
- » Technological contexts and histories
- » Production histories
- » Media theory
- » Aesthetic theory
- » Other (please elaborate below)

A respondent would see only those frameworks he or she had selected as relevant earlier in the questionnaire.



In cases where full interactive access to a complex digital art object is not possible, what documentation strategies work the best for your research purposes? Please cite specific examples if possible.

You indicate that you are an educator. In what contexts and subject areas does your teaching involve interactive digital media artworks and artifacts? (Select any that apply.)

- Post-secondary education (colleges, universities, post-secondary art or technical schools); please describe subject areas
- Primary or secondary educational institutions (ages 5-18); please describe subject areas
- Museum, gallery, or archival settings; please describe below
- Short-duration workshops, seminars, or lectures; please describe below
- Other; please describe below

Questions marked in blue appeared only to respondents who described themselves as educators earlier in the questionnaire.

What kinds of interactive digital artworks or artifacts do you emphasize in your teaching?

How do you use, assign, or reference such materials in your teaching?

What impediments have you found to teaching interactive digital media materials?

What aspects of these works are most important for your students to understand?

What kinds of technical and archival support would most enhance your teaching with digital media materials?

As an artist, do you create interactive digital media artworks? Please elaborate on your response as desired.

Yes

No

Questions marked in purple appeared only to respondents who identified themselves as artists earlier in the questionnaire.

What preservation measures have you taken with your own digital work?

What are your biggest concerns about preserving your own media artwork?

Note that these questions target preservation concerns in general. The project team is creating a follow-up interview questionnaire for artists in the Goldsen collections, to address more specific technical and aesthetic concerns.

How do practices of archiving and accessing complex digital media artworks affect you most, with regard to your own creative and professional work?

As a curator, are you responsible for interactive born-digital artworks and artifacts? If yes, what kinds of works are in your collections?

Yes

No

Questions marked in pink appeared to respondents who identified themselves as curators earlier in the questionnaire. This section is very short, as most of our curator-focused questions appear in the second part of the questionnaire ("Museums/Archives/Libraries").

As a curator, what are your biggest concerns about preserving access to older digital artworks?

Please use the space below to share any additional comments about this survey or its subject matter.

Would you be willing for members of this preservation and access initiative to contact you in the future?

Yes--use this contact information:

I would rather not be contacted.

Thank you for your responses. You may stop here, or retake the questionnaire on behalf of a library, archive, museum, or other cultural heritage institution with which you are affiliated.

Finish questionnaire.

Retake questionnaire as a representative of a library, archive, museum, or other cultural heritage institution.

#### MUSEUMS / ARCHIVES / LIBRARIES

The following series of questions appeared to respondents who selected "respond on behalf of an archive..." in the introductory question.

What is your name?

What institution do you represent?

What is your role there?

Does your institution include born-digital interactive media artworks and artifacts in its holdings?

Yes

No

Respondents' selection here determined which questions they would see next.

You indicate that your institution does not include born-digital interactive media artworks in its holdings. What are the main reasons why not? (Check any that apply.)

- Such materials fall outside collecting scope
- Intellectual property questions too complex
- Procedures for providing access too complex or unsustainable
- Procedures for preservation or conservation too unclear or technologically challenging
- Lack of technological infrastructure
- Lack of specialized staff support
- Lack of audience interest
- Other \_\_\_\_\_

These selections marked in blue appeared only to respondents who answered 'no' to the previous question

Other (please describe below)  

Use this space to elaborate on your responses from the previous question, if desired.

What kinds of interactive born-digital materials do you have, and approximately how many items fall into these categories?

How do you currently provide access to these materials?

These questions marked in pink appeared only to respondents who answered 'yes' to the earlier question.

What preservation strategies do you currently employ for these materials?

What conservation measures do you take, and when?

What measures do you take to secure access, preservation, and migration rights to digital materials in your holdings? (This questionnaire aims to learn more about your interactive born-digital media art preservation practices, but other kinds of collections may be relevant here as well.)

These questions marked in green appeared to all respondents, regardless of whether their institutions hold collections of born-digital interactive media art.

What kinds of patrons or audiences seek out your media collections?

Do you support online access to such materials? If so, how?

Do you support on-site access to such materials? If so, how?

What are the most serious institutional challenges you face, with regard to providing continued access to born-digital interactive artifacts in your holdings?

These questions marked in pink appeared only to respondents whose institutions hold collections of born-digital interactive media art

When you have questions about preservation, conservation, or providing access to interactive born-digital media objects, where do you look for answers?

Please use the space below to share any additional comments about this survey or its subject matter.

Would you be willing for members of this preservation and access initiative to contact you in the future?

Yes--use this contact information:

I would rather not be contacted.

Thank you for your responses. You may stop here, or retake the questionnaire from your perspective as an individual researcher rather than a representative of an archival institution.

Finish questionnaire.

Retake the questionnaire, responding as an individual researcher.

## **APPENDIX C: ARTIST QUESTIONNAIRE INSTRUMENT**

### **C1: Sample Message Distributed to Artists to Elicit Responses to the Initial Artist Questionnaire**

Dear \_\_\_\_\_,

We remain so appreciate of your participation in the Rose Goldsen Archive of New Media Art. Since its founding in 2002, when your initial support meant so much to the future success of the Archive, the Goldsen Archive has developed into one of the premier international resources for the archiving and preservation of video and new media art. We invite you to peruse our website (<http://goldsen.library.cornell.edu>), which we are currently updating, to familiarize yourself with the breadth and extent of our holdings, with major holdings and special collections in video art, internet art, CD-ROM and DVD-ROM interactive art, as well as work samples and dossiers across the electronic and digital media. We hold some of the most substantial special collections in Chinese and American new media art, as well.

Among our proud accomplishments have been three partner grants with Turbulence.org provided by the U.S. National Endowment for the Arts on the preservation of Turbulence internet art. We are now concluding, as well, a major two-year grant from the U.S. National Endowment for the Humanities on preservation and access of CD-ROM interactive art. As part of this grant, we are conducting a survey of selected artists represented in the Goldsen Collection. We would be very appreciative if you were able to take some time to respond to our questions.

We might mention that we are always interested in developing our dossiers of individual artists' work, and would continue to welcome contribution of any artist samples, publications, or projects that you might be willing to send along for this very successful, international collaborative effort. Please don't hesitate to contact either of us if you have any questions or suggestions. We so welcome your support and collaboration.

Best wishes,

Tim Murray, Curator ([tcm1@cornell.edu](mailto:tcm1@cornell.edu))  
Mickey Casad, Associate Curator ([mir9@cornell.edu](mailto:mir9@cornell.edu))  
Rose Goldsen Archive of New Media Art  
Division of Rare and Manuscript Collections  
2B Carl A. Kroch Library  
Cornell University, Ithaca, NY, 14853

**C2: Initial Interview Questionnaire Distributed to Artists**

**The following is a PDF version of the online questionnaire we distributed to artists.**

### Block 3

Thank you, \${m://FirstName}, for responding to this general questionnaire about your artworks in the Rose Goldsen Archive of New Media Art. Your responses will help guide our implementation of new archival practices developed with the support of an NEH preservation and access grant. We appreciate your input.

We would like to begin by asking some questions about specific artworks. Please verify that the titles below are your artworks, and that you are willing to share your recollections about them with us.

- \${e://Field/Work1}
- \${e://Field/Work2}
- \${e://Field/Work3}

The survey instrument generates a personalized version of the questionnaire for each artist in the Goldsen database. Artwork specific questions will repeat for each title the artist selects.

### Block 1

What type of computer hardware was used to create this artwork? (eg: Apple/Mac, PC/Windows, Linux, Other)

What software or programming language was used to create this artwork? (eg: Macromedia Director, or other software, or some specific programming language/framework)

What hardware and software were optimal for running this artwork when it was new?

Do you recall any specific technical problems with this artwork when it was new?

Do you still have any of the working files (including source code) you used to develop this artwork?

- Yes (feel free to add any relevant notes)
- No
- The working files may be held by another person or institution (please elaborate below)

Would you consider depositing your working files in the Rose Goldsen Archive?

- 
- Yes
- Yes, with conditions (please note below)  
[ ]
- No (feel free to add any relevant notes)  
[ ]
- 

If this work included links to external websites (not stored on the disc), would you like to discuss possibilities for archiving and providing stable access to those sites?

- Yes
- No
- Not applicable
- 

Have you created new versions or updates of this artwork?

- Yes -- please describe below  
[ ]
- No
- 

Would you consider depositing these updates in the Goldsen Archive?

- Yes
- Yes, with conditions (please note below)  
[ ]
- No
- 

May we contact you to discuss newer versions of your artwork and their relationship to the version we have archived?

- Yes
- No
- 

What were some of your initial thoughts in creating this artwork?

[ ]

---

How was your artistic vision shaped by / constrained by the technologies you used?

---

In what other ways does your artwork reflect the technology of its historical moment?

---

## Block 2

We have found virtual machine emulation to be an effective strategy for providing access to interactive digital artworks to researchers. Running older artworks in an emulation environment may involve changes to the look and feel of the original artwork. Our default access strategy is likely to involve:

- **current commercial-grade hardware and peripherals (mouse, screen, keyboard, etc.)**
- **color shift associated with the change from CRT to LED monitor screens**
- **possible changes in the speed of animation and interactive responses**
- **possible changes to audio resolution and / or quality**
- **presentation of digital surrogates rather than physical materials (discs, booklets, cases, etc.)**

With these considerations in mind, the following questions seek your guidance in developing the best possible access strategy for your artworks.

Please see the potential alterations outlined above. How might these changes affect the initial design and implementation of your work?

We expect to present users with a general statement about the effects of our emulation environments on the rendering of an artwork.

If you would like to author a more specific statement about how these changes may affect your work, we can provide researchers with this information as well. In some cases, we may also be able to provide documentation of original rendering conditions. Please let us know if you would like to discuss these possibilities further.

- I would like to write a statement to provide to researchers
- I would like to discuss additional documentation possibilities with curators
- I am satisfied with the general disclosure described above

With these qualifications in mind, may we have your permission to provide research access to your artwork in an emulation environment? (This could be the only means with which we can make your artwork available to researchers.)

- Yes
  - Yes - with qualifications (please note below)
  - No
- 

Occasionally, users of the Goldsen Archive request access to material for the purposes of creative reuse. Under what circumstances would you support such creative reuse of material from your artwork?

- Creative Commons licensing
  - Case-by-case negotiation between copyright holder and requester
  - Other conditions (please note below)
  - I would not support creative reuse of material from my artwork
-