

Shaun M. Trujillo. Living Legacies: Recovering Data from 5¼” Floppy Disk Storage Media for the Commodore 64. A Master’s Paper for the M.S. in I.S. degree. November, 2011. 97 pages. Advisor: Christopher Lee

In an attempt to investigate the challenges of recovering and preserving digital objects from legacy systems this case study focuses on working with a particular storage medium and computing hardware. This study illustrates the physical and representational challenges that result from recovering data created with a Commodore 64 computer and stored on 5¼” floppy disks. A system for classifying types of digital objects found in the sample of recovered data was developed. This study contributes to the discourse of collecting institutions engaged in digital preservation and provides examples of ad hoc solutions for working through the challenges of recovering meaningful information from legacy systems. The issues that come to light in this study can be extended beyond the context of the Commodore 64 to include other types of digital resources and computing artifacts that will potentially cross the archival threshold in the near future.

Headings:

Data recovery (Computer science)

Digital curation

Personal archives

Emulators (Computer programs)

Commodore 64 (Computer)

LIVING LEGACIES:
RECOVERING DATA FROM 5¼" FLOPPY DISK STORAGE MEDIA FOR THE
COMMODORE 64

by
Shaun M. Trujillo

A Master's paper submitted to the faculty
of the School of Information and Library Science
of the University of North Carolina at Chapel Hill
in partial fulfillment of the requirements
for the degree of Master of Science in
Information Science.

Chapel Hill, North Carolina

November 2011

Approved by

Dr. Christopher Lee

CONTENTS

1. INTRODUCTION	2
2. LITERATURE REVIEW	6
2.1 E-Manuscripts / Digital Curation	6
2.2 Data Recovery and Levels of Representation	8
2.3 Characterization, Affordances, Genre Classification and Triage	12
2.4 Emulation as Digital Preservation	15
3. PROFILE OF THE COMMODORE 64	17
3.1 System Features and Affordances	18
3.1.1 Connectors, Ports and Peripherals	18
3.1.2 Commodore Keyboard and Graphics	20
3.1.3 BASIC Programming Language and the CBM Disk Operating System	23
3.1.4 VIC-1541 Disk Drive and 5¼” Floppy Disks	24
3.2 Characterization of Files	32
3.3 Cultural Context & Resources	33
4. METHOD	35
4.1 Recovery Process	36
4.1.1 Acquiring hardware and software	38
4.1.2 Testing hardware	40
4.1.3 OpenCBM build and ZoomFloppy	41
4.1.4 Creating disk images with OpenCBM	43
4.2 Results	47
4.2.1 Breakdown of Accessions	47
4.2.2 Analyzing the Content of Disk Images	50
4.2.3 Constraints to Accessing User Created Content	58
4.2.4 Maintenance	61
4.3 Triage System	62
4.3.1 Designing the Classification System	62
4.3.2 Administering the Intercoder Agreement Test and Results	65
5. DISCUSSION	67
ENDNOTES	72
REFERENCES.....	75
APPENDIX A – Disk Image Results From Four Accessions of 5¼” Floppy Disks ..	80
APPENDIX B – Online Communities and Resources for C64 Enthusiasts	93
APPENDIX C – OpenCBM Error Codes and Matching CBM DOS Messages	95

1. INTRODUCTION

Before ever opening a file saved to a hard drive or USB thumb stick I accessed data stored on 5¼” floppy disks. My first computer was a Platinum Apple IIe, a “gift for the family” that was purchased by my parents in 1989. A year prior to the arrival of the IIe my best friend was given a Commodore 64 for his birthday, at which point it became my mission to use all of my leveraging power towards acquiring a home computer. My computing routine at the time amounted to loading and playing classic video games like *Dig Dug* and *Burger Time*, and, depending on the minor holiday in question, occasionally printing ugly monochromatic cards with 8-bit pictures of shamrocks or Abe Lincoln’s profile on them. I am indulging in anecdotal reverie to loosely illustrate a point about the early consumer market for computers. Researching this project I have come to appreciate, mostly through the lens of devoted retro-computing fansites and computing publications of the age, that the possibilities offered by these early commercial microcomputers were far richer than I had ever realized. The machines I had miscast as glorified gaming consoles were in fact robust computing systems with scores of specialized devices and software that could equip users to tailor complex and customizable information interactions. Much like the devices of today, the first generation of personal computers were intended for more than playing games; they allowed for the creation and manipulation of unique data and complex digital objects, and, by modem or by mail, they empowered a hybrid network of communication and discourse for the enthusiasts of the day. The continued spark of interest in enthusiast communities regarding the computers

of the 1980s is rooted primarily in nostalgia, but I believe that contemporary users are still interested in pushing the capabilities of these older systems in part because they were so versatile in their own right and many users had not previously been able to experience the full scope of their capabilities. With this work I hope to qualify some of the enduring value of these systems for the broader cultural heritage community.¹

The catalyst for this study was a book authored by Matthew G. Kirschenbaum titled *Mechanisms: New Media and the Forensic Imagination*. In his meditation on the techniques of digital inscription, Kirschenbaum frames digital media as a writing technology and effectively places objects encountered by digital curators within the continuum of literary analysis and forensics. The thrust of Kirschenbaum's analysis relies on the premise that information is *in reality* bound to a physical entity, and that although digital information is medium-independent it can never be wholly intangible.

Philosophically speaking, Kirschenbaum's forensic imagination contrasts the Platonists' position in which the real is situated outside of and beyond phenomenal reality in the realm of supersensible ideas. Rather, in the digital domain, Kirschenbaum locates the real beneath everyday human reality in the realm of subsensible mechanics. In both positions the source of reality is not immediately accessible to individuals who are separated from it by varying degrees of abstraction.

Computer systems are designed to hide what is most essential to their functioning, allowing human users only a requisite margin of interaction and manipulation of the underlying processes that determine a machine's output. As computer technology has grown in complexity over the last century, the abstractions between the human programmer and the inner workings of the machine have broadened as well. Accessing

information via a particular configuration of hardware and software binds it to a set of affordances that act as both ingress and terminus of human interpretation. Archivists working with born digital acquisitions will be charged with characterizing the advantages and limitations of various formats and media types and will be responsible for peeling back the layers of representation that digital objects possess. *Digital archaeology* is a term used to designate the work of curators employing post hoc strategies to recover information from legacy systems and formats (Arms, 2000; Ross & Gow, 1999). Ross and Gow (1999) define digital archaeology as an approach “to accessing digital materials where the media has become damaged (through disaster or age) or where hardware or software is either no longer available or unknown” (p.iii). Comparing aspects of digital curation with archaeology is by no means arbitrary; data can be stratified within obsolete formats, corrupt files are essentially sealed off in a state of disintegration, orphan files remain unrenderable except by means of forgotten ancestral operating systems, and deleted information long believed to be lost can be fossilized on a magnetic disk (Farmer & Venema, 2005). I discovered one remark in an online message board that I believe succinctly sums up all the intricacies, frustrations and tribulations of data recovery and digital forensics: “It's like trying to reanimate a frozen Siberian mammoth.”²

Searching for traces of the creator along with the original function and provenance of records are all tasks familiar to archivists. By reexamining digital information, we can divest from it the reputation of a mutable and chaotic distribution of electrons and can place it within the category of sound physical artifacts. This does not mean that digital objects will conveniently fit into all models of archival preservation and arrangement; that would be too easy. Most of the records preserved in the archives

heretofore were paper based, and could be handled and interpreted directly with very few boundaries other than semantic, linguistic, and dialectical unknowns, loss of integrity of information due to physical disrepair and decay, and possibly poor handwriting. The challenges that archivists face today and tomorrow will be more numerous than ever before, will require skills borrowed and adapted from other disciplines and will foster new alliances across institutions.

In an attempt to investigate the challenges of recovering and preserving digital objects from legacy systems I have conducted a case study that focuses on working with a particular storage medium and computing hardware. In this study I have worked to illustrate the physical and representational challenges that result from capturing data created with a Commodore 64 (C64) computer and stored on 5¼" floppy disks for nearly two decades. *Figure 1* shows the C64 hardware used during the course of this project.



The issues that come to light in this study can be extended beyond the context of the C64 to include other types of digital resources and computing artifacts that will potentially cross the archival threshold in the near future. After collecting the data from the case study, I developed a system for classifying the types of digital objects that I found in the body of data. The purpose of this case study, and the triage system more generally, is to contribute to the discourse of collecting institutions engaged in digital preservation and to provide an example of developing ad hoc solutions for working through the challenges of recovering meaningful information from unfamiliar legacy systems.

2. LITERATURE REVIEW

2.1 E-Manuscripts / Digital Curation

Increasingly, collecting institutions are encountering acquisitions which include born-digital materials. In much of the professional literature, the future of archival collecting is figuratively overshadowed by an ominous cloud of digital media that threatens to produce a deluge of volatile information. This data is at risk of being lost if certain issues are not immediately addressed by the professional community. In recent years any number of collecting institutions have needed to come to terms with the reality of digital acquisitions, including museums and cultural heritage organizations such as historical societies, public libraries, college and university archives, government archives, corporate and institutional archives, digital libraries, and other small records environments. If we add to that list the curation of personal digital objects by individuals in the wider social sphere, material that Jeremy Leighton John – the curator of eMANUSCRIPTS at the British Library – describes as “archives in the wild” (John,

Rowlands, Williams, & Dean, 2010), then we begin to see the wide ranging influence that born-digital media will hold for the future of archives and cultural memory in general.

Personal digital archives (PDA) and electronic records created by individuals for personal use have only recently found a place in the larger professional dialogue of electronic records management. As far back as 1994, Adrian Cunningham, of the National Archives of Australia, had written an article examining the difficulties of preserving personal digital information. Almost six years later Cunningham wrote a follow-up article that addressed the near total lack of response that his article had received, due in part to an ongoing trend in the discourse on electronic records, “which was, and sadly still is, completely dominated by the corporate/organizational records perspective. This lopsided discourse is something I have over the years criticized as ‘corporate myopia’, a phrase denoting the failure to recognize that private individuals also create records” (Cunningham, 1999, ¶3). Notably there has also been little urgency on the part of the manuscript and special collections community regarding digital assets that appear in the collections of current donors, or the variety that are bound to appear in future accessions. Since the early 1990s advances in Internet and computing technologies have led to the increased digital output of individuals from all walks of life. This widening stream of personal digital information has become preponderous enough to command the attention of collecting institutions.

So what are some of the options for collecting institutions already haunted by expanding backlogs and diminishing resources to allocate towards processing analog collections? Can archives afford to appraise the digital records of individuals in order to support access to their relevant and meaningful content, or should they accept everything

and ask questions later in accordance with the notion of benign neglect? Storage media are cheaper every year and thereby increasingly available; it might actually be feasible for archives to store many types of digital objects that were created in recent memory for an interim period during which they are fundraising and creating preservation policies. However, when considering the astronomical increase in the creation of data and speed of its proliferation (and duplication), archivists will inevitably be forced to take in more “bytes than they can chew” if they do not proactively determine a system for appraising and retaining digital objects (Cook, 1991-92; Cook, 2004). Eric Schmidt, a CEO of Google, estimates that the current rate of information being generated every two days roughly equals the sum of all information from the beginning of civilization to the year 2003.³ By Schmidt’s estimation, user-generated content is largely responsible for that staggering statistic, as more and more people are using personal computers and mobile devices to create and organize their electronic lives (Strodl, Motlike, Stadler, & Rauber, 2008).

2.2 Data Recovery and Levels of Representation

Among the concerns that are central to the preservation and curation of born-digital materials is the problem of providing users with access to data created with outdated computing platforms and stored in legacy formats. Transferring this data from its obsolescent storage media into current contexts in which it may be accessed and used is the basic premise of data recovery (Ross & Gow, 1999). Data recovery, or digital capture (John, 2008), utilizes tools and strategies that rely on the physical realities of digital data in order to preserve the integrity and authenticity of digital information on a bit-by-bit basis. The cultural and commercial rhetoric regarding technology, personal

computing and the accelerating growth of information relies, to some degree, on equivocations about the imperceptible nature of information; that is, it confounds the difference between information's invisibility to the human eye and inaccessibility to human touch with actual intangibility. In an amusing article on techniques for destroying paper-based information, Burnett and Randolph explain that "information is... medium independent... (but) definitely not 'immaterial'" (p.71).

Data recovery – the process that results in the accurate reproduction and access of digital objects from data stored on fragile and at-risk media – relies on interacting with and evaluating a series of physical conditions that determine numerous means for interpreting a digital object. In 2002, Kenneth Thibodeau defined digital objects as "information object[s], of any type of information or any format, that [are] expressed in digital form" (p.6). According to Thibodeau, digital objects have characteristics derived from three levels of representation which are described below:

All digital objects are entities with multiple inheritance; that is, the properties of any digital object are inherited from three classes. Every digital object is a physical object, a logical object, and a conceptual object, and its properties at each of those levels can be significantly different. (p.6)

Accordingly, each layer of a digital object's representation presents specific challenges to recovery and preservation practices.

At the most fundamental level, digital information can be interpreted as a form of physical writing, i.e., binary inscription on the surface of a disk in the form of magnetic flux reversals (Kirschenbaum, 2008; Thibodeau, 2002; Woods, Lee, & Garfinkel, 2011). Guttenbrunner, Becker and Rauber (2010) cite the UNESCO guidelines for the preservation of digital heritage to explain that the properties of a digital object inherited

from its physical layer are threatened by decay to the storage medium, i.e., bitrot, as well as by the absence of the necessary hardware to access the medium (p.66).

Kirschenbaum (2008) describes the next class “above” the physical representation of bits as a formal level of executing rationalized code, i.e., “data as it is recognized and interpreted by particular processes and applications software” (p.3). A machine that is processing code is actively reading and writing data in order to perform some action, which could involve the erasure, copying, addressing or modification of a file or other form of data. Here again digital objects are essentially grounded in inscription practices. Notably this second level of representation accounts for the arrangement and allocation of the units of storage, such as blocks, clusters and sectors, which are characteristic to digital storage media (Kirschenbaum, 2008). This arrangement of digital bytes is largely a function of a computer’s operating system and generally takes place outside of the direct control of the user.

Finally, there is the high-level conceptual representation of digital information, which usually takes shape as the user-facing end product of an operating system’s interpretation of digital bits. This level of representation is synonymous with the user interface and is the means of accessing an information object. Some examples of the conceptual layer of an information object are games running on specific platforms, an MP3 played with a particular application, or an uncompressed image being accessed by means of a graphical editor.

The Reference Model for an Open Archival Information System (OAIS) provides a parallel schema for interpreting the levels of representation mentioned above and for understanding the components of an information object (CCSDS, 2002). Based on the

OAIS model, a *data object* can be composed of either a physical object or a digital object. The data object is interpreted by means of associated *representation information*. The representation information can then be variously reinterpreted or translated using a secondary set of representation information or metadata, or it can designate its final output, the *information object*, which is the basic denomination of human-information interactions. In addition to the three classes of representation mentioned above Guttenbrunner et al., again citing the UNESCO guidelines, acknowledge a fourth layer of a digital object. This layer is described as the *essential elements* of a digital object and consists of “the context in which a digital object has been created. This information describing a digital object is usually referred to as *metadata*” (p.66). The OAIS model distinguishes similar contextual information for a digital object’s content and labels it *Preservation Description Information*. PDI includes unique reference identifiers for the content of digital objects along with information for documenting the provenance, context and fixity of such content (OCLC/RLG, 2002).

The conceptual level of a digital object is necessarily integrated with and dependent on its means of access, and therefore accessing a digital object in a new context, whether by converting the data or rendering the original data via emulation, will likely result in changes to the object’s characteristics and will necessitate reinterpretation of its essential qualities. Nonetheless, since determining what qualifies a digital object as genuine and accurate is a judgment call that usually relies on the topmost level of interpretation (i.e., on the level of the information object), multiple possible encodings, or formats, of the same object are possible and can equally preserve the essential characteristics of a digital object (Thibodeau, 2002). One of the challenges of digital

curation is identifying what the essential characteristics of a digital object actually are and then securing those characteristics over time. In the end preserving the essential aspects of a digital object may not be feasible for several reasons including high costs associated with analyzing recovered data and developing systems for emulating legacy files, loss of *ambient data* (Kirschenbaum, 2010) and metadata during migration of files, or simply an inability to capture all of the relevant inheritances that comprise the object. Approaching new and unfamiliar sources of digital information will require that archivists develop methods for drawing out important characteristics of information objects and determining what qualities can be sacrificed and what qualities should remain in order to maintain the fidelity of the preserved object to its original source (Holdsworth & Sergeant, 2000; Heslop, Davis, & Wilson, 2002).

2.3 Characterization, Affordances, Genre Classification and Triage

Abrams et al., cite Adrian Brown in describing characterization as “information that describes a digital object’s format-specific character or significant nature” (p.16). Characterization can refer to the specifications of the medium on which a digital object is stored, the encoding standard of the file types of which it is composed, and the qualities of its use within a specific computing context. Understanding the characteristics of a digital object can provide grounds for interpreting the object in different contexts and help in prioritizing the aspects of the object that should remain across various translations.

Characterization of a digital object is closely linked to the affordances of the particular computing environment by which it is accessed. Affordances are the perceivable qualities of an artifact that enable it to be used in specific ways and which

highlight the nature of human practices surrounding it (Gibson, 1979; Sellen & Harper, 2001; Kirschenbaum, 2008). Tools and artifacts are designed by humans with specific uses in mind, and these uses are integrated into the affordances an object provides. The affordances of an object are footholds for both deeper inference concerning its meaning and use and, in the case of digital objects, for the possibility of automating the creation of characterization information (Kim & Ross, 2007).

Characterization information, similar in many ways to Thibodeau's representation information, is considered highly important to digital repository frameworks for ingesting digital objects (Abrams, 2009; Esteva, Xu, Jain, Lee, & Martin, 2011). In order to preserve the coherence of a digital object over time and across contexts, characterization information must be preserved alongside the object. This means that information important to the integrity of the object, such as header and file type information, checksums, and other authority metadata, must be extracted from various representational levels of the object and stored peripherally in the archives. There are numerous proprietary and open source tools⁴ that can be used to gather and extract preservation metadata from digital objects. Special forensic software such as AccessData's Forensic Toolkit⁵ and Guidance Software's EnCase Forensic⁶ software can be used to create bitstream copies of storage media, aid in analyzing disk images by arranging the bitstream into its original sectors, provide a hexadecimal view of the bytes in a file, and allow for searching across the bitstream for signature file structures and data characteristics.

Simultaneously, larger assertions about the use and characteristics of a digital object can be drawn from its conceptual representation, i.e., its human interface, which

can then be used to organize the object into predefined genre classifications. Developing a system for assigning genres to digital accessions is a high-level approach to understanding the contents of digital collections and efficiently applying an archives' resources towards their further description, processing and preservation. In 2008, the Digital Curation Center published a briefing paper by Abbott and Kim that defined genre classification as "the process of grouping objects together based on defined similarities such as subject, format, style, or purpose" (¶1). Elsewhere, Kim and Ross have conducted several studies that look at automatic genre classification as a means of enhancing the management and preservation of digital objects. In a 2007 study they defined (automatic) genre classification as:

The automatic grouping of documents into distinctive document structures, to be followed by focused metadata extraction from single structural types, as a means of creating a tool capable of extracting metadata across many domains at different semantic levels... Identifying the genre first provides a mechanism to limit the scope of document forms from which to extract other metadata. (p.50)

The fundamental purpose of genre classification is to inform a system of triage that can reduce the space of interactions and analysis of data objects. Additionally, by applying genres to known object types and sets of characteristics, effectively separating those objects from the larger aggregation of data, the unknown quantities within an accession are distinguished and brought to light. In describing the benefits of characterizing digital objects, Esteva et al. explain, "learning what is not known about a collection is an important part of its assessment" (p.46). Additionally they describe the purpose of an assessment system as allowing curators to "go from overviews to detailed examinations, combine variables of interest, identify risk priorities, detect patterns and make preservation recommendations" (p.46). Finally, a triage system can efficiently underscore the parts of an accession that are inessential to the overall collection or even

to the collecting institution at large and can conserve energy that would be exhausted on uncovering the meaning of such objects through complex analysis, effectively reinforcing local collection policies (Kim & Ross, 2007).

2.4 Emulation as Digital Preservation

Emulation is seen as one of several partial solutions to the various problems confronting digital curators. In most cases emulation is discussed in contrast with migration as a strategy for preserving the integrity of born-digital files across time. Migration is a technique in which data in obsolete formats is saved in new formats that are compatible with modern operating systems and software. Migration is a cheaper solution to preserving digital files in the short term, but due to the constant evolution of software and computing technology it involves regular reiterations of the process in order to maintain the data's accessibility (Rothenberg, 1999; Granger 2000). A second shortcoming of the migration strategy resides in its tendency to alter the "look-and-feel" of old files, often losing aspects of original born-digital documents such as fonts, formatting, associated images and graphical data, and other *ambient data* (John et al., 2010). Alternatively, emulation preserves the bitstream of the original media and accesses the legacy data using virtual machine software installed on modern computers. Although emulation tools tend to be more expensive to develop than the initial cost of migrating files to modern formats, emulation has the potential to maintain a higher fidelity to the "look-and-feel" of the original computing environment, along with preserving the integrity of the data in the archival science sense of provenance, authenticity and operational context (Hedstrom, Lee, Olson, & Lampe, 2001; Heminger & Kelley, 2005).

Attempts have been made to empirically test user perception of the benefits of emulation over migration. In 2001, Hedstrom and Lampe reported on a study that asked users to describe their interactions with a retro-computer game as played on an emulator as well as a migrated file on a modern PC, but only after first playing the game on the original computing system. Based on the users' responses there was no major qualitative difference in their satisfaction with the emulated or migrated version of the game, however users resoundingly preferred either version to playing the game in its original context on the actual BBC Micro hardware and peripherals.

Concepts of user satisfaction are commonly linked to studies concerned with preserving outdated digital objects. In 2006, Hedstrom, Lee, Olson and Lampe focused on the acceptance of "look-and- feel" as an indicator of overall user satisfaction when interacting with emulated files, and looked at office documents as well as instances of retro-gaming for sources of insight. An emphasis on user comfort has been incorporated into recent emulation projects and case studies. Technology research initiatives such as the PLANETS project⁷ and the CAMiLEON project⁸ have advanced the concept of remote emulation, which sidesteps the necessity of installing operating systems and applications that are necessary to run legacy executables on local machines, and instead allows users to access files remotely. In addition, it is useful to note that easy access to emulated files will not necessarily account for quality user experiences. Von Suchodoletz and van der Hoeven, 2009, and Woods and Brown, 2010, both point out the necessity of contextualized knowledge of the original operating system and software in order to meet performance expectations of installation and execution of legacy files. Simply put, users forget "how to drive" old systems and will need assistance in accessing and operating

legacy files beyond the initial setup of a virtual machine and emulator (Woods & Brown, 2010).

3. PROFILE OF THE COMMODORE 64⁹

Commodore International Limited, the parent company of Commodore Business Machines (CBM), released several 8-bit microcomputers preceding the C64 including the Commodore PET and the VIC-20. The C64 was released in 1982 and rapidly took over the personal computing market. The main feature that the C64 offered that no other microcomputer in its price range included at the time was 64Kb of random-access memory (RAM). The C64 allowed access to its built-in 64Kb of RAM, 20 Kb of read-only memory (ROM) and numerous input/output (I/O) ports, along with RAM or ROM cartridge extensions, all within a 16 bit address space, by means of its Programmable Logic Array (PLA) chip that allowed for bank switching (Evans, 2010a).¹⁰ In essence bank switching is a technical method for expanding the available memory in a system beyond what is directly addressable by the processor by temporarily repurposing, or switching, the role of memory elements in the system such as ROM or RAM.¹¹ This all goes to illustrate that for its time the C64 was an incredibly versatile machine with powerful processing capabilities available to consumers at a relatively inexpensive retail price of \$595 (Mullish & Kruger, 1984). Researching the C64, I came across numerous reminders that it is, in fact, ranked by Guinness Book of World Records as the best selling computer model of all time, due in part to its long stint on the market, from 1982-1994, during which time CBM sold between 17 and 30 million units (Edwards, 2008). This claim to market dominance gives further proof of the value of this study in the context of digital curation and archives, since a large swath of the personal digital

information generated during the seminal years of personal computing was created with C64 systems and is now at high risk of permanent loss.

3.1 System Features and Affordances

In this section I have included a brief overview of some of the system specifications for the Commodore 64 and introduced some of the primary modes of data creation, transmission and retrieval available to users of the original C64 machines. This is by no means an exhaustive list of hardware devices or interface and programming options for operating the C64, with some of the glaring omissions including the C64's groundbreaking 6581 Sound Interface Device (SID) sound chip and the use of "sprites" for programming graphical animation. Essentially, the list below reflects the ground level interactions and operations one would first expect to encounter when approaching the C64 system from a generalized non-computer science background.

3.1.1 Connectors, Ports and Peripherals¹²

- The **cartridge expansion slot** for ROM-based software and, later in the C64's 12 year run, for various system expansions, aftermarket add-ons and utilities.
- The C2N **cassette interface**, designed to connect the C64 to Commodore 1531 Datassette tape drive. This drive was also compatible with the Commodore PET and VIC-20 models (Evans, 2010a). Unfortunately, I was unable to incorporate the 1531 into my study and the mysteries of storing data on tape will persist until a later date.¹³
- An **RCA port** for video and audio that transmits a modulated radio frequency (RF) signal.¹⁴ Establishing the video signal from a C64 to a television or monitor can be somewhat of a challenge. Initially, while conducting hardware testing for this

project, I assumed that an ordinary RCA cable would work, and plugging the C64 directly into an NEC monitor's video/audio input returned no result, leading me to believe that there was something wrong with the monitor or the computer, rather than the connection. In reality the modulated RF signal requires that the RCA cable be attached to a switchbox that converts the signal to TV antenna. I describe this issue further in the Method section of this study.

- An **8 Pin A/V port** that requires a special composite cable, i.e., the Commodore video cable. This option allows for high quality 16-color video and sound and was ideally designed to connect the C64 to a Commodore 1701 monitor (Evans, 2010a).
- The **Serial port (aka IEC bus)** for communication between the C64 and external devices such as disk drives and printers. The IEC bus was designed to be a proprietary serial-bit version of the industry-standard IEEE 488 bus which was featured on the PET and the VIC-20, and was significantly slower in transmitting data.¹⁵ The slowness of the serial I/O speed was long considered one of the C64's glaring weak spots and was later addressed by the programming community with the introduction of fast loader software that replaced the built-in ROM of the VIC drives as well as expansion cartridges and complimentary fast loader routines included with specific software (Evans, 2010b).¹⁶ The serial connection was designed to meet the TALK/LISTEN protocol of the CBM DOS file system which Jim Butterfield describes in a July 1983 article in *Compute!* magazine:

The bus, like the IEEE, has two modes of operation: Select mode, in which the computer calls all devices and asks for a specific device to remain connected after the call ("Jones, would you stay in my office after the meeting?"); and Data mode, in which actual information is transmitted ("Jones, I've decided to give you a raise")... the main part is: you select a device, and then you send to it

or receive from it. Finally, you shut it off. All devices are connected, but only the one you have selected will listen or talk (as cited in Derogee, 2008, p. 4).

The IEC bus was capable of connecting multiple peripherals through daisy-chaining, with the maximum number of concatenated devices reaching five disk drives and one printer.¹⁷

- The 8-bit **User port** allows for “free style interfacing” (Evans, 2010a), and was designed by CBM to be used with communication devices such as the VICMODEM and the RS 232 cartridge.¹⁸
- Two 9-pin **Game ports (aka Control ports)** that are compatible with Atari joysticks.

3.1.2 Commodore Keyboard and Graphics

According to the *Commodore 64 User's Guide*, the C64 16-color display combines 40 columns by 25 rows to hold up to 1000 characters or memory locations in 320x200 pixel resolution.¹⁹ Each location consists of a byte and thereby can contain a number from 0 to 255 for a total of 256 values that correspond to a graphic character in the PET Standard Code of Information Interchange (PETSCII). *Figure 2* shows the range of available PETSCII characters on the C64 and was taken from the Wikimedia Commons.²⁰

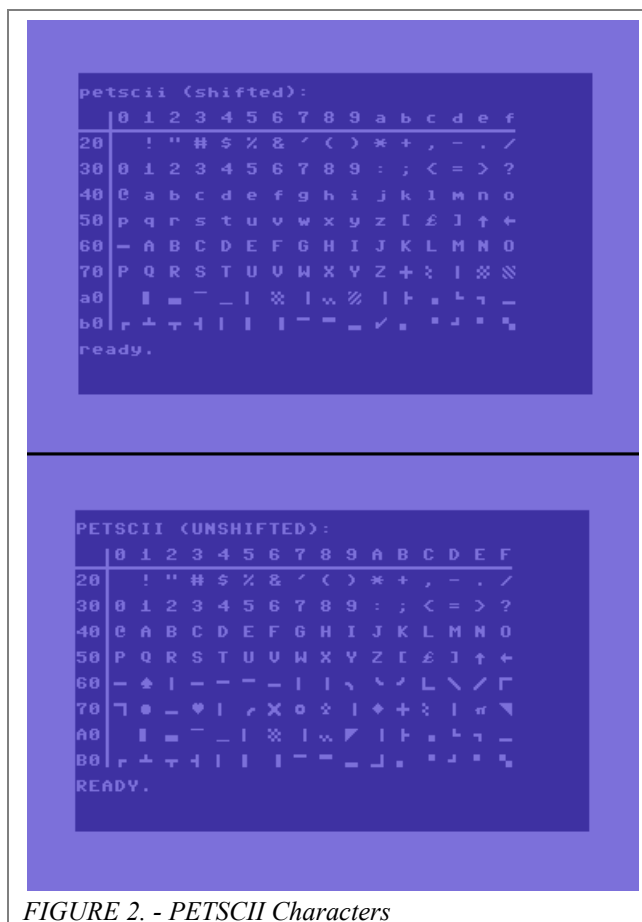


FIGURE 2. - PETSCII Characters

The Commodore 64 uses the PETSCII character encoding set for both textual and graphical purposes, with PETSCII representing CBM's attempt at customizing the 1963 version of ASCII in order to work around graphics limitations of the Commodore PET (Østerby, 2009). Nearly all other information systems that use the ASCII character set are based on the 1967 version.²¹ Like the built-in BASIC Ver. 2.0 programming language, PETSCII for the C64 is a relic from a previous CBM micro-computer, and though PETSCII does allow for creative use of characters for generating graphics as well as screen control codes for reversing characters and changing display colors, converting text encoded in PETSCII to standard ASCII can be problematic and can interfere with some aspects of emulation. The C64 keyboard is made up of the standard 62-key QWERTY

layout overlaid with shift accessible PETSCII characters and bonus programmable function keys. *Figure 3* shows the C64 keyboard and a detail depicting the PETSCII characters displayed on the keys.



FIGURE 3. C64 Keyboard with PETSCII Characters

3.1.3 BASIC Programming Language and the CBM Disk Operating System

As defined by Silberschatz, Peterson and Galvin (1991), an operating system is “a program that acts as an intermediary between a user of a computer and the computer hardware” (p.3). The C64 has two main components to its native operating system - the BASIC programming language and the Commodore Disk Operating System (CBM DOS) which is stored in ROM on the various Commodore disk drives. When one turns on the C64 a simple blue terminal takes shape on the monitor. This is the full-screen command line interface where users implement the Commodore BASIC programming language (aka PET BASIC) in order to write programs and routines, to operate peripherals and most importantly to load software. The built-in BASIC Ver. 2.0 language is the same as what CBM implemented with the VIC-20, and is based on Microsoft’s BASIC language. In 1986, Commodore released the Graphical Environment Operating System (GEOS), a graphical user interface that was also used by the early Apple II family of computers.²² Although worthy of note, the GEOS remains outside of the scope of this project as both the method of testing, extracting, and emulating data was performed with BASIC and CBM DOS commands. Some of the essential disk commands for BASIC, used for writing to and reading from disk devices which were signified with an address number from 8 to 11, are listed below. In the examples below all of the commands are being executed on a floppy drive located at address number 8:

- **Format a new disk:** OPEN 1,8,15,“N0: DISKNAME, ID”:CLOSE 1
- **Load a file:** LOAD “FILENAME”,8
- **Load 1st disk file:** LOAD “*”,8

- **Load and run file:** LOAD “FILENAME”,8,1
- **Save a file:** SAVE “FILENAME”, 8
- **Save and replace:** SAVE “@0:FILENAME”, 8
- **Verify disk file:** VERIFY “FILENAME”, 8
- **Directory listing:** LOAD “\$”,8 (followed by) LIST

CBM DOS is the software that the C64 uses to communicate with and control its various disk drive devices, or, in similar terms CBM DOS abstracts user commands and manages secondary storage media and the data stored on them (Silberschatz, Peterson, & Galvin, 1991).²³ Among the primary tasks of the DOS is the management of free-space on storage media as well as the systematic allocation and retrieval of data to and from memory. The operating system is also responsible for creating and deleting files and directories, and supporting “primitives for manipulating files and directories” (Silberschatz et al., 1991, p.44). The specifics of CBM DOS’s file mapping and space management as it relates to this project will be described in the following section.

3.1.4 VIC-1541 Disk Drive and 5¼” Floppy Disks

The VIC-1541 (aka CBM 1541) is one of several floppy disk drives that were compatible with the C64 and was probably the most popular of the drive options during its heyday. The 1541 is the heart of the data recovery process explored in this study and, in broader terms, the development of firmware and communication protocols between the varying Commodore disk drives and modern day platforms is central to ongoing preservation efforts in the Commodore enthusiast community. The Commodore arrived on the scene shortly before IBM popularized internal hard drive technology with the IBM PC/XT unit.²⁴ For most other PCs of the age, all the action of reading, writing and storing

data took place in an external drive mechanism that existed outside of the main chassis that housed the central processing unit (CPU). Today, someone encountering the C64 for the first time may think that he or she is looking at a keyboard and may wonder where the rest of the computer is hiding. For users who are used to laptops and towers, the separation of disk drive and computer makes less sense than the separation of keyboard and computer. The fact that the CBM disk drives remain easily accessible to human manipulation and are not relegated to the internal space of the computer or hermetically sealed within a drive case is one of the main differences in terms of the affordances that affect a user's behavior in creating and interacting with data. In other words, the role of the user as a physical intermediary in the circuit of data manipulation and the specifics of how the external drives handle data - in terms of the discrete units of storage dictated by optimal performance and space and the means of addressing, accessing and transmitting data from such storage media to the user - affects the way in which data was produced and used.

The VIC-1541 was considered a "smart device" in that not only did it receive commands from the C64, but the drive also had its own integrated CPU chip, 2 kB of RAM, and CBM DOS v2.6 hardcoded in 16kB of ROM.²⁵ The 1541 included two serial ports, one for connecting to the C64 and one for daisy-chaining with another external device. A drive could be assigned a specific number from 8 to 11 with the DIP switch on its back to differentiate it from other connected drives. Most importantly, the single-sided 1541 drive was designed to accept 5¼" floppy disks, the most ubiquitous storage medium of the early 1980s before they were superseded by 3½" floppies. Even though the 1541 read one side of a disk at a time and was therefore originally intended for single density

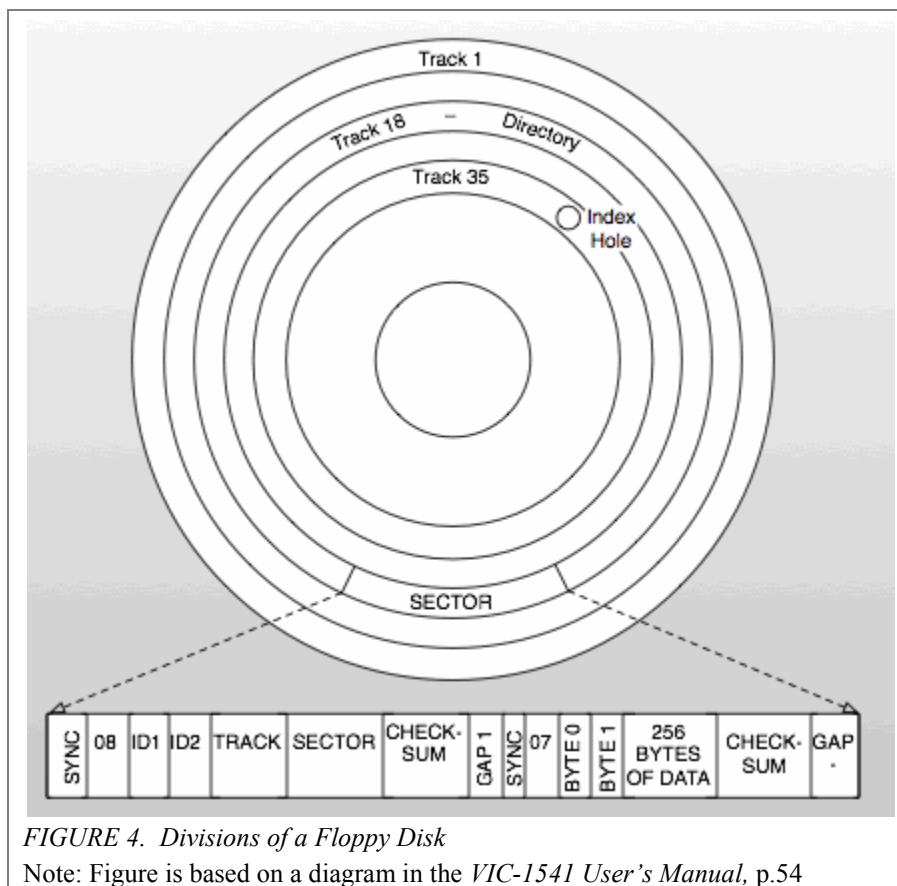
(SD) single-sided floppy disks, many commercial and private-use floppies were double-sided. Most floppy manufacturer's warned against writing data to both sides of a single-sided disk claiming that data loss was accelerated by friction from the internal magnetic disk rotating in the opposite direction than intended.²⁶ However, most users ignored this warning altogether and figured out a simple way of converting single-sided disks into fully functional double-sided disks. While revisiting the 5¼" floppy format, Kirschenbaum (2008) notices the mark of this (almost literal) hack on one of his own disks and infers about the information behavior that resulted from the affordances of the floppy medium:

A portion of the disk envelope alongside the label has been crudely cut away... By mimicking the professionally cut write tab in a symmetrical location I double-sided the disk, signaling to the drive mechanism that the reverse side was also available for data inscription (to access the reverse side you would place the disk in the drive upside down). This was a common trick, and one I was quick to appreciate once I learned that disks had a finite capacity, and... that more disks were what enabled me to expand my software collection (primarily games)... In this instance I can surmise that I double-sided this disk at some point well after its initial purchase in order to store the Koala picture files I was then generating with that piece of software (itself stored on a separate disk), hastily inking in an addition to the label which would allow me to locate them. (p.33)

Here Kirschenbaum has illustrated how the physical characteristics and limitations of a particular medium inform the ways in which people manipulate it and, in turn, these manipulations have an effect on information practices that are identified with the medium. It is important to note that a single program and its supporting files could be stored on either side of a double-sided floppy, as well as on various sides of multiple floppies. Therefore, the means to access a complete digital object may require that the user maintain several floppy disks.

To better understand the process of reading, writing and storing a file generated with the C64, a closer examination of the CBM DOS file system and the storage

specifications of 5¼” floppy disks follows. CBM DOS tells the 1541 disk drive’s controller to inscribe bits of information to the floppy’s magnetic disk in a particular way, known as the group code recording method,²⁷ effectively organizing bytes according to specific physical conditions, i.e., the spacing of magnetic flux reversals. These bytes are then further bundled together, by means of a file system, into logical address spaces that the DOS can recognize and “remember” when the drive is called upon to retrieve or (over)write data. Every proprietary operating system organizes these bundles slightly differently as they require partitioning a specific physical storage medium, e.g., floppy disks, to garner specific results such as speed of retrieval or storing the most amount of data in a limited amount of space. The most basic unit of bundled bytes on a floppy disk is called a *block*, and every file is made up of one or more blocks. Blocks are comprised of 256 bytes, with four blocks equaling 1kB of data. The magnetic disk is divided into discrete concentric sections called *tracks* which are further subdivided into *sectors* once a disk is formatted. The specific track number combined with the specific sector number gives the address of a block of data. *Figure 4* illustrates the divisions of tracks and sectors on a floppy disk and shows the breakdown of header information and data on a single sector:



Commodore's particular schema for Group Code Recording allowed for 174,848 bytes (roughly 170kB) of space on one side of a single-density floppy disk, which was broken into a total of 664 available blocks distributed along 35 tracks divided into 683 sectors.²⁸ Double density floppies held roughly 80 tracks but the VIC drives' read/write heads were too wide to register all of the tracks, so instead they wrote data to every other track, maintaining a total of 35 tracks. Manufacturers of proprietary software took advantage of this physical limitation to create a kind of primitive copy protection for their software, dubbed 'fat tracking', by assigning data to nonstandard tracks and using programmed commands, sometimes included in dongle keys, to force the head to read the disk tracks in an unconventional alignment (Østerby, 2009).²⁹

Peter Schepers (2005) describes the way in which the different sized tracks (smaller on the inside and larger on the outside) were assigned different numbers of sectors, “Commodore, in their infinite wisdom, varied the number of sectors per track and data densities across the disk to optimize available storage” (¶2). *Figure 5* is based on a chart by Schepers that explains the distribution of sectors across tracks:

Storage Distribution Across Tracks			
Tracks	Sectors Per Track	# Sectors	Storage in Bytes
1-17	21	357	7820
18-24	19	133	7170
25-30	18	108	6300
31-40*	17	85	6020
*Tracks 36-40 only on 40 track images		683 (for a 35 track image)	

FIGURE 5. Storage Distribution Across Tracks

The disk drive is a random access device, and therefore the CBM DOS does not assign blocks of a file to successive sectors, but rather assigns data based on an algorithm for conserving space and enhancing the speed of retrieval that depends on the physical mechanics of the spinning drive. This method of file assignment is termed ‘interleaving’ and when CBM DOS looks for available blocks to store a file’s data it skips to every third sector on a track, e.g., t1/s1, t1/s4, t1/s7, etc.³⁰ Interestingly, CBM DOS tries to write as much of a file on a single track as possible for the sake of efficiency. If a particular disk has files with blocks scattered across various tracks it can be inferred that at some point files were erased and overwritten, probably when the user ran out of room on the disk. File blocks that are scattered across multiple tracks represent a simplified form of disk fragmentation,³¹ and also provide a forensic investigator or archival researcher a clue for uncovering hidden information that may be lurking in the *slack*

space of an orphan data block. CBM DOS is able to keep track of each block of data and its relation to other blocks in a 'file chain' as well as locate blocks for writing new data by creating a Block Availability Map (BAM) on every disk. Each floppy disk formatted by CBM DOS has a special track reserved for the BAM, which is always located on the 18th track situated directly in the "middle of the diskette, halfway between the center hub (track 35) and the outer rim (track 1)" (*VIC-1541 Manual*, p.11). Again, the location of the BAM within the center track increases the efficiency of recalling files, since the disk drive read arm will only have to move a maximum of half the entire disk's width in order to retrieve the first block in a file starting from the BAM. More specifically, the BAM is located within sector 0 of track 18 along with the disk name and ID number. Sectors 1-18 of track 18 contain the contents of the disk directory, which includes the names of all the files written to the disk. The BAM and file directory take up a total of 19 sectors. Each of the directory's 18 sectors is capable of holding 256 bytes of data divided among eight 32 byte entries, so that 8 entries by 18 sectors results in a total of 144 separate entries per disk directory. In bytes 0 and 1 of t18/s0 the BAM contains a pointer to the first track and sector of the directory, namely track 18/sector 1. Bytes 4-143 of the BAM keep a record of all the blocks still available for tracks 1-35.³² Byte 0 of a directory entry contains one of five specific files types that correspond with the particular file; these file types are briefly described in the following section. Bytes 1,2 of a directory entry are pointers that tell the file system where to find the specific track and sector of the file's first data block. Finally bytes 3-18 contain the name given to a file by the user at the time it was saved to disk. Using the BASIC command LIST after loading a disk will bring up the disk directory on screen. The character of the first line of the directory consist of either a 0 or

a 1 to indicating the drive number, with 1 only appearing if the disk was loaded from a dual drive unit. The drive number is followed by the disk name, ID number and a number code for the version of DOS used to create the disk. Most 5¼" disks formatted on a C64 have the number 2A in the file directory to indicate that they were created with DOS version 2.6. The following directory entry lines give the file size in blocks, followed by the name of the file and the three letter file type code. The final line in the directory gives the amount of free space in blocks remaining on the disk. *Figure 6* shows the directory for a disk entitled "Games 30," as displayed when the disk is accessed via the C64's BASIC interface and when an electronic version of the disk, i.e. a disk image, is accessed directly with the *Power64* emulator:

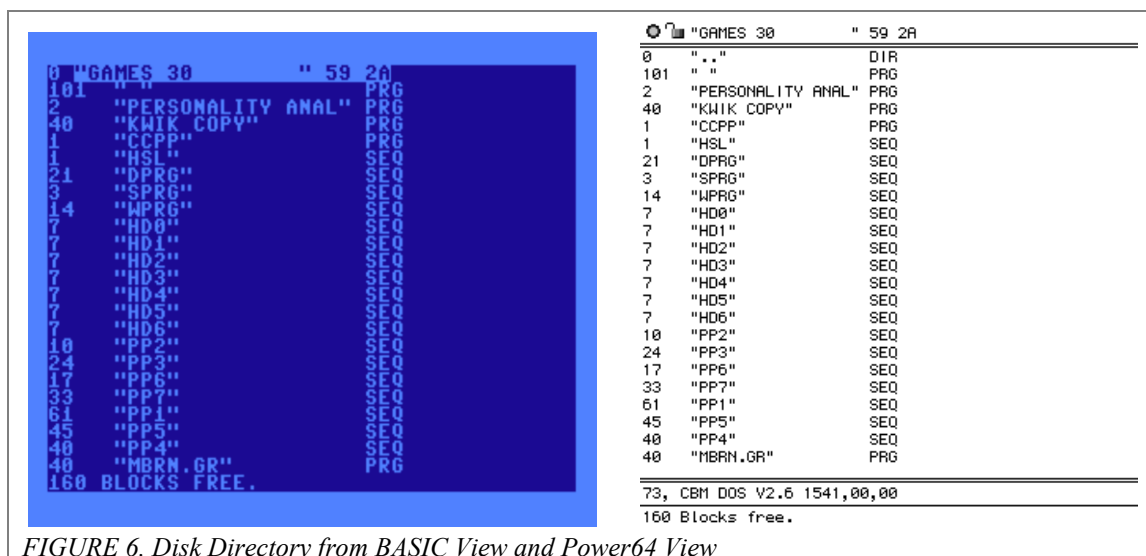


FIGURE 6. Disk Directory from BASIC View and Power64 View

An important omission from the metadata contained in a directory entry and the directory header (bytes 0, 1 and 2) is the time of file creation. This information would be valuable to an archivist attempting to preserve the authenticity and context of a digital object, but because Commodore chose not to include it (due most likely to concerns of conserving limited storage space) preservationists will have to look elsewhere for this

characteristic of the data. Additionally, the relatively small size of SD 5¼” floppies (176 kB) presents further limitations in gathering valuable information associated with files into one place. In most cases commercial programs came with a hardcopy manual for users to understand the various functions and options available for running the program. In many cases these manuals may not be presented with an accession of disks, so that the responsibility of tracking down the supplementary information necessary to fully access files on a disk rests with the archivist. In cases when file names in a directory bear no relation to the software used to create and access them, and when the disk name is obtuse, it may be an intensive process just to identify the original software. In section 3.3 I discuss some of the registries, databases, forums and other resources that can aid in tracking down supporting programs and manuals.

3.2 Characterization of Files

According to the *VIC-1541 manual* (1981), “to the disk drive all files are created equal” (p.19). All C64 files are written the same way using the BASIC programming language. The difference in file types is more or less a formality to signal to the operating system how to handle the contents of a file. The primary difference among the file types listed below is that the PRG files, which usually represent top level applications and application subroutines, may be bootable whereas the others are always handled like flat files and read by the OS during the course of other routines.³³ PRG files also make up the vast majority of files found on a floppy disk as demonstrated in *Figure 13* in the results section.

- **PRG Files** - Program files. They can be either bootable top-level programs or derivative files that run during the course of another routine.

- **SEQ Files** - Sequential files. Linear files that are read from beginning to end.
- **REL Files** - Relative files. “Internal files” (not readily viewable from BASIC command line).
- **USR Files** - User files. Customizable file types intended for DOS expansion.
- **DEL Files** - Undocumented file type for internal use commonly used to customize the look of a disk directory.
- **SCR Files** - Scratch files (not readily viewable from BASIC command line).
Interesting from a forensic perspective since SCR files are basically labeled slack spaces, i.e. blocks of space that are available to be overwritten but that have lingering data allocated within them from a previous deleted file.
- *** Files** - File types preceded by an asterisk denote files that were not properly closed upon their creation. These files are known as poison files and are common causes of data corruption since they crosslink blocks in the BAM and cause file collisions.

3.3 Cultural Context & Resources

Among the identified sources of tools potentially available to digital curators outlined in the *Digital Lives* report, the ancestral computer community is by far the least discussed (John et al., 2010). Perhaps this is due in part to the fact that, barring a few instances of computer history museums and retro-computing exhibitions, not much overlap has occurred between the cultural heritage community and the ancestral computing community thus far. Another reason might be that many communities of enthusiasts that shared knowledge of ancestral systems at one time have become inactive and are disappearing, or the traces of their activity have been dispersed across networks

which would be difficult to collate without considerable research. In many cases online communities can be understood as temporary information grounds (Fisher, Durrance, & Hinton, 2004), and suffer from the same issues of dispersal of content and obsolescence of context that plague PDA curation in general.

Archivist will want to consider how to preserve the contributions of groups like computer enthusiasts, who have tackled relevant problems in the past that are only now confronting archives. Another potential role for digital curators and archivists may be that of network anthropologists, i.e. researchers who trace the communication and exchanges of technology-focused communities across time and work to preserve and interpret their output for present and future uses. It should be a goal of archivists to engage members of these specialist communities to weigh in on considerations for digital capture, appraisal and access.

The number of storage media types that have proliferated from the onset of the personal computing revolution into the last decade is staggering. A short list of fading and forgotten media that can be potentially encountered in PDAs includes but is not limited to: Audio MiniDisc; Jaz cartridges; VHS tapes; Advanced Intelligence Tapes; Iomega's Click! drive; ID-1 Digital Tape; SyJet cartridges; compact cassettes; SmartMediaCards; 3.5" diskettes; BetaMax tapes; CompactFlashCards; and of course 5¹/₄" floppy disks. ³⁴ In an ideal world capturing the authentic data contained on these media is a matter of setting up the right workflows and policies within standardized forensics applications and storage technologies; even so, being able to access the media will present considerable challenges to anyone without knowledge of the original systems and hardware. Furthermore appraising the content of these media may be impossible

without insight into the operations of their original use environments. As mentioned previously, individuals “forget how to drive,” old systems and an operating system merely a decade old can appear alien to users accustomed to modern day interfaces (Woods & Brown, 2010).

The C64 has a devoted following in Germany, Britain and the United States and stands apart from most other retro systems in terms of the level of community engagement and enthusiast networking that still goes on today. There are a huge number of online resources for finding information about Commodore systems, tracking down games, downloading utilities, sharing new custom programs and files, and pursuing other related interests. It’s important to note that new applications and devices are still being developed for the C64 by its hardcore devotees. In addition new products are being introduced that reimagine and derive aspects of the C64 in modern computing contexts or entirely separate cultural domains.³⁵ For more information about resources available to Commodore neophytes and elites alike see *Appendix B* for a list of useful sites.

4. METHOD

This section recounts the process used for recovering data from 122 floppy disks that were attained from four separate sources. I present the results of the recovery process which yielded electronic versions of the original floppy disks in the form of D64 disk image files. Finally, I introduce a triage system for classifying the contents of D64 disk images to inform digital preservation practices, and will test this classification schema using a Kappa statistic for measuring intercoder agreement between two research assistants.

4.1 Recovery Process

In recent years many advances have been made in the development of hardware and software aimed at solving the problems of access, transmission and recovery of data at high risk of loss. These efforts fall under the larger banner of digital preservation, and, while a large portion of those engaged in recovery practices fall into the camp of computer enthusiasts and hobbyists, new technological offerings in the world of data recovery and preservation are increasingly being tailored to the contexts of archives and cultural heritage communities (Elford, Pozo, Mihajlovic, Pearson, Clifton, & Web, 2008).³⁶ The purpose of this project has always been to approach a collection of digital artifacts from the perspective of a small collecting institution, a situation that would require ad hoc solutions to access and preservation challenges and that would not rely so much on building an exhaustive infrastructure for ingest, preservation and long-term access. Nonetheless the implications of my study could be applied to any archives, small or otherwise, that do not have an explicit system in place for recovering data that falls into the domain of digital archaeology. For instance, in 2009 John Blythe examined the workflow for processing digital media at the University of North Carolina's Southern Historical Collection and found that certain media storage types fell outside of the SHC staff's ability to access them. Blythe describes how issues surrounding access to 5¼" disk contributed to the uncertain state of SHC policies for accessioning and processing future collections of floppy disks and similar media types:

The SHC staff member planned to copy the data from the 5¼" floppy disks to CDs... Unfortunately, only two of the 30 floppy disks were readable. The SHC staffer... (was) unable to determine whether the disks had become corrupt or whether there was a problem with the 5¼" drive. After consultation with SHC supervisors, staff decided to leave the disks unmigrated. The brief experiment... was not thorough enough for staff to draw any reliable conclusions about data

capture from such a disk type. Nevertheless, the experiment revived earlier discussions within the SHC about whether the archive should limit the types of digital media that it collects. (p.34-35)

Out of necessity I could not approach these challenges from the prospective of a seasoned specialist, computer scientist or technology enthusiast. Instead I needed to start at the bottom and to search out a case scenario and set of solutions that would be feasible for a digital preservation novice, and to draw out the aspects of that scenario that spoke strongly to larger concerns of archival theory and practices.

Fine tuning the scope of the case study informed the requirements for a technical method of transmitting and capturing legacy data on a modern computing platform. Many of the options for data capture designed for C64 floppies, some of which are listed in *Appendix B*, use software that was developed for Microsoft Windows and that requires some hardware customization of the 1541 drive. I decided to use the ZoomFloppy device because it required no specialized installation of an I/O port, as its firmware takes advantage of the native serial connection on the C64, and because it was a newer mechanism that had benefitted from the knowledge developed while testing earlier iterations of similar intermediary devices. Lastly, the C64 emulation software I found to be the most user-friendly, at least in terms of installation and basic navigation, was the Mac-based *Power64* emulator. Therefore, I tried to consolidate as much of the data recovery process and subsequent data analysis as possible to the Mac interface. Later, I will discuss how making use of some of the specialized software created by Commodore enthusiasts required that I also use Windows as a means to manipulate D64 disk images and individual files.

4.1.1 Acquiring hardware and software

I began with the goal of working in some capacity with the Commodore 64 and data on 5¼" floppies. The initial stage of preparation involved gathering the actual legacy hardware and software from a variety of sources. I had already obtained two C64's, a VIC-1541 drive, joystick controllers and a small set of floppies at a yard sale the year prior to entering the Master's Program at UNC. The gear had been sequestered in my closet ever since and had never been tested to see if it would awaken from its slumber. In a way this is the dilemma of any one, either hobbyist or professional, who sets out to do something they have either never attempted before or have not tried in some time. In the setting of the archives or museum one may have hardware available that could potentially access software and various data stored on legacy formats, but the question remains of whether that equipment will even function today, and more importantly, whether one is capable of getting it to work or remembering how to operate it.³⁷

Parallel to the uncertainty surrounding the state of available legacy hardware, the prospect of acquiring sets of random 5¼" floppy disks provides dual concerns in terms of the media's integrity and provenance. Testing floppies (and other media generally) requires knowing how the disks were originally formatted, i.e., knowing what machine(s) and system(s) the disks initially ran on and with which they are compatible. Even if the disks have survived over the years with a significant amount of data intact, there is still no guarantee that one will be able to read the data, unless one makes special arrangements to acquire multiple drive devices that can read versatile formats or a single drive that is format agnostic. Looking at a lot of nameless floppy disks for auction on eBay I was left to wonder if they were created by a machine that was PC DOS or Apple DOS based, or

whether they were actually written on a CBM machine, or "a CP/M machine, like a Kaypro. Or (they) could be from a Microbee, which almost every public school had in the 80s" (Lavo, 2010). It is also possible for the disks to be completely blank or in such bad disrepair that they are completely unreadable. Østerby (2009) cites numerous sources that estimate the outside lifespan for floppy disk media is roughly 15 years, which means that disks created in the twilight years of the Commodore market have already surpassed their drop-dead date (Levy, 1998; Rothenberg, 1999).

Fearing that, after jumping headlong into a data recovery project, I had already misstepped by putting off proper testing of the hardware at hand, I preemptively bought another C64, 1541 drive, and bonus NEC character display monitor from someone I had contacted on the Raleigh/Durham/Chapel Hill chapter of Craigslist. I was warned when I bought the gear that "everything turns on, but the monitor doesn't seem to work; maybe it's a broken video chip in the C64?" The full extent of testing that this system had received before I acquired it was a basic power check, which is often the case for hardware bought remotely via eBay. I bought all three pieces for \$30. As extra incentive I asked if there were any floppies left over the owner would be willing to part with and received a set of 41 disks, the contents of which the owner explained, "got him through high school," presumably in the mid 1980s. In the end buying this backup system proved to be completely unnecessary, but on the upside the disks that came with the setup turned out to be, from an archival perspective, the most interesting collection of files that came into my possession.

At the start of this project my entire inventory of equipment amounted to three Commodore 64 computers; two VIC-1541 disk drives (one Alps Drive and one

Neutronics Drive);³⁸ one NEC Character Display Monitor; one ZoomFloppy device; and 122 random floppy disks from four separate ‘accessions.’

4.1.2 Testing hardware

Before connecting the 1541 to my MacBook via the ZoomFloppy, I needed to test whether the drives even worked. If I had installed the supporting software for interfacing with the 1541 first and the drives were inoperable then there would be no way to tell whether the problem was due to the bad drive or a bad installation. I needed to logically discern the best order of operations for testing and connecting all of the components of the ‘recovery station.’ It turned out that there were dependencies between the linked hardware and that each device relied on another device being functional in order to be tested. The chain involved first testing the monitor, then testing the C64 computer, and finally testing the most important component, the 1541 drive.

Testing the NEC monitor presented an issue. As the seller predicted, no picture came up on the display once everything was turned on. Anticipating that something could be wrong with the C64 from the Raleigh setup I decide to test the original ‘yard sale’ C64 with the Raleigh (Alps) 1541 drive. The Raleigh unit came with an adapter that converts the Commodore A/V 15 DIN output to an early form of S-Video by separating out the chroma signal from the luminosity signal. The reason that the monitor was not picking up any signal was because the adaptor arrangement was not complete. After searching the forums I discovered that another connector, commonly found at Radio Shack, was needed to recombine the video signal into a single composite output that could then be connected to most monitors.³⁹ I did not have the composite connector on hand, so I decided to go a different route by locating a modulated radio frequency (RF) output converter to test the

picture using the C64's RCA output. Additionally, since the classic NEC computer monitor had no RF input, I needed to test the video signal using a television that accepted an antenna input.

Finally I could turn on the C64 and get a picture. The Commodore turned on without a hitch and slowly the classic blue and white BASIC command terminal came into focus. A mere second or two after flipping the On/Off toggle the system is ready to run a program.⁴⁰ I had now confirmed that one of the Asheville C64s worked, but the Alps 1541 drive's red error light was flashing. After a new round of forum crawling I decided that this drive was either permanently damaged or beyond my immediate ability to repair. My only hope was that the Neutronics drive that came with the original 'yard sale' system would work; otherwise I would be forced to acquire more drives via eBay. Luckily the Neutronics drive functioned perfectly, as I was able to load a disk, view the contents of its directory and take a (rather extensive) personality test.

In the end testing revealed that all three C64 computers, the NEC monitor, and the Neutronics 1541 disk drive were functional. The only lemon was the Alps 1541 drive and, after verifying that I had a working disk drive, I no longer even needed the C64 computers to perform data recovery. I could now move on to building the OpenCBM library and ZoomFloppy interface necessary for transferring data from the floppy disks to my MacBook.

4.1.3 OpenCBM build and ZoomFloppy

The ZoomFloppy device allows users to connect an IEC (Serial) device such as the VIC-1541 to a modern computer via the USB port. The ZoomFloppy is manufactured by Jim Brain and is built from a version of the xum1541 firmware (pronounced "zoom")

developed by Nate Lawson with help from Wolfgang Moser, Spiro Trikaliotis, and Christian Vogelgsang and provided under the GNU General Public License.⁴¹ In order to control the ZoomFloppy it is necessary to compile the OpenCBM library that includes command line tools that communicate with the VIC drive's IEC bus "at the level of simple TALK and LISTEN commands, similar to the one [sic] provided by the Commodore kernel routines" (Klien, Trikaliotis, & Moser, 2007). Compiling OpenCBM for Macs requires an installation of MacPorts and it is important to run an MD5 checksum on the portfile after downloading the MacPorts and OpenCBM packages in order to confirm that the installation file is both complete and identical with the file described in the installation guide. My first attempt at compiling OpenCBM failed because I had downloaded a PC version of the OpenCBM portfile from SourceForge not realizing that, at the time, the Mac version was only hosted at Chris Vogelgsang's website.⁴² By comparing the MD5 hash of the download with the one provided in the installation documentation it was clear that I had downloaded the wrong file. Once I had successfully compiled OpenCBM and connected the 1541, the ZoomFloppy and the MacBook, the real fun of the process began. *Figure 7* shows the ZoomFloppy device connected to a 1541 disk drive via an IEC serial cable and a MacBook via a mini USB to USB cable.

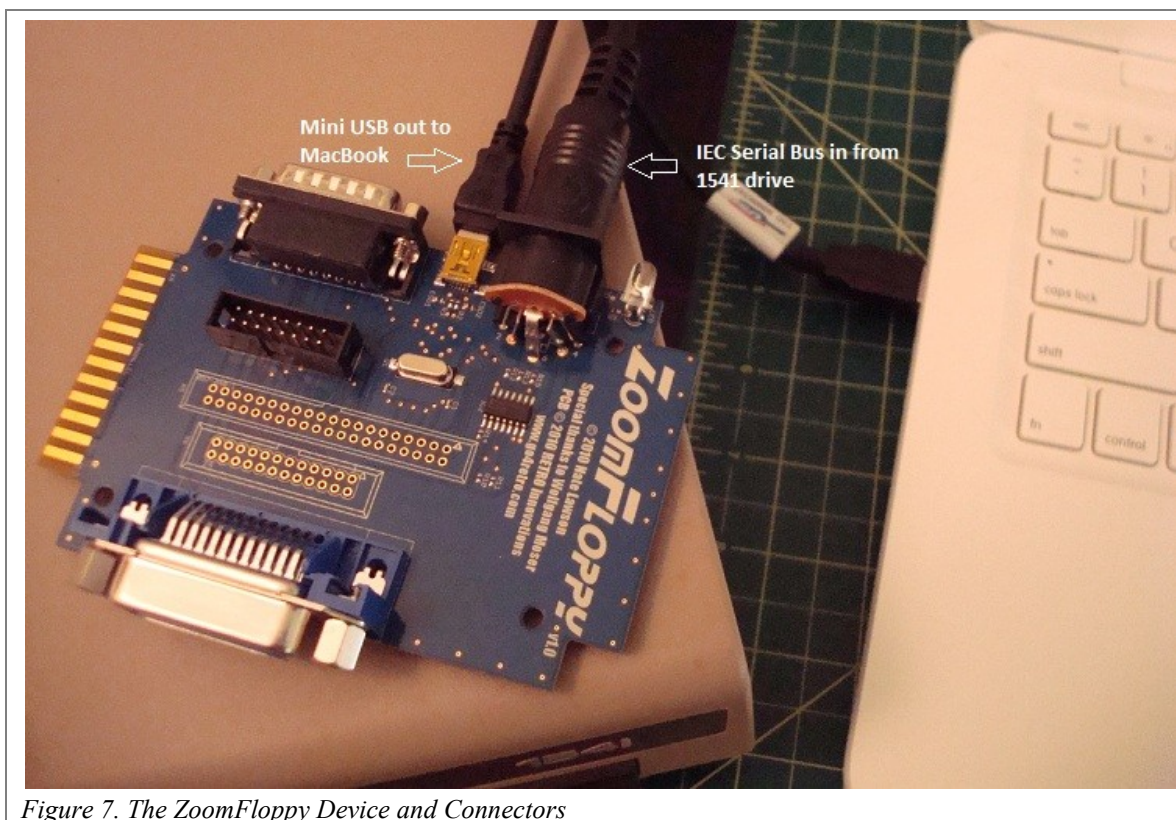


Figure 7. The ZoomFloppy Device and Connectors

4.1.4 Creating disk images with OpenCBM

The basic electronic equivalent of a 5¼” floppy disk is a D64 file. This is the file format that OpenCBM creates when it copies the bitstream of a disk on the local drive of the Mac. Peter Schepers (2005) discusses the D64 disk image among other formats and explains that it “is basically a sector-for-sector copy of a 1540/1541 disk... The standard D64 is a 174848 byte file comprised of 256 byte sectors arranged in 35 tracks with a varying number of sectors per track for a total of 683 sectors” (¶1). D64 images are not the only kinds of electronic floppy disk files used by emulation programs, but they are probably the most common C64 files available on the Web, with most of the online archives and enthusiast sites discussed in section 3.3 sharing recovered floppies in the D64 format. There are alternative 5¼” image formats that are proprietary to specific

emulators or that highlight different aspects of the original Commodore files or VIC drive types.⁴³ The main (negative) difference between the D64 and its floppy disk source is the loss of header information for individual sectors, which includes the ID number and checksum of the sector. The checksum of the sectors are taken into account at the time of conversion, when the disk drive is reading the actual floppy disk, but are not preserved in the D64 bitstream. The exclusion of the ID number can also have an effect on copy protected software, making emulated versions of the disk unusable.⁴⁴

With the use of OpenCBM commands on the Mac Terminal it is very simple to create D64 disk images. The regimen of OpenCBM commands consists of “cbmctrl detect” and “cbmctrl status” for checking the connection to the 1541 drive; “cbmctrl dir 8” for reading the contents of the disk’s directory; and finally “d64copy 8 *filename.d64*” for copying the contents of the number 8 device, i.e., the disk drive, to a D64 file within the Mac user’s local folder. *Figure 8* below shows the Terminal output resulting from the creation of a D64 disk image with OpenCBM.

```

Last login: Tue Aug 9 18:14:51 on ttys000
ubik:~ shauntru$ cbmctrl detect
 8: 1540 or 1541
ubik:~ shauntru$ cbmctrl status 8
00, ok,00,00
ubik:~ shauntru$ cbmctrl dir 8
0 ."games 30      " 59 2a
101 " "          prg
 2  "personality anal" prg
40  "kwik copy"    prg
 1  "ccpp"        prg
 1  "hsl"         seq
21  "dprg"       seq
 3  "sprg"       seq
14  "wprg"       seq
 7  "hd0"        seq
 7  "hd1"        seq
 7  "hd2"        seq
 7  "hd3"        seq
 7  "hd4"        seq
 7  "hd5"        seq
 7  "hd6"        seq
10  "pp2"        seq
24  "pp3"        seq
17  "pp6"        seq
33  "pp7"        seq
61  "pp1"        seq
45  "pp5"        seq
40  "pp4"        seq
40  "mbrn.gr"    prg
160 blocks free.
00, ok,00,00
ubik:~ shauntru$ d64copy 8 disk.d64
[Warning] growing image file to 683 blocks
 1: *****
 2: *****
 3: *****
 4: *****
 5: *****
 6: *****
 7: *****
 8: *****
 9: *****
10: *****
11: *****
12: *****
13: *****
14: *****
15: *****
16: *****
17: *****
18: *****
19: *****
20: *****
21: *****
22: *****
23: *****
24: *****
25: *****
26: *****
27: *****
28: *****
29: *****
30: *****
31: *****
32: *****
33: *****
34: *****
35: *****
683 blocks copied.
100% 683/683

```

FIGURE 8. Terminal Output for Disk Image Creation

As the D64 file is created the Terminal displays the sector-by-sector progress - an asterisk appears if the sector is copied successfully and a question mark appears if the sector is skipped. If all goes well, the user sees 683 asterisks divided accordingly between 35 tracks and a final message stating “683 blocks copied.” When there is a read error the OpenCBM reports the track and sector where the error occurred and attaches an error code for the sector at the end of the disk image.⁴⁵ The error code matches the built-in CBM DOS error codes for the 1541 drive, and effectively replaces the missing sector header information mentioned above, but does so without creating a new checksum for the sector. If OpenCBM encounters continuous errors on a single track it will eventually give up copying the track and move on to the next. See *Appendix C* for the full list of

CBM DOS error codes. *Figure 9* below shows an example of an attempt to create a D64 image with OpenCBM that resulted in read errors.

```

ubik:~ shauntru$ d64copy 8 easyscript.d64
[Warning] growing image file to 683 blocks
 1: -*-----?-----      0%    1/683[Warning] read error: 01/0c: 4
 1: -*-----?-----      0%    2/683[Warning] read error: 01/02: 4
 1: -*?-----?-----      0%    3/683[Warning] read error: 01/0d: 4
 1: -*?-----??-----     0%    4/683[Warning] read error: 01/00: 4
 1: ??-----??-----     0%    5/683[Warning] read error: 01/0e: 4
 1: ??-----???-----    0%    6/683[Warning] read error: 01/03: 4
 1: ??*?-----???-----   1%    7/683[Warning] read error: 01/0f: 4
 1: ??*?-----????-----  1%    8/683[Warning] read error: 01/04: 4
 1: ??*???-----????----- 1%    9/683[Warning] read error: 01/10: 4
 1: ??*???-----?????----- 1%   10/683[Warning] read error: 01/05: 4
 1: ??*???-----?????----- 1%   11/683[Warning] read error: 01/11: 4
 1: ??*???-----?????----- 1%   12/683[Warning] read error: 01/06: 4
 1: ??*???-----?????----- 1%   13/683[Warning] read error: 01/12: 4
 1: ??*???-----?????----- 2%   14/683[Warning] read error: 01/07: 4
 1: ??*???-----?????----- 2%   15/683[Warning] read error: 01/13: 4
 1: ??*???-----?????----- 2%   16/683[Warning] read error: 01/0a: 4
 1: ??*???-----?-?????----- 2%   17/683[Warning] read error: 01/08: 4
 1: ??*???-----?-?????----- 2%   18/683[Warning] read error: 01/14: 4
 1: ??*???-----?-?????----- 2%   19/683[Warning] read error: 01/09: 4
 1: ??*???-----?-?????----- 2%   20/683[Warning] read error: 01/0b: 4
 1: ??*???-----?????----- 3%   21/683[Warning] giving up...
 1: ??*???-----?????-----
 2: *****
 3: *****

```

FIGURE 9. Terminal Output for Read Errors

In some cases the 1541 drive is unable to access the contents of a disk, either due to degradation of the media or because the disk is either unformatted or formatted in a way that cannot be interpreted by CBM DOS. Trying to access the directory of a problem disk will result in the 1541 transmitting an error code via the ZoomFloppy to be displayed in the Mac Terminal. These errors are synonymous with those listed in *Appendix C*.

4.2 Results

Below are the results of the entire data recovery process. In order to avoid confusion when discussing parts of the collection, I refer to the total corpus of 122 disks as the project “library,” and have kept the library divided in four discrete accessions.

4.2.1 Breakdown of Accessions

For each set of floppy disks I received I assigned an accession number and maintained the original order of the disks. *Figure 10* shows the quantity of disks for each accession, as well as the total number of disks and discrete sides in the library.

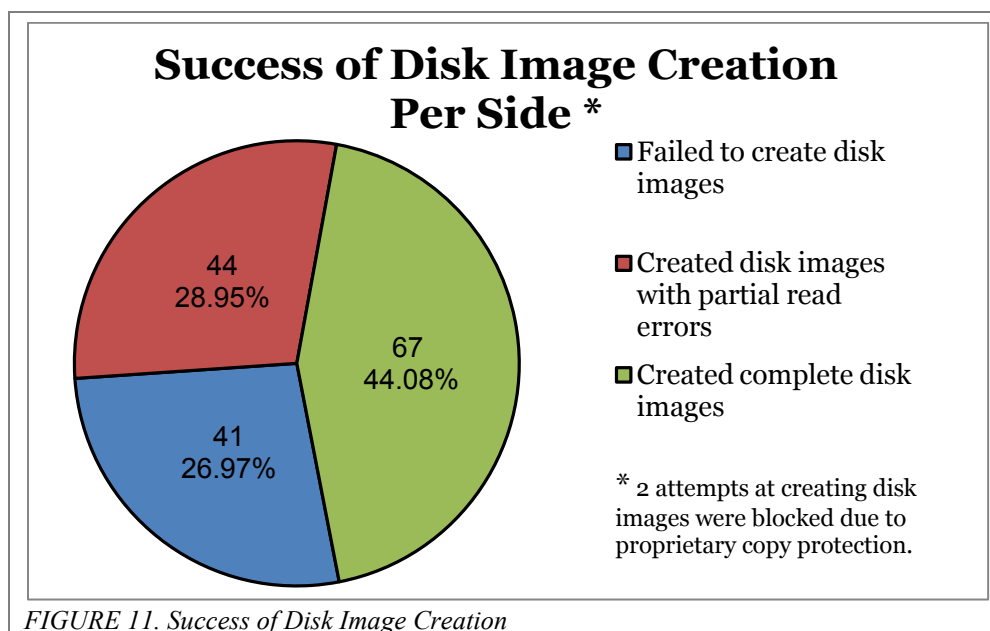
Total No. of Disks and Sides				
	Single Sided	Double Sided	Total Disks	Total Sides
Acc. #1	6	3	9	12
Acc.#2	10	5	15	20
Acc.#3	45	12	57	69
Acc.#4	31	10	41	51
Total	92	30	122	152

FIGURE 10. Total No. of Disks and Sides

Accession #1, which was purchased at a yard sale along with a C64 system in October of 2009, included nine 5¼” disks, three of which were double-sided. Accession #2 and #3, both won on eBay in April of 2011, contained fifteen and fifty-seven 5¼” disks respectively. Accession #2 included five double-sided disks, whereas accession #3 included twelve. Finally, accession #4, found on Craigslist and purchased in Chapel Hill in August of 2011, included a total of forty-one disks, ten of which were double-sided. The combined library amounted to 122 disks with a total of 152 discrete sides.

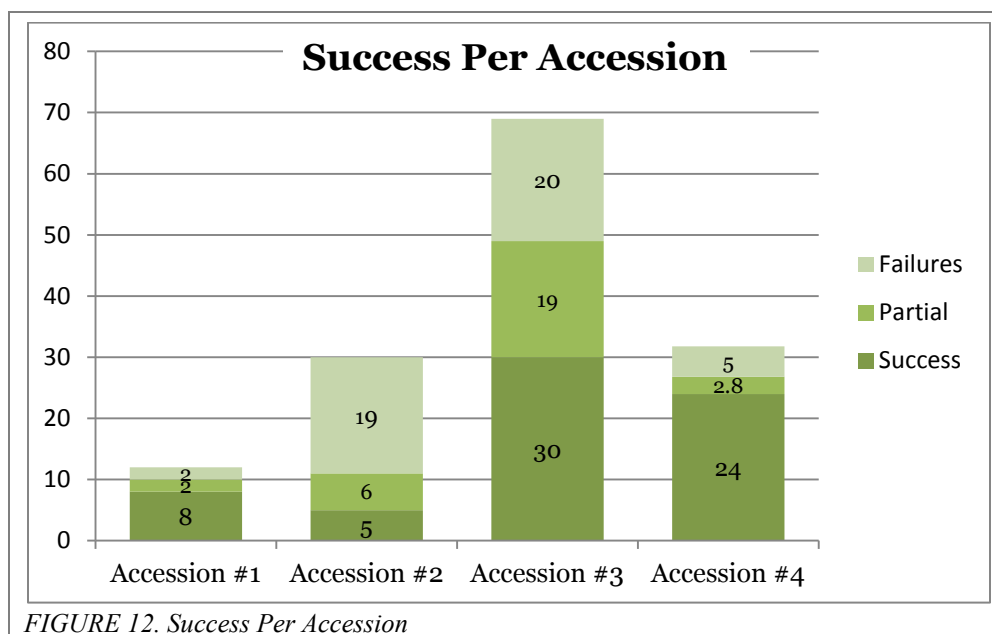
I made at least one attempt to create a D64 disk image for the contents of every side of every disk in the library. *Figure 11* shows the resulting percentages of successful

image creation, corrupted image creation due to partial read errors and failure due to total read errors for each.



Of the total 152 sides of data I was able to create 111 disk images. Sixty-seven disk images represent complete copies of the bitstream from the original floppy disk. Forty-four of the images include one or more sectors of corrupt data that could not be confirmed by checksums during the transfer process; the component files may still be accessible, however, in part or in whole via emulation. Twenty-seven percent of the disk library, or forty-one disks, remained completely inaccessible and failed to be copied. The overall success rate for complete disk image creation was 44.07 percent while the overall failure rate was 26.97 percent.

Breaking down the library to its subsequent accessions we see varying rates of successful data recovery for each. *Figure 12* shows the total number of entries, i.e. sides, in each accession and the number of successfully completed disk images.



Accession #1 included twelve possible entries, eight of which were fully copied for a 66.7 percent success rate and only two of which failed for a 16.7 failure rate. Five out of thirty entries in Accession #2 were completely copied for a 16.7 percent success rate while nineteen failed resulting in a 66.3 failure rate. Accession #3, which included the most entries, yielded thirty complete disk images out of a possible sixty-nine for a 43.48 percent rate of success, while twenty entries remained inaccessible for a failure rate of 28.98 percent. Accession #4 consisted of fifty-one entries, twenty-four of which were perfectly copied for a 47.06 percent success rate, and nine of which failed resulting in a 17.64 failure rate. The average success rate between all four accessions was 43.46 percent, slightly higher than the overall success rate for the library. The average failure rate was 32.42 percent, again slightly higher than the overall failure rate. Accession #1 represented an obvious outlier to the rate of success, most likely because it consisted of a high ratio of out-of-the-box proprietary software, with six out of nine of the disks (66.6%) having official manufacturer's labels on them. Floppy disks are known to be an

unreliable medium in general, and in many cases users made multiple back-ups of important data to protect against loss due to sectors of a disk going bad after repeated use. It was not uncommon for users to create use copies of developer-made software on blank disks and to store the original disks for long-term backup.⁴⁶ Since proprietary disks were expensive and rarely overwritten or erased to store new data, there is less likelihood that sectors were damaged or bits were lost from data being poorly copied to the disk. In turn accession #2 represented an outlier to failed results; a potential reason being that many of the disks returned specific error messages indicating that the disk was either blank (unformatted) or not formatted for CBM DOS. Out of fifty-seven disks only five (8.8%) were proprietary; of those five, three failed to copy, one returned read-errors and one was successfully copied.

4.2.2 Analyzing the Content of Disk Images

One of the first things I noticed when looking at the directories of the fully intact disk images was the abundance of files on each disk. This is not to say that there were a large amount of programs per disk, rather there were on average four to five bootable programs each of which were followed by a series of dependent files that were necessary to run the program. Even disks that were dedicated to a single program could have upwards of 100 files. User-generated content created by a program was rarely stored on the same disk as the program and some disks were dedicated to nothing but subcomponents of other applications such as font and clip-art files, spreadsheets and data templates. Some program subcomponents, such as individual fonts, were saved as USR files and could only be viewed or accessed using specific programs. *Figure 13* below shows a breakdown of the number of files by file type present in each accession.

	Disk Images	PRG	SEQ	REL	DEL	USR	SCR	*PRG
Acc. #1	8	562	18	0	0	0	9	0
Acc. #2	5	42	1	0	3	0	6	0
Acc. #3	30	476	54	0	2	137	24	0
Acc. #4	24	588	29	0	42	20	16	4
Total	67	1668	102	0	47	157	55	4

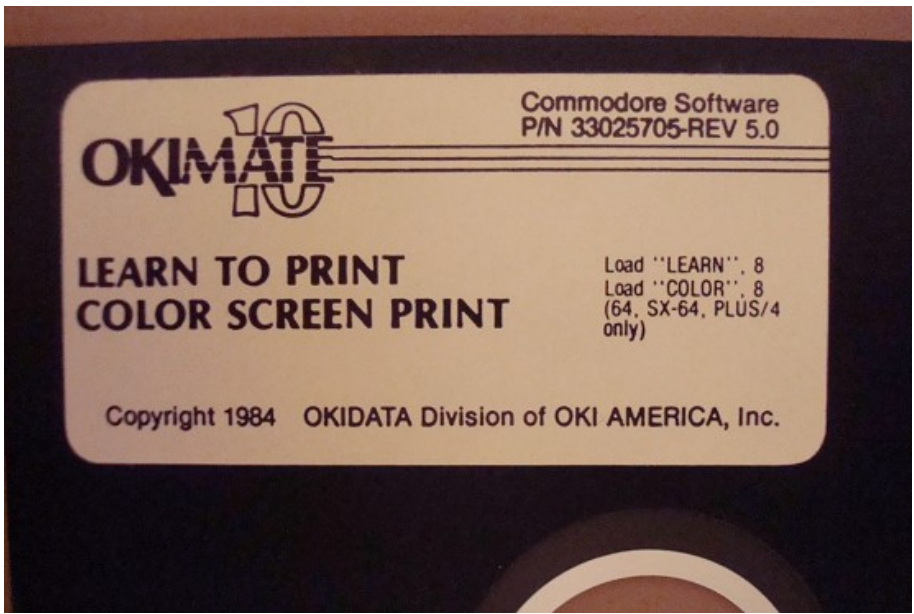
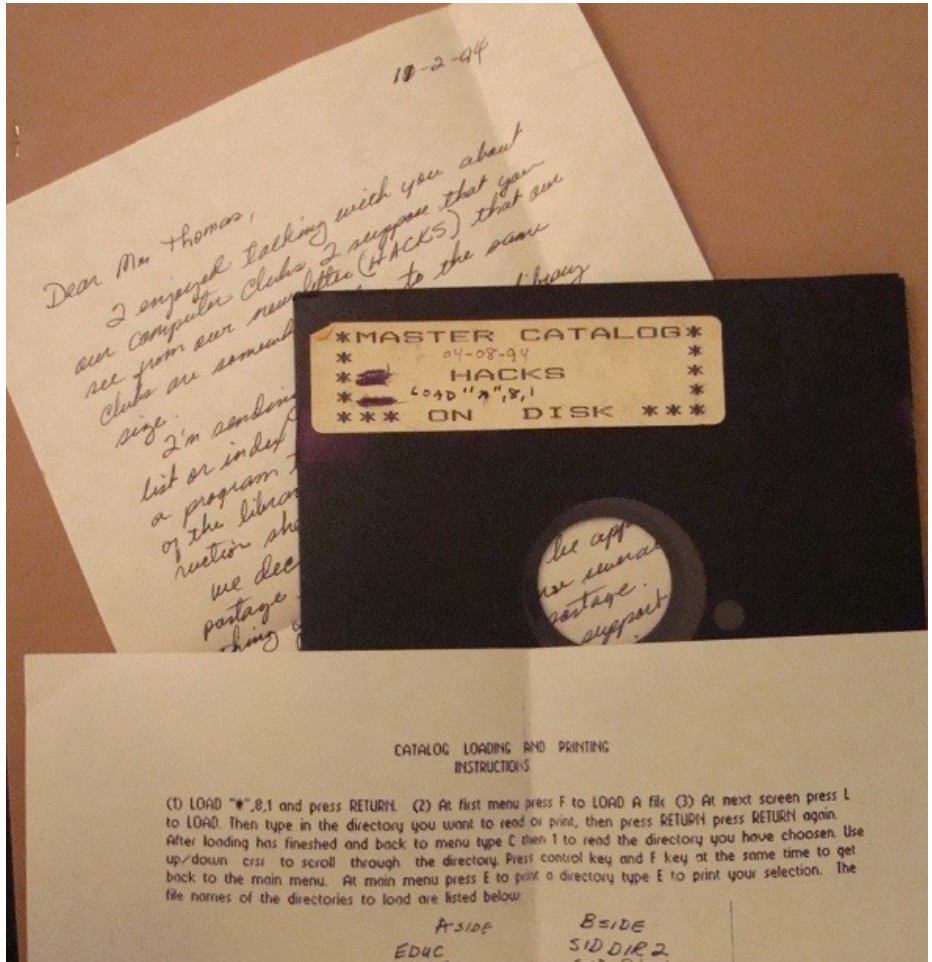
	PRG	SEQ	REL	DEL	USR	SCR	*PRG
Acc.#1 Avg./Side	70.25	2.25	0	0	0	1.13	0
Acc.#2 Avg./Side	8.4	0.2	0	0.6	0	1.2	0
Acc.#3 Avg./Side	15.9	1.8	0	.07	4.6	.8	0
Acc.#4 Avg./Side	24.5	1.2	0	1.75	.83	.67	.17

FIGURE 13. Files Types by Accession and Average File Type per Side

With only sixty-seven complete disk images, each carrying at the most 167 kB of data, I was still confronted by the difficulty involved in interpreting their content. Using the *Power64* emulator was useful only for listing the contents of a disk's directory and for booting certain games and utilities. Very few, if any of the disks provided directions for using their programs or accessing their files. Traces of useful metadata describing the function and use of a particular file at the *conceptual level* (Thibodeau, 2002), i.e., instructions and contextual information within programs or menus, were also difficult to find. Where this was the case, I had to search for hints that could lead me to the additional resources required to access particular files.

The question boiled down to "how could I fill in the blanks?" In the end there were three primary sources I turned to in order to make headway with a file. The first source was the disk label, which may seem obvious, but could be overlooked once the

digital contents of a disk were separated from the physical artifact of the floppy disk. The label on the floppy disk was always a good place to start making inferences about the contents of a disk. Among other examples of disk label metadata, *Figure 14* shows an instance in which the disk sleeve of a particular floppy contained a handwritten letter from one Commodore enthusiast club to another that when read carefully explained the context of the disk's creation, its contents and the exact instructions for interacting with the files on the disk. All the examples below include information regarding the content and use of the disks. The metadata include instructions for operating programs, booting programs, updates to program features, and listings of the disk's directory contents.



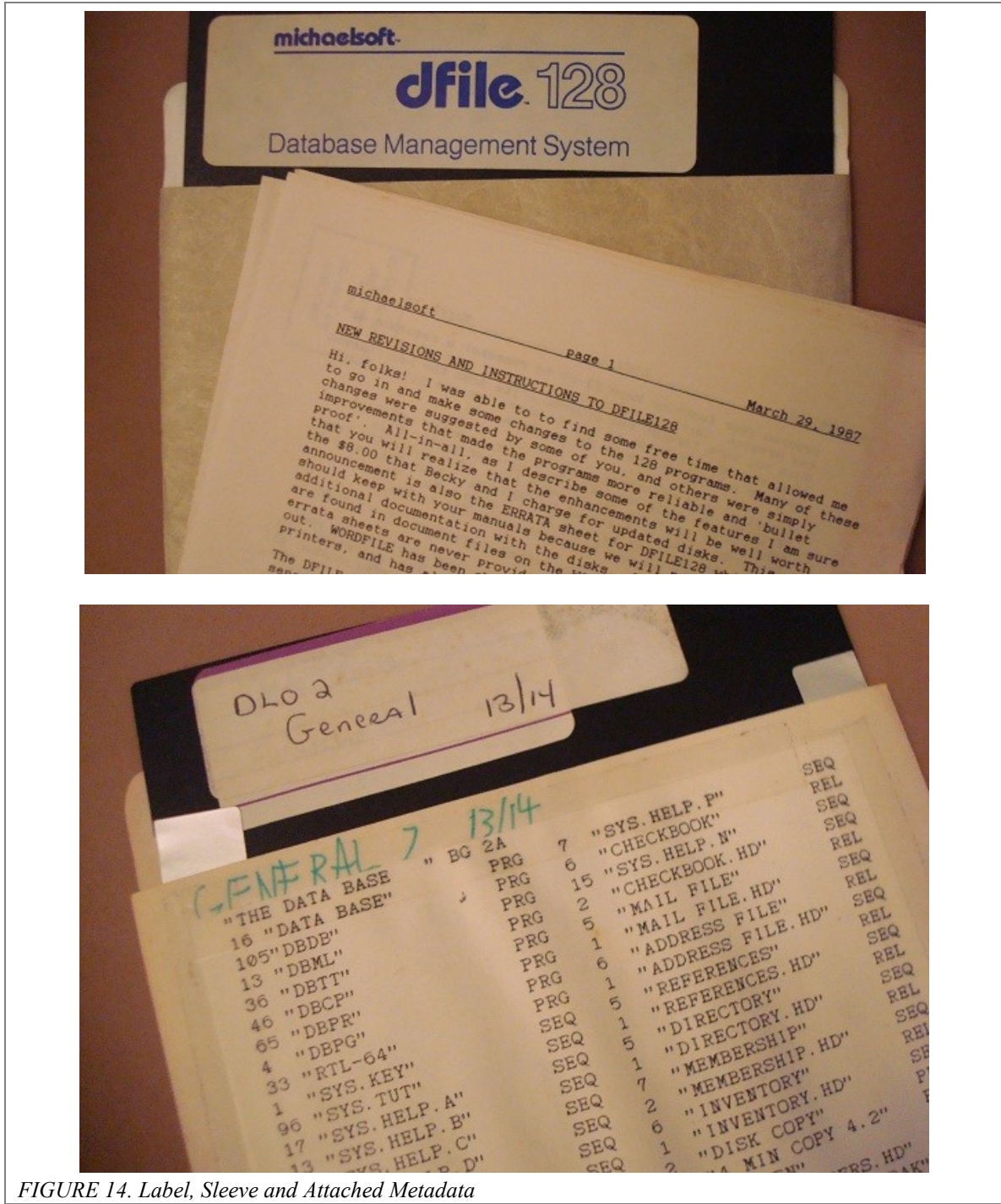


FIGURE 14. Label, Sleeve and Attached Metadata

In some cases a disk would contain mostly user generated files, or templates, or fonts, without any indication within the disk directory as to how they were created. It never hurt to take a look the disk's label where I found, more often than not, that the

name of the application of origin had been scribbled down. Sometimes the label also contained directions for how to boot the primary program on a disk, which is not always the first PRG file in a directory. *Figure 15* below provides the specific number of disks per accession that included labels, load information and additional supplementary metadata. The disks have been separated into two types – disks with manufacturer labels that were likely out-of-the-box proprietary software and ordinary user copies of software. In some cases user-created working copies would include detailed copyright information and would indicate that the contents of the disk were proprietary. Nonetheless it was too difficult to categorize the disks in terms of proprietary content so I relied on the distinction between original manufacturer-direct floppies with carefully printed and designed labels and floppies created by users with handwritten labels instead. For a complete list of the contents of individual disk labels see *Appendix A*.

Disk Label Information by Accession								
	Blank	L1	L2	L3	M1	M2	M3	Total Disks
Acc. #1	1	1	1	0	2	1	3	9
Acc.#2	2	11	1	0	1	0	0	15
Acc.#3	4	26	15	5	3	3	1	57
Acc.#4	7	27	2	0	4	1	0	41
Total	14	65	19	5	10	5	4	122

L1=	Basic (written) label w/o load info.	L3=	Single unpackaged disk includes supplementary material in disk sleeve	M2=	Manufacturer disk w/ load info.
L2=	Basic label w/ load info.	M1=	Manufacturer disk w/o load info.	M3=	Manufacturer disk includes original packaging and manual

FIGURE 15. Prevalence of Labels on Disks by Accession

The next logical resource for interpreting the D64 images was the Internet. Often knowing the name of an application was enough to find an scan of the original user's

manual on a forum or at least find an overview of the program that could help me understand if I was looking at a database management system, a word processing application or a disk utility. In some instances I would have files created by a particular program, but was either unable to create a working copy of the actual program from an accompanying disk or the accession did not include the program in the first place. Here, as with associated metadata like user manuals, enthusiast communities provide an invaluable resource since so many of the C64's applications have been diligently converted into D64 files and archived online.

Finally, in instances when there were no external cues or resources for understanding the contents of a disk, I had to turn to the individual files themselves and access them “beneath” the user-interface level. To accomplish this I discovered software called *D64 Editor* that is similar in some ways to forensic applications like *FTK Imager* and generic hex editors, but is designed specifically to access D64 image files and alter their files at the bit level. With this program it is possible to access the disk directory and BAM to see how files are partitioned across the disk, and to analyze the hexadecimal content of files to find familiar ASCII characters. *D64 Editor* also allows users to read all files as if they were SEQ files, which translates the hexadecimal code into PETSCII characters. *Figure 16* below shows the BAM and a block of data from the directory rendered by *D64 Editor* in decimal and hexadecimal form respectively.

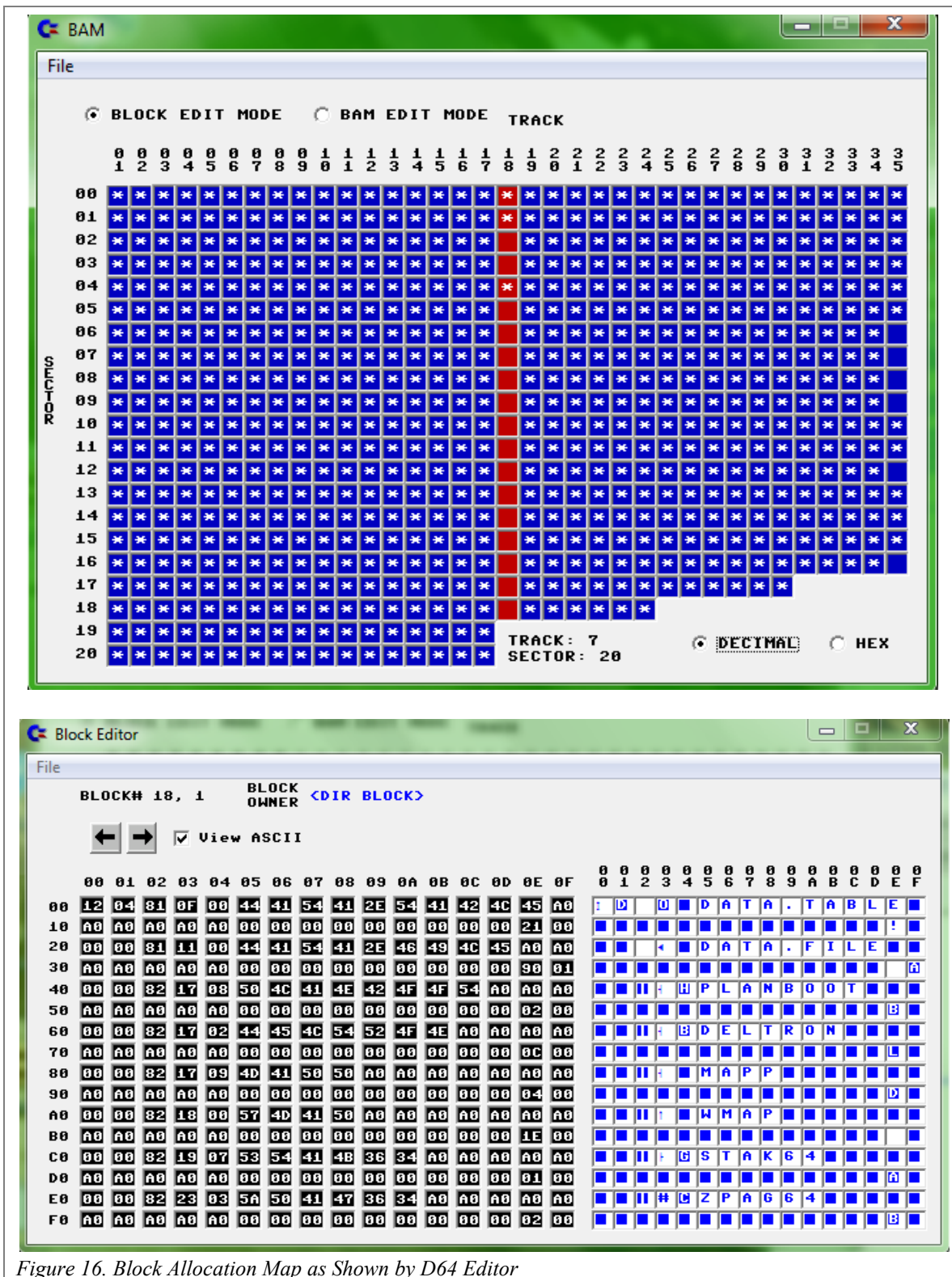


Figure 16. Block Allocation Map as Shown by D64 Editor

4.2.3 Constraints to Accessing User Created Content

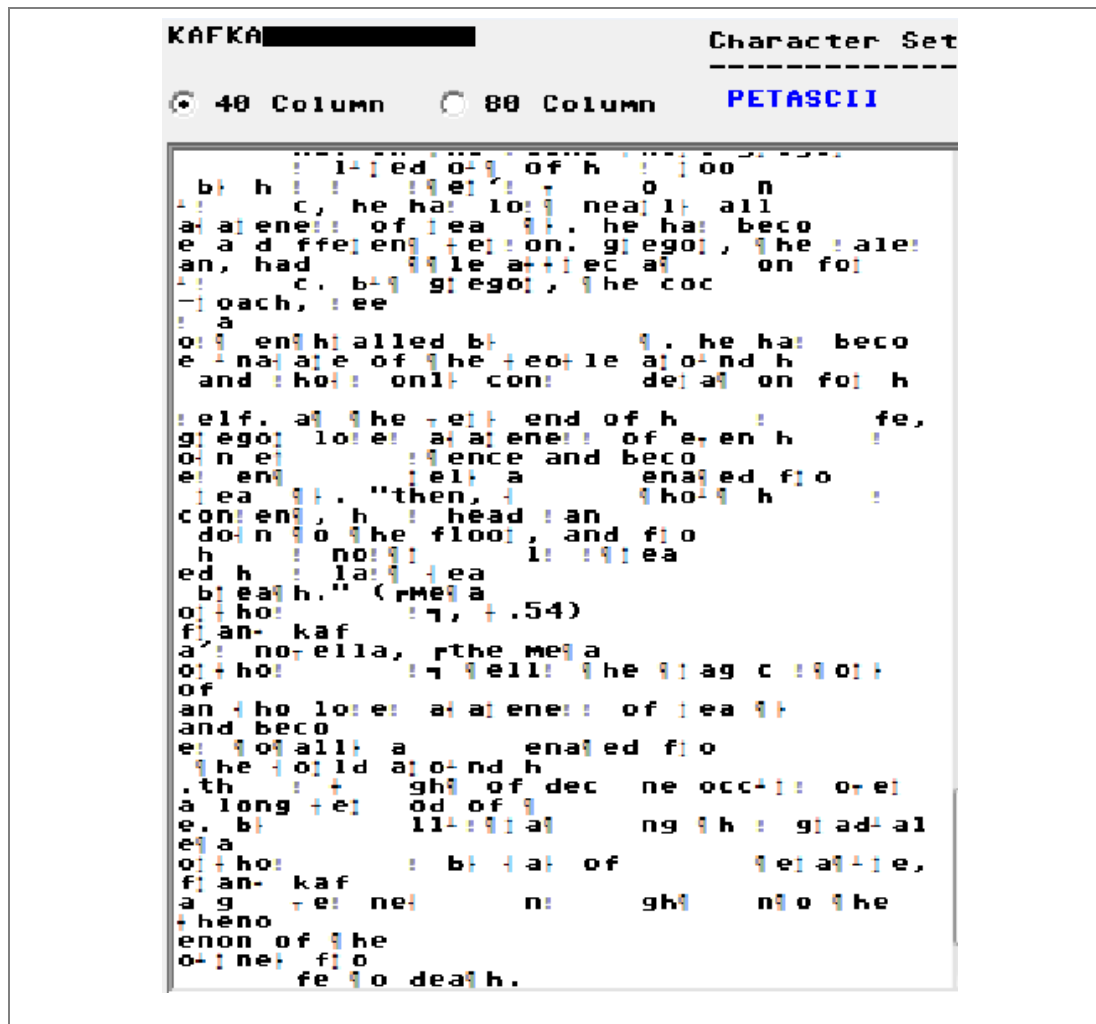
In the previous section I discussed some of the strategies that I implemented to deal with constraints to accessing disk content. In this section I would like to illustrate several specific instances of access constraints and their broader implications to digital curation practices.

Supporting Software Constraint: Accession #4 included several disks of user-created word processing documents that, based on the physical labeling of the disks and the disk name encoded in the BAM, I was able to infer were created with the PaperClip program.⁴⁷ I could view portions of the content of the files with a hex editor, but the ASCII conversion of the text did not capture all of the PETSCII characters, let alone the proper formatting of the document which was originally implemented in PaperClip by the user with complex coding. In order to reduce the noise of the hex view and to access a version of the original file that met a sufficient standard of integrity I needed to find an electronic version of PaperClip online.⁴⁸ If I were unable to find this program I would be forced to accept the ASCII interpretation of the file and lose much of the information in the process. At what point in similar scenarios does a digital curator decide that a file is either too important to allow for a compromised representation of its content or that a file is not worth preserving if it cannot be properly accessed?

Supporting Metadata Constraint: Fully accessing and manipulating the word processing documents created by PaperClip required that I download the user's manual.⁴⁹ In these particular instances reading the manual allowed me to "print" the user-generated documents which the *Power64* emulator could then export as text files, resulting in a simplified method for migrating the documents. What if this manual was not readily

available online? What characteristics of a file or a collection would justify experimenting with a program or vigorously researching its use until positive results were obtained? The difference between a portion of a word processing document as represented with a hex viewer and as represented in its original context is illustrated in

Figure 17.



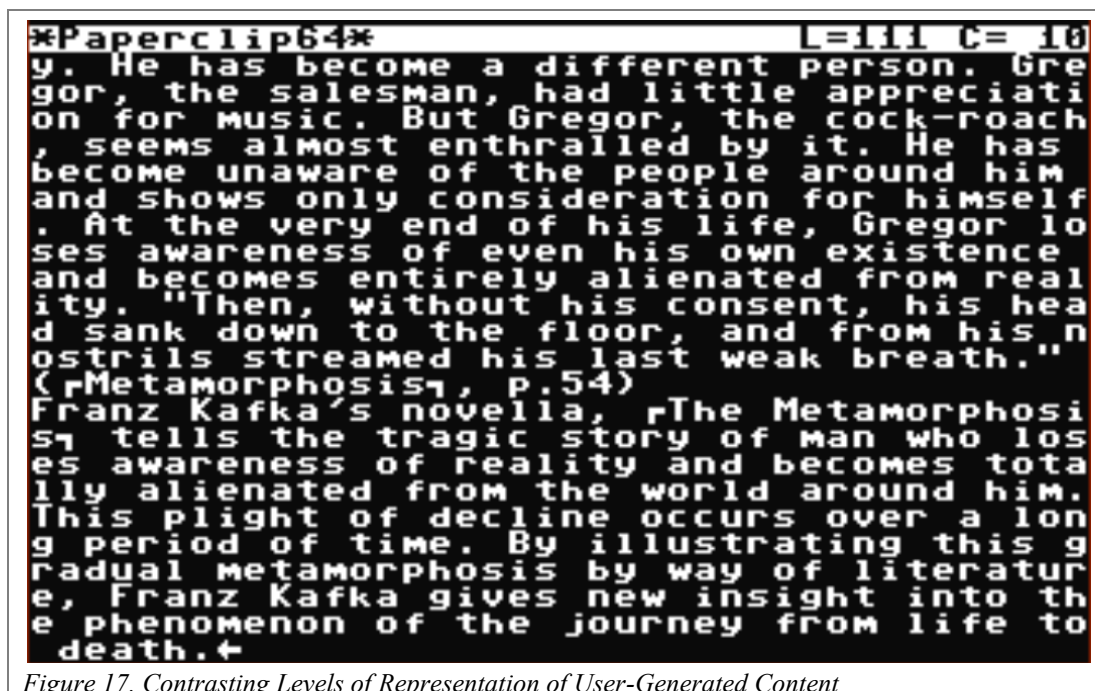


Figure 17. Contrasting Levels of Representation of User-Generated Content

Logical Representation Constraint: Certain word processing files on the pclip3.d64 disk image were inaccessible within the *Power64* emulation environment. The issue occurred when there were file names that include the PETSCII characters for ♠, which could not be properly entered into the PaperClip application when attempting to load a word processing document. Because the emulation software did not perfectly map the Macintosh keyboard to the original C64 keyboard, certain characters could not be produced in specific contexts. There are several ways to work around this problem. As mentioned earlier, if high-level integrity of file representation is not required, piecing together a general approximation of the file's content is possible by following the file chain within the hex view of *D64 Editor*. However if the file was of high priority, representing mission critical or historically valuable material that required the highest possible fidelity to the original conceptual level document, there are two more intensive solutions. The first would involve overwriting the offending filenames in the C64 BASIC

command line via an emulator, effectively altering the authenticity of the disk image copy.⁵⁰ Essentially this is a solution that affects the second “logical” level of the digital object’s representation. The final option is a more complex version of the previous renaming strategy that operates at the primary level of a file’s representation, by using a hex editor to “flip the bits” representing the offending characters in the filename, changing them into something suitable to the PaperClip software as it exists in the context of the *Power64* emulation software.

4.2.4 Maintenance

While browsing the Commodore troubleshooting forums I came across several mentions of simple maintenance techniques for increasing the performance of the 1541 drive and accessing grime-encrusted floppies.⁵¹ During the course of the data recovery process it was necessary for me to periodically open up the 1541 drive and clean the read head with isopropyl alcohol. Surprisingly this technique was extremely effective and after learning that one could clean the magnetic disk inside of a floppy case using alcohol as well, I decided to test this cleaning process on some of the disks that returned read errors during transfer. After cleaning eight partial read-error disks, five of them were able to be fully copied. In one case I was able to establish a perfect copy of a disk that had completely failed to be read by the 1541 after cleaning it. Interestingly after copying this disk it turned out to be proprietary software by Electronic Arts that was write protected with fat tracking. One other noteworthy result involved a cleaned disk that still returned read errors. However the second attempt at copying the disk resulted in read errors on different sectors of the disk, meaning that comparing the sectors of the first partial disk

image with the second could enable a patient curator to manually refresh the bytes with a hex editor resulting in a third complete copy!

4.3 Triage System

The purpose of developing a triage system for assessing the content of disk images is to facilitate an efficient process of file ingestion into a digital repository and to aid in the subsequent designation of repository resources towards ensuring the long-term access and preservation of heterogeneous collections of digital objects. What follows is a first attempt at creating such a system as it would apply to digital objects created by the C64. The proposed system is in many respects conceptual and was tested manually using a statistical method of intercoder agreement to measure the accuracy of identified genre types applied to a subset of files from the body of data recovered during this project. The results of the recovery portion of the project provided a unique test sample that I used to draw conclusions about systematically processing digital content and explore challenges that arise from interpreting unique digital objects by means of their format and content.

4.3.1 Designing the Classification System

Reviewing the content of the recovered files I was able to identify predominate genre types or classes that appeared across each accession of 5¼” floppies. These classes were outlined as they would apply to individual files represented in functioning form via an emulator or original C64 interface, i.e. as interpreted by a human being and not a computing system. A significant difficulty that arose while considering the heterogeneous types of data present on a disk was determining both the lowest and highest level of granularity that data could be grouped and still be relevant to the triage system. Part of making this decision required understanding what the intended outcome of organizing

and prioritizing these entities would be. In the end I settled on catering to two key outcomes for the triage system - identifying materials that were unique to specific accessions, i.e. user-generated content, and supporting long-term access to those materials. In order to properly attend to these outcomes it was necessary to designate a primary set of categories intended to classify the contents of *individual files* on a disk, as well as a secondary set of categories for differentiating whether a *disk* was dedicated to a single type of content or whether it contained heterogeneous files. Additionally, the primary and secondary categories were given sets of possible attributes to address specific characterizations such as ownership rights found within a file and the physical affordances of disks that signify sources of additional information. A list of both sets of categories and attributes and their descriptions is provided below in *Figure 18*.

Major Categories for Individual C64 Files:

Educational Software - Includes programs designed to teach aspects of math, science, computer science and other topics typically covered in primary and secondary schools.

Games - General forms of entertainment and gameplay.

Productivity Tools - Includes word processing software, spreadsheets, database software and other office and financial tools.

Utilities - Programs designed to increase the functionality of the computing system. Includes fastbooters and performance enhancing software, disk cleaners and other system analysis tools.

Communication/Networking - Early networking terminals for access Bulletin Board Systems (BBS) and the early internet. Also includes communication between groups of C64 programs and users (sceners) sharing information about available software, hacking techniques, etc.

Demos - General demonstration programs, usually created by a programmer to illustrate the capabilities of a system or particular software. Includes special software for creating demos.

User Generated Content - Includes anything that was not provided within a software package,

and usually was the end result of productivity tools or educational software. Includes word processing documents, music files, image files, customized scripts and programs, etc.

System/Derivative File - Designates a file that is part of a larger program and is not intended to be accessed directly by users.

Unknown

Secondary Attributes for Individual C64 Files:

Public Domain - believed to be free for copying and distribution.

Proprietary - licensed and protected by copyright.

Copy Protected - physically protected by a method of copy protection so that the file must be “cracked” to be copied.

Cracked/Hacked - proprietary software that has been altered by users in order to be copied and distributed freely (and illegally). Many cracked programs include a title sequence that boasts of the hacker’s accomplishment and attributes the crack to a particular hacker or group.

Unknown

Major Categories for C64 Disks:

Heterogeneous - An assortment of programs and genre types on one disk. These disks are usually created by users.

Dedicated - A disk dedicated to a single, generally larger, program. These tend to be proprietary software that are write protected from further alteration.

Unknown

Secondary Attributes for C64 Disks:

Labeled, Single-Sided, Free

Labeled, Double-Sided, Free

Labeled, Single-Sided, Write Protected

Labeled, Double-Sided, Write Protected

Unlabeled, Single-Sided, Free

Unlabeled, Double-Sided, Free

Unlabeled, Single-Sided, Write Protected

Unlabeled, Double-Sided, Write Protected

FIGURE 18. Triage Categories and Attributes for Files and Disks

4.3.2 Administering the Intercoder Agreement Test and Results

In preparation of the test I identified a series of eleven disk images with content spanning all genre types listed in *Figure 18*. For each disk image three files were chosen from the directory that could fall into one or more categories. When the test was administered the categorization system was explained to the research assistants and they were introduced to two options for accessing the files. The first possibility was to access the files with an emulator. This method was useful for identifying bootable applications like games or educational software, but was ineffective for accessing derivative files that required additional software to be opened or system files that could only be accessed while running other routines. The second possibility was to examine the contents of the files using the *D64 Editor* program. Additionally, the researchers were allowed to look at the physical 5¼” floppy disk source for each disk image for clues regarding the contents of the disk and its use. Specifically they were asked to examine the original label metadata and the presence of single-sided or double-sided notches and write protection tabs on the disks. Researchers were given as much time as needed to inspect the contents of the files and make their designation. Researchers were asked to give a major content categorization and a secondary ownership attribute for each file in the test. After assigning a category and attribute to each set of three files the researcher was also asked to categorize the entire disk based on its assumed uniformity of content. The purpose of this disk-level categorization was for the researcher to make inferences about the inter-dependency of the files on a disk and to identify whether a disk consisted of files dedicated to one program or use context, or if the files on the disk were components of multiple programs or contexts. Finally, researchers also gave the disk a set of secondary

attributes based on its physical affordances. These affordances can be understood as the visible signs that one encounters when handling the physical floppy disk. The disk-level qualities I asked the researchers to identify included whether the disk had label information, was single or double sided, and whether the disk was write-protected or free to be modified.

The results of the four intercoder tests were as follows. For the first test – assigning major categories for content of individual files – the raters agreed on twenty-five out of thirty-three cases, with an overall agreement percentage of 75.75% and a free-marginal Kappa measurement of 0.727272. In the second test – assigning minor attributes for identifying ownership of individual file content – the raters agreed on twenty-four out of thirty-three cases, with an overall agreement percentage of 72.72% and a free-marginal Kappa measurement of 0.65909. The third test – assigning categories for indicating the uniformity of disk content – returned eight agreements out of twelve cases for an overall agreement percentage of 66.66% and a free-marginal Kappa measurement of 0.500001. The last test – attributes of a disk’s physical affordances – returned twelve agreements out of twelve cases for a perfect agreement measurement of 1.00.

Tests one and two both resulted in acceptable rates of agreement with free-marginal *kappas* above .70. These tests concerned the genre type of specific files and their assumed ownership. The third test, which was based on determining whether a disk contained uniform content, i.e., content dedicated to one application or context, yielded the lowest *kappa* measurement possibly because the researchers were asked to look at only three files per disk and the inter-relationship of files was not immediately apparent from viewing the disk directory. Finally, the fourth test yielded perfect agreement due

largely to the fact that categories for this test were not qualitative but relied on observing definable physical aspects of a floppy disk.

5. DISCUSSION

This project represents a first attempt at addressing an area of digital preservation and data recovery from legacy system that is not extensively addressed in the archives and digital curation literature. Generally, digital preservation models and policies tend to discuss the characterization of file formats and data objects in terms of relatively common file types, e.g., PDF files, Microsoft Word files, Computer Assisted Design files, common image formats like JPEG, TIFF, etc. These file types are predominantly the product of increased processing power and efforts towards interoperability across systems, but their predominance also relies in part on the medium of their capture and storage, namely high-capacity hard disk drives. Furthermore, digital preservation initiatives tend to focus on ingesting digital objects into repositories in the form of secured bitstreams rather than considering the means of recovering digital objects from storage media. Very little is written on files and formats predating the staples of modern computing, and in the case of Commodore's BASIC files there are no examples of recovery and preservation recommendations that I have come across in the schemas developed by large-scale preservation initiatives.⁵²

Perhaps this is partly because the C64 storage media, 5¼" floppies, have exceeded their expiration date while data created by systems in recent memory, which are suited to abundantly available memory and storage, necessarily take precedence over a relatively small body of personal digital information. Perhaps too the Commodore's reputation as a gaming system overshadows its role in introducing many individuals to

personal computing in the 1980s and providing an early testing ground for post-market consumer devices, system customization and other hardware and software add-ons.

Several of the enthusiast archives online claim to have created disk images for 99% of the proprietary software offered for the C64 as well as a sizable chunk of the specialized software created by the devoted cottage industry of freelance programmers. If the legacy of the C64 is interpreted primarily to be a seminal entertainment platform, perhaps due diligence has been performed already and integrating special characterization of the C64's data types into preservation models would be an effort of diminishing returns.

However, the fact that user-generated content and personal digital information created with the C64 is minimized alongside the commercial offerings of the system, and that in large part it is the software and programs that have survived rather than the products of their use, underscores a broader concern regarding most of the micro and personal computing platforms from the late 1970s to the early 1990s. My goal in looking at the contents of an assortment of 5¼" floppies was to show that the data that will ultimately be the most interesting to archives and to preserving cultural heritage will also be the least available and the most difficult to recover and access. In the end this point was demonstrable because the C64 has an incredibly rich base of support in the form of online enthusiast networks that actively preserve not only the C64's programs, but also the supporting materials such as manuals, utilities and even the original hardware. I was able to identify problematic and at-risk personal files - to access and in some cases migrate them - because I could find the necessary software and metadata to do so easily online. I can see much greater challenges for preserving the output of less popular systems of the age, abandonware media,⁵³ and platforms that enjoyed far less time on the

market with little fanfare or development before discretely relocating to the trash heap. How far back and how extensively will archives be willing to search for meaning encased in obscure digital contexts? Is it a fair compromise to allow some systems, and their dependent body of user-generated data, to go extinct or to be relegated to the role of unused artifacts in a hardware museum while others enjoy an extended afterglow of use and reuse online and in the archives via emulation and migration? Will the contents of these kinds of media warrant the effort it takes to recover them?

In many ways these are questions that transcends media types altogether and are thoroughly grounded in discourses in archival science, differing only slightly in that more emphasis is placed on access in archival contexts than on recovery. Archivists have long asked what, according to the quality and character of each collection, are the reasonable ends towards making documents and materials accessible. To answer this question archivists rely in part on their instincts about the inherent value of a collection and defer to a complex paradigm relating the environment of their institution, its patrons and the contents of the archives. In the world of digital forensics and data recovery, however, the question revolves primarily around how far one is willing to go in terms of *techne*. The potential for automatically retrieving certain kinds of content or characterization information from a mass of digital bits is feasible if the proper technical groundwork is in place. The potential advancement of recovery processes provides an ongoing hypothetical scenario wherein the reproduction of digital objects that attain a high level fidelity to their various representational inheritances is almost certain with increased technical perspicacity. Digital accessions may provide an alternative answer to the archival formula

that states that less process equals more product (Greene & Meissner, 2005). In the digital landscape it may be that more process equals more product.

Moving forward from this study, possible next steps would be to improve the results of the recovery process by applying more advanced technologies and tools than addressed here. Using the serial (IEC) bus to create disk images via the ZoomFloppy device is not an ideal configuration for high-quality and controlled data recovery. According to many C64 enthusiasts, including the creators of the ZoomFloppy device, a better approach would be to optimize the outcome of creating D64 disk images by customizing the 1541 disk drive to include a parallel port for faster and more intensive data recovery.⁵⁴ The ZoomFloppy is equipped to accept a parallel cable as well as a serial cable and can still be used as a transfer mechanism, but, with a customized parallel port installed, a 1541 can connect directly to parallel enabled PCs. Using *NibTools* software, which requires a parallel connection, copies of the content of a disk can be created at the level of a nibble of data (4 bits) at a time rather than a sector. This method is faster, provides more options for packaging the files with representational metadata and is more effective at duplicating copy protected disks.⁵⁵

Finally, I would propose further development of a characterization system for file types representing a broader range of retro-computing systems beyond the domain of the Commodore 64. Furthermore a series of recommendations for maintaining and preserving files that fall within specific categories should be explored. The outcomes of the intercoder test showed that a relatively high level of consensus concerning the grouping of digital files is possible. However, it remains to be seen if triage recommendations assigned to specific categories would actually streamline the process of

identifying user-generated content and improve the recovery and preservation of files relevant to that content. Future categories should be fine-tuned to address the relationships between derivative and primary files, in an attempt to better understand the various levels of preservation that must be applied to a “single” digital object. A formalized method of triage that preempts the ingestion of digital objects into a repository should also be tested to see if genre-based characterizations are accurate and useful. Lastly aspects of the classification system that met with high percentages of agreement between coders should be considered in development of a system of automated characterization for digital resources.

ENDNOTES

- ¹ Edwards, B. (2011) Can you do real work with the 30-year-old IBM 5150? *PCWorld*. Retrieved from http://www.pcworld.com/article/237878-4/can_you_do_real_work_with_the_30yearold_ibm_5150.html
- ² tintinaujapon. (2010, May). 5.25" floppy disk [Msg 10]. Message posted to <http://www.mactalk.com.au/28/84520-5-25-floppy-disk.html>
- ³ Siegler, M.G. (2010, August 4). Eric Schmidt: Every 2 days we create as much information as we did up to 2003. *TechCrunch*. Retrieved from <http://techcrunch.com/2010/08/04/schmidt-data/>
- ⁴ See the Forensic Wiki for descriptions of available options: Main Page. (n.d.). Retrieved November 12, 2011 from Forensics Wiki: http://www.forensicswiki.org/wiki/Main_Page
- ⁵ AccessData Group, LLC. (n.d.). Forensic Toolkit (FTK) computer forensics software. Retrieved from <http://accessdata.com/products/computer-forensics/ftk>
- ⁶ Guidance Software, Inc. (n.d.). Computer Forensic data collection for digital evidence examiners. Retrieved from <http://www.guidancesoftware.com/forensic.htm>
- ⁷ Farquhar, A., Billenness, C., & Yarwood, A. (2007). PLANETS: Home. Retrieved from <http://www.planets-project.eu/>
- ⁸ CAMILEON. (n.d.) Retrieved November 12, 2011 from Wikipedia: <http://en.wikipedia.org/wiki/CAMILEON>
- ⁹ Many of the technical specifications discussed in this section were researched online using both retro-computing forums and Wikipedia. The reason I chose to use Wikipedia as a source is that the information provided was comprehensive and correlated with information I found on enthusiast and expert sources such as user's manuals. Rather than cite Wikipedia in the bibliography, I have usually chosen to provide a link to the relevant entry in the endnotes.
- ¹⁰ Programmable logic array. (n.d.) Retrieved November 12, 2011 from Wikipedia: http://en.wikipedia.org/wiki/Programmable_logic_array
- ¹¹ Bank switching. (n.d.). Retrieved November 12, 2011 from Wikipedia: http://en.wikipedia.org/wiki/Bank_switching
- ¹² For a visualization of the C64's I/O ports see – File:C64 interfaces.jpg. (n.d.) Retrieved November 12, 2011 from Wikipedia: http://en.wikipedia.org/wiki/File:C64_Interfaces.jpg
- ¹³ For an overview of the C2N tape storage technology see Østerby, 2009, p. 5.
- ¹⁴ Farquhar, D. (2005) How to connect a Commodore 64 to a television. *The Silicon Underground*. Retrieved from <http://dfarg.homeip.net/2005/01/how-to-connect-a-commodore-64-to-a-television/>
- ¹⁵ Commodore 1541. (n.d.) Retrieved November 12, 2011 from Wikipedia: http://en.wikipedia.org/wiki/Commodore_1541
- ¹⁶ Fast loader. (n.d.) Retrieved November 12, 2011 from Wikipedia: http://en.wikipedia.org/wiki/Fast_loader
- ¹⁷ Commodore Business Machines (1981). *VIC-1541 User's Manual*, p.4.
- ¹⁸ Commodore Business Machines (1984), *Commodore 64 User's Guide*, p.3.
- ¹⁹ Ibid. p.62.
- ²⁰ File:C64 Petscii charts.png. (n.d.) Retrieved November 12, 2011 from Wikimedia: http://commons.wikimedia.org/wiki/File:C64_Petscii_Charts.png
- ²¹ PETSCII. (n.d.). Retrieved November 12, 2011 from Wikipedia: <http://en.wikipedia.org/wiki/PETSCII>
- ²² GEOS (8-bit operating system). (n.d.). Retrieved November 12, 2011 from Wikipedia: http://en.wikipedia.org/wiki/GEOS_%288-bit_operating_system%29
- ²³ Disk operating system. (n.d.) Retrieved November 12, 2011 from Wikipedia: http://en.wikipedia.org/wiki/Disk_operating_system
- ²⁴ History of hard disk drives: 1980s, the PC era. (n.d.) Retrieved November 12, 2011 from Wikipedia: http://en.wikipedia.org/wiki/History_of_hard_disk_drives#1980s.2C_the_PC_era
- ²⁵ See: Evans, E. (2010, April 25). Show 132: The Commodore 64, part I. *The Retrobits Podcast*. Podcast retrieved from <http://retrobits.libsyn.com/show-132-the-commodore-64-part-i;>

VIC-1541. (n.d.) Retrieved November 12, 2011 from C-64 Wiki: <http://www.c64-wiki.com/index.php/VIC-1541>; Tempelmann, T. (2011, July 15). The Story of FCopy for C64 -. [Web log post]. Retrieved from <http://www.pagetable.com/?p=647>

²⁶Commodore64Scene.nl. (n.d.) 1541 disk drive maintenance. Retrieved from http://www.c64scene.com/c64/?page_id=1230

²⁷Group code recording: GCR for floppy disks. (n.d.). Retrieved November 12, 2011 from Wikipedia: http://en.wikipedia.org/wiki/Group_code_recording#GCR_for_floppy_disks

²⁸ Ibid. See also: VIC-1541. (n.d.) Retrieved November 12, 2011 from C64-Wiki: <http://www.c64-wiki.com/index.php/VIC-1541>; mlonnqvist. (2005, August 1). Filename questions [Msg 5]. Message posted to: <http://www.lemon64.com/forum/viewtopic.php?t=17498>; Victoria. (2010, February 15). Media recognition: Floppy disks part 2. Retrieved from <http://futurearchives.blogspot.com/2010/02/media-recognition-floppy-disks-part-2.html>

²⁹ Rittwage, P. (2011). Copy protection methods. Retrieved from <http://c64preservation.com/dp.php?pg=protection>; Rittwage, P. (2011). Electronic Arts C64 fat track loader. Retrieved from <http://c64preservation.com/files/EaLoader.txt>

³⁰ Schepers, P. (2004). Disk file layout (D64, D71, D81). Retrieved from <http://ist.uwaterloo.ca/~schepers/formats/DISK.TXT>

³¹ Fragmentation (computer). (n.d.) Retrieved on November 12, 2011 from Wikipedia: http://en.wikipedia.org/wiki/Fragmentation_%28computer%29

³² *VIC-1541 Manual*, p.55

³³ Commodore DOS. (n.d.) Retrieved on November 12, 2011 from Wikipedia: http://en.wikipedia.org/wiki/Commodore_DOS

³⁴ For an extended list of obsolete media formats visit the Lost Formats Preservation Society: Experimental Jetset. (2000, December). Lost formats. Retrieved from <http://www.experimentaljetset.nl/archive/lostformats.html>

³⁵ For instance, an Android mobile phone application released in 2009 emulates the C64, and a wave of chiptunes musicians have taken to acquiring C64s solely to repurpose their SID sound chips as components for custom synthesizers.

³⁶ See also: The Software Preservation Society. (2010). Kryoflux – USB floppy controller. Retrieved from <http://www.kryoflux.com/>; Artefactual Systems Inc. (2006) Archivemata. Retrieved from http://archivemata.org/wiki/index.php?title=Main_Page; Jansen, G. (2011). UNC-Libraries/Curators-Workbench. Retrieved from <https://github.com/UNC-Libraries/Curators-Workbench>

³⁷ In large part this is why it is essential to move dated media to an emulation environment and to go to the trouble of automating some of the process of accessing media within the emulation environment. See Woods & Brown, 2010.

³⁸ For specifics on the two difference see: Sullivan Geoff. (n.d.). Servicing the 1541 drive. Retrieved from <http://home.comcast.net/~safeharborbay/tcr/cbm/1541.html>; Overdoc. (2001, December 29). Older Alps mechanism on 1541 [Msg 3]. Message posted to <http://www.lemon64.com/forum/viewtopic.php?t=25217&sid=91b3dbf09973a97313ab096a9f7e1c7e>

³⁹ The Rosetta Stone of the entire testing process can be found here: Farquhar, D. (2005) How to connect a Commodore 64 to a television. *The Silicon Underground*. Retrieved from <http://dfarq.homeip.net/2005/01/how-to-connect-a-commodore-64-to-a-television/>

⁴⁰ It's interesting to note that the speed of booting up the C64 system is somewhat alarming for someone who is used to waiting a minute or so for a graphical user interface OS to login and allow the user to access his or her desktop.

⁴¹ The GPL allows enthusiasts to develop their own devices, free of charge, for achieving similar results as the ZoomFloppy. See: Lawson, N. (2011). xum1541 development page and firmware. Retrieved from <http://www.root.org/~nate/c64/xum1541/>; Lawson, N. (2009, January 21). Introducing xum1541: The fast C64 floppy USB adapter. Retrieved from <http://rdist.root.org/2009/01/21/introducing-xum1541-the-fast-c64->

[floppy-usb-adapter/](#); Brain. (2004, June 20). New Zoom Floppy [Msg 3]. Message posted to <http://www.lemon64.com/forum/viewtopic.php?t=36245>; *ZoomFloppy Manual* (2011)

⁴² See: Vogelgsang, C. (2009). OpenCBM on Mac. Retrieved from <http://lallafa.de/blog/opencbm-on-mac/>

⁴³ An in-depth discussion of many original Commodore file formats and their emulated equivalents can be found on here: Schepers, P. (2005). Introduction to various emulator file formats. Retrieved from <http://www.unusedino.de/ec64/technical/formats/>

⁴⁴ Ibid.

⁴⁵ See Forster, J. (2005, April 22). OpenCBM: Detail:1188112 – Error info block contains invalid codes. Message posted to http://sourceforge.net/tracker/index.php?func=detail&aid=1188112&group_id=122047&atid=692219

⁴⁶ Kozierok, C. M. (2001). Floppy disk reliability. Retrieved from <http://www.pcguides.com/ref/fdd/confReliability-c.html>

⁴⁷ See review of PaperClip software: Ciruolo, M., & Friedland, N. (1985, May) Paperclip. *Antic*, 4 (1), 24. Retrieved from <http://www.atarimagazines.com/v4n1/paperclip.html>

⁴⁸ Details of download here: Loge, K. (n.d.) Documen details | PaperClip Word Processor. Retrieved from http://dreamsteep.com/component/docman/doc_details/171-paperclip-word-processor.html?tmpl=component&Itemid=30

⁴⁹ I referred to this forum to find the manual: hurminator. (2010, November 22). Paperclip 64 [Msg 2]. Message posted to <http://www.lemon64.com/forum/viewtopic.php?t=35936&sid=d3955870964397f3ef692d072a9df079>

⁵⁰ BASIC command for renaming files: OPEN 1,8,15,"R:New_Name=Old_Name":CLOSE 1

⁵¹ See: retrofloppy. (2007, February 13). What to use to clean 1541 read head [Msg 2]? Message posted to <http://www.lemon64.com/forum/viewtopic.php?t=22418&view=next&sid=683a61170872b53c865a9db7d3fb80fb>; Cleaning a 5.25" disk / disc [video file]. Retrieved from <http://www.youtube.com/watch?v=dlijfp3FBWIA>

⁵² File registries also have little to bear on Commodore PRG, REL, USR, etc. files created with BASIC. PRONOM includes no file types related to the Commodore system nor does the Library of Congress registry located at <http://www.digitalpreservation.gov/formats/intro/intro.shtml>. Preservation modules such as JHOVE also overlook early micro-computing legacy formats.

⁵³ Abandonware. (n.d.). Retrieved on November 12, 2011 from Wikipedia: <http://en.wikipedia.org/wiki/Abandonware>

⁵⁴ For parallel installation instructions see: Schepers, P. (2010). Installing the 1541 Parallel Port. Retrieved from <http://ist.uwaterloo.ca/~schepers/1541port.html>

⁵⁵ For notes on copy protection and nibblers see – Rittwage, P. (2011). Copy protection methods. Retrieved from <http://c64preservation.com/dp.php?pg=protection>

REFERENCES

- Abbott, D., & Kim, Y. (2010). *DCC Briefing Paper: Genre classification*. Digital Curation Centre. Retrieved from <http://hdl.handle.net/1842/3367>
- Arms, W. Y. (2000). *Digital libraries*. Cambridge, Mass: MIT Press [Electronic version]. Retrieved from <http://www.cs.cornell.edu/wya/diglib/MS1999/index.html>
- Blythe, J. A. (2009). *Digital dixie. Processing born digital materials in the Southern Historical Collection* (Master's thesis). School of Information and Library Science of the University of North Carolina, Chapel Hill, NC.
- Brown, J.S., & Duguid, P. (2000). *The social life of information*, 173-205. Boston, MA: Harvard Business School Press.
- Burnett, D.J., & Randolph, S. (2011, Summer). The memory hole has teeth. *Cabinet*, 42, 70-77.
- Commodore Business Machines (1981). *VIC-1541 single drive floppy disk user's manual*. West Chester, PA: Commodore Business Machines.
- Commodore Business Machines. (1984). *Commodore 64 user's guide*. West Chester, PA: Commodore Business Machines.
- Consultative Committee for Space Data Systems. (2002). *Reference model for an Open Archival Information System (OAIS)*. Washington, D.C: CCSDS Secretariat.
- Cook, T. (1991-1992). Easy to byte, harder to chew: The second generation of electronic records archivists. *Archivaria*, 33, 202-216.
- Cook, T. (2004). Byte-ing off what you can chew: Electronic records strategies for small archival institutions. *Archifacts*, 1-20.
- Cunningham, A. (1994). The Archival Management of Personal Records in Electronic Form: Some Suggestions. *Archives and Manuscripts*, 22 (94).
- Cunningham, A. (1999). Waiting for the ghost train: Strategies for managing personal electronic records before it is too late. *Archival Issues*, 24 (1), 55-64.
- Derogee, J. (2008). IEC-bus documentation as used for the 1541-III. Retrieved from http://jderogee.tripod.com/project1541_downloads/IEC_1541_info.pdf

- Duranti, L., Eastwood, T., & MacNeil, H. (2002). Preservation of the integrity of electronic records. Dordrecht: Kluwer Academic Publishers.
- Duranti, L. (2005). The InterPARES Project: the long-term preservation of authentic electronic records: the findings of the InterPARES Project. San Miniato (PI), Italia: Archilab.
- Eastwood, T. (2004) Appraising digital records for long-term preservation. *Data Science Journal*, 3, 202-208.
- Edwards, B. (2008, November 4). Inside the Commodore 64. *PCWorld*. Retrieved from http://www.pcworld.com/article/152528/inside_the_commodore_64.html
- Elford, D., Pozo, N.D., Mihajlovic, S., Pearson, D., Clifton, G., & Webb, C. (2008, August). Media matters: developing processes for preserving digital objects on physical carriers at the National Library of Australia. *Proceedings of the 74th IFLA General Conference and Council*. Québec, Canada: IFLA.
- Esteva, M., Xu, W., Jain, S.D., Lee, J.L., & Martin, W.K. (2011). Assessing the preservation condition of large and heterogeneous electronic records collections with visualization. *International Journal of Digital Curation*, 6 (1). Retrieved from <http://www.ijdc.net/index.php/ijdc/article/view/162>
- Evans, E. (2010, April 25). Show 132: The Commodore 64, part I. *The Retrobits Podcast*. Podcast retrieved from <http://retrobits.libsyn.com/show-132-the-commodore-64-part-i>
- Evans, E. (2010, August 29). Show 135: The Commodore 64, part II. *The Retrobits Podcast*. Podcast retrieved from <http://retrobits.libsyn.com/show-132-the-commodore-64-part-i>
- Farmer, D., & Venema, W. (2005) *Forensic discovery*. Upper Saddle River, NJ: Addison-Wesley.
- Fisher, K. E., Durrance, J.C., & Hinton, M.B. (2004). Information grounds and the use of need-based services by immigrants in Queens, New York: A context-based, outcome evaluation approach.” *Journal of the American Society for Information Science and Technology*, 55, 754–766.
- Gibson, J. J. (1979). *The ecological approach to visual perception*. Boston, MA: Houghton Mifflin.
- Granger, S. (October 01, 2000). Emulation as a digital preservation strategy. *D-lib Magazine*, 6, 10.

- Greene, M. A., & Meissner, D. E. (January 01, 2005). More product, less process: Revamping traditional archival processing. *American Archivist*, 68, 2.
- Guttenbrunner, M., Becker, C., & Rauber, A. (2010). Keeping the game alive: Evaluating strategies for the preservation of console video games. *International Journal of Digital Curation*, 5 (1). Retrieved from <http://ijdc.net/index.php/ijdc/article/view/147>
- Hedstrom, M.L., & Lampe, C.A. (2001). Emulation vs. migration: Do users care? *RLG Diginews*, 5 (6). Retrieved from <http://webdoc.gwdg.de/edoc/aw/rlgdn/preserv/diginews/diginews5-6.html#feature1>
- Hedstrom, M.L., Lee, C.A., Olson, J.S. & Lampe, C.A. (2006). 'The old version flickers more': Digital preservation from the user's perspective. *American Archivist*, 69 (1). 159-187.
- Holdsworth, D., & Sergeant, D. M. (January 01, 2000). A blueprint for representation information in the OAIS model. *NASA Conference Publication*, 209888, 413-428.
- Hyry, T., & Onuf, R. (1997). The personality of electronic records: The impact of new information technology on personal papers. *Archival Issues*, 22 (1), 37-44.
- John, J.L. (2008, September). Adapting existing technologies for digitally archiving personal lives: Digital forensics, ancestral computing, and evolutionary perspectives and tools. *Proceedings of the Fifth International Conference on Preservation of Digital Objects (iPRES '08)*, London, UK.
- John, J.L., Rowlands, I., Williams, P., & Dean, K. (2010). *Digital lives: Personal digital archives for the 21st century: an initial synthesis*.
- Kim, Y., & Ross, S. (December 02, 2008). "The naming of cats": Automated genre classification. *International Journal of Digital Curation*, 2 (1).
- Kirschenbaum, M. G. (2008). *Mechanisms: new media and the forensic imagination*. Cambridge, MA: MIT Press.
- Kirschenbaum, M. G., Oviden, R., Redwine, G., & Donahue, R. (2010). *Digital forensics and born-digital content in cultural heritage collections*. Washington, DC: Council on Library and Information Resources.
- Klien, M., Trikaliotis, S., & Moser, W. (2007). *OpenCBM 0.4.2 users guide*. Retrieved from <http://opencbm.trikaliotis.net/>
- Lavo. (2010, May). 5.25" floppy disk [Msg 10]. Message posted to <http://www.mactalk.com.au/28/84520-5-25-floppy-disk.html>

- Levy, D.M. (1998). Heroic measures: reflections on the possibility and purpose of digital preservation. *Proceedings of the Third ACM Conference on Digital Libraries (DL '98)*. New York, NY: ACM. DOI=10.1145/276675.276692
<http://doi.acm.org/10.1145/276675.276692>
- Marshall, C. C. (March 01, 2008). Rethinking personal digital archiving, Part 1. *D-lib Magazine*, 14.
- McKemmish, S. (1996) Evidence of me. *Archives and Manuscripts*, 24 (1), 28-45.
- Mullish, H., & Kruger, D. (1984). *At home with BASIC: The Simon & Schuster guide to programming the Commodore 64*. New York, NY: Computer Book Division, Simon & Schuster.
- OCLC/RLG Working Group on Preservation Metadata., OCLC., & Research Libraries Group. (2002). *Preservation metadata and the OAIIS Information Model: A metadata framework to support the preservation of digital objects*. Dublin, Ohio: OCLC/RLG Working Group on Preservation Metadata.
- Østerby, J. M. (2009). *Methods for preserving digital heritage from legacy platforms: A case study on preservation of Commodore 64 material*. Århus: Aarhus Universitet, Datalogisk Institut.
- Ross, S., & Gow, A. (1999). *Digital archaeology: Rescuing neglected and damaged data resources, a JISC/NPO study within the electronic libraries (eLib) programme on the preservation of electronic materials*. London, UK: Library Information Technology Center. Retrieved from
<http://www.ukoln.ac.uk/services/elib/papers/supporting/pdf/p2.pdf>
- Rothenberg, J., & Council on Library and Information Resources (USA). (1999). *Avoiding technological quicksand: Finding a viable technical foundation for digital preservation*. Washington: Council on Library and Information Resources. Retrieved from <http://www.clir.org/pubs/reports/rothenberg/contents.html>
- Schepers, P. (2005). D64 (Electronic form of a physical 1541 disk). Retrieved from
<http://www.unusedino.de/ec64/technical/formats/d64.html>
- Siegler, M.G. (2010, August 4). Eric Schmidt: Every 2 days we create as much information as we did up to 2003. *TechCrunch*. Retrieved from
<http://techcrunch.com/2010/08/04/schmidt-data/>
- Silberschatz, A., Peterson, J. L., & Galvin, P. B. (1991). *Operating system concepts*. Reading, Mass: Addison-Wesley.

- Strodl, S., Motlik, F., Stadler, K., & Rauber, A. (2008, December). Personal & SOHO archiving. *Proceedings of the ACM International Conference on Digital Libraries*, 115-123.
- Thibodeau, K. (2002). *Overview of technological approaches to digital preservation and challenges in coming years*. Presented at The State of Digital Preservation: An International Perspective. Washington, DC: Council on Library and Information Resources.
- von Suchodoletz, D., & van der Hoeven, J. (2008). Emulation: From digital artefact to remotely rendered environments. *Proceedings of the Fifth International Conference on Preservation of Digital Objects, (iPRES 2008)*. London, UK: The British Library.
- Woods, K., & Brown, G. (2010). Assisted emulation for legacy executables. *International Journal of Digital Curation*, 5 (1), 160-171.
- Woods, K., Lee, C. A., & Garfinkel, S. (2011, July). Extending digital repository architectures to support disk image preservation and access. *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL '11)*. New York, NY: ACM. DOI=10.1145/1998076.1998088
<http://doi.acm.org/10.1145/1998076.1998088>

APPENDIX A – Disk Image Results From Four Accessions of 5¼” Floppy Disks

Acc #	Disk Image	Transfer Success	Date Created/ Attempted	Disk Label	Additional Information	Genre
1	1.A	100%	8/16/2011	The Print Shop COMMODORE 64 SIDE A (SIDE B on Back) Broderbund Software™ 17 Paul Drive > San Rafael > California 94903-2101	Proprietary Software. Manual and Packaging Included.	Productivity
1	1.B	100%	9/24/2011	[Same as Above]	[Same as Above]	Productivity
1	2.A	100%	8/16/2011	The Print Shop FOR THE COMMODORE 64/128 SIDE A (SIDE B on Back) Broderbund Software™ 17 Paul Drive > San Rafael > California 94903-2101 COMDSK93	[Same as Above]	Productivity
1	2.B	100%	9/25/2011	[Same as Above]	[Same as Above]	Productivity
1	3.	100%	8/16/2011	commodore model 1541 test/demo-diskette single sided 48 tpi	Proprietary Software. Utility.	Utility
1	4.	Partial Read Error	8/16/2011	OKIMATE Commodore Software P/N 33025705-REV 5.0 LEARN TO PRINT COLOR SCREEN PRINT Load "LEARN", 8 Load "COLOR", 8 (64, SX-64, PLUS/4 only) Copyright 1984 OKIDATA Division of OKI AMERICA, Inc.	Proprietary Software. Utility.	Utility
1	5.	100%	8/9/2011	LANIER® Diskette No. U-268 LANIER BUSINESS PRODUCTS, INC. ATLANTA, GEORGIA USA LTD-3 SR40 LOAD "PERSONALITY ANAL", 8 RUN	Label later marked with a "KO" next to diskette no.	Games, Entertainment, Misc.

1	6.	Partial Read Error	8/16/2011	LANIER® Diskette No. U-268 LANIER BUSINESS PRODUCTS, INC. ATLANTA, GEORGIA USA LTD-3 SR3 8 FAST HACKEM II	Label later marked with a "KO" next to diskette no. "SR3 8" on label is struck through.	Games, Misc.
1	7.A	100%	8/16/2011	LEADER BOARD™ World Class REVERSE SIDE OF DISK CONTAINS COURSE DATA ©1987 ACCESS SOFTWARE · 2561 SOUTH 1560 WEST · WOODS CROSS, UT 84087	Proprietary Software.	Games
1	7.B	100%	8/16/2011	[Same as Above]	[Same as Above]	Games
1	8.	Failed. Read Error	8/16/2011	[Blank/Missing]	Technically blank disk image. Read Error 27 (checksum error in header) Read Error 21 (no sync character)	Productivity, Unknown
1	Null	Failed. Read /Write Blocked.	8/16/2011	10th FRAME PRO BOWLING SIMULATOR By Roger & Bruce Carver © 1986 ACCESS Software Inc. 2561 South 1560 West Woods Cross, Utah 84087	Security Key required for Read/Write permissions. No disk image created at time of attempt. Proprietary Software. Manual and Packaging Included.	Games
2	Null	Failed. Read Error	8/18/2011	PacDude	Read Error 74 (drive not ready). Disk is likely not formatted for the Commodore.	Unknown
2	Null	Failed. Read Error	8/18/2011	Business ACCT SYSTEM NO DD (40TR)	Read Error 74 (drive not ready). Disk is likely not formatted for the Commodore. Sleeve Label: WORK (erased)	Unknown
2	Null	Failed. Read Error	8/18/2011	Data Base	Read Error 74 (drive not ready). Disk is likely not formatted for the Commodore.	Unknown

2	1.A	Partial Read Error	8/18/2011	A. "SPITFIRE" HELLCAT B. CHESS	"SPITFIRE" on label is struck through.	Games
2	1.B	100%	8/18/2011	[Same as Above]	[Same as Above]	Games
2	2.A	100%	8/18/2011	LOADSTAR A & B A. Zorro, Pipeline, etc B. Helicopter	"LOADSTAR A & B" on label is struck through.	Games
2	2.B	100%	8/18/2011	[Same as Above]	[Same as Above]	Games
2	3.A	Partial Read Error	8/18/2011	A. Easy Script B. Easy Spell (Load from A).		Productivity. Programming.
2	3.B	Partial Read Error	8/18/2011	[Same as Above]	[Same as Above]	Productivity.
2	Null	Failed. Read Error	8/18/2011	[Blank]	Read Error 74 (drive not ready). Disk is likely not formatted for the Commodore.	Unknown.
2	4.A	Partial Read Error	8/18/2011	A. STAR RANK BOXING B. KIDS' GAMES SPIT 40 SUPER HUEY	"KIDS' GAMES SPIT 40" on label is struck through.	Games
2	4.B	100%	8/18/2011	[Same as Above]	[Same as Above]	Games
2	Null	Failed. Read Error	8/18/2011	[Blank]	Read Error 74 (drive not ready). Disk is likely not formatted for the Commodore.	Unknown
2	Null	Failed. Read Error	8/18/2011	LET 1 BAK FB 1	Read Error 74 (drive not ready). Disk is likely not formatted for the Commodore. "LET 1 BAK" on label is struck through.	Unknown
2	5.A	100%	8/18/2011	1)XEVIOUS 2)WIZARD OF ARKYX 3)CAULDRON II KNIGHTS OF THE DESERT	Disk has two labels, one for each side, with the second label beginning with "KNIGHTS..." Hole Punch used to make disk double sided.	Games
2	5.B	Partial Read Error	8/18/2011	[Same as Above]	[Same as Above]	Games
2	Null	Failed. Read Error	8/18/2011	DISK	Read Error 74 (drive not ready). Disk is likely not formatted for the Commodore.	Unknown

2	Null	Failed. Read Error	8/18/2011	WORDSTAR 2000 PLUS GRAPHICS 1 Release 3.00 0419A8266	Read Error 74 (drive not ready). Disk is likely not formatted for the Commodore. Wordstar was a proprietary word processing software for IBM systems running DOS 2.0 or later. Sleeve label: SSBK 0907 BKup 4/10/87	Word Processing. Productivity. Unknown
2	Null	Failed. Read Error	8/18/2011	I-C Working Language	Read Error 74 (drive not ready). Disk is likely not formatted for the Commodore.	Unknown
2	Null	Failed. Read Error	8/18/2011	21 A - RDC KOALA PIX B - STATUS KOALA PIX	Read Error 74 (drive not ready). Disk is likely not formatted for the Commodore. However KOALA was a popular Commodore graphics editor, so the disk may be damaged, or may have been reformatted at some point.	Unknown
3	Null	Failed. Read Error	8/10/2011	Cut + Paste - Document Disk #1	Read Error 74 (drive not ready). Disk is likely not formatted for the Commodore.	Unknown
3	1.	100%	8/10/2011	H.E.R.O., Boulder Dash, GCS Zaxxon, Frogger Boot (water vers.) Choplifter, Ski, Pool, Int'l Soccer, Gyruss, Copy-Q		Games
3	Null	Failed. Read Error	8/10/2011	Hacker Congo Big Mac Ringo Potty Pidgeon	Read Error 74 (drive not ready). Disk is likely not formatted for the Commodore. Names of games match those for C64, so the disk was likely formatted for C64 at some point.	Games. Unknown

3	Null	Failed. Read Error	8/10/2011	GAME Load "*", 8, 1	Read Error 74 (drive not ready). Disk is likely not formatted for the Commodore. Label includes BASIC loading command, so the disk was likely formatted for C64 at some point.	Unknown
3	Null	Failed. Read Error	8/10/2011	Diagnostics S1 + S2	Read Error 74 (drive not ready). Disk is likely not formatted for the Commodore. Sleeve label: Laura	Unknown
3	2.	100%	8/18/2011	michaelsft™ dfile™ 128 Database Management System	Proprietary software. Includes manual and order form in disk sleeve.	Productivity. DMS
3	Null	Failed. Read Error	8/18/2011	Mavis Beacon Teaches Typing! Copyright 1987 The Software Toolworks Commodore 64/128 Sides A and B	Proprietary software. Read Error 74 (drive not ready). Disk is likely no longer formatted for the Commodore.	Unknown
3	Null	Failed. Read Error	8/18/2011	Bits, Pieces and Clues 1. African Adventure 2. Pirate Adventure 3. King Tut's Tomb LOAD'N'GO!™ SOFTWARE LOAD 'N' GO! SOFTWARE OPERATING INSTRUCTIONS Enter as follows: LOAD "MENU", 8, 1 (Press Return) This disk is for use with Commodore 64/128	Proprietary software. Read Error 74 (drive not ready). Disk is likely no longer formatted for the Commodore.	Unknown
3	3.	Partial Read Error	8/10/2011	Mavis Beacon Teaches Typing! Copyright 1987 The Software Toolworks Commodore 64/128 Sides C and D	Proprietary software. Many read errors during creation of disk image. Largely unusable.	Productivity

3	Null	Failed. Read Error	8/10/2011	Raid on Bungeling Bay Load "ROBB", 8, 1 Back Up	Read Error 27 (checksum error in header). Disk is likely no longer formatted for the Commodore.	Unknown
3	4.	Partial Read Error	8/10/2011	Miner 2049' ER Blue Max Bruce Lee Chinese Juggler	d64copy process stuck in loop, but disk image was completed. see opencbm terminal output for more.	Games
3	Null	Failed. Read Error	8/10/2011	Outrageous Pages Page Disk	Read Error 74 (drive not ready). Disk is likely no longer formatted for the Commodore.	Unknown
3	Null	Failed. Read Error	8/10/2011	Outrageous Pages A. Templates - Calendar - Invitation, Name Tags B. 2 column Diamond Newsletters 3 column Newsletters	Read Error 74 (drive not ready). Disk is likely no longer formatted for the Commodore.	Unknown
3	Null	Failed. Read Error	8/10/2011	Outrageous Pages Page Disk Page Disk	Read Error 74 (drive not ready). Disk is likely no longer formatted for the Commodore.	Unknown
3	5.	100%	8/10/2011	RX-50 FORMAT Turtles Teenage	"Teenage" on label struck through. Sleeve label: Paperboy DPaint Yellow Pages Relayer Menu	Games. Productivity
3	Null	Failed. Read Error	8/10/2011	SUPER ZAXXON RIVER RAID ---> to Load 1st load ACT Wizard of Wor GHOST BOOT	Read Error 74 (drive not ready). Disk is likely no longer formatted for the Commodore.	Unknown
3	Null	Failed. Read Error	8/10/2011	"PC Publisher" Backup 64mode Load "PCPUB", 8, 1 Disk ID P2	Proprietary software. Read Error 74 (drive not ready). Disk is likely no longer formatted for the Commodore.	Unknown
3	6.	100%	8/10/2011	CALKIT LOAD "0:*"; 8, 1 made with "Backup on this disk		Productivity

3	Null	Failed. Read Error	9/10/2011	Cut + Past / Letter document disk Flag	Read Error 74 (drive not ready). Disk is likely no longer formatted for the Commodore.	Unknown
3	7.	Partial Read Error	9/10/2011	PaperClip III (C-64) Program (System) Disk Load "START" + Run		Word Processing. Productivity
3	8.A	Partial Read Error	9/11/2011	*MASTER CATALOG* 04-08-94 *HACKS* LOAD "*" , 8, 1 *ON DISK*	Includes printed "Catalog Loading and Printing Instructions." Includes a handwritten letter from Tommy Tucker to Mr. Thomas.	Networking. Hacking. Library List
3	8.B	Partial Read Error	9/11/2011	[Same as Above]	[Same as Above]	Networking. Hacking. Library List. Unknown
3	9.	100%	9/10/2011	"PCSPELLER" ID C4 (128) also printer files		Productivity
3	10.	100%	9/10/2011	"Paperclip III" 128 - No line feed on Okimato (Version 2) Load "0:*" , 8 Disk ID C3		Word Processing. Productivity
3	11.	Partial Read Error	9/10/2011	Household Manager 1. Home Expense 2. Receipe Box (sic) 3. Vital Data 4. Word Master Senior	Includes photocopy of original software package.	Productivity
3	12.	Failed. Read Error	9/10/2011	Spreadsheet Storage Disk	Technically blank disk image. See 9-10TO.txt for more.	Productivity. Unknown
3	13.A	Partial Read Error	9/10/2011	#2 Outrageous Pages A – Fonts B - Clip Art		Word Processing. Productivity
3	13.B	100%	9/10/2011	[Same as Above]		Word Processing. Productivity
3	14.	100%	9/10/2011	Cal Kit	Sleeve label: Load "0:*" , 8, 1	Productivity
3	15.A	100%	9/10/2011	OutRageous Pages #1 A Program + Print B Template #3 Full certified 1/2 Letterhead	Sleeve label: #1 Outrageous Pages A - Program B - Template #3	Word Processing. Productivity

3	15.B	100%	9/10/2011	[Same as Above]	[Same as Above]	Word Processing. Productivity
3	16.	100%	9/10/2011	ZORK I Back Up		Games
3	17.	100%	9/10/2011	Defender Boot, 1A <Rocket><Ball> Rock N' Bolt Crystal Castle Popeye Boot		Games
3	18.A	100%	9/10/2011	Load this side up Load "TT", 8 (Typing Tator) or "LI", 8 (Letter Invaders Game)	Label is on Side B of Disk. Sleeve label: Side 1 + 2 Load "TT", 8 Load "LI", 8 Side 2 works Load "TT", 8 Typing Tator Load "LI", 8 Letter Invaders Game	Productivity. Games
3	18.B	100%	9/10/2011	[Same as Above]	[Same as Above]	Productivity. Games
3	19.	100%	9/10/2011	SKY TRAVEL	Sleeve Label: Shift Stop F3 at same time	Education
3	20.	100%	9/10/2011	SOME STUFF c64menu	Includes print out of disk directory	Games. Misc.
3	21.	100%	8/16/2011	(Mandelbrot) Graphics	Sleeve label: Mandelbrot [Print out of directory taped to sleeve]	Graphics. Visualization. Mapping Software. Science
3	22.	100%	9/10/2011	[Blank]	Sleeve label: REU Laura [Illegible] *P Utilities Load "*", 8 in 128 mode	Utility
3	23.	100%	9/10/2011	Desterm 128/80 Local BBS	See http://www.dickestel.com/arti cles/novaterm_desterm.txt	Networking. Terminal. Internet
3	24.A	100%	9/10/2011	[Blank]	Sleeve label: S1 - Popples - greeting with name - Load "Popples", 8, 1 S2 - Jingle disk - with card maker - Don't use fast load cartridge - Load "Jingle", 8, 1	
3	24.B	Partial Read	9/10/2011	[Same as Above]	[Same as Above]	

		Error				
3	25.	100%	9/10/2011	1541 SPECTRUM 128		
3	26.A	100%	9/10/2011	Classic Concentration Load "**", 8, 1		Games
3	26.B	Partial Read Error	9/10/2011	[Same as Above]		Games
3	27.	100%	9/10/2011	Datafile Rev. 3.1 A Database Management System by Mike Konshak for C64 + 1541		Productivity. DBMS
3	28.A	Partial Read Error	9/10/2011	Card Sharks	Sleeve label: Card Sharks	Games
3	28.B	Partial Read Error	9/10/2011	[Same as Above]	[Same as Above]	Games
3	29.	Partial Read Error	9/10/2011	Mavis Beacon Data Disk	Includes Instructions from Mavis Beacon Teaches Typing software package	Productivity
3	30.A	Failed. Read Error	9/10/2011	Evelyn Woods Dynamic Reader	Technically blank disk image	Productivity
3	30.B	Failed. Read Error	9/10/2011	[Same as Above]	Technically blank disk image	Productivity
3	31.	Partial Read Error	9/10/2011	Evelyn Woods Data Disk		Productivity
3	32.	Partial Read Error	9/10/2011	Bridge 5.0 Artworx®	Proprietary software.	Games
3	33.	100%	9/10/2011	Delphi's Oracle Type Load "BOOT", 8, 1	Sleeve label: Delphi's Oracle - needs no dongle	Games
3	34.	Partial Read Error	9/10/2011	five Delphi's Oracle		Games
3	35.	Failed. Read Error	9/10/2011	Delphi's Oracle write errors m t	Technically blank disk image	Games
3	36.	100%	9/10/2011	Backstreet[?] Data Disk Database Manager Load "**", 8 menu will appear on screen no "docs" needed		DBMS
3	37.	100%	9/10/2011	Datafile Rev. 3.1 A Database Management System by Mike Konshak for C64 + 1541		Productivity. DBMS

				(Michaelsoft) 1985		
3	38.A	Partial Read Error	9/10/2011	DLO 2 General 13/14	Sleeve label: GENERAL 2 13/14 Print out of disk directory taped to sleeve	DBMS
3	38.B	100%	9/10/2011	[Same as Above]	[Same as Above]	DBMS
3	39.A	Failed. Read Error	9/10/2011	[Blank]	Technically blank disk image	Backup
3	Null	Failed. Read Error	9/10/2011	[Same as Above]		Backup
3	40.A	Partial Read Error	9/10/2011	[Blank]		Games. Backup
3	40.B	Partial Read Error	9/10/2011	[Same as Above]		Games. Backup
3	41.A	100%	9/11/2011	[Blank]		Games. Backup
3	41.B	100%	9/11/2011	[Same as Above]		Games. Backup
3	42.	Partial Read Error	9/11/2011	dFile 128 A Database Management System		DBMS. Backup
3	43.	100%	9/11/2011	dFile 128 DBMA Sile Utility Program		DBMS. Utility
3	Null	Failed. Read Error	9/11/2011	[Blank]	Read Error 74 (drive not ready). Disk is likely not formatted for the Commodore.	
4	Null	Failed. Read Error	8/28/2011	ZORK		Games
4	1.	100%	8/28/2011	Pclip programs		Word Processing. Productivity
4	2.A	Partial Read Error	8/28/2011	Winter Games front	Read error pattern looks like the disk may be copy-protected.	Games
4	2.B	Partial Read Error	8/28/2011	Winter Games back	[Same as Above]	Games
4	3.	100%	8/28/2011	[Blank]		Educational. Science

4	Null	Failed. Read Error	8/28/2011	No. 1 Windows 3.1	Read Error 74 (drive not ready). Disk is likely not formatted for the Commodore.	
4	4.	100%	8/28/2011	[Blank]		Hacking. Nibbler. Utility
4	5.	Failed. Read Error	8/28/2011	[Blank]	Technically blank disk image	
4	6.	100%	8/28/2011	[Blank]	Actually a copy protected Activision game, using XEMAG 2.0 Copy protection. Game is "The Great American Cross Country Road Race." See http://c64preservation.com/dp.php?pg=xemag2	Hacking. Copy Protection. Games
4	7.	Partial Read Error	8/28/2011	[Blank]		Productivity. Utility
4	8.	Failed. Read Error	8/28/2011	SARGON II 03430 COMMODORE 64 Copyright © 1983 by HAYDEN BOOK COMPANY INC. HAYDEN SOFTWARE	Technically blank disk image.	
4	9.A	Partial Read Error	8/28/2011	Chestnut A1 ID Jumpman Load jumpman, 8, 1		Games
4	9.B	Partial Read Error	8/28/2011	Disk Bonus, A1		Games
4	10.	Failed. Read Error	8/28/2011	[Blank]	Technically blank disk image.	
4	11.	100%	8/28/2011	[Blank]		Games
4	12.	100%	8/28/2011	[Blank]		Games
4	13.	100%	8/28/2011	Games, G1 Public Domain Disk		Games
4	14.	100%	8/28/2011	GAMES G2		Games
4	15.	100%	8/28/2011	GAMES G3		Games
4	16.	100%	8/28/2011	Computer Science, CA		Educational. Science
4	17.	100%	8/28/2011	BUSINESS B1 Public Domain Disk PROGRAMS		Productivity

4	18.	Partial Read Error	8/28/2011	SCIENCE S1 Public Domain Disk		Educational. Science
4	19.	100%	8/28/2011	SCIENCE S2 Public Domain Disk		Educational. Science
4	20.	Partial Read Error	8/28/2011	Science S3		Educational. Science
4	21.	Partial Read Error	8/28/2011	SCIENCE S4 Public Domain		Educational. Science
4	22.	Partial Read Error	8/28/2011	GEOGRAPHY G1 Public Domain Disk		Educational. Science
4	23.	100%	8/28/2011	Paperclip - Spellpack Working Copy LOAD "PCLIP", 8, & Run		Word Processing. Productivity
4	24.	100%	8/28/2011	Pclip II Programs		Word Processing. Productivity
4	25.	100%	8/28/2011	Spell Pack Dictionary Working Copy		Word Processing. Productivity
4	26.	Failed. Read Error	8/28/2011	Talledega	Technically blank disk image.	Unknown
4	27.	100%	8/28/2011	commodore VIC-1541 TEST/DEMO 48TPI 1S		System. Hardware. Utility
4	28.	100%	8/28/2011	SpellPack 4040/2031/1541 DISK DRIVE & C64 TO USE TYPE- LOAD"0:*" ,8,1 -PRESS "RETURN" BATTERIES INCLUDED ©1984		Word Processing. Productivity. Proprietary
4	29.	100%	8/28/2011	SPELL PACK DICTIONARY 4040/2031/1541 FORMAT ©1984 BATTERIES INCLUDED		Word Processing. Productivity. Proprietary
4	30.	Failed. Read Error	8/28/2011	PaperClip 64 4040/2031/1541 FORMAT ©1982 BATTERIES INCLUDED march 84 64D		Proprietary. Word Processing.
4	31.	100%	8/28/2011	Pclip Programs III		Word Processing. Productivity

4	32.A	Partial Read Error	8/28/2011	Park Patrol Slap - Shot Track/Field Archive Copy/All		Games
4	32.B	Partial Read Error	8/28/2011	Summer Games Decathlon Toy Bizarre		Games
4	33.A	Partial Read Error	8/28/2011	Baseball		Games
4	33.B	Failed. Read Error	8/28/2011	Football	Information exists on disk image	Games
4	34.A	100%	8/28/2011	Bungling Bay		Games
4	34.B	Partial Read Error	8/28/2011	Pitfall II		Games
4	35.A	100%	8/28/2011	Bridge Motor - Mania Q-Bopper frogger Hellcat Ace		Games
4	35.B	100%	8/28/2011	Dragon Fly Sea Saw Catastrophe		Games
4	36.A	Partial Read Error	8/28/2011	Pole Position Ghostbusters Spyhunter Scramble Zenji Pegasis Laser - Zone		Games
4	36.B	Failed. Read Error	8/28/2011	Night Mission Pinball	Technically blank disk image.	Games
4	37.A	Partial Read Error	8/28/2011	Games I		Games
4	37.B	100%	8/28/2011	Games II		Games
4	38.A	Partial Read Error	8/28/2011	Sky fox		Games
4	38.B	Partial Read Error	8/28/2011	Beach + Head		Games
4	39.A	Partial Read Error	8/28/2011	Utilities I		Utilities
4	39.B	100%	8/28/2011	Education I		Educational

APPENDIX B – Online Communities and Resources for C64 Enthusiasts

Forums and Wikis

Lemon64 - <http://www.lemon64.com/forum/>

C64Wiki - http://www.c64-wiki.com/index.php/Main_Page

Recovery Options

ZoomFloppy - <http://store.go4retro.com/products/ZoomFloppy.html>

OpenCBM - <http://www.trikaliotis.net/opencbm>

1541U - <http://1541ultimate.net/content/index.php>

CatWeasel - http://www.jschoenfeld.com/products/catweasel_e.htm

Kryoflux - <http://www.kryoflux.com/>

Recovery Services - <http://www.floppyrecovery.net/>

Online Archives

C64 Preservation Project - <http://c64preservation.com/>

Project64 - <http://project64.ath.cx/index.htm>

Replay Resources - http://ar.c64.org/wiki/Main_Page

RETROPORT - <http://retroport.de/>

C64.COM - <http://www.c64.com/>

Commodore Server - <http://www.commodoreserver.com/>

N2DVM Commodore site - <ftp://n2dvm.com/Commodore/>

COCOS, the Commodore Computer Site list - <http://www.c64.cc/>

SearchDBC64 - <http://noname.c64.org/csdb/release/?id=76967>

PokeFinder - <http://ftp.pokefinder.org>

Enthusiasts and Technical Specs

C= Computer Connection - <http://home.comcast.net/~safeharborbay/tcr/cbm/>

Bo Zimmerman's Commie Page - <http://www.zimmers.net/commie/index.html>

SID in-depth information site - <http://sid.kubarth.com/>

FairLight64 - <http://www.fairlight.to/>

Commodore.ca - <http://www.commodore.ca/manuals/default.htm>

Emulation Options and Tools

Power64 - <http://www.infinite-loop.at/Power64/>

Vice - <http://www.viceteam.org/>

StarCommander - <http://sta.c64.org/>

Frodo - <http://frodo.cebix.net/>

64Copy Central - <http://ist.uwaterloo.ca/~schepers/personal.html>

CCS64 - <http://www.ccs64.com/>

Stores

64 And More - <http://www.64andmore.com/catalog/>

4Jays Video Games - <http://4jays.net/store/>

APPENDIX C – OpenCBM Error Codes and Matching CBM DOS Messages

Code	Error	Type	1541 error description
03	21	Seek	No SYNC sequence found.
02	20	Seek	Header descriptor byte not found (HEX \$08, GCR \$52)
09	27	Seek	Checksum error in header block
0B	29	Seek	Disk sector ID mismatch
02	20	Seek	Header block not found
04	22	Read	Data descriptor byte not found (HEX \$07, GCR \$55)
05	23	Read	Checksum error in data block
0F	74	Read	Drive Not Ready (no disk in drive or no device 1)
06	24	Write	Write verify (on format)
07	25	Write	Write verify error
08	26	Write	Write protect on
0A	28	Write	Write error
01	00	N/A	No error.