UNITED STATES COPYRIGHT OFFICE

Long Comment Regarding a Proposed Exemption Under 17 U.S.C. § 1201

[ ] Check here if multimedia evidence is being provided in connection with this comment

ITEM A. COMMENTER INFORMATION

Andrew Berger
Senior Digital Archivist
Computer History Museum
1401 N. Shoreline Blvd
Mountain View, CA 94043

I am writing as an individual practitioner in the fields of software and digital preservation. Although in this comment I draw on my experience working as a digital archivist, my opinions are my own and do not necessarily reflect those of my employer, the Computer History Museum. I include my employment address for contact and identification purposes only.

ITEM B. PROPOSED CLASS ADDRESSED

Proposed Class 9: Computer Programs – Software Preservation

ITEM C. OVERVIEW

I am writing in support of the proposed exemption for libraries, archives, museums, and other cultural heritage institutions to circumvent technological protection measures on lawfully acquired computer programs for the purposes of preserving computer programs and computer program-dependent materials.

After reading the previously filed comments, especially those filed in opposition to this exemption, I believe there is a need for more detailed information about what the work of software preservation actually looks like. I base my comments on real world examples where technological protection measures pose a barrier to preservation.

I first describe two case studies from my personal experience where technological protection measures affected preservation and research activities. Next, I summarize two articles written by other practitioners on the subject of copy protection, both of which are relevant to this proceeding. Finally, I address some of the concerns previously expressed in the opposition comments on this exemption.

ITEM D. TECHNOLOGICAL PROTECTION MEASURE(S) AND METHOD(S) OF CIRCUMVENTION

In my work as a digital archivist, I have encountered technological protection measures multiple times. The following are two cases in which researchers requested access to historic software that, upon examination, turned out to employ TPMs.

[Note: In the interest of protecting patron privacy, I have not identified the researchers who made the requests.]

Case study 1: Software that will only run during a specific date range

Title: Real G2 Next Generation
Format: CD-ROM
Date: 1998
Location: Computer History Museum, catalog number 102685637

Background

In 2016, a researcher requested access to the 1998 CD-ROM title, Real G2 Next Generation. This CD-ROM was issued as part of a developer program for the RealNetworks company. It contains a preview release of RealSystem G2, which was intended to allow developers to become familiar with the software in advance of its full public release. The CD-ROM contains both extensive documentation and executable application files.

When asked whether they wanted to simply view the files on the CD-ROM, or to see the application in a running state, the researcher replied that they would like to be able to run it. As the Real G2 Next Generation CD-ROM was released for Windows in 1998, setting up an environment in which to install it required running a copy of Windows 98.

Preservation actions

Ordinary CD-ROMs like this one do not have a long shelf life, and each use can shorten that life. In the interest of preservation, a disk image – essentially a complete copy – of the CD-ROM was created to serve as a preservation master. This master copy can be used either to create new physical CDs or "virtual" CDs for use with virtual machine programs.

Creating a disk image copy is only the first step in preservation, however. To fully verify that a valid copy has been created, it is necessary to verify that the files can be opened and the applications installed. And to do that in this case, it was necessary to create a Windows 98 operating system environment.

Setting up a Windows 98 environment on current hardware could itself be a case study in software preservation, but since I was able to install a genuine copy of Windows 98 in a virtual machine with a valid certificate of authenticity, I leave those details out. The real roadblock came when I attempted to install the RealSystem G2 applications.

Technological protection measure encountered

Two applications, RealPublisher and RealPlayer, both failed to install on Windows 98. Both returned the same error message indicating that the versions on disk had "expired" and that the user should install a new version. But of course a new version is not the version I was attempting to preserve, and not the version the researcher had requested to see. The researcher wanted to examine RealSystem G2, not a later update or reissue.

After further analysis, it became clear that both applications check for the current system time and date. If the date reported is later than a certain date in 1998, then the applications cannot be installed. But if the system time is changed to report a date from the summer of 1998, it would be possible to install and run the applications. Windows 98 allows users to change the system time, which provides a potential path around the protection measure.

This protection measure makes sense in its original context, that context being 1998. RealNetworks had a legitimate need to protect their preview release from unauthorized use as a replacement for the full release, which was then still to come. Nearly twenty years later, with RealSystem G2 having long since been superseded, this protection is no longer a market necessity. Yet preserving and providing research access to the software still requires circumventing the technological control.

Case Study 2: Software that is dependent on a program that employs TPMs

Title: Building Design and Construction
Format: 5.25" floppy disk
Date: 1983
Location: Computer History Museum, catalog number 102639462

Background

In 2017, a researcher studying the history of the use of computers in architectural design requested to view Building Design and Construction, a 1983 title by Software Arts, Inc. When asked whether the request was to view the files on the program disk or to view the application in a running state, the researcher replied that they wanted to be able to interact with the program, if possible.

The documentation for Building Design and Construction indicates that the disk was intended for use only with a program called TK!Solver, mathematical software first released in 1982, also by Software Arts, Inc. Essentially, Building Design and Construction provides a set of models that TK!Solver manipulates. A copy of TK!Solver 1.2 is also in the Computer History Museum collection (catalog number 102639467). Released in 1983, this version of TK!Solver should be compatible with Building Design and Construction.

Preservation actions

As with the Real G2 Next Generation CD-ROM from the first case study, best practice for preserving software floppy disks entails creating disk images and then verifying these copies. In this case, disk images were created for both Building Design and Construction and TK!Solver, but despite making multiple attempts to run TK!Solver on multiple versions of DOS running in virtual machines, along with attempts using the DOSBox emulator, none met with success. It is

possible that the program could have been run on a restored 1983 PC, but none was available for that purpose.

Without verification, neither the copy of Building Design and Construction nor the copy of TK!Solver can be said to have been fully preserved.

Technological protection measure encountered

The disk for Building Design and Construction does not seem to employ any special technological protection measures. However, the disk for TK!Solver, does employ such measures. I have not identified the exact nature of the protection employed by TK!Solver. Nevertheless, the TK!Solver user manual clearly states that the disk is copy-protected: "If you try to make a backup copy of your program diskette, the copied program will not work."[1]

Given the legal and the technical barriers to identifying how to circumvent the protection, I decided to provide the researcher with access only to the files on the Building Design and Construction disk. This access was provided in person, on a modern computer in the museum's reading room. Many of these files can be read using a text viewer, but this level of access pales in comparison to being able to see the software in action.

An additional point to be made here is that the inability to run TK!Solver adversely affects any attempt to preserve and access other contemporaneous programs that require that same version of TK!Solver. Building Design and Construction, the subject of the initial research request, is only one of a number of "solver packs" created to go with TK!Solver in 1982 and 1983.[2]

Looking at the bigger picture, TK!Solver is itself only one of many examples that could be given of programs that are required to access other programs. Indeed, the same could be said of the version of DOS required to run TK!Solver. Software preservation is full of these types of cascading dependencies.

Other research on software TPMs

Two recent articles identify additional TPMs created in the 1980s that continue to impact preservation efforts today. These articles address specific computer systems, but many of the techniques described are not unique to those systems.

Peter Ferrie's 2016 article, “A Brief Description of Some Popular Copy-Protection Techniques on the Apple][Platform,” identifies so many different copy-protection techniques as to defy easy summary.[3] Ferrie outlines over a dozen different categories of copy-protection, many of which contain multiple examples. Ferrie notes that even this extensive list is not fully complete.

---

[1] In the interest of full disclosure, I should note that I cannot be sure that the TPM is what is preventing the program from running. The physical disk could be damaged, or there could be errors in the disk image. It is not always clear why a given disk image cannot be accessed, which is why attempting to access disk images and then analyze any failures is such an important part of preservation.

[2] Other titles include: Introductory Science, Mechanical Engineering, and Financial Management.

[3] Peter Ferrie, “A Brief Description of Some Popular Copy-Protection Techniques on the Apple][Platform,” *PoC || GTFO*, no. 0x10 (2016): 39–74.

It appears that just about any component of the Apple II system could be exploited for copy-protection purposes.

As an archivist, my two main conclusions from reading Ferrie's article are that, first, it can be difficult to predict which TPM you are going to find on a given disk, given the sheer number of different techniques that were created; and second, historical TPMs are themselves a valid subject of research. It is difficult to see how one could study this topic without also studying circumvention techniques. Copy protection, and the contemporaneous efforts to defeat it, is an important part of computer history.

Along similar lines, John Aycock and Andrew Reinhard's 2017 article, "Copy Protection in Jet Set Willy: Developing Methodology for Retrogame Archaeology" is a deep analysis of copy protection for a single software title, Jet Set Willy.[4] Jet Set Willy was released in 1984 for the ZX Spectrum in the now obsolete form of a software cassette tape.

The specific technique described here is the use of a physical card, issued along with the cassette containing the software, for authentication purposes. This is just one example of a TPM that relies on external media for protection, a category described in the article as a "what you have" strategy.[5] Methods for circumventing these types of strategies include identifying how the program validates user input when the user is asked for information found on an accompanying object. In order to check the user's answer against the "correct" answer, the program either needs to store the answer itself, or needs to store a method of calculating the answer when needed.

Aycock and Reinhard's article is also notable for the way the authors describe the provenance of their digital copy of the program, and for how they cite it. Normal citation practice generally identifies a source such as a published work in a way that enables readers to locate it should they wish to evaluate it for themselves. But concerns about copyright lead Aycock and Reinhard to identify their copy using a checksum, which is based on the content of the copy, rather than citing a location. This makes it possible for future researchers to compare their own copies with Aycock and Reinhard's to see if they are exact matches, but such researchers would need to independently acquire those copies.

This is a particularly dramatic example of copyright concerns adversely affecting scholarly practice. In the ideal case, a researcher would be able to cite a copy of Jet Set Willy held in an institution such as a library, archive, or museum, and the holding institution would be able to document and describe how the original and its preservation copy were acquired and created. Instead, the researchers ended up relying on a copy of uncertain provenance.

ITEM E. ASSERTED ADVERSE EFFECTS ON NONINFRINGING USES

Building on the preceding discussion, I see the most significant adverse effects of the current prohibition on circumvention as the following:

[4] John Aycock and Andrew Reinhard, "Copy Protection in Jet Set Willy: Developing Methodology for Retrogame Archaeology," *Internet Archaeology*, no. 45 (2017), https://doi.org/10.11141/ia.45.2

[5] Ferrie also briefly discusses strategies where the user is required to provide information found outside of the program, such as information taken from a manual. See Ferrie, 39-40.

1. It prohibits activities necessary for preservation. Such activities include:

- The act of copying data from physical media, when the TPM is designed to prevent making copies.
- The act of running the software from a preservation copy in order to verify that that copy is a valid one, when the TPM is activated as part of the program's execution.

2. It discourages institutions from engaging in dedicated work to preserve protected software.

Given limited resources, libraries, archives, and museums are more likely to devote more time to unprotected software than to protected software, as unprotected software carries fewer risks. Protected software may be left in a partially preserved state – lacking full verification – or not addressed at all. Considering the fact that physical media in historic software collections has already begun to degrade, there is an added urgency to carry out this work sooner rather than later.[6]

Over time this could actually lead to a distortion in the way software history is studied, as software titles that employ TPMs may, contrary to whatever historical significance they may have had, end up receiving less attention in the years to come. It would be an unfortunate irony if historically influential pieces of software could not be studied easily in the future because the protections that helped maintain their value during their years of active use and support remained in force decades after they reached "end of life."

3. The prohibition adversely affects those seeking to conduct historical research on protected software, as legitimate copies with clear provenance may not be available to them.

The effect of this is that those seeking to carry out noninfringing research are often forced to turn to websites that distribute copies of software of uncertain provenance. These copies may have been modified in ways that are not obvious or documented, which may have a negative effect on research.[7]

Response to opposition arguments

Finally, I will close by addressing two of the objections raised in the comments filed in opposition to this exemption. Some of these comments demonstrate an apparent lack of familiarity with both software preservation and with research in software history.

First, there is the objection that the exemption is too broad. The reality is that the sheer variety of historical technological protection measures makes it difficult to enumerate them all in advance of encountering them in real world collections. A collection of historical software is not going to

[6] For a study of the success rates of imaging disks, see Denise de Vries and Melanie Swalwell, "Creating Disk Images of Born Digital Content: A Case Study Comparing Success Rates of Institutional Versus Private Collections," *New Review of Information Networking* 21, no. 2 (July 2, 2016): 129–40, https://doi.org/10.1080/13614576.2016.1251849 Note that disks in private collections appear to be significantly more at risk, making institutional support all the more important. This study took place in Australia, but given the physical uniformity of computer media worldwide it is applicable to collections in the United States.

[7] Aycock and Reinhard discuss this problem at length.

come organized by the method of technological protection and it is not realistic to expect collections to be reorganized in that manner.

From a collection management standpoint, it is far more efficient to work through a collection based on other factors, such as age, format, operating system, or subject. In the course of doing so, software preservationists are likely to encounter a whole range of technological preservation measures. They need to be able to address these issues as they arise, even if the specific TPM is not one they have encountered before.

Second, there is the objection that the existence of reissued, updated, or backwards compatible software somehow eliminates the need to preserve historic software.[8] While for some purposes it may be enough to be able to convert a file to an updated or compatible format, this does not change the fact that there are situations where migration or conversion is not acceptable.

These situations include, but are not limited to:

- Software-dependent materials that have no contemporary target format.

  For example, the 1983 Building Design and Construction, described above, needs to be run using a period-appropriate version of TK!Solver.

- Software-dependent materials where conversion to a new format entails an unacceptable loss or alteration of information.

  Conversion is not always a perfect operation. Sometimes conversion errors affect only form, such as a font or a layout in a document. But sometimes the errors can alter the meaning of documents.[9] And sometimes fonts or layouts are themselves historical subjects, making changes in form unacceptable for the research purpose at hand.

- Research where a particular version of the software is itself the subject.

  The fact that software can be updated or reissued in a new version does not change the historical fact of its existence in an earlier form. Asking someone to use a current version of a software program as the basis for research on an earlier version of that program is as absurd as asking someone to write a history of a film using only a later remake.

  Some of the most innovative, historically-based work being done today is in the area of software history. Access to preserved copies of original software, rendered faithfully to the period in which it first appeared, is essential to making that work possible.

[8] A related claim, that the release of MS-DOS source code for versions 1.1 and 2.0 means that DOS-based programs are no longer at risk, is also mistaken (SIIA comment, 2). Source code is not a replacement for an executable application, and DOS had many more versions than 1.1 and 2.0.

[9] For examples, see the 2012 Archives New Zealand report, "Rendering Matters", http://archives.govt.nz/rendering-matters-report-results-research-digital-object-rendering (last accessed 3/13/2018).

DOCUMENTARY EVIDENCE

[No documentary evidence is being submitted in this section, but evidence has been cited throughout in the footnotes.]