

5th International Digital Curation Conference

December 2009

Assisted Emulation for Legacy Executables

Kam Woods and Geoffrey Brown
Indiana University
School of Informatics and Computing

July 2009

Abstract

Emulation is frequently discussed as a failsafe preservation strategy for born-digital documents that depend on contemporaneous software for access. Yet little has been written about the contextual knowledge required to successfully use such software. The approach we advocate is to preserve necessary contextual information through scripts created during the preservation workflow designed to control the legacy environment. We describe software designed to minimize dependence on this knowledge by offering automated configuration and execution of emulated environments. We demonstrate that even simple scripts can reduce impediments to casual search and use of the digital objects being preserved. This can help eliminate both a dependence on physical reference workstations at preservation institutions, and provide users accessing materials over the web with simplified, easy-to-use environments. Our implementation is applied to an existing collection of over 4000 virtual CD-ROM images containing thousands of custom binary executables.



Emulation and Access

Emulation is a key strategy for ensuring long-term access to born digital materials, yet it has a vulnerability that is rarely discussed – its dependence upon original software and the corresponding assumption that future users will remember how to “drive”. Even a seemingly simple task such as installation of software can be a formidable impediment to use. We describe a project to build tools that will assist these future users by automating tasks such as software installation that could otherwise compromise usability. Additionally, we discuss software to accommodate other common tasks, such as exporting data and generating reports using these environments. The work assumes emulation is a necessary strategy for preservation of some materials and that the underlying technology is sufficiently well understood that preservation of emulation environments is likely to be viable. The specific issues that our work addresses are automating access to materials within an emulation environment, preserving and sharing emulation environments, and the technology required to automate emulation

Our software provides users with access to legacy executables in automatically configured virtual machines using simple installation scripts. Scripts and relevant documentation are stored alongside the original digital objects and metadata. Our goal has been to address basic questions about preserving any contextual knowledge associated with legacy executables. We began with simple automation of common tasks, and additionally considered “wrapping” older (DOS-based and early Windows GUI) applications to generate reports or export data in useful ways.

Automation allows users to quickly browse and obtain information from these programs much as they would traditional static document types, without the need to learn arcane installation procedures, risk contamination of an existing environment, or be burdened with manual reconfiguration of the virtual machine for individual items. We show that with a small amount of coding - less than 600 lines of C# code to guide the user, configure an emulation environment, and run associated install scripts, and less than 100 lines of Java to link browsing of a web archive to the local application – we can create stable, repeatable environments for virtually any application type.

We believe digital archives with a mandate for open access must inevitably implement strategies of this type. Modern users are accustomed to having broad levels of access to a variety of archival materials over the Internet, but do not necessarily have the expertise to reconstruct the environments necessary to fully use these materials. A survey of more than 4000 legacy CD-ROMs held in one collection at Indiana University exposed more than 1900 unique binary executables distributed without source. These programs encode historically significant scientific, economic, legislative, environmental, and social data – data that is media-rich and frequently cannot be migrated, reprocessed, or viewed in a static document form without information loss.

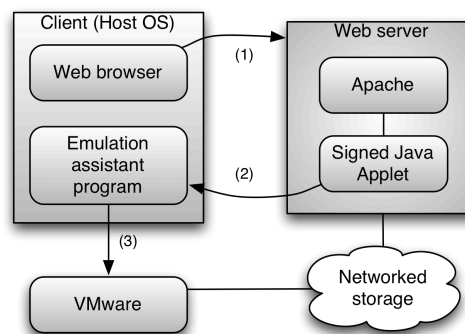
The development of archival strategies and the software tools that support them has increasingly become focused on a consideration of future access -- that is, who will be looking for these materials, how they will find them, and under what conditions they will be used. Many of the assumptions made in these strategies are predicated on

the idea that the majority of these materials will be stored as traditional documents -- frequently word processing, spreadsheet, database, or presentation formats.

Complex digital objects such as executables present a significant access and search problems. Modern indexing schemes are based primarily on document metadata and, to a lesser extent, on document content. As complexity of the digital object increases, facilitating end-user access to these materials becomes as critical a preservation problem as retention of the metadata used to describe them. Assisted emulation schemes such as the one presented here are a first step in addressing this issue.

Approach

Our approach focuses on improving user navigation of and access to programs located within a networked collection of CD-ROM ISO images presented as a single filesystem and accessible through a web interface (<http://www.cs.indiana.edu/svp/>). When a user clicks on a link to a virtual disk image for which an installation script is available, a Java applet executes a helper application on the user's workstation taking the form of a "wizard" that informs the user and provides the option of mounting the ISO image within a guest Windows XP image.



[Caption] Figure 1: Client request for networked resource.

Figure 1 illustrates the basic interaction between a user with an appropriately configured workstation and a remote site with legacy materials held on networked storage. A user navigating an online archive via a web browser locates a resource by entering search terms and selecting a desired ISO image from the result list. The user may browse the contents of the image directory structure in the browser, but for those images containing executables, two links are provided for direct access: one which executes the local helper application to configure a virtual machine using the appropriate network resource (.iso file mounted as a "virtual" D: drive), and one which provides the option of downloading the entire image.

There are numerous advantages to this approach. ISO images of CD-ROMs are large (typically hundreds of megabytes), placing considerable overhead on the network and subjecting users on typical network connections to a lengthy wait. However, for large resources like .iso files stored in a distributed, networked filesystem such as AFS, a mount of the resource as a virtual drive in a local VM results in transfer over the

network of *only those parts* of the image that are required.

These two features - automated virtual machine configuration and networked resource storage - are designed to eliminate the need for libraries to maintain physical “reference workstations”, or hardware platforms with customized software environments to support access to legacy electronic materials. We focus in particular on executable software originally distributed on legacy media, including CD-ROMs and floppy disks. We envision future use profiles in which the experience of a user interacting with a particular collection using modern hardware is identical irrespective of their physical location or local software environment.


For an increasing body of legacy executable software, however, the user may find that lack of knowledge about an outdated operating system -- or the time required to learn or reacquire that knowledge -- hinders their ability to effectively browse the available materials. This loss of knowledge is already occurring for older Windows and MS-DOS environments. What percentage of modern users browsing an archive remember (or were ever familiar with) the intricacies of editing AUTOEXEC.BAT and CONFIG.SYS files, or could diagnose silent installation failures due to a hardware incompatibility? Careful documentation and operational metadata can mitigate the problem somewhat, but eventually these instructions become little better than incantations that the user must follow on faith, with little recourse should they fail.

There are further problems of environmental integrity. A dedicated hardware workstation must either be preconfigured with every available piece of legacy software within an archive (a daunting thought, not to mention problems associated with conflicting requirements and system security), or make available to the user installation and execution permissions that would lead to inevitable corruption or misuse. This can be overcome with the use of a virtual machine “snapshot” that is returned to a base configuration at the end of a browsing session, but this again places the burden of installation and execution on the user.

By providing a software-assisted method for automating interaction with the virtual machine (including configuration and launch of the actual VM and installation of the desired software), we eliminate the problem of requiring extensive environmental knowledge from non-technical users. Our implementation provides direct control over a variety of VMware products, access to network disk resources, automated installation and execution of legacy binaries. The publicly available C# source code can be readily adapted to integrate into a wide variety of environments, and the associated Java applet code can be used to link user activity in a browser to execution of a local VM. As the associated automation scripts are linked to individual ISO images and assume the presence of a “clean” virtual machine snapshot, the environment is effectively “frozen”, eliminating concerns of future environmental integrity or software changes. Likewise, additional scripts can be added to the archive as necessary without requiring alteration of the helper application.

Background

Emulation Platform



In order to be effectively deployed in a production preservation setting, an emulation platform must implement features not only for technical preservation activities, but also for administration, configuration, and maintenance. These facilities should be organized as management and automation APIs that can be used to normalize a virtual machine environment irrespective of external factors such as where it has been deployed.

Emulation tools prepared by the digital preservation community remain largely primitive in terms of both technical and administrative features. Commercial products including VMware, Parallels, and CrossOver largely set the standard for feature sets that will arguably be required in any large scale preservation solution, including robust APIs for 3rd-party development, support for a wide variety of virtualized I/O, network, and hardware devices, and automatic methods for deployment.

For this project we used VMware Workstation and VMware Server, along with the VIX automation API due to its relative stability and widespread use. However, our solution uses fewer than 100 lines of C# code to provide the needed automation support (start and stop of the VM, device mounting, and file operations), and could readily be ported to a fully open source solution such as QEmu. While configuration of products such as QEmu is somewhat less “user friendly”, APIs such as libvirt¹ provide equivalent functionality to available commercial packages.

The maturity and stability of these existing commercial and open source products - and the fact that they are routinely used in mission-critical environments to provide users with flexible access to customized environments irrespective of the underlying hardware – suggests that they meet the current requirements of the digital preservation community.

Related Work


Development efforts within the digital preservation community have focused on the creation of virtualization platforms that are flexible, robust, and can be preserved and readily executed on future platforms. Projects with an emphasis on “write once, run anywhere” code, such as the JPC and Dioscuri, remain primitive with respect to emulation of many necessary features of legacy platforms.

Additional evidence of the need for automated emulation environments linked to web-accessible archives comes directly from practicing scientists. At sites like nanoHub², a distributed system enables scientists to conduct simulations in software tools written for legacy platforms via a VNC client accessible via a website. For those applications for which a GUI was not created, a software API is provided to generate an appropriate interface matching the original input and output requirements. It should be noted that many legacy (interactive) applications do not have a straightforward I/O design, limiting the extensibility of this approach.

Much of the work on emulation that has emerged from the preservation community has focused on questions that largely ignore the applicability of modern

¹ <http://libvirt.org/architecture.html>

² <http://www.nanohub.org>



commercial products. These include “Will a reliance on emulation adversely affect other preservation strategies?”, “How can we ensure that the emulator itself is preserved?”, and “What vulnerabilities are exposed when relying on imperfect emulation methods?”. We believe that the issue of customization on demand – that is, whether we can reliably provide users with emulated environments to suit their needs on a per-resource basis, is of equal importance. This is not just a *technical* issue, but one of building a sensible access requirements for existing and future archives.

In addressing the issues, the OAIS model makes specific reference to legacy Windows operating systems, and notes the following:

“[I]t may not be possible to fully simulate all of the old hardware dependencies and timings, because of the constraints of the new hardware environment. ... [D]etermining that some new device is still presenting the information correctly is problematical and suggests the need to have made a separate recording of the information presentation to use for validation. Once emulation has been adopted, the resulting system is particularly vulnerable to previously unknown software errors that may seriously jeopardize continued information access. Given these constraints, the technical and economic hurdles to hardware emulation appear substantial.”

The suggestion that a “reference recording” be retained to verify software operation is oddly naïve. It suggests a focus on software that operates with a minimum of interaction, and does not branch significantly. The vast majority of software could not be effectively verified by such a method. More to the point, the majority of legacy software for Microsoft operating systems was designed to run on a diverse range of hardware conforming to a compatible standard, to which a mature emulation platform such as QEmu or various VMware products (along with the requisite virtual machines, device drivers, and configuration options) comply.

Software

Overview

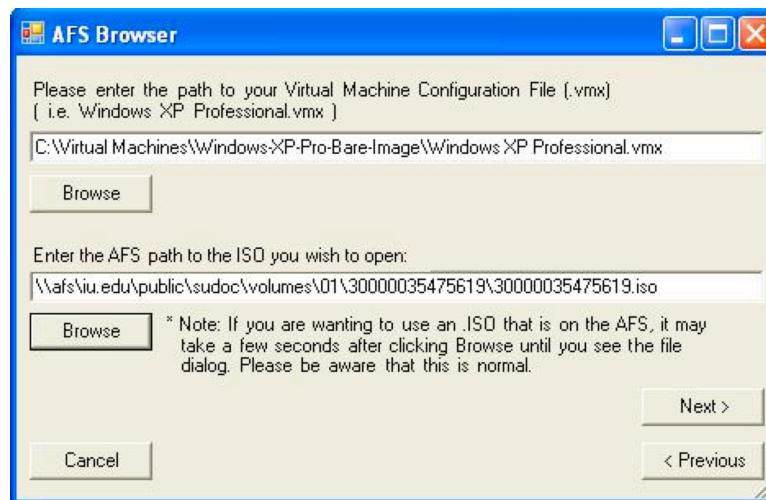
Our software includes four basic components: a small “wizard” application to provide the user with assistance in configuration and execution of an emulation environment linked to a networked archival resource, a (signed) Java applet to execute the wizard on the workstation as necessary when the user navigates to a resource of interest on a given archival website, scripts to support installation and execution of legacy applications held on media images within the archive, and scripts to support data extraction via export facilities within the applications.

When a user clicks on a link to a virtual disk image for which an installation script is available, the Java applet executes a software application on the user’s workstation taking the form of a “wizard” that informs the user and provides the option of mounting the ISO image within a guest Windows XP image with or without the installation.



[Caption] Figure 2: Emulation platform selection.


Selection of a resource via a clicking an appropriate link to a signed Java applet (running server-side) executes a helper application installed on the local workstation. This application first prompts the user to select the desired emulation package (Figure 2). For this implementation, we tested the software with both VMware Server 1.0.2 and VMware Workstation 6.0.4, to ensure compatibility of the API calls with a variety of products.



[Caption] Figure 3: Client request for networked resource.

The user is presented with a final confirmation of their selections, including the location of the requisite VM (the application automatically searches for paths corresponding to valid virtual machines and presents a default choice in an intermediate dialog) and path to an ISO image corresponding to their original selection on the web site. The application (shown in this stage in Figure 3) may also be run in a “standalone” mode, in which the user may run it directly and interact with a virtual machine and virtual disk image of their choosing.

Once these selections have been made, the software helper on the local workstation identifies the appropriate commands in the VMware configuration file, rewriting `ide1:0.filename` and `ide1:0.device type` to point to the appropriate ISO



stored in the AFS. No further changes are required, although the application performs some basic tests to ensure that the user has not requested a missing resource. Finally, connects to the appropriate logical volume on the AFS to determine if any install scripts are available for the requested ISO. After a final user confirmation, the application starts the virtual machine from the existing snapshot, copies over any associated installation script, and executes it.

On-site use

The software described above is intended primarily to replace or augment dedicated workstations maintained on-site at libraries where legacy collections are accessible over a network. Our implementation generates automatically Microsoft Windows virtual machines with CD-ROM ISO images “mounted” as virtual drives over a network connection to an AFS domain where the images reside. With minor modifications, it can accommodate any VM supported by the emulation software (VMware) and any form of media capable of being mounted in the VM.

This has many advantages from an administrative perspective. Administrators can deploy this solution (or a modified one using the available source) simply by adding a applet-based link that points to the desired resource within existing pages. In our testbed, these links were generated automatically in the following form:

```
<applet code="RunExecutable.class" archive="RunExecutable.jar"
width="120" height="35">
<param name="exePath" value="C:\Legacy Emulation Assistant\Legacy
Emulation Assistant.exe">
<param name="afsPath"
value="//afs\iu.edu\public\sudoc\ volumes\04\30000038669341\3000003
8669341.iso">
</applet>
```

The Java applet simply reads two parameters; one corresponding to the expected path of the helper application on the user’s workstation, and one corresponding to the networked resource. These can easily be customized depending on the site requirements; the applet exists only to provide a “trusted” link between the resource provider (online archive) and the workstation.

Remote use

A major advantage of this method is that the experience of any remote user (running a compatible version of the VMware emulation software) browsing the Web-accessible archive is essentially identical to that of a user browsing in the “controlled” on-site environment. Because modern browsers operate in a “sandbox” designed to shield the user from malicious sites, we used a signed Java applet to link browsing activity with the local executable necessary to configure, boot, and perform required installations in a local virtual machine.

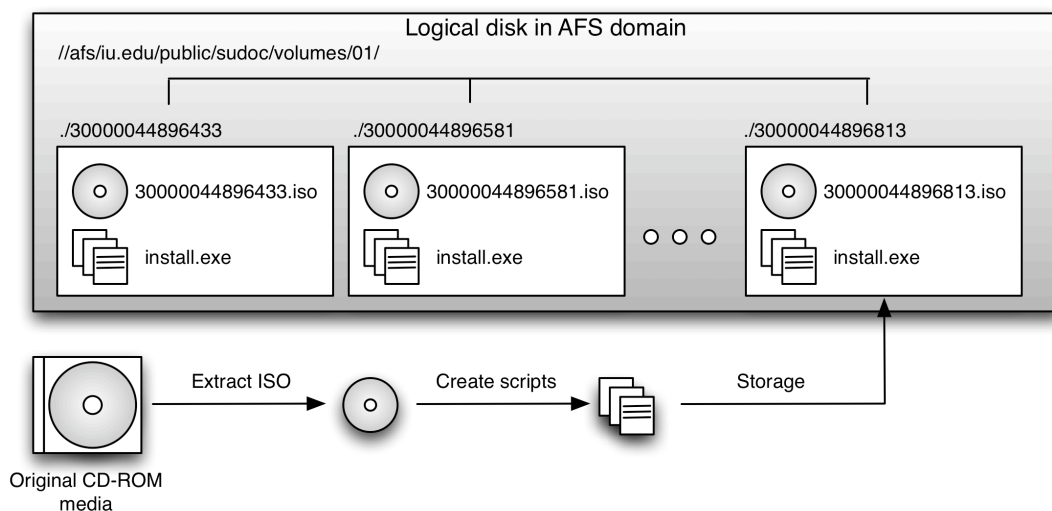
The implementation can easily be adapted to the needs and capabilities of a variety of users or user bases. The Java applet can be adapted to examine a variety of paths for available helper applications, and the local application itself can poll the system to determine what virtualization platforms are available. In the current implementation, this extends to the VMware Server and Workstation products.

However, it can be modified with a minor amount of development effort to search for additional platforms (such as QEmu) and execute the relevant API calls to configure them.

Preparing the Archive

Scripted Installation Support


We selected a set of approximately 150 candidate executables for scripting from a listing of more than 1900 available programs in our existing repository. A work-study student completed each installation in a “clean” Windows XP virtual machine, recording the necessary GUI actions, compatibility modifications, and (where necessary) alterations to the original installation procedure. Each action was recorded and encoded as an appropriate AutoIt command. The scripts were subsequently compiled as executables and stored alongside the original archival materials in the networked Andrew File System disk array.



[Caption] Figure 4: Organization of the virtual archive.

Due to variability in installation routines and the fact that most legacy Windows executables are distributed as compiled binaries without source code, installation and configuration cannot be handled programmatically through the Windows API. Fortunately, a variety of high quality Windows GUI scripting languages exist for the express purpose of automating user-level tasks. We used the AutoIt tool, which provides a simple BASIC-style syntax for automation of the GUI. The following code (taken from a sample collection developed as part of our testbed) demonstrates automation of a simple Windows 95-era setup procedure:

```
Run("D:\Win95\SETUP.EXE")
WinWaitActive("Welcome", "&Next >")
Send("!n")
WinWaitActive("Choose Destination Location", "&Next >")
Send("!n")
WinWaitActive("Select Program Folder",
```



```

                                "&Program Folders:")
Send("!n")
WinWaitActive("Start Copying Files", "&Next >")
Send("!n")
WinWaitActive("Setup Complete", "Finish")
ControlClick("Setup Complete", "Finish", 1)

```

The installations scripts are frequently even simpler than this, particularly for DOS-era programs for which only one or two commands are necessary, or for which the installation assumes a default path without informing the user. For those DOS-based applications that required additional changes to configuration and initialization files, help documents were included to describe the process of preparing the scripted installation.

Scripted Data Extraction

We observed that many of these legacy applications serve a single purpose – to reformulate data stored in a common file format (such as dBase or Microsoft Excel) and present the resulting “views” to the user. In one common case, raw data such as economic or census figures is stored in dBase files upon which various joins and filters are performed based on selections made by a user in a Windows 3.1-era GUI application. The number of available views is often limited, and the application provides a facility for exporting a columnar view as a text file.

While it is possible to migrate the dBase data directly to a more modern format, the end result is essentially meaningless since without the various data operations performed in the GUI application, it is unclear what the actual intended use of the data was. Macro scripts such as those described in the previous section can provide a simple method for exporting this kind of data.

Discussion

We describe a software-assisted method to provide users with a simple and reliable method to install and execute legacy software in an emulated environment when browsing legacy archival materials. Our implementation has been extensively tested on the collection of legacy CD-ROM images available at <http://www.cs.indiana.edu/svp/>. The executables and source used to provide emulation assistance are freely available via SourceForge³. We have also made available the Java code used to generate the server-side applet for implementation at other sites. Automation and data extraction scripts for the materials in the SVP archive can be accessed via an AFS client pointed to the publicly available archival space.⁴

Our work directly addresses the problem of retaining the contextual information required to install, execute, and interact with legacy software. By automating both the process of configuring the virtual machine and of installing legacy software, we remove basic impediments to access that current and future users might face when browsing these types of materials. In future work, we plan to modify these tools to

³ <http://www.sourceforge.net/FOO-BAR>

⁴ [//afs/iu.edu/public/sudoc/volumes](http://afs/iu.edu/public/sudoc/volumes)

further assist in extraction of information from specialized legacy applications where traditional migration strategies limited.

Acknowledgements

The researchers would like to acknowledge Mitchell Lutz, who prepared the configuration scripts for the testbed, in addition to final programming and revision of the virtual machine automation routines.

[[[(Witten & Frank, [2005](#))
Santini ([2004b](#))]]]

References

- [journal article]Mellor, Phil. CaMiLEON: Emulation and BBC Doomsday. *RLG DigiNews*. 2003 7 (2).
- [book][[please leave the categories before your refs](#)]Borgman, C.L. (2007). *Scholarship in the digital age: Information, infrastructure, and the Internet*. Cambridge, MA: MIT Press.
- [proceedings]Borgman, C.L., Wallis, J.C., & Enyedy, N. (2006). Building digital libraries for scientific data: An exploratory study of data practices in habitat ecology. *10th European Conference on Digital Libraries*. Alicante, Spain: Berlin: Springer.
- [report]Digital Curation Centre. (2005). *Digital curation and preservation: Defining the research agenda for the next decade*. In Report of the Warwick Workshop, November 7-8, 2005. Digital Curation Centre: Warwick, UK. Retrieved July 27, 2007, from http://www.dcc.ac.uk/events/warwick_2005/Warwick_Workshop_report.pdf
- [proceedings]Esanu, J., Davidson, J., Ross, S., & Anderson, W. (2004). Selection, appraisal, and retention of digital scientific data: Highlights of an ERPANET/CODATA workshop. In *Data Science Journal*, 2004, 3. Retrieved July 30, 2007, from http://www.jstage.jst.go.jp/article/dsj/3/0/227/_pdf
- [journal article]Mayernik, M.S., Wallis, J.C., & Borgman, C.L. (In press). Adding context to content: The CENS Deployment Center. *Journal of the American Society for Information Science & Technology*. Milwaukee, WI: Information Today.
- [journal article]Mayernik, M.S., Wallis, J.C., & Borgman, C.L. (In press). Adding context to content: The CENS Deployment Center. *Journal of the American Society for Information Science & Technology*. Milwaukee, WI: Information Today.
- [report]National Science Foundation. (2003). *Report of the National Science*



Foundation Blue-Ribbon Advisory Panel on cyberinfrastructure. Retrieved September 30, 2006, from
http://www.nsf.gov/publications/pub_summ.jsp?ods_key=cise051203

[Internet journal]Payette, S., Staples, T., & Wayland, R. (2003, April). The Fedora Project: An Open-source digital object repository management system. *D-Lib Magazine* 9,(4). Retrieved September 30, 2006, from
<http://www.dlib.org/dlib/april03/staples/04staples.html>

[proceedings]Santini, M. (2004a). A shallow approach to syntactic feature extraction for genre classification. *Proceedings of the 7th Annual Colloquium of the UK Special Interest Group for Computational Linguistics*.

[report]Santini, M. (2004b). *State-of-the-art on automatic genre identification*. (Technical Report ITRI-04-03). University of Brighton, UK, Information Technology Research Institute (ITRI).

[Internet journal]Smith, M., Barton, M., Bass, M., Branschofsky, M., McClellan, G., Stuve, D., et al. (2003, January). DSpace: An open source dynamic digital repository. *D-Lib Magazine* 9,(1). Retrieved September 30, 2006, from
<http://www.dlib.org/dlib/january03/smith/01smith.html>

[book]Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques*. (2nd ed.). San Francisco: Morgan Kaufmann.