STATSBIBLIOTEKET

DET KONGELIGE BIBLIOTEK

# Handling file formats

May 2004

The State and University Library, Århus, Denmark
The Royal Library, Copenhagen, Denmark

Main author:
Lars R. Clausen
The State and University Library
Universitetsparken
8000 Århus C
Denmark
Email: lc@statsbiblioteket.dk

# Table of contents

# 1. Introduction

This report describes a part of “Internetbevaringsprojektet” (the Internet preservation project), which is a joint project between Statsbiblioteket (The State Library) and Det Kongelige Bibliotek (The Royal Library) in Denmark. The analyses in the report were done in the period from Oct. 1, 2003 to Feb. 1, 2004.

The aim of the project is to archive the Danish part of the Internet. The part described in this report is how to handle file formats in a long-term archival situation.

A strategy for handling file formats is an essential part of long-term preservation of digital objects. Very few digital objects can be read without some kind of interpreter, and it is uncertain which, if any, of the current interpreters will be available and functioning after 50 or 100 years. Thus we will have to consider ways to ensure that digital objects can still be read and understood after such a time span.

An incalculable amount of data has been already lost due to problems in reading the media and understanding the file formats. NASA has lost as much as 20% of the scientific data gathered on Mars by the Viking 1 & 2 missions due to tape decay and obscure file formats[1]. The grandiose BBC Domesday Project[2][3][4] in England was 15 years after its creation on the verge of unreadability, and only through significant effort was it possible to ensure future access.

Ensuring that the media storing the data is uncorrupted and that machinery for reading it is available is only the first step in the process of ensuring long-term availability of digital objects. Even if a perfect bit stream is preserved, the interpretation of said bit stream may still pose problems if the file formats required are no longer in use. We will not consider the problems of preserving the underlying bit stream in this report, but concentrate on the problems of interpreting the bit streams.

It can be argued that unless an object is accessible, it cannot be said to be preserved, as an inaccessible chunk of zeroes and ones is of no use whatsoever. Thus, any talk of preserving digital objects must include ways to access the objects.

This document is structured as follows: Section 2 describes a categorization of formats that is useful for discussing the relevant aspects of preservation and conversion. Section 3 introduces various aspects of files that may be preserved through various techniques, while section 4 considers some criteria for assessing the medium-term usability of file formats. Section 5 discusses a specific problem of preservation, namely digital rights management systems. In section 6, we discuss the five main ways to preserve access to information stored as digital objects:

1. Capture, either on analog media or on very simple file formats
2. Sequential conversion to new formats
3. Conversion-on-demand to new formats.
4. Emulation of the system currently used to read the formats.
5. Preservation of current hardware and software used to read the formats.
6. Filming of current usage of the files.

Section 7 describes our suggested strategy for handling file formats, and section 8 concludes.

# 2. Categorization of formats

The number of file formats defined by various companies, organizations and sometimes individuals is staggering. The File Extensions collection[5] claims to index over 15,000 file name extensions. The more detailed indexes[6][7][8] contain about a thousand descriptions of file formats. To simplify further discussion, we will introduce a categorization of formats in order to better discuss aspects of related file formats.

## 2.1. Mime types

The official categorization of file formats is the MIME type, handled by IANA[9]. They define 548 separate MIME types in the following major categories (the number of registered types shown in parentheses, the second number being the vendor-specific formats).

- application (90/260)
- audio (46/17)
- image (12/20)
- message (10/0)
- model (3/9)
- multipart (13/0)
- text (18/18)
- video (21/11)

There is some limited overlap in these categories, for instance the RTF format exists in both "application" and "text" categories. There is some more overlap within categories as some formats are listed in vendor-specific versions as well as generic versions. This overlap pales in comparison with the mime-types returned from the servers. In the "Uge46" harvest of selected Danish websites done in week 46 of 2003 (see section 2.3), 97 different mime types were encountered (ignoring case, but keeping different spellings). Out of these, 42 were x- forms[1], and only 14 of them were types defined by IANA. These 14 types accounted for 94,41% of the total files, while 3,91% of files had no specified type. Encountered formats for which no MIME type is registered include Javascript and Shockwave/Flash.

Apart from the problems of randomness of mime-types, the categorization is not very useful for our purposes. For instance, placing the RTF format under "applications" omits the crucial point that RTF documents have no user interaction and generally have a reasonable printed representation.

## 2.2. A useful categorization of formats

For the purpose of discussion in this document, we will use the following categorization that avoids the catchall 'application' category and that allows us to discuss what preservation means for various kinds of files.

- Document-like (PDF, DOC, PS, DVI, HTML…)
- Image formats (GIF, PNG, JPG, …)
- Sound formats (MP3, OGG, …)
- Movie formats (MPG, AVI, …)
- Data formats (more or less raw data from experiments)
- Structured graphics formats (CAD, VSD, QXD, …)

[1] About which RFC2046 says "Any format without a rigorous and public definition must be named with an "X-" prefix, and publicly specified values shall never begin with "X-".".

- Spreadsheets (XSL, …)
- Databases (DBF, DDF, …)
- Collections (tar, zip, …)
- Configurations & metadata (CSS, …)
- Program-supporting formats (TTF, game saves, …)
- Program file formats (Javascript, Java, SWF …)

This classification gives us an idea of the various types of expression stored in files. Note that the specific file formats are intended as examples only, and may even change over time. For instance PDF seems to be moving from a purely document-like format towards a spreadsheet-like format with its facility for inline form entry.

The most problematic group is the last one, programs. This group contains some of the most fragile objects, as they are typically highly complex binary objects. At the same time, these objects are the most complex and application-specific, frequently depending on undocumented features or even on bugs. The program-supporting files are usually not documents in themselves, and may be closely tied to particular programs. They can be necessary for viewing other files or for running emulators.

## 2.3. File formats found in Uge46

In week 46, 2003, the Danish Royal Library crawled a number of selected Danish sites of cultural interest. Of the downloaded documents, 688029 documents specified a MIME-type. The following types cover 99.9% of the documents:

| Percent | Mime-type | Category |
|---:|---|---|
| 66,78% | text/html | Document-like |
| 19,17% | image/gif | Image |
| 10,12% | image/jpeg | Image |
| 1,11% | text/css | Configuration & Metadata |
| 0,87% | application/x-javascript | Program |
| 0,68% | text/plain | Document-like |
| 0,29% | application/x-shockwave-flash | Program |
| 0,24% | image/png | Image |
| 0,20% | audio/x-pn-realaudio | Sound |
| 0,11% | application/octet-stream | Unknown |
| 0,07% | image/pjpeg | Image |
| 0,07% | image/x-icon | Image |
| 0,04% | audio/x-ms-wma | Sound |
| 0,03% | video/x-ms-asf | Video |
| 0,03% | image/bmp | Image |
| 0,02% | text/xml | Configuration & Metadata or Data |
| 0,02% | application/msword | Document-like |
| 0,01% | audio/midi | Sound |
| 0,01% | video/x-ms-wmv | Video |
| 0,01% | video/x-ms-wvx | Video |

This includes 3 document-like formats, 6 image formats, 3 sound formats, 3 video formats, 2 Configuration & Metadata format (of which text/xml might be Data or something else), 2 program formats, and one unknown format.

## 2.4. Categorizations in file format repositories

There are several on-line repositories of file formats. Three of the most comprehensive are Wotsit's Format[6], My File Formats[7], and File Format Encyclopedia[8]. These all document about a thousand file formats with varying accuracy. They are maintained by individuals, and so cannot be considered reliable in the long term. The British National Archive's file format and software repository, operative since March 2002, opened its web access in January 2004[10]. It contains 550 file format descriptions as of February 1st, 2004, but does not allow direct access to any specifications they may have stored. Further information on format repositories can be found in [11].

Some of repositories use their own categorizations of file formats. These categories are for ease of navigation for users rather than a categorization intended to clarify aspects of preservation, but can nonetheless serve as an example of the varied kinds of formats in use. Below is a listing of the categories found in the three repositories mentioned above compared with the categorization shown in section 2.2.

| Category | Wotsit | MyFileFormats | File Format Encyclopaedia |
|---|---|---|---|
| Image/ structured graphics | 3D Graphics Files | 3D Graphics files | 3D Formats |
| | Graphics Files | Graphics files | Graphics |
| Movie | Movies/Animations | Animations and Movies | Animation |
| Collections | Archive Files | Archive files | Archive |
| Program files | Binaries | Binary files | Binary |
| | Comms Formats | Comm files | Communication |
| Spreadsheets | Spreadsheet/Database | Spreadsheet files | - |
| Databases | | Database files | Database |
| Document-like | Text Files/Documents | Documents and Text files | Text |
| Program files | - | - | Emulator |
| Data formats | Financials/Stocks | - | - |
| Program supporting | Font Files | Font files | Font |
| Program supporting | Game Files | Game files | Games |
| Data formats | GIS[2] Formats | GIS files | - |
| - | Hardware Formats | Hardware formats | - |
| - | Internet Related | Internet files | - |
| - | Miscellaneous | Miscellaneous | - |
| Music | Sound and Music | Music and Sound files | Sound |
| - | Windows Files | Windows files | Windows |
| Document-like | Printer Formats | - | - |
| Music | - | - | Modules |

Several of these groups (Comms formats, Hardware formats, Internet related) are not file formats at all, but rather descriptions of hardware or protocols. The Windows files groups are a selection of formats from other groups with a relevance to Windows

## 2.5. Conclusions and recommendations

The bewildering amount of file formats in present (and past) use complicates discussion of file format handling. In this section, we have described a categorization of formats that we will use in the remainder of the document, and compared it to categorizations used by file format repositories.

[2] Geographical Information Systems

As part of long-term preservation planning for file formats, it is essential that information about the formats be preserved. Therefore, we suggest that international cooperation on file format repositories be supported, preferably through participation in development.

# 3. Aspects of preservation quality

Before we can consider which preservation strategy (or strategies) is the most appropriate, we must consider what aspects of digital documents are the most important to preserve. It may be pointless to preserve a static image of an object whose main purpose is to provide certain functionality, but visual design researchers may consider it better than nothing. Obviously, one would want the most exact preservation possible, but the resources for that may not be available, or unforeseeable developments in computer technology may render a perfect copy perfectly unreadable. Given that even today, many viewers have their own quirks, errors and omissions, is may be difficult to determine what should be considered a perfect copy even now. There are several aspects of a digital object that can be the goal of preservation, some requiring more resources than others. When deciding on a strategy for preservation, the desired aspects and the associated risks and costs are cornerstones of the consideration.

## 3.1. Aspects

We use the following five aspects as a basis for discussing what to preserve. Note that the aspects are not necessarily in increasing order of complexity or quality, in particular the functionality aspect may have little correlation with the other aspects.

**Readability:** A minimum requirement must be that the core elements can be read. For documents with text, a simple text extractor can do this. For images and sound, some kind of viewing/playing is a minimum, though a significant amount of loss is acceptable at this level. Movies may be represented with some number of still shots. This is generally the least costly aspect to preserve, but has the greatest amount of loss. This is useful for those merely interested in the content. Example uses could be: Proving the existence of a text on a website at a certain time, tracing the usage of new words on websites, and examining fashion trends as seen in online pictures. However, users should be aware of the risk of missing or distorted information posed by only having this aspect.

**Comprehensibility:** Most text documents have more to them than just the raw text. Data may be lined up in columns, arrows may point at important features, text attributes may indicate particularly important words, etc. These elements can be as important as the text itself, and losing these could render a document meaningless or even misleading. For images and sound, some errors may be introduced, but not enough to interfere with easy comprehension. Movies would be low-resolution or with significant artifacts, but still viewable as a movie. Having this aspect preserved would be enough for most research and legal uses, and would allow research based on files with significant formatting.

**Appearance:** Some attributes of a file format are not necessary to understand the meaning of a file, but are part of the overall impression. Correct kerning and anti-aliasing of text, exact rendition of colors in images, and noise free audio played in correct stereo, for example, all give a better impression, but would only rarely be the dividing line between a comprehensible object and a chunk of useless data. This quality preservation is rarely essential unless researching the quality of documents created at a certain time. However, a good rendition of the file adds to the confidence in the data, and gives a better overall impression of what the original document looked like. This aspect would be required for researching the form that material has been presented in, or for art history research, and would be preferable, though not required, for general viewing.

**Functionality:** Unlike analog objects, digital objects often have functionality beyond that of looks and sounds. Spreadsheets contain formulas that are not shown in printouts. Many formats now include hyperlinks, even when the format is designed for a paper representation. PDF includes functionality to fill out forms, including ways to check legality of the input. CAD files may include constraints that are not visible in a printed version. Some objects make little sense without their functionality, for instance chat clients and games. Including the functionality of an object can be an essential requirement for preservation of some files. This aspect would be beneficial for researching how people use the internet, or for retrieving lost files with important functionality.

**"Look & Feel":** A perfect copy of a digital object would preserve not only the appearance and functionality of the original, but the entire "look & feel," i.e., the design and operational quirks of GUI elements, the resolution of the monitor, even the speed of the machine. While this may be overkill for most preservation, arcade game enthusiasts have gone to great lengths to achieve this. For formats that are not closely tied to a particular program, it may be difficult to decide what the look & feel is, as different current viewers may provide very different results.

The partitioning given above is not a strict level partitioning. A viewer for Word documents may recognize links and give them the required functionality, yet not be able to render arrows. The visual parts of a movie may be perfectly rendered even if sound is missing. In particular, significant functionality part might be converted even in a barely comprehensible conversion (e.g. link extraction), or a picture-perfect conversion may discard all functionality (e.g. printing).

## 3.2. Considerations on aspects

We do not know what aspects will be considered important in the future. When the Danish newspaper archives were started, most expected the news articles to be the significant part, but current researchers are at least as interested in obituaries and advertisements. Similarly, a future researcher may be interested in current layout techniques, current interaction models or other features that we don't even think of.

We face a trade-off between how much we can preserve and the resources we can spend on preserving it. It would make little sense to allocate many resources to correct preservation of a file format that appears only a few times in a billion-object archive. The overview in section 2.3 of file formats found on harvested objects shows that 96% of the files found are HTML, JPEG or GIF. Clearly, preserving the more widespread formats must take higher priority, but at the same time, the most widespread formats are the ones most likely to have viewers available in the future. The most problematic formats may well be ones that are widespread enough that losing them would lose significant amounts of data, but not widespread enough that we can feel sure that somebody will always be there to create a viewer.

## 3.3. Example aspects for categories

We here give some examples of what the aspects can mean for the various formats defined in section 2.2. These are examples only, and not intended to provide an exhaustive list of what is required to preserve a particular aspect of a category.

| Category | Readability | Comprehensibility | Presentation | Functionality | Look & Feel |
|---|---|---|---|---|---|
| Document-like | Text | Text with some markup | All markup and graphics | Links work | |
| Image | Low resolution | Medium resolution | High resolution | - | |
| Sound | Lowest bit rate/sample space | Medium bit rate/sample space | High quality | - | Includes player |
| Movie | Some images | Low resolution, some artifacts | High resolution, few artifacts | DVD menus work | Includes viewer |
| Data | Text extract | Text in columns | - | - | - |
| Structured graphics | Text extract | Image capture | Vector format | Connections, checks etc | Same interface |

| Category | Readability | Comprehensibility | Presentation | Functionality | Look & Feel |
|---|---|---|---|---|---|
| Spreadsheets | Text extract | Text in correct positions | Correct text and graphics | Formulas | Same interface |
| Collections | Index | Separate files | Archive | - | - |
| Configurations | Text extract | Structured text | - | - | - |
| Programs | Screenshots or film | Semi-functional emulation | - | Full emulation | Program runs |

When selecting formats for conversion or emulation, we should make note of what important aspects can be identified in the format, and how different converters or emulators may preserve these aspects. That will not only help archivists focus their efforts on the desired aspects, but also give users some idea of what aspects they can expect to find preserved.

# 4. Assessing the future usability of file formats

Before choosing what to do in terms of converting, we will need a way to assess whether a format is likely to be viewable a long time into the future, or if it is in danger of becoming obsolete already.

When negotiating the format of deposited (as opposed to harvested) material, we should ask for the material in a form that is likely to be useable for a long time, to minimize the risk of loss of data. Additionally, if we decide to convert some files to new formats, the conversion would be a waste of time if the new format does not live longer than the original. Thus, we should maintain a set of “preferred formats” that we expect will remain usable for a significant amount of time.

The set of preferred formats should be small, to preserve resources, but should cover the various file categories that we want to archive as well as different aspects that we may consider important. For instance, we may chose to have both PDF and HTML as preferred formats for document-like files, since PDF can handle complex documents and some functionality, but HTML is more likely to be comprehensible in the long term.

In this section, we describe a number of criteria for assessing the long-term viability of a format. These criteria are not all-or-nothing criteria that must all be fulfilled for a format to be used for archival purposes. Nor are they a simple checklist where fulfilling more means that a format is better than another. Some criteria are trade-offs against each other, and some are predictions on future events that are necessarily subjective. These criteria are issues that must be considered when selecting formats, but in the end, the decision is a subjective and predictive decision, which should be based on, but not restricted by, the criteria below.

These criteria should be adjusted periodically to ensure that they reflect the relevant issues in assessing the future accessibility of formats.

## 4.1. Openness criteria

Formats that are described by publicly available specifications or open-source source code can, with some effort, be reconstructed at a later time, whereas proprietary formats risk becoming unreadable if the company owning them goes out of business or decides to stop supporting the formats. The following criteria list various ways in which a file format can be considered open.

1. *Open, publicly available specification.* This allows a later creation of viewers even if viewers and systems are unavailable.
2. *Specification in Public Domain.* A specification not encumbered by patents or copyright issues is more likely to have free viewers made for it in the medium term. In the long term, patents and copyrights will eventually expire.
3. *Viewer with freely available source.* This also allows creation of viewers, even if the source cannot compile anymore. A working viewer may in some cases be more useful than a specification, as specifications are not always obeyed in practice.
4. *Viewer with GPL’ed source.* A viewer that is under the GPL license[12] cannot be extended and closed off by companies, but will always be freely available, if available at all. This is an extra insurance that the viewer source does not stop being freely available.
5. *Not encrypted.* A format that requires a special encryption key to read is doubly at risk of becoming obsolete, as the key may be lost as well.

## 4.2. Portability criteria

A format that makes extensive use of specific hardware or operating system features is likely to be unusable when that hardware or operating system falls into disuse. A format that is defined in an independent way will be much easier to use in the future.

1. *Independent of hardware.* Hardware dependency is particularly dangerous in a format, as hardware changes particularly fast.
2. *Independent of operating system.* Operating systems tend to have a longer lifetime than hardware, but cannot be expected to last for centuries. Operating system dependencies also restrict what systems can be used to view the files.
3. *Independent of other software.* Any extra software, like compression or encryption libraries, is another part that could be lost, and stand-alone formats should be preferred. Each extra software requirement should be evaluated separately, and the format should be considered only as stable as the least stable piece of software involved.
4. *Independent of particular institutions, groups or events.* A format made to suit a particular organization might contain peculiarities for that organization that lessens its future usability.
5. *Widespread current use.* Widespread use of a format indicates that others consider it useful and important. It also means that more work has gone into creating viewers and tools, and that there is a greater overall pressure to create viewers in the future.
6. *Little built-in functionality.* The more functionality a format contains, the harder it is to create a correct viewer or to later convert to other formats. Some kinds of functionality, like embedded programs, are at a high danger of becoming unusable. A format that allows the same expression in a simpler, more explicit form, is to be preferred.
7. *Single version or well-defined versions.* A format that comes in many versions is harder to understand, particularly if the difference in versions is not immediately obvious.

## 4.3. Quality criteria

The quality of the format is an issue that can be reasonably estimated at the current time, as it pertains to how well the format fulfills its task today. However, a number of the criteria here work against each other, so a number of trade-offs will be encountered.

1. *Low space cost.* If significant amounts of data are expected to be archived in a particular format, the sheer cost of a space-consuming format may prove prohibitive. Smaller files are also easier for tools to handle.
2. *Highly encompassing.* A format that can be used as a target for a greater number of other formats saves resources otherwise necessary to maintain other formats.
3. *Robust.* A format that is unreadable if a single bit is flipped or a header is misunderstood is more likely to be unreadable in the future. A format should preferably be robust both against random bit errors and loss of parts of the file. Note that compressed formats are particularly vulnerable to bit errors.
4. *Simplicity.* The simpler the format, the more likely that new viewers can be created in the future that will handle the format correctly.
5. *Highly tested.* A format that has been put to a number of different uses and/or has been used for a significant amount of time has been put to many independent tests of its quality. Thus, widespread and long-term use gives us more assurance that the format is of high quality.
6. *Loss-free.* If converting a loss-free format into a lossy format, some information will obviously be lost. Preferably, lossy formats should only be used for conversion from other lossy formats, and only if the conversion does not incur significantly more loss.
7. *Supports metadata.* While we plan to archive metadata outside the files themselves, metadata support may allow us to gain metadata about the source of the files that would not otherwise be available, and also provides a redundancy of metadata in case the externally stored metadata are lost.

## 4.4. Monitoring obsolescence

Information gathered through regular web harvesting can give us some information about what file types are approaching obsolescence, at least for the more frequently used types. When the number of files of a certain file formats starts dropping, it is a sign that the format is not in active use anymore and will soon be obsolete. At that point, an effort must be made to ensure the further availability of files in that format.  We have no current information on how this has developed in the past, but it could possibly be obtained from The Internet Archive[13].

# 5. Specific preservation issues

In this section, we discuss some specific issues that complicate file preservation. These issues must be dealt with either through technological solutions or by use of specific legislation.

## 5.1. Digital Rights Management technologies

Digital Rights Management (DRM) technologies are systems that control the ways in which users may access digital material. Through the use of encryption and special viewers, the content producers can specify various limitations on usage of material, such as a limited number of viewings, copy-protection, inability to print or mail the material, or a limited duration of usability. DRM technologies have been tried since the advent of the home computer, but only a few systems have seen widespread acceptance. Some well-known examples include Microsoft's product activation system[14] and Apple's iTunes music store DRM[15].

From an archival point of view, DRM technology is very bad news. In order to view a DRM-protected file, the computer will typically have to contact a specific server that validates the license and checks that it is only used on authorized computers. Such a check is necessary to avoid the file simply being copied bit-for-bit. However, it is virtually certain that the authorizing server is either gone or changed beyond use within a century. Even if it isn't, the (often proprietary) software that handles the authorization on the client side would only run in an emulator, with concomitant restrictions and risks of loss.

Even if the authorizing server should be available, restrictions on usage make the file much less useful. A time limit on usage essentially makes the file unarchiveable, while limitations on the number of times it can be read makes it of minimal use in future research. Other limitations, such as not allowing printing or cut-and-paste, also hamper effective use. A file protected by DRM technology cannot be said to be usefully archived in that form.

The Danish legal deposit laws state that content providers are required to provide us with content suitable for archiving. In particular, we may request an unprotected version of copy-protected published material.

We need to monitor the archive for files that have DRM protection on them. When we find DRM-protected files, we need to contact whoever placed the files on the website we archived. They will then have to provide us with the appropriate unprotected files that we can add to the archive. This process needs to be as automatic as it can be without endangering the integrity of the archive. A system where the copyright holders can submit the unprotected files would be useful.

A further risk of any increase in the use of DRM protection is that the DRM-capable file formats may be proprietary, and so harder to convert even when devoid of DRM-protection. It is currently unclear how significant this problem will become. We may need to request the unprotected version in a different format altogether.

# 6. Preservation strategies

In the international preservation community, the use of conversion to preserve digital objects has been questioned in recent years[11], and some alternatives have been appearing[16][17]. In this section, we describe the various approaches for handling the problem of format obsolescence.

## 6.1. Capturing with “flat” formats

The simplest way to ensure that digital objects remain readable in the long term is to make them non-digital. Printing to paper or microfilm preserves the look of the object for as long as the medium itself survives, and there is a large body of research on preservation of such media from several hundred years of archiving in libraries. The two main problems with this approach are indexability and the loss of functionality.

### 6.1.1. Analog capture

Indexability is the ability to find and retrieve documents in a timely fashion. While analog formats can be indexed at their creation time, retrieval would be dependant on manual intervention, leading to access times that can be several hours or even days.

Functionality is all the things a digital object can do that an analog version obviously cannot: Contain hyperlinks, register user input, show pop-up notes, calculate formulas etc. This aspect can be the target of research itself, or it can be the aspect most desired when recovering an old file. This aspect would obviously not be possible to capture in flat formats.

Additional concerns are the sheer size of such a conversion, the time required to handle the items, and the fact that some analog formats (especially for sound and video) suffer from obsolescence problems just like digital formats.

### 6.1.2. Digital capture

A related approach is to convert digital objects to a much simpler form, where the form itself contains enough description for an interpreter to be recreated. Statens Arkiver (The State Arkives) currently require that all documents be delivered in TIFF and ASCII formats[18], and keep any relations between the objects in a separate index. While this solution solves some of the indexability problems and the size concerns, it has the same problems as paper when it comes to functionality, and additionally has problems of existing on media that may itself become obsolete. Reduction to TIFF is an appropriate approach for Statens Arkiver, where the digital objects are generally devoid of functionality, but merely a digital version of a paper document. However, when it comes to archiving documents from the Internet, TIFF is only slightly more usable than a paper printout.

## 6.2. Early conversion to other formats

The obvious approach to the problem of file format obsolescence is to convert files into standard formats. While this approach has been under some discussion (see sections 6.3 and 6.4), it has the advantages that action can be taken early to preserve data, and that it requires no long-term maintenance of special programs nor significant changes to the presentation system. However, it is dependant on having some way to convert the file, a task that may be impeded by the complexity of the file or lack of information about the file format.

The decision to convert, and in particular what to convert to, is strongly influenced by what aspects of preservation are required. Some users may find a conversion to raw text sufficient, while others would demand the most exact conversion possible. Since several of the parameters in the conversion decision are predictions of future developments, it may be prudent in some cases to perform several conversions of a given file. A conversion to a simple format expected to last a long time may provide the readability aspect, while full appearance or look & feel preservation may be done to a less reliable format. By using a multi-pronged approach to conversion, we can avoid an all-or-nothing scenario of preservation.

While high-quality conversions give the best impression when viewing the individual files, a conversion to a simpler format may allow automated analysis and search. In particular with regards to files containing text, extracting the text into ASCII may allow free-text search in the archive with very simple tools and with a very low risk of losing the information. Other automated analyses like word frequency counts may also be performed on the simpler formats without requiring knowledge of a host of different formats.

Regardless of the conversions applied, it is essential that the original file be preserved without any transformation at all. While it may become unreadable at some point in time, a concerted effort by interested parties might later yield a better conversion program. Retaining the original also opens the possibility of applying other strategies for preservation, as will be shown in the coming sections. The suggestion for handling conversion in the ARC file format, as described in VTL, retains the original and adds the converted version as a separate but related object.

### 6.2.1. Handling conversion loss

Except for a few situations like lossless bitmap formats, conversion is likely to entail some loss of information. Different formats contain a different selection of possible information. Some formats allow rectangles with rounded corners, others do not. Colors can be represented in different color spaces. Different bit-sampling rates may be allowed, or different compression algorithms may be available. Even within a format, lossy formats by their nature lose information every time they are resaved, even within the same format. Different programs may also have different implementations of a format, intentionally or unintentionally. For complex formats, many programs implement only the subset interesting for that program.

Sequential conversion accumulates errors. For each conversion an object passes through, some errors particular to the conversion program will be added. Accumulated errors can be reduced by keeping every converted version around and performing new conversions from the version that gives the best result. It may be difficult to determine what the “best result” is, though, particularly if the original viewer is no longer available. See [11] for examples of how unchecked sequential conversion can cause serious errors over time.

## 6.3. Conversion on demand

The Camileon Project[19] suggests an alternative solution to sequential conversion, namely conversion on demand (also known as “migration on request”[17]). In this strategy, a set of conversion utilities is maintained indefinitely, and conversion from the original data into the best format available is done at access time. This approach avoids the problem of accumulated errors, but adds the burden of maintaining a suite of conversion tools indefinitely into the future as well as having to integrate the conversion routine into the presentation system.

This approach has some of the same fundamental problems as sequential conversion, that some formats may be difficult to convert correctly, or no meaningful conversion may be possible at all.

This can be either because of lack of information about the file format, or because of lack of a target format that can handle the complexity of the source format. However, given converters of a reasonable quality, this approach will yield a higher quality result. It also offers better authenticity, as the converted object has only been processed once, and is necessarily still present in its original form. This method can also save machine resources, as only the materials requested for viewing will be converted, and the results of the conversion may be cached. On the other hand, it may cause delays when viewed for the first time, as a potentially time-consuming conversion process must run while the user is waiting.

While a functioning conversion-on-demand system would be the optimal conversion solution, it depends on the indefinite maintenance of a set of conversion tools. The implementation done by the Camileon Project has longevity as an important requirement that influenced a number of decisions, and can, if properly maintained, be expected to be usable for a long time period. However, no progress seems to have happened on the project for over a year, so it is doubtful whether it can be considered to be a live project.

A conversion-on-demand system would have to be based on tools that we can keep and modify in the future. Commercial tools may do the job at the present, but we cannot assume that the company will continue producing the required tools or even exist in the long term. Thus, we would have to possibly create and probably maintain these tools ourselves. If the tools at some point cease functioning, we will be no better off than had we done nothing.

Building and maintaining converters for a sufficient number of formats is a daunting task, outside the scope of the Danish libraries alone. Only insofar as international cooperation in the field succeeds in creating a comprehensive project with active maintenance should we consider conversion on demand as a primary preservation strategy. However, by ensuring the preservation of the metadata necessary for a future conversion utility, we will be able to adopt this strategy at a later stage with only a slight use of present resources.

## 6.4. Emulation of current viewers

Emulation as an approach to object preservation was seriously attempted for the first time in connection with the Domesday Project[2]. Instead of attempting to convert the complex existing system, which included a number of compiled programs, they created an emulation system that allows the original bit-streams to be executed as on the original computer. This shows that emulation can be a viable approach to preserving digital objects. Jeff Rothenberg of the Rand Corporation suggested in 1999 that emulation is the only viable strategy for long-term archival[16].

Emulation has one obvious advantage over conversion: emulation is possible without any knowledge about the internals of the file format. For secret formats, emulation or printing may be the only viable alternatives. For highly complex formats, emulation may be the only way to get reasonable preservation.

Emulation is likely to give the exact look & feel aspect of the original viewer. This is beneficial for those researching media and presentation issues, but may be a hindrance for those who just want the readability or comprehensibility of the original document. Some care must be taken to pick the right viewer for a format; the most widespread viewer may not be the best, as can be seen by Internet Explorers sub-par handling of PNG files. Integrating the emulator with other viewing tools like browsers may also prove difficult, and can cause a lessened usability of the emulation approach. In the Domesday project, the entire system is emulated, giving a flawless interface; it is unlikely that such can be achieved for Internet pages archived over many years.

### 6.4.1. Current emulators

There are already now several working emulators of complete systems on very different hardware. MAME[20], an emulator for arcade machines, emulators for Spectrum, Commodore 64, Atari and other early machines, and the emulation of 68000 code on the PowerPC Macs[21]. These emulators are all made by interested hobbyists, except the MacOS emulator, which was designed as an upgrade solution to get away from an outdated CPU platform.

The most interesting project in this connection is MAME, as it had perhaps the most adverse conditions to work under: The hardware was totally proprietary, with no description available to the general public, few machines were available, and the available machines were not easy to examine. Despite these odds, MAME now emulates over 2000 different games, from Space Invaders to Mortal Kombat 3. The MAME developers are very particular about doing a correct emulation, including getting the correct look of flickering, blurry colors of the original machines.

Some current emulation systems like Crossover[22] perform a form of software emulation. They are able to run programs designed for different operating systems by emulating the interface the programs connect to. This kind of emulation assumes that the software runs on the same underlying hardware, e.g., x86 compatible machines. For a long-term solution, we cannot assume that compatibility with current hardware will be present, not to mention correct.

Raymond A. Loria at IBM Almaden Research Center proposes creating a Universal Virtual Computer that can be used to create both a generic emulator and archiveable viewer programs for non-program files[23]. However, no information about the actual creation of such a system can be found at the present.

### 6.4.2. Emulation as a general preservation approach

Unlike the conversion approaches, emulation requires a separate program to show the files, a program that may include starting a full emulation of an operating system. It may be difficult to properly embed such a program in a viewer. If for instance the GIF format is only viewable through emulation, Web pages that currently show GIF files inline may instead have to show them in separate windows. Integrating the emulators with the current viewing system would require an extra effort.

Like the conversion on demand strategy, emulation requires that certain programs be maintained to permit viewing. Whereas conversion on demand requires at least a module for each file format, emulation requires one emulator for each underlying system. Additionally, appropriate viewers and system software must be stored and made available at viewing time. Maintaining these emulators is no easy task, and is beyond the scope of what the Danish libraries can be expected to support.

Another problem with emulation is that it is an all-or-nothing approach to a greater degree than conversion. Conversion is a step-by-step process, in which some elements may be lost while others are preserved. Emulation, on the other hand, creates the system on which files can be viewed, and an error in the emulator is more likely to make viewing impossible than to leave parts of the file viewable. Emulation leaves the file as a black box, for which we only have one, heavyweight way of examining it. We cannot perform text search in the files, unless we happen to have stored a text searching program that we can emulate as well. Performing text search across a multitude of emulated formats would be highly impractical, if possible at all.

### 6.4.3. Emulation as a supplement to other preservation strategies

Certain formats may be near impossible to convert to other formats, or not enough information may be publicly available to create new viewers. For such formats, emulation would be the only viable way to ensure availability in the long term. In particular programs are very difficult to convert in a meaningful way, but other formats of a highly proprietary and complex nature, such as 3D models, may also require emulation.

For such formats where we cannot obtain or create converters of a reasonable quality, emulation provides a way to increase the chance that the files are viewable in the future. Since locating and storing viewers and surrounding system components of appropriate versions is a resource-demanding task, this should only be done for formats that is found in significant numbers in the collected material.

The archive should be monitored for formats that we may want to retain viewers for. An example would be the Flash file format[24], which is 1) proprietary but disclosed, 2) binary and highly complex, 3) of highly complex functionality, 4) only converters to video or still images are available[25], and 5) Flash is the 7th most common format found in the download described in section

2.3. Thus, storing a Flash player along with necessary systems components would be helpful if support for Flash formats disappears in the future.

The main criteria for choosing to obtain and store the data necessary for emulation are:

1. Is the format proprietary and/or undisclosed?
2. Is the format binary and/or highly complex?
3. Is the functionality of the format highly complex (e.g. program files)?
4. Are converters currently unavailable?
5. Is a sufficient number of files of this format found in the archive to warrant the resources involved?

In order to run viewers, we must in some way emulate the system that they depend on. Since the aforementioned software emulation cannot be assumed to function in the long term, we must store enough of the system to run hardware emulation. This would entail either a full installation of an operating system capable of running the viewer, or appropriate software to create such an installation.

## 6.5. Preservation of current hardware

The most basic way of doing emulation is by keeping an appropriate selection of machines. While this has the advantage of giving the most exact replication of the original experience, it has the serious drawback that computers are physical machines, and as such subject to wear and tear. The most vulnerable parts are the mechanical parts like drives and fans, but chips can also start failing after a relatively short time. An additional problem is the storage space required to store this number of machines, and the problems in selecting which are important to keep. For these reasons, hardware retention cannot be considered a valid long-term solution.

An additional and serious failure of the hardware retention strategy is that the objects no longer can be considered entirely digital. The hardware required to view a file must be considered a part of the archive in some sense, and restricts the options available for backup, refreshing and viewing. For example, pictures that were originally embedded in a web page might under this strategy require that the user move to a different machine in order to view the picture. This is not desirable as a general preservation strategy, but may be used as a last resort. Some institutions, like Teknisk Museum in Copenhagen and British Museum have collections of old hardware that could be used if necessary.

## 6.6. Filming

For highly complex formats where user interaction plays an important part, we could provide a record of the usage and looks of the files by filming a user using the file. This filming could either be in the format of continuous screen capture, or by physical filming, possibly including comments. This may be particularly interesting for systems combining a number of different components that may otherwise be difficult to convert or emulate, or where the actual use is more important than the program itself (such as chat rooms).

Filming is one of the most resource-intensive forms of archiving. It requires creating of a setup that allows filming, and one or more people to perform the filming itself (at least one user and zero or more archivists). The cost may be reduced by using automatic screen capture of users in their normal environment, though significant editing must still be done. Additionally, the size of the film may be significantly larger than the program itself. Filming would mostly be of interest to researchers interested in computer use, interface design or the like. It would be of limited use as a source for textual material.

## 6.7. Summary of risks and benefits

The table below summarizes the risks and benefits of the various strategies for handling file formats.

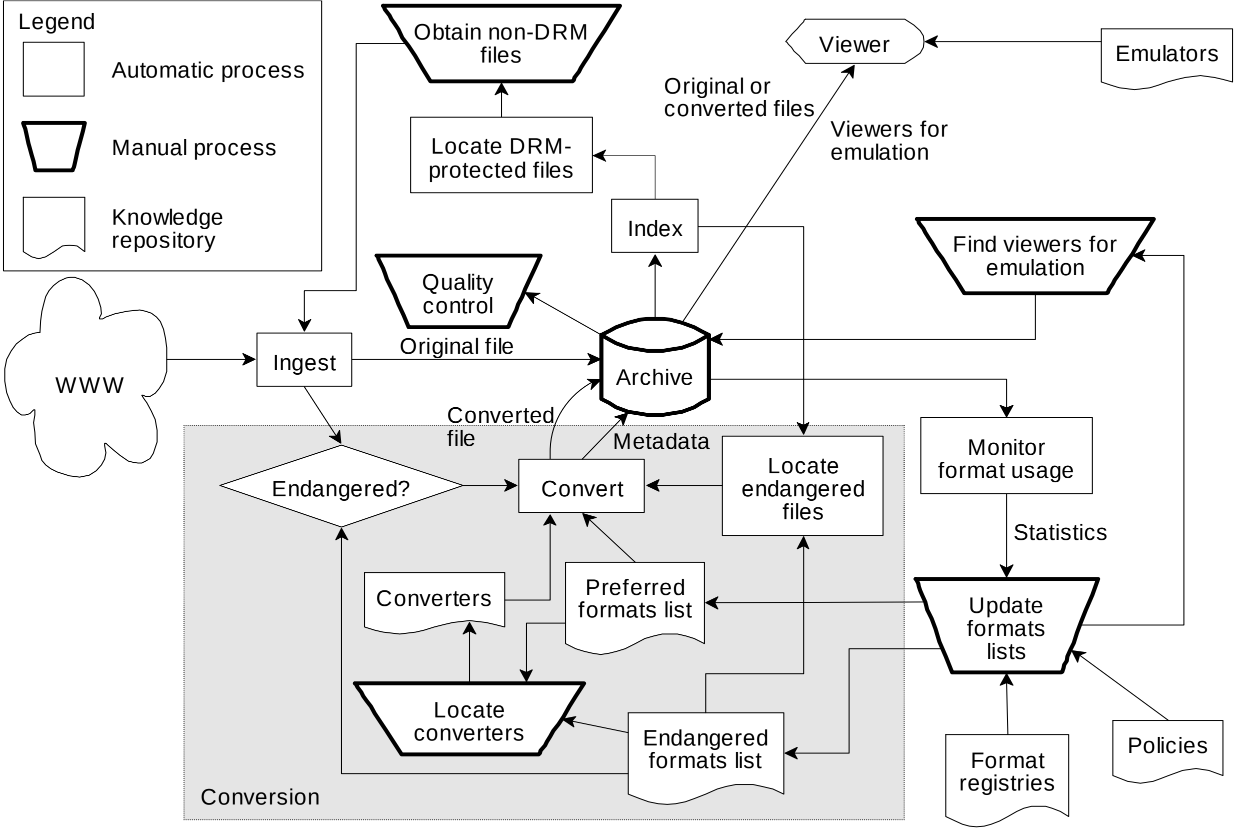
| Strategy | Risks | Benefits | Resource usage |
|---|---|---|---|
| Flat | Loss of functionality, impractical to use | Format independence | High: Manual handling and indexing |
| Early conversion | Cumulative conversion errors, some loss of functionality | Independent of software, multi-aspect conversion possible, viewing is easy | Medium: Monitor format evolution, obtain and use converters when needed |
| Conversion on demand | Conversion errors, some loss of functionality, software dependency,<br>viewing delay | Viewing is easy, fewer errors than early conversion | High: Create and maintain converter suite |
| Emulation | All-or-nothing preservation, may be impractical to use, software dependency | Near-perfect look & feel, no conversion errors | High: Create and maintain emulators |
| Hardware retention | Mechanical failures, very impractical to use | Perfect look and feel | Very high: Retain physical machines |
| Filming | Loss of functionality, impractical to use | Shows functionality and actual use, format independence | Very high: Run recording sessions |

# 7. Suggested strategy

In this section, we describe our suggestion for the practical handling of file formats in an archival system. Important parts of this include tracking developments in file format usage, administering conversion process, and preparing for emulation when needed.

## 7.1. File preservation workflow

In the diagram below, we show the suggested workflow for converting files that can be converted and storing information required to run emulators for inconvertible files.



The diagram identifies five manual tasks that must be performed with some regularity:

1. *Update format lists*. Both the list of preferred formats and the list of endangered formats must be kept up-to-date to avoid loss of data. Input for this comes partly from statistics gathered from the archive itself, and from international format registries.
2. *Locate converters*. When a decision has been made to convert a format to another, we must obtain an appropriate converter and put it to use. Converters may be either commercial tools, open source tools, or if necessary created in-house. International cooperation on this area is important, as creating a converter is a resource-intensive task.
3. *Find viewers for emulation*. For formats that we consider endangered and cannot currently convert, we must store appropriate viewers and their required system components to allow emulation at a later date. We do not plan to build the emulators ourselves, but to join international development efforts when necessary.
4. *Perform quality control*. The various automated processes must regularly be checked to see if they perform their tasks well enough. In particular, the conversion process and the chosen converters must be check and kept up-to-date, as that part has the greatest risk of loss.

5. *Obtain non-DRM files*. For all files with DRM-protection found in the archive, we must obtain an unprotected version and add that to the archive as well. This will primarily be done through contact with content producers.

Outside the archive, there are four knowledge repositories that we must maintain ourselves:

1. *Preferred formats list*. A short list of formats that we consider the best current format for various format categories and aspects, as described in section 3.3.
2. *Endangered formats list*. Here we list the formats that we consider in danger of becoming unreadable. They should be prioritized by both assessed risk and importance of retaining the data stored in these formats.
3. *Converters*. For those formats that we deem necessary to convert, we must have converters at hand to either convert at ingest time, convert as a later batch run, or both.
4. *Policies*. The overall policies that form the basis of the other tasks and lists must be kept up to date with technical and societal developments. Policies include what aspects are considered important for what categories (section 3.3) and what criteria are used to select preferred or endangered formats (section 4).

Two knowledge repositories are external, preferably maintained through international cooperation:

1. *File format registry*. A list of known file formats, with as must detail as possible on the structure of the format, the practical usage, converters and viewers.
2. *Emulators*. These are typically created by other organizations and individuals interested in the system at hand, and will be used in the viewer process as required. Creating an emulator is a very resource-intensive task that we would not be able to undertake ourselves, but we should keep abreast of available emulators.

## 7.2. Prototypes implemented

“Internetbevaringsprojektet” uses the ARC format[26] designed by the Internet Archive[13] for its archival systems. We have extended the ARC format to allow storing of converted files. We have implemented a conversion tool that performs batch conversion of files stored in ARC files, given a tool that can convert the raw file. The converted files are then stored in new ARC files.

In order to benefit from the converted files, a presentation tool must be able to ask for the kinds of files that it can display, and the archive must be able to find the converted file when asked for a URL in a format that the viewer cannot display. We are implementing this in a proxy-based viewer prototype.

# 8. Conclusion and recommendations

Handling file formats is an essential part of a long-term archive. History shows us that most file formats fall out of fashion within a few decades, and unless action is taken at an early stage, many archived files will be incomprehensible blocks of bits and so essentially lost. Traditionally, conversion to newer formats has been considered the solution to this technological rot, but the problem of accumulated errors have led to other proposals, such as conversion-on-demand or emulation of current viewers.

We must make sure that we do not place all our archived eggs in one technological basket. An archive solution that is based on a single or a few essential programs runs the risk of losing everything if those programs are faulty or not maintained in the future. Rather, we should take a multipronged approach that will give us some chance of a perfect conservation but also a high probability of at least some useful conservation.

## 8.1. Recommendations

In order to maximize the usefulness of the archives, we give the following recommendations for handling file formats:

- Any file received for archiving must be preserved in its original form in addition to any conversions that may take place, to allow for higher-quality conversion or emulation at a later stage.
- The archive must continuously be monitored for developments in the amount of files in different formats.
- The archive operators must be notified when new formats attain widespread use and when old formats show a declining number of uses.
- The set of criteria for predicting the longevity found in section 4 must be maintained and used as the basis for selecting formats for the two lists below.
- A running list of formats that are obsolete or in danger of becoming so must be maintained and available for automatic processing (endangered formats list). The list must specify what formats to convert from and to, and the reasons for performing the conversion, as well as giving an overview of which of the aspects identifiable in the format are likely to be preserved with different converters or emulation.
- A short list of high-quality formats should be maintained (preferred formats list). These formats should be of expected medium-term usability at least, and should cover the desired types of files and aspects of preservation with a minimum of overlap.
- The preferred formats should be a starting point for negotiations on receiving archive material directly from producers.
- Endangered formats mast be converted into preferred formats at an early stage of archiving, possibly as part of the harvesting system. If resources allow, non-endangered formats should also be converted into preferred formats, to minimize the risk of unnoticed obsolescence.
- Conversion of some formats into multiple formats should be considered when choosing conversions, so as to allow different risk factors for the different aspects being preserved.
- DRM-protected files in the archive should be automatically noticed, and unprotected copies should be obtained and added to the archive by contacting content producers and allowing easy submission of unprotected copies.
- For formats that are particularly hard to convert and considered sufficiently important, suitable viewers and system components should be stored to allow an emulation approach in the future.

- The conversions performed must be subject to periodic quality control. If a conversion does not achieve the desired quality, the conversion programs and formats involved must be investigated to ensure sufficient quality. A switch to emulation for that format must also be considered, in particular if suitable conversion programs or formats are hard to acquire.
- Developments in international cooperation for conversion-on-demand should be followed, and viable projects should be supported, preferably through participation in development.
- Developments in international cooperation on emulation projects should be followed, and viable projects should be supported, preferably through participation in development. In particular emulation of popular formats that have no good conversion alternatives should be supported.
- International cooperation on registration of file formats and their specifications should be supported, preferably through participation in development.

# 9. References

[1] Laura Tangley, “Whoops, there goes another CD-ROM”, U.S. News & World Report, February 16, 1998.

[2] Andy Finney, “The Domesday Project”, 2003., URL: http://www.domesday.org.uk

[3] “The Domesday Project – November 1986”, URL: http://www.atsf.co.uk/dottext/domesday.html

[4] “BBC Domesday”, URL: http://www.si.umich.edu/CAMILEON/domesday/domesday.html

[5] “The top file extensions Windows site”, URL: http://www.icdatamaster.com

[6] “Wotsit’s Format”, URL: http://www.wotsit.org

[7] “My File Formats – the programmers file format collection”, URL: http://www.myfileformats.com

[8] “File Format Encyclopedia”, URL: http://pipin.tmd.ns.ac.yu/extra/fileformat/

[9] Internet Assigned Numbers Authority, “MIME Media Types”, URL: http://www.iana.org/assignments/media-types/

[10] Public Record Office Pronom, URL: http://www.records.pro.gov.uk/pronom/

[11] The Representation and Rendering Project, University of Leeds: ”Survey and assessment of sources of information on file formats and software documentation, Final Report”, 2003, URL: http://www.jisc.ac.uk/uploaded_documents/FileFormatsreport.pdf

[12] Free Software Foundation, Inc., “GNU General Public License”, URL: http://www.gnu.org/copyleft/gpl.html, 1991

[13] The Internet Archive, URL: http://www.archive.org/

[14] Microsoft Corp., “Microsoft Product Activation”, December 2003, URL: http://www.microsoft.com/piracy/basics/activation/

[15] Apple, Inc, “iTunes 4: About Music Store Authorization and Deauthorization”, February 2004, URL: http://docs.info.apple.com/article.html?artnum=93014

[16] Jeff Rothenberg, “Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation”, January 1999

[17] Phil Mellor, Paul Wheatley, and Derek Sergeant, ”Migration on Request – a Practical Technique for Preservation”. In *Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries,* pages 516-526. Springer-Verlag, June 2002

[18] “Aflevering af elektroniske arkivsystemer til Statens Arkiver”, Statens Arkiver, 2000

[19] The Camileon Project, URL: http://www.si.umich.edu/CAMILEON/

[20] Multiple Arcade Machine Emulator. URL: http://www.mame.net/

[21] Randy Thelen, “Under the Hood: The Power Mac’s Run-Time Architecture”, BYTE Magazine, April 1994.

[22] CodeWeavers Software, “CrossOver”. URL: http://www.codeweavers.com/site/products/

[23] Raymond A. Lorie, “A Project on Preservation of Digital Data”, RLG DigiNews, 5(3), June 2001, URL: http://www.rlg.org/preserv/diginews/diginews5-3.html#feature2

[24] MacroMedia, Inc, “Macromedia Flash File Format Specification (SWF)”. URL: http://www.macromedia.com/software/flash/open/licensing/fileformat

[25] iTinySoft, “Magic Swf2Avi”. URL: http://www.itinysoft.com

[26] Mike Burner and Brewster Kahle, “WWW Archive File Format Specification”, September 15, 1996, URL: http://pages.alexa.com/company/arcformat.html