SELECTING ELECTRONIC DOCUMENT FORMATS

Gary Cleveland[1]
IFLA UDT Core Programme
National Library of Canada
gary.cleveland@udt.ifla.org

The number of electronic document formats available on the Internet is large and growing. Whether documents are stored and displayed as images or as text, the range of formats to choose from can be bewildering. This paper examines the characteristics of some of the more popular formats and provides a rough guide for their selection.

1. Introduction

During its brief history, the Internet has become a global repository of information without precedence, containing a large and wide-ranging quantity of materials that continues to grow unabated. One of the side effects of this ever-increasing amount of information is the multiplication of document formats and standards used to store and deliver it. There are image formats, text formats, structured formats, and presentation formats—with names such as TIFF, JPEG, ASCII, RTF, SGML, XML, HTML, and PDF. Given the number of differing electronic document formats available for delivering information over the Internet, how does one decide which format is appropriate for a particular library application?

The answer to this question is not a simple one. There are not any fixed decision tables or simple rules, because in selecting an appropriate format, you must consider a large number of interacting factors. These factors include library mandates, local or national policies about electronic information, available funding, skill levels of staff, technical support, available technical infrastructure, user groups and their expected usage patterns, material types that are the target of the application, and the application itself. To make matters more complicated, each factor has several levels. For example, funding could be generous, adequate, or minimal; user groups could be academic, secondary school students, K-12, or the general public; material types could be fixed media (e.g., paper texts, photographs, art work) or digital (MS Word files, collected SGML documents). At least theoretically, there is an appropriate format for every conjunction of factors and values. Expressed another way, there is an appropriate format for every particular application in a particular environment.

[1] The author would like to thank Terry Kuny and Susan Haigh of the National Library of Canada for insightful discussions on this issue and valuable comments on earlier drafts.

Yet, it is clearly impossible to express all of the various combinations and permutations of the factors involved, yielding a appropriate document format at each point, in such a short paper. Nonetheless, a general tool for aiding in the selection of appropriate document formats can be constructed by focusing on only two, primarily technical, factors:

- **the application** to be developed, including the types of materials that will be the target of the project and the functionality required in the application
- and **the document format**, including its characteristics that enable specific types of functionality

Simplified in this way (albeit artificially), the decision of which format to use becomes a process of matching the application design with the capabilities of the document format. This paper provides background for this matching process. It first describes common electronic document format characteristics and their relationship to potential functions within an application. It then goes on to briefly describe some of the more prominent document formats of interest to librarians, creating a profile for each based on the degree to which they exhibit each characteristic. The profiles are based on a review of the literature, subjective judgements of current usage, and approximate estimates of costs. The result is a rough guide that will help, at least in part, in narrowing down the choices among many alternative electronic document formats.

2. Characteristics of electronic document formats

At their simplest, electronic document formats are a series of zeros and ones which are capable of being stored and manipulated by computers and delivered through networks. However, at a higher level, document formats are much more complex. They can be of multiple types, each of which has unique characteristics that determine the way it can be used for information creation, storage, access, and delivery. The primary characteristics of document formats are listed below, along with a discussion of how each relates to factors to consider in the design of a library application for delivering electronic information.

2.1. Machine-readability

While all digital formats can be read by a computer, machine-readability refers specifically to a computer's ability to recognize text within a document without additional processing (i.e., Optical Character Recognition—or OCR). For example, a bitmapped image may contain human-recognizable lettering in its pattern of dots, but without OCR, a computer cannot recognize the same patterns. Electronic document formats that are machine-readable are necessary if the content within them is to be processed by indexing systems, screen readers for the visually impaired, and other text-processing applications. Content already in electronic form will remain machine-readable. However, if the target materials are in a fixed medium, such as paper, then they must be scanned and processed further with OCR.

Questions to ask:

- Must the contents be searchable in the application?
- Must the contents be available to the visually impaired?
- Is there a requirement that scanned material be machine-readable?

2.2. Display of multilingual characters

This characteristic refers to the format's ability to display the manifold characters of the world's written languages. In the context of this paper, this characteristic includes both support of international machine-readable character codes, such as Unicode (see 3.2), as well as the ability to present human-readable characters on screen or paper. Thus, a bitmapped image, because it can display any letter capable of being scanned, could be considered a multilingual format in this context.

Questions to ask:

- In what languages are my materials?
- What scripts do they appear in?
- Does my application require the display of non-Latin-based language materials?

2.3. Layout retention

Layout retention refers to the degree to which a document format preserves the look of the original document. Some formats preserve layout completely, some approximately, and others not at all. For example, a scanned bitmap of a paper page retains the look and feel of the original, but if it is OCRed and stored as ASCII (see 3.2), it loses its original layout. Another example are documents that are produced with desktop publishing (DTP) software. If these complex, laid-out documents are converted to HTML (see 3.4.3), only an approximation of the original layout is possible.

Questions to ask:

- Is original layout important, or is access to only the content sufficient?
- Must the page layout be retained?
- Is verisimilitude to the original important for scholarly purposes (e.g., in the case of scanned paper-based historical materials)?

2.4. Editability

Some document formats can be edited, while others either cannot, or only with a unusual degree of intervention. While less important in library applications where the ideal is

control over the integrity of materials, in some applications (e.g., internal document management) there may be a requirement for editability.

Questions to ask:

- Must the content be able to be edited?

2.5. File size

The same content rendered in different electronic document formats will yield substantially different files sizes. For example, one scanned page (a TIFF – see 3.1.1) containing one column of text is approximately 706 KB uncompressed. The same page as Adobe Acrobat PDF (see 3.3.2) is 76 KB. File size has implications both for storage and for the time it takes to download a document over a network. If a characteristic of your user group is that they typically have slow modems and poor network bandwidth (e.g., in developing countries), then reduced file size is an important criterion.

Questions to ask:

- What resources are there for file storage?
- What is the speed of document server?
- What is the bandwidth of library's network connection?
- What is the speed of target users' connections?

2.6. Multiple-page

This characteristic refers to whether a document format supports inclusion of all "pages" of a document within one file. For example, if a ten-page paper-based article is scanned and stored as GIFs, ten separate files will result.

In an application, such separation complicates document storage, management, and network delivery. Storage and management is made more problematic because the separate files will have to be kept together and their order somehow preserved. For access and delivery, each page will have to be individually requested, viewed, and printed.

Questions to ask:

- How the user will want to use the document?
- Will the user be willing to download and print each page if it is broken up into many sub-components (sometimes a problem with HTML documents), or across many document images?
- Will the user be willing to download a large multi-page file?

2.7. Structured or non-structured

Structured formats explicitly identify document elements, such as titles, authors, sections, headings, and paragraphs. Formats like SGML and XML (see 3.4.1 and 3.4.2), allow structure to be imposed on the content. Be aware that it is generally time-consuming, skill-intensive, and expensive to use structured formats.

Questions to ask:

- Is searching on the full text of a document not sufficient?
- Is there a requirement for searching on specific document elements?
- Is the information reused in other documents?
- Are the documents updated often?
- Are the materials large, complex documents?
- Are multiple authors involved?

Information reuse, frequent updating, management of large, complex documents, and multiple authors are key reasons for selecting structured documents.

2.8. Multimedia

Multimedia formats contain more than one media type, typically text, graphics, audio, and video. Some formats support multiple media while others do not.

Questions to ask:

- Are the materials multimedia?
- Must they be presented in an integrated way online?

2.9. Supports links

Links—like Uniform Resource Locators (URN), HyTime, and XLink—support documents with compound architectures, allowing sub-components to be linked, and multimedia content to be included. They also give documents some interactivity, allowing users to browse from one document to another and select among alternative choices. Not all formats have linking capability.

Questions to ask:

- Will the application have to support interactive content?
- Must the users be able to fill in forms, select among alternatives, browse from document to document?

If so, a format that supports links, like SGML, XML, HTML, or PDF, is required.

2.10. Screen display

All electronic document formats are capable of being displayed on screen, however, some present better than others. Users tend not to like reading long texts on screen. They will, however, read short texts, navigate, browse, and make document selections based on what is displayed.

Questions to ask:

- Will the users be primarily viewing the documents on screen?
- Are the documents Web pages that provide an overall framework for a site—such as a homepage or top-level index pages? This type of content typically is not printed; good screen display is a priority.

2.11. Printing

This characteristic refers to how well, or how easily, the format prints on paper. Most readers will print documents like articles or white papers they wish to read closely or keep in their files.

In designing an application, consider how important printing will be. Another consideration is whether the display version will also be the print version. With image formats, for example, a document optimized for screen display will not print well. There are problems with sizing as well as colour. If printing is a priority, a format optimized for printing will also have to be created. But will you want to be required to store and manage two versions, one for display and one for printing? Some formats, such as PDF, do both well with one version.

Questions to ask:

- Will the users want to print the documents?
- Will they want to print whole documents or parts of documents?

2.12. Availability to search Internet engines

At this time, not all electronic document formats can be indexed by Internet search engines, such as Alta Vista and HotBot. Most are now designed to index only HTML and basic text documents (see 3.2). However, PDF documents are reportedly soon to be among indexed formats. What documents are indexable by Internet search engines will change with time. This characteristic is important if your information must appear in large Internet indexes.

Questions to ask:

- In the application, must the materials be accessible through Internet search engines?
- Are the formats you are considering currently indexed by Internet search engines?

2.13. Resource overhead

There is a wide variation in the resources required to create, store, manage, provide access to, and deliver different document formats. This characteristic involves a number of factors, outlined below.

2.13.1. Tools needed to create and manage the format

The use of a particular document format will affect how extensive the technical set-up will be. The range starts at the low end with simple text editors for encoding HTML, to the high end with sophisticated SGML authoring and management tools for SGML documents.

Questions to ask:

- What tools are required to create and manage the document format?
- What are the underlying infrastructure requirements to support candidate formats (e.g., storage, indexing, document management, backup)?

2.13.2. Complexity of preparation

Documents formats also vary in how difficult they are to prepare. Some can be created with the push of a button, as in the case of PDF, and other only through a long markup process as with SGML or XML.

Questions to ask:

- How complex and time-consuming is the process to put the document in the format?

2.13.3. Skills/training needed for preparation

Skill and training is an often overlooked or minimized aspect of planning for applications.

Questions to ask:

- What specialized skills are required to create and maintain the format?
- How well do these skills match with those of your staff?

- Can staff be trained, or will outside expertise be required?

If it is a small project, and there is no time or resources for extensive training, consider using a format that is straightforward to produce, such as PDF or HTML.

2.13.4. Cost

Some formats cost more to create and maintain than others. SGML, for example, is a very expensive format to implement. Costs for producing materials is HTML or PDF format, on the other hand, are minimal relative to SGML.

Questions to ask:

- What is the overall cost in terms of software, hardware, training and staff time to create and maintain the format?
- What funds are available for developing and maintaining the application?

2.14. Degree of usage

Finally, the last characteristic is the degree of usage of the format on and off the Internet. This characteristic relates to how confident one can be about wide-spread and lasting usability of the format. While there are no guarantees that a particular document format will be usable by a large community, and remain so over the long-term (either for formally recognized standards or *de facto* industry standards), the more common its usage, the higher the probability that software supporting it will readily available over time.

Questions to ask:

- How common is the use of the format?
- Is it widely implemented in IT applications?
- Is it a *de facto* standard?
- Is the software required to view and manipulate the format readily available?

3. Common electronic document formats

The spectrum of electronic document formats ranges from bitmapped images to simple unstructured text to complex, structured, multimedia documents that link to external items like audio and video. As described above, each format possesses a set of characteristics that enable specific types of functionality. This functionality must be matched with the requirements of the target materials and with the intent and purpose of the application. What are the document formats that may be appropriate for your particular application? This section introduces the document formats that are of interest to librarians and, in many cases, are in common use on the Internet. Based on the characteristics listed in Section 2,

a profile that will aid in selection is created for each. The profiles are based on a review of the literature, rough estimates of cost, and subjective estimates of current usage. Table 1 at the end of the section presents a summary of all formats and their associated characteristics.

3.1. Image formats

Image formats are typically used to display digital images of text pages, photographs, illustrations, artwork, and other graphical material. When people talk about "digitization," they are usually referring to digital images of paper pages. Common image formats in use on the Internet include TIFF, GIF, and JPEG.

3.1.1. TIFF (Tagged Image File Format)

TIFF, denoted by the ".tif" extension, was developed by Aldus and Microsoft as a common format for image scanner vendors and DTP software. It is currently the standard intermediary file format for transferring files among scanning, paint, imaging, and DTP programs. The main advantage of TIFF is that it uses a "loss-less" compression scheme wherein no information is thrown away when compressed (in contrast to JPEG, which uses a "lossy" compression scheme. See 3.1.3 below).

Information in TIFF images is not available to text processing applications without additional OCR processing, thus it cannot be indexed by internal nor external Web search engines. Because it is a bitmap format, it resembles the original exactly, retaining layout features, graphics, and characters of any type (it can support human-readable presentation of any kind of character because they are simply rendered graphically). TIFF files are not editable, in the sense that written text within them cannot be easily changed. TIFF uncompressed is a very large file. Multiple TIFF images can be put into one file, but this is not often done because of resulting unwieldy file sizes. Like all image formats, it does not support the inclusion of structure information, a multimedia content, or links to external files.

While files can be presented on screen, their unwieldiness makes them a poor presentation format. In addition, the screen display version cannot easily serve as the print version because of sizing difficulties. On a relative scale, the resource overhead for using TIFF images is low to moderate, requiring scanning and image processing equipment, as well as considerable storage for images. It requires some skill to produce an acceptable image. Because of its relatively large size, and the fact that it cannot be viewed within Web browsers, it is not commonly used as a delivery format over the Internet. It is, however, very widely used as an image capture, exchange, and archival format and is supported by almost all popular imaging software.

Typical uses: A base format used in the scanning process from which other versions for other purposes are derived; an archival format; an interchange format.

3.1.2. GIF (Graphics Interchange Format)

GIF was developed by CompuServe Incorporated in 1987 (the GIF87a format), and improved in 1989 (the GIF89a format). Recognized by the ".gif" extension, it is one of the most common formats for graphical information on the Internet. GIF supports only 256 colours or shades of gray and, although this translates into limited resolution, it also means that GIF file sizes are relatively small. At one time, only GIF images could be viewed as in-line images on Web pages and this fact contributed to its widespread use on the Internet. Since that time, JPEG has been added to the types of images that Web browsers support natively.

Like all image formats, information is GIF images is not available to text processing applications without additional processing with OCR, and neither is text within it editable (without undue manipulation). It resembles the original exactly, acting like a digital "picture" of the original. While it supports limited colours, it is an uncompressed image format, thus file sizes are smaller than TIFF, but larger than JPEG. Multiple GIF images can be put into one file, but again this is not commonly done. It does not support the inclusion of structure information, multimedia content, or links to external files. GIF files can be presented on screen and, in fact, have been used in library-related projects such as JSTOR[2]. The screen display version cannot easily serve as the print version. On a relative scale, the resource overhead for using GIF images is similar to TIFF in that it requires scanning and image processing equipment, as well as considerable storage for images. GIF is supported by almost all image software and, because it has been the preferred file format for inline images in Web browsers, it is one of the most common on the Internet.

Typical uses: An interchange format; inline images in Web browsers; a delivery format for document page images.

3.1.3. JPEG (Joint Photographics Expert Group)

Named after the group that development the standard, JPEG is an image coding standard that has been optimized for continuous tone images, such as photographs. It supports 16 million colours and, because of high compression ratios, is an excellent, small-sized format for delivering photographs over the Internet. However, the compression scheme used is lossy, in that information deemed non-essential to the image is discarded. Typically, three ratios can be chosen when saving a JPEG image, selecting image quality versus depth of compression used. The higher the compression, the lower the quality of the resulting image. The two dominant Web browsers, Microsoft Internet Explorer and Netscape both support JPEG images as in-line images. JPEG files are denoted by the ".jpg" or ".jpeg" extension.

To make text information in JPEGs available to text processing applications, like screen readers or indexing systems, OCR must be performed. JPEG retains the look-and-feel of

---

2 See http://www.jstor.org/.

the original, thus layout features, graphics, and characters of any type are displayed. JPEG's three compression options typically include: 1) high quality/low compression; 2) high quality/good compression; and 3) low quality/high compression. Even at the high quality/low compression setting, the file size is smaller than a GIF. JPEG only supports one image per file. Like the other image formats, it does not support the inclusion of structure information, multimedia content, or links to external files. In addition, images optimized for screen display many not necessarily be adequate for printing, and visa versa. The resource overhead for using JPEG images is similar to TIFF and GIF, however, because JPEG images are much smaller, storage will be somewhat less. JPEG is supported by almost all image software and, because it is now a file format that is supported natively in Web browsers, it is also in very common use on the Internet.

Typical uses: Compression format for colour photographs; inline images for photographs on the Web; larger, more detailed images linked to thumbnails; not recommended for black and white images or text.

3.2. Basic text formats

Text document formats are the simplest form of electronic document. Often denoted by the ".txt" file extension, these documents contain only a simple string of characters and are devoid of more complex information. For example, they do not include information about structure (e.g., explicitly identified titles, paragraphs, authors) or page layout (e.g., font size), or other, more sophisticated elements (e.g., diagrams, tables, binary pictures, sound), or links to other documents. The most common encoding standards for text are ASCII and UNICODE:

- **ASCII.** ASCII—or American Standard Code for Information Interchange—was developed 30 years ago and has become the de facto standard for encoding textual data. ASCII files contain no formatting—only text, spaces and carriage return codes. When a file is in this format, it can be retrieved into many other programs without additional conversion, since the software-specific codes have been removed. Virtually all PC applications accept ASCII data and can transfer data in the ASCII format. Problems with ASCII as a document format is that it strips the file of typographic, graphic, and layout elements. Further, it supports only Western (Latin) characters.

- **Unicode** A likely candidate for replacing ASCII is the Unicode standard. Developed by the Unicode Consortium (IBM, Novell, Microsoft, DEC, Apple and other industry leaders) Unicode is a 16-bit code that allows the representation of 38,885 characters, although it has space for a total of 65,536. These characters cover the major written languages of the Americas, Europe, the Middle East, Africa, India, and the Asia-Pacific regions. Since UNICODE has strong industry support, it is well positioned to become the new standard character-encoding system for all the world's languages.

As a base format, the contents of text documents are available to all text processing applications, such as indexing, word-processors, and screen readers. However, whether multilingual characters are supported depends upon the encoding scheme used. Its layout retention is nil, because when saved as pure text, most formatting is lost. Because there is no additional code or information included in the file, text is the smallest document format. It is multi-page, though pagination is not easily maintained unless typed in manually by the user. Text is not a structured format, does not support multimedia content, nor links to external files. Although a little dull, text presents well on the screen and on paper using the same file. On a relative scale, the resource overhead for using text is very low, requiring only text editors that are typically bundled free with most operating systems. Basic text is supported by all text processing applications and Web browsers and, as such, is an extremely common format.

Typical uses: An interchange format; email newsletters; a delivery format (more commonly in the Internet Gopher era); output from an OCR process.

3.2.1. Rich Text Format (RTF)

RTF is a text format developed by Microsoft that allows some minimal kinds of formatting (e.g., such as bold, italics and underlined characters) to be saved along with the full text of a document. It was developed as an exchange format to allow documents created with different operating systems and different software applications to be readily interchanged, preserving some formatting elements. Many software applications now allow documents to be translated easily to and from RTF. An RTF file consists of unformatted text, control words, and control symbols.

RTF is basic text with some formatting codes added, thus its contents are machine-readable. It has also been designed to use Unicode, if necessary, thus it can support multilingual characters As stated above, it provides some layout retention, though more complex formatting may be lost. Because there is are additional codes included in the file, RTF is the slightly larger in file size than ASCII. It is multi-page, and pagination is supported. RTF is not a structured format, does not support multimedia content, nor links to external files. RTF presents well on the screen and on paper using the same file. On a relative scale, the resource overhead for using RTF text is also very low, and can be produced by most word-processing packages. RTF is not supported by Web browsers and, although one does find RTF documents on the Web, they are not very common.

Typical uses: An exchange format; sometimes as delivery format on the Web.

3.3. Presentation formats

Presentation formats are those formats that have been developed for on screen display or printing (also sometimes called page description formats). They are typically static, single file formats that do not contain any structure information. The most common presentation formats are Adobe PostScript and Adobe Acrobat Portable Document Format (PDF).

3.3.1. PostScript

Adobe PostScript is a page description language—a programming language that is used to specify precisely the location and nature of graphical elements on an output page. PostScript, a proprietary language developed by Adobe Inc., is a hardware- and application-independent language that allows documents containing high-quality graphics and typography to be printed on any PostScript-compatible printer.

While text characters are embedded within PostScript code, they cannot be used by text processing applications[3]. PostScript can display any kind of character set that can be produced by the application in use. Because PostScript is a page description language, it reproduces exactly the look-and-feel of the original. PostScript code can be edited manually, but only by experts who understand the language. Postscript files are fairly large. For example, one page of text output in MS Word 97 format is 20K. As PostScript code, it is twice as large at 40K. (Note that the same file in PDF format, discussed below, is almost half the size of the original Word document at 12K). It is not a structured format, does not support multimedia content, nor links to external files. PostScript is specifically a language developed for printing, and is less useful as a screen display format. Viewers are available, but they can be difficult to work with. On a relative scale, the resource overhead for using PostScript text is very low—it can be produced by most word-processing packages with the push of a button. The use of PostScript is fairly common on the Internet, or at least once was before the advent of PDF.

Typical uses: Delivery format to printing houses; network delivery, though use has fallen off because of the rise in popularity of Adobe Acrobat.

3.3.2. Adobe Acrobat PDF (Portable Document Format)

Adobe's PDF format is a relatively new format, based on PostScript, that supports the on screen display and printing of documents containing complex text and graphical information. PDF is device- and application- independent, allowing files to be displayed or printed on any type of system with the use of the Adobe Acrobat Reader, a free program available from Adobe. Producing PDF files, on the other hand, requires a for-fee program called the Adobe Acrobat Suite. The advantages of using PDF is that it:

---

3 One source states that character strings in a PostScript file are often broken up, sometimes into lengths of one character. Thus, while the character codes are there, they cannot be used intelligently.

- retains the look-and-feel of the original document
- bundles multiple pages in a single file
- supports a zoom feature
- supports internal hypertext links and tables of contents
- supports thumb-nail views of document pages
- supports multimedia content
- supports forms
- can be made keyword searchable
- can be viewed within a Web browser through the use of a plug-in.

Using the Acrobat Suite, PDF files can be created in two ways: 1) from the direct output of software applications, such as word-processing or DTP programs; and 2) converted from the output of a scanning process (typically TIFF files).

3.3.2.1. PDF output from software applications

PDF code has been designed to be machine-readable. Adobe's Acrobat Reader, for example, allows searching of text within a PDF document, while their Catalog product will index a collection of PDF files. The state of indexing PDF files by Web search engines is less clear, with only one search engine, Verity, claiming to index the PDF format[4]. PDF files created from applications look identical to the output that would have been created by the application itself, including the production of Roman and non-Roman characters. In fact, PDF files are created from the print stream of the application, either directly, or indirectly through PostScript as an intermediate format.

PDF does an excellent job of both screen presentation and printing—which is, in fact, what is was designed for. PDF can be easily edited manually with the use of Adobe Exchange. As regards file size, PDF files are relatively small compared with other formats that provide that same functionality. It does not yet have the ability to represent document structure, but it can include multimedia content, as well as both proprietary internal links and URL links to external files. On a relative scale, the resource overhead for using PostScript text is moderate. With the Adobe Acrobat Writer, PDF can be produced as easily as a paper printout. The use of PDF is growing quite rapidly on the Internet, and there are freely available plug-in for Web browsers.

3.3.2.2. PDF converted from scanned output

There are three basic types of PDF files created from scans: 1) PDF image only, in which a bitmap scan has been turned into a vector image; 2) PDF image and text in which the text portion is OCRed, and hidden "behind" the vector image; and 3) PDF "normal," in which the document fonts, layout, and text have been OCRed.

---

[4] The author has yet to see a PDF file be returned as a hit in a Web search.

PDF image

Like all three versions of this type of PDF format, the first step is to convert bitmapped TIFF files to vector images. For the straight PDF image version, no further processing is done by the Adobe Capture software. Thus, it has very some characteristics of image formats—non-machine-readable and good layout retention (including non-Roman characters). However, it differs from image formats in that multiple images can easily be concatenated into one file, the file size is much smaller, they can be resized, and the same file can be used for on screen display and printing. Links and multimedia material can also be supported. The resource overhead for using the PDF image format is fairly low, adding an extra stage in the scanning process to convert scans to PDF. The use of PDF image as an alternative to delivering bitmapped images of text is growing.

PDF image and text

The PDF image and text version also begins with a TIFF conversion, however, the Adobe Capture software does further OCR processing. The recognized characters are hidden "behind" the image so that the exact "digital picture" of the original is retained, yet the text is now searchable. All other characteristics of a PDF output from an application apply.

Normal PDF

The PDF normal version has even more processing done to it. With this version, not only are the characters recognized and converted to their machine-readable equivalent, but the fonts and layout of the document are as well. The Capture program tries to match and replace the original font with a system font in its library. Where it fails to recognize a feature of the original image, the program snips out that portion and pastes it into the final output. This version of PDF can be edited manually with the use of Adobe Exchange. All other characteristics of a PDF output from an application apply.

Typical uses: Delivery format for application output; as a replacement for delivering document images; growing in use.

3.4. Structured formats

Structured formats as defined in this paper are those formats that support explicit tagging of document elements. Formats falling into this category are SGML, XML, and HTML.

3.4.1. Standard Generalized Markup Language (SGML)

Standard Generalized Markup Language (SGML) emerged 25 years ago out of the publishing industry as a method of easily exchanging text among typesetting systems. Essentially, SGML provides a framework for describing the logical structure of documents. It sets out a grammar and method for: a) explicitly identifying the structural, or content, elements of documents using tags; and b) codifying the hierarchical relationships among them.

The SGML standard also contains general rules for describing specific types of documents. These descriptions, called Document Type Definitions (DTD), set out the elements allowed in a given document type as well as the relationships among the elements. For example, a journal article can be defined as having a title, followed by an author name, followed by an abstract, followed by an indeterminate number of headings and paragraphs in the body, followed by references. In short, the DTD is an abstract model of the document. Two common DTDs of interest to librarians are the Text Encoding Initiative (TEI) DTD for the markup of scholarly texts, and the Encoded Archival Description (EAD) DTD which provides a framework for encoding archival finding aids[5].

One of the key characteristics of SGML documents is that the content and the layout of the document are separated. SGML tags specify only the content of a document, and not its layout. In propriety desktop publishing (DTP) applications, content and layout are typically merged within a document. That is, the codes for a paper's appearance is contained within the file along with the content. This conflation causes problems when the document is to be produced in a different format (e.g., HTML), because the original layout instructions must be removed and replaced with instructions appropriate to the new medium. Because SGML separates content from layout, alternative layouts can be easily applied to the same structured content.

With SGML, information in documents becomes more than a long string of undifferentiated characters. Each element instead becomes a uniquely identifiable "object" that can be independently stored, accessed, inserted, deleted, revised, hidden, protected, searched, and reused. This feature of SGML is what makes it so powerful in managing large and complex documents and document collections.

SGML consists of tagged text of any type. It is, therefore, both machine-readable and supportive of multilingual character sets. Because its structure is explicitly identified, searching options can be much more powerful than simple text. As stated above, SGML is concerned with the structure of a document, rather than its appearance, thus layout depends entirely on the output specifications created for it. By itself, SGML does not support layout. SGML documents are editable, in fact, this is one of its strengths in that it is well-suited to environments where many authors must contribute to a work. Because SGML is simply tagged text, relatively speaking, it has a small file size. It supports multimedia content, as well as links to external materials. There are many tools available with which to view SGML documents on screen, however, there is only one freely available SGML viewer, called Panorama, and its development has fallen behind its commercial counterpart. Printing and output on other media depends, again, upon the output specifications designed for the document in question.

---

[5] See http://www-tei.uic.edu/orgs/tei/ and http://www.loc.gov/ead/ead.html for more information.

Regarding cost, SGML is one of the most complex and expensive formats to use. Thus, one must be very careful about what applications it is chosen for. SGML is a complex tool and, therefore, requires a large investment in skills development. SGML skills are needed on many levels, including a thorough understanding of the standard; document analysis and DTD design; training of personnel in the workings of the re-engineered document processes, as well as in any specialized SGML tools they may use; and the selection, setup, and maintenance of an SGML document-processing system.

Typical uses: Documents suited to SGML encoding tend to be large, complex, with multiple authors. They typically undergo frequent revision, have components that are reused in other documents, and are output on multiple media. SGML is used most successfully in document management applications where documents are massive, number in the millions, and involve many parties at both the input and output stages (e.g., in the defense, aviation, and pharmaceuticals sectors).

3.4.2. Extensible Markup Language (XML)

XML is a simplified subset of SGML intended for use over the Web. It is a grammar, like SGML, rather than a fixed tag set, for creating tags and adding structure to documents. It is intended to be more powerful than HTML in giving documents structure, but easier to implement than full-blown SGML, and thus lies somewhere between SGML and HTML on the spectrum of complexity. Specifically, it has been designed to be:

- easier to implement over the Web
- easier to define document types
- easier to create supporting software
- interoperable with both SGML and HTML.

Like SGML, XML supports a strong separation of content and presentation. To control appearance, a standard for output specifications called Extensible Style Language (XSL) is being proposed for use with XML. It combines formatting features from both DSSSL[6] and CSS[7]. The proposed standard for linking documents in XML is XLink. Both the style sheet and linking standards have yet to be finalized.

As a simplified subset of SGML, XML shares many of its characteristics. However, because it has been designed to be easier to implement, one assumes that the overall costs of use will be lower. This assumption may, or may not, be borne out with time. It has also been designed to use Unicode as the character set, ensuring handling of a wide range of Roman and non-Roman characters. There is still very little use of XML on the Web. The current release of Microsoft's Internet Explorer supposedly offers limited XML

6 DSSSL, or the Document Style and Semantics Specification Language, is the international standard style sheet for SGML.

7 CCS, or Cascaded Style Sheets, is the style sheet developed for HTML.

support, but XML documents only view as raw code in the browser. The upcoming Netscape 5.0 is purported to be able to import and display XML documents.

Typical uses: Not yet widely used; intended niche is Web documents that require more structural definition than HTML can provide.

3.4.3. Hypertext Markup Language (HTML)

While SGML and XML are both languages for creating tag sets for structured documents, HTML is a *specific* SGML DTD, providing a fixed set of tags and an architecture that defines the elements and structure of an HTML document. These elements include items such as titles, paragraphs, citations, and lists. While the original intent with HTML was to also separate content from layout, the HTML specification was diluted over time by the addition of tags that dealt directly with appearance.

Linking in HTML is accomplished through the Uniform Resource Locator (URL) whose four-part structure includes the protocol in use, the machine address, the document path, and the document file name (e.g., http://www.nlc-bnc.ca/ifla/VI/5/udt.htm).

HTML shares many characteristics of SGML and XML. It is machine-readable and indexing engines that recognize HTML are widely available. One difference is that the appearance of a document depends upon an interaction of the HTML structure tags used, any additional layout tags used, and the manner in which a particular browser is programmed to display text within them. Thus, the same HTML code will appear slightly differently from browser to browser. By desktop publishing standards, a HTML document's layout is rudimentary. When documents produced with word-processing or DTP systems are transferred to HTML, much of the sophisticated formatting is lost. The simplicity of HTML, however, is one of the factors that has made the Web the large repository of information it has become. It is easy to learn and to perform HTML markup. Low cost tools to markup HTML abound. HTML is, by far, the most common document format on the Internet.

Typical uses: Web documents; skeletal or "administrative" Web pages that link to other document types (e.g., PDF); interfaces to databases and library catalogues.

4. Conclusion

Because every library and every application is unique, the selection of appropriate electronic documents is not just matter of picking from a list. Many factors have to be taken into consideration—differing sets of policies, available funding, skill levels of staff, material types, technical support, and user groups will all affect the decision. This paper has focused on the more technical aspects of this problem, examining two factors only: application design and associated material types, and the capabilities of the document format. The rough guide for selecting appropriate document formats presented in this paper will allow librarians who are considering building Internet-based library applications to begin to narrow the choices among the wide range of document formats in use on the Internet today.

Selected sources

Alschuler, L. (1995). *ABCD... SGML: a user's guide to structured information*. London: International Thomson Computer Press.

Coleman, J. and Willis, D. (1997). *SGML as a framework for digital preservation and access*. Washington: The Commission on Preservation and Access.

Cover, R. (1998). Extensible Markup Language (XML). Available at: http://gopher.sil.org/sgml/xml.html

Flynn, P. (1998). *Frequently Asked Questions about the Extensible Markup Language*. Available at: http://www.ucc.ie/xml/

Goodman, R. (1997). *Document Formats: A Discussion Paper*. Project Acorn's Reports and Documentation. Available at: http://acorn.lboro.ac.uk/reports/formats.htm.

Mace, S. Flohr, U., Dobson, R., and Graham, T. (1998). *Weaving a better Web*. Byte (Online). http://www.byte.com/art/9803/sec5/art1.htm

Murray, J.D., and Van Ryper, W. (1994). *Graphics file formats*. Sebastopol, CA: O'Reilly and Associates.

Silver, B. (1997). *Bringing paper to life: Adobe Acrobat Capture unlocks corporate memory*. Industry Trend Reports, April.

Table 1: Summary of electronic document format characteristics

Characteristic (including questions to consider)	TIFF	GIF	JPEG	ASCII	RTF	Post-Script	PDF app. output	PDF Image	PDF Image + text	PDF normal	SGML	XML	HTML
Machine-readability Must the contents be searchable in the application? Must the contents be available to the visually impaired? Is there a requirement that scanned material be machine-readable?	No	No	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes
Multilingual characters In what languages are the materials? What scripts do they appear in? Does application require the display of non-Latin-based language materials?	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Layout retention Is the layout is important, or is access to only the content sufficient? Must the page layout be retained? Is verisimilitude to the original important for scholarly purposes?	Yes	Yes	Yes	No	Some	Yes	Yes	Yes	Yes	Yes	No, by itself	No, by itself	Some
Editability Must the content be able to be edited?	No	No	No	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
File size What resources are there for file storage? What is the speed of document server? What is the bandwidth of library's network connection? What is the speed of target users' connections?	Large	Medium	Small	Very small	Very small	Large	Small	Medium	Medium	Medium	Small	Small	Small
Multi-page How the user will want to use the document? Will the user be willing to download and print each page if it is broken up into too many sub-components, or across too many document images?	Possible	Possible	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Structured/non-structured Must the structure be machine-identifiable? Is searching on the full text of a document not sufficient? Is the information in them reused in other documents? Are the documents updated often? Are the materials large, complex documents? Are multiple authors involved?	Non	Non	Non	Non	Non	Non	Not yet	Non	Non	Non	Yes	Yes	Yes
Multimedia Are the materials multimedia? Must they be presented in this way online?	No	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 1: Summary of electronic document format characteristics (continued)													
	TIFF	**GIF**	**JPEG**	**ASCII**	**RTF**	**Post-Script**	**PDF app. output**	**PDF Image**	**PDF Image + text**	**PDF normal**	**SGML**	**XML**	**HTML**
Supports links Will the application have to support interactive content? Must the users be able to fill in forms, select among alternatives, browse from document to document?	No	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Screen display Will the document only be displayed?	Poor	Good	Good	Good	Good	Poor	Good	Good	Good	Good	Poor	??	Good
Printing Will the documents be printed?	Good	Good	Good	Good	Good	Good	Good	Good	Good	Good	Good	Good	Adequate
Avail. to Web search engines Must the materials be accessible through Internet search engines? Are the formats under consideration currently indexed by Internet search engines?	No	No	No	Yes	Yes	No	??	No	??	??	No	No	Yes
Resource overhead	Low – Moderate	Low – Moderate	Low – Moderate	Low	Low	Low	Low	Low	Low	Low	High	Moderate	Low
Tool requirements What tools are required to create and manage the document format? What are the underlying infrastructure requirements to support candidate formats?	Low – Moderate	Low – Moderate	Low – Moderate	Low	Low	Low	Low	Low	Low	Low	High	Moderate	Low
Complexity of preparation How complex is the process to put the document in the format?	Low – Moderate	Low – Moderate	Low – Moderate	Low	Low	Low	Low	Low	Low	Low	High	Moderate	Low
Skills/training needed What specialized skills required to create and maintain the format? How well do these skills match with those of your staff? Can staff be trained, or will outside expertise be required?	Low – Moderate	Low – Moderate	Low – Moderate	Low	Low	Low	Low	Low	Low	Low	High	Moderate	Low
Cost What is the overall cost in terms of software, hardware, training and staff time to create and maintain the format? What funds are available for developing and maintaining the application?	Low – Moderate	Low – Moderate	Low – Moderate	Low	Low	Low	Low	Low	Low	Low	High	Moderate	Low
Degree of usage on Internet	Low	High	High	High	Low	Moderate	Growing	Growing	Growing	Growing	Low	Low	High
Degree of usage, other apps	High	High	High	High	High	High	Growing	Growing	Growing	Growing	-	-	-