Role of PDF and Open Data

James C. King
Senior Principal Scientist
Adobe Systems Incorporated
March 3, 2013

Introduction

There is widespread belief that once data has been rendered into a PDF format (Portable Data Format – ISO standard ISO 32000-1[1]) any hope to access or use that data for purposed other than the original presentation is lost. While there is an element of truth to that belief, it is not nearly as dire as most believe.

Common Use of PDF

PDF is arguably the most widely used file format for representing documents in a portable and universally deliverable manner. The ability to capture the exact appearance of output from nearly any computer application in a form that can subsequently be viewed or printed on nearly any computing device has made it invaluable for the presentation of content for which the author wishes to have total control of the presentation. The traditional print/publishing world was and is obsessed with the appearance of content and PDF suits their electronic needs quite well.

The challenge has been to find ways to have your cake and eat it too: to have a highly controlled and crafted final presentation and yet keep the ability to reshape the same content into some other form. We know of no perfect solution/format for this problem but there are many ways in which PDF can contribute to solutions.

In our presentation we intend to review the best ways in which PDF can contribute to the distribution and use of Open Data.

Preexisting PDFs

The most difficult challenge is to attempt to harvest content from preexisting PDF files that were made in some unknown way and most likely without any concern to provide for such harvesting. The wide range of PDF creation tools with an equally wide range of quality levels make approaching preexisting PDF file a risky business. For harvesting such files, although difficult and not always perfect, there are tools and techniques readily available. Basically the methods must reconstruct information about the content that may have been lost when converted to the most elementary forms of PDF. Such as:

- Converting glyph references into characters. This includes things like removing line-end hyphens, breaking single ligature glyph references into multiple character references, etc. (See this blog post[2] for more details.)
- Determining the reading order of any text that is presented within the content. It is possible to create PDF files where the order of the glyph references is not in

the same order as needed to read the content. Tools must determine the reading order by where on the page the various glyphs are placed, using that information to determine words, columns, headings, etc.

- Discovering the repetitive nature of tabular content and converting it to CVS or some other such form. This also is best done after determining the positioning of the content on pages.
- Discovering images or graphics that can be extracted. This is actually one of the more easily accomplished feats since that material is more clearly identified as such with PDF files.

There are many products and open source tools that do these things and do them surprisingly well. For some reason many of the people that could make use of them are not aware of what is readily available.

PDFs Planned for Harvest

It is possible to make PDF files from which the content can be reliably extracted and reused. Additional, optional information, not essential for quality rendering, can be included within a PDF which provides the structural information needed to determine things like headings and reading order without the need to rediscover it. Additional information for converting glyphs to characters can also be embedded within the PDF file. Such files are called Tagged PDF[3] and make use of the Structured PDF[4] constructs to include this additional information.

Again, tools and products are available for taking advantage of this information, when present, to extract content from PDF files with precision. For example, it is now routinely possible to round trip files between PDF and Microsoft Word formats.

PDF attachments

Perhaps the most powerful and useful way for PDF to contribute to the Open Data initiatives is to use it as a very flexible enveloping mechanism to hold and describe the original raw data using the file attachment feature of PDF. The PDF standard allows for any number of any kind of attachments to a PDF file to be embedded within the file itself. An interesting side note is that PDF/A-3[5] also allows this when creating archival PDFs.

PDF attachments can be compressed when embedded within a PDF file using the same compression method (Flate) as used by ZIP[6] and PNG[7]. This is key technology for being able to keep the file sizes small especially when embedding XML, which usually compresses exceptionally well using this method. And finally, any of the extra PDF services like digital authentication, digital signatures, forms, and encryption/access control that can be used on the PDF files also encompasses the embedded attachments.

Hybrid PDF

This PDF attachment feature has been used effectively by Open Office software to produce "hybrid" documents[8], which are PDFs that include the OpenDocument[9] files that were used to create the PDF. The OpenDocument software will open the attached files whereas a PDF viewer will display the PDF.

This feature is also provided for Microsoft Word using Adobe's create PDF software for Word.

Enveloping Raw Data using PDF

Another very powerful technique is for each graph or spreadsheet presented within the PDF to attach the originating file. A clickable button can be placed on or near the presentation, such that clicking it will provide an interface for extracting that particular "source" file. This provides the ability of an author not only to offer nicely formatted and interpreted presentations of data but also to include the original data used to create those presentations. Of course, this technique is not limited to graphs or tables but can be applied to specific textual content items as well.

Raw Data – Too Raw

A bunch of numbers held within a CVS file may or may not be useful as Open Data. Exactly what each number represents and how they may interrelate is important to being able to do subsequent processing of those data. In most cases, the raw data needs some accompanying documentation in order to accurately interpret and process it.

A very effective solution to this can be provided by using a PDF file to document the properties and interpretation of the data, carefully explaining the format, semantics, source, etc. of the data that is then attached to that PDF file. This provides an all-in-one package that not only provides the data but also explains how it can be subsequently used. An example of such a file can be found here[10].

In the case of XML data, a schema can be a side-by-side attachment to the XML attachment in the PDF file can provide the additional semantic information that is missing, yet absolutely needed for further processing. A more lengthy discussion of this can be found in this blog article[11].

Summary

Recognizing that PDF remains the preferred choice for electronic document representation, an understanding of the limits and abilities for PDF within the Open Data world will prove useful. We have tried to show the depth and breadth of PDF concepts, tools, and products that can become very effective in Open Data workflows.

References

1. ISO 32000-1
   Standard: http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/PDF32000_2008.pdf
2. Text Content in PDF Files: http://blogs.adobe.com/insidepdf/2008/07/text_content_in_pdf_files.html
3. Tagged PDF: http://www.pdfa.org/2011/10/the-value-of-tagged-pdf/
4. Structured PDF: http://wac.osu.edu/pdf/what-is-accessible-pdf.html
5. PDF/A-3: http://en.wikipedia.org/wiki/PDF/A
6. ZIP: http://www.zlib.net/
7. PNG: http://www.libpng.org/pub/png/book/chapter09.html
8. Hybrid Documents: http://extensions.services.openoffice.org/project/pdfimport
9. OpenDocument: http://www.openoffice.org/
10. Sample PDF Document: http://blogs.adobe.com/insidepdf/files/2010/11/LeadershipPacs_2010.pdf
11. XML data: http://blogs.adobe.com/insidepdf/2011/10/my-pdf-hammer-revision.html