How to modify PDF documents?

Seth Kenlon

The Portable Document Format, or PDF, is a document standard developed by Adobe to help ensure that when a user prints, they get exactly what they see on the screen. They are used as "pre-flight" tests in graphics design, or as convenient ways to sent documents (with embedded fonts and graphics) across a network.

PDFs also get abused quite a lot; they are often considered an e-book format even though they don't feature the re-sizing and re-flowing capabilities of true e-book formats like e-pub. PDFs are, as their name states, meant to be digital versions of paper with all of its advantages as well as disadvantages.

Sometimes you'll a need to modify PDFs. Adobe itself successfully pushed PDF as a universal, cross-platform standardized format, and yet their Acrobat Pro application, which allows a user to open and modify PDF documents, is not available on Linux. As a result, many Linux users have opted for e-pub (based on entirely free technology like html, css, and zip) over PDF, but PDF still does have distinct advantages when you need a document to be an inflexible print-ready proof. Luckily, there are a number of tools that will allow you to work with, create, and modify PDFs on Linux.

Evince

As a part of the GNOME Desktop Environment (which Ubuntu's Unity uses as its base), Evince is the default PDF viewer on Ubuntu Linux. It allows you to do all of the usual PDF things like read, rotate, resize for viewing, and well as a few advanced features.

One problem with PDFs is that there are many different ways to create them but not much of a way to tell what feature set a particular document actually contains. For instance, it's possible to embed text into a scanned set of images in a PDF (using OCR technology), but not everyone does this. So you might open one PDF document of a scanned text book and discover that Evince can highlight and copy all of the text on a page. You might then open a separate document that looks basically the same and yet Evince will be unable to select or copy the text. It's important to know that this is not something Evince is or is not doing; it's information that is or is not embedded invisibly into the PDF file itself.

Another confusing feature are PDF Forms (FDF). You might download a document that you are required to fill out (such as a job application or a school form) and find that Evince allows you to click into each field and type in the data. You could then save the PDF with the form data included and submit it back to the organization requesting the information. You might then open a different form and find that Evince will not allow you to fill in the data. Once again, this isn't Evince arbitrarily deciding whether or not you can fill out a form; it's how the PDF itself was created. Some have form data while others do not, and there's no good way to know for sure which is which except by trying to perform an action and watching the results.

Figure 1. *The Evince viewer brings Adobe PDF functionality to your desktop*

For all the common tasks, however, you'll find Evince a fine PDF viewer that easily matches Adobe Acrobat Reader in features and performance.

Modification

Even though Adobe doesn't bother releasing Acrobat Pro for Linux, there are plenty of tools that we Linux users can use to modify PDFs.

The first is Inkscape, a digital illustration application that also happens to translate PDFs into their graphical and textual components. Retaining the layout of the page, Inkscape is able to open PDFs into fully modifiable page layouts.

Inkscape is a traditional illustration program that is powerful and yet intuitive. Install it via the Ubuntu Software Center; for a full, free series on using the tools of Inkscape, see *screencasters.heathenx.org*

If the PDF contains multiple pages, then you may need to stitch the pages back together. For instance, if you modify the second page of a three page document, you could open page 2 in Inkscape, modify it, and export it as a stand-alone modified page 2, but then you'll want to integrate it back into the original three page document.

Once you've made the modifications to the PDF page, you can export it page out as a PDF just as you would with any other application; choose Print from the *File* menu, and choose to *Print to File*.

There is a handy commandline tool for this slicing and dicing of PDF files called pdftk (PDF Tool Kit), available from the Ubuntu Software Center. There's a lot you can do with pdftk, including splitting up the pages in a PDF and then re-constructing it.

To break the PDF into pages, you can use the burst option:

```
pdftk bigfile.pdf burst
```

To stitch it back together again with the new page 2:

```
pdftk pg_0001.pdf modified_2. ↵
   pdf pg_0003.pdf cat output ↵
   newfile.pdf
```

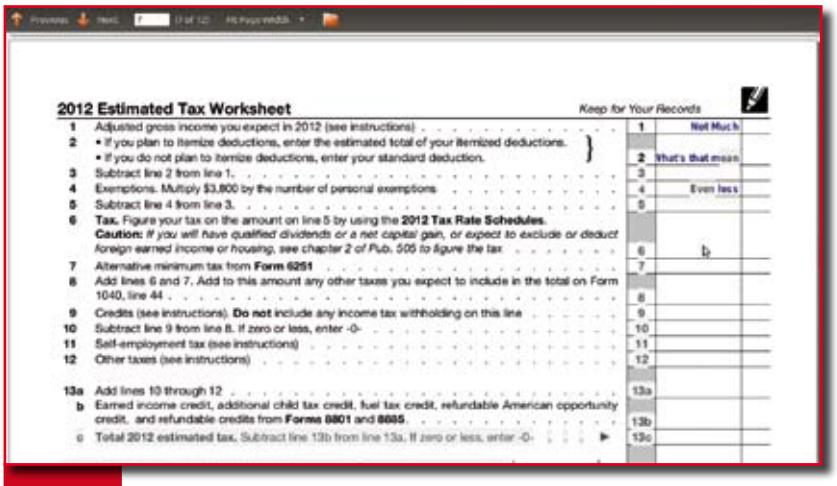

Figure 2. *Filling out Form Data with Evince*

And this will leave you with a single PDF file ("newfile.pdf" in the example) as if though nothing had changed.

Generating a PDF

There are many ways to send your own documents out in the PDF format. Nearly every program that can print can also "print" to PDF; in other words, the applications thinks it's printing, but instead of printing to paper it saves the results to a file.

This is a great way to quickly get PDF versions of anything that can be printed. If you do any kind of document authoring either in LibreOffice, Open Office.org, Scribus, or even just a basic text editor, you have this method to produce attractive and functional digital documents. The disadvantage to this, however, is that it does not take advantage of PDF's ability to do internal linking or follow external hyperlinks; i.e., if you reference a web page in your document, the user cannot click on the weblink and be whisked off to that webpage in their browser. At best, they'd need to select the text in the PDF and open a browser and paste the text into the URL bar; this is simple enough on a computer but can be difficult on a mobile device.

The answer is an html-like markup language called docbook, which allows you to create documents in any plain text editor like Gedit or Emacs, process the document with a stylesheet, and then output to a PDF that will open in PDF viewers with all hotlinks enabled.

To install the docbook toolchain, visit the Ubuntu Software Center and install the following programs:

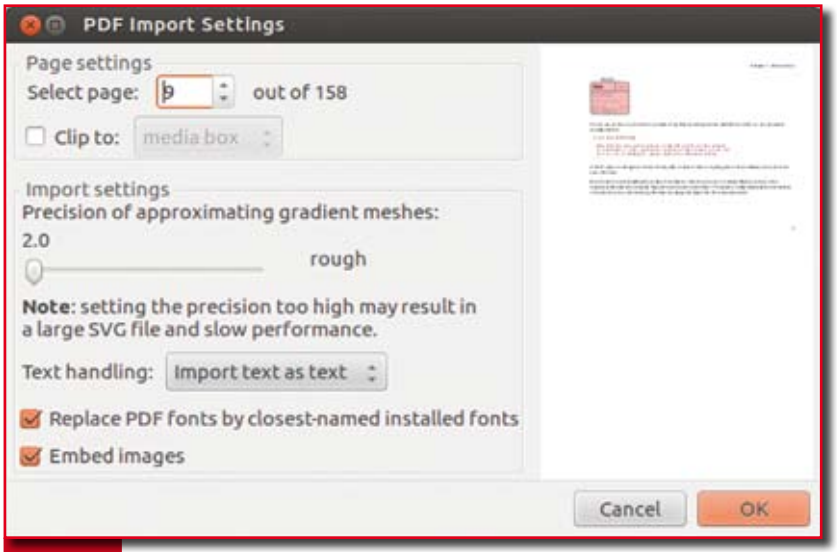

Figure 3. *Open any PDF page in Inkscape*

Figure 4. *Modifying PDFs with Inkscape!*

```
sudo apt-get install fop xmlto ↵
    docbook
```

If you've ever used HTML, then docbook will come naturally. If not, then you'll probably find docbook a little technical at first but once you've tried it for a few basic documents, you'll find the learning curve pretty minimal.

The idea behind docbook is to use obvious markup tags that label significant elements in a document. These include *<para>* for paragraphs, *<itemizedlist>* for bullet lists, *<orderedlist>* for numbered lists, and so on. Once you know a few tags, it's fairly intuitive.

To get started on a basic document, open gedit and type in the following sample text:

```
<?xml version="1.0" ?>

<!DOCTYPE article PUBLIC "-// ↵
   OASIS//DTD DocBook XML ↵
   V4.1.2//EN" "docbook/d ↵
   ocbookx.dtd">

<article>
<title>
Linux Identity Sample PDF
</title>

<para>
Linux Identity is an informa ↵
   tive magazine. Visit ↵
   their <ulink url="http:// ↵
   linuxidentity.
   com">website</ulink> ↵
   today!
</para>
</article>
```

That is the basic structure of a basic docbook document. To process it, you must first add a header lien or two so that the stylesheet processor knows how to interpret it, so add these two lines to the very top of your document:

```
<?xml version="1.0" ?>

<!DOCTYPE article PUBLIC "-// ↵
   OASIS//DTD DocBook XML ↵
   V4.1.2//EN" "docbook/d ↵
   ocbookx.dtd">
```

Save the file as magazine.xml to your Documents folder, and you're ready to process the document to apply some (very) basic styles:

```
xmlto fo ~/Documents/magazine. ↵
   xml -o ~/Documents/fo
```

The "fo" filetype is a PDF-ready format that would look mostly like gibberish if you were to look at it. So the final step is to process the ".fo" document that xmlto has just created into a proper PDF:

```
fop ~/Documents/fo/magazine. ↵
   fo ~/Documents/magazine.pdf
```

Now open your file manager, Nautilus and take a look in your *Documents* folder. You'll find magazine.pdf there, which you can open in evince. It won't be much to look at, since it is, after all, very basic, but try clicking the hyperlink and notice how it automatically opens your web browser and takes you to the appropriate website.

Additional features of docbook include embedding media like graphics, providing an automatically hotlinked Table of Contents and Index, blockquotes, code boxes, and obviously all the styles and fonts you could ever want. It has been used to create ebook and printed versions of school textbooks, technical manuals, articles (including this one), scientific papers, works of fiction, and much more.

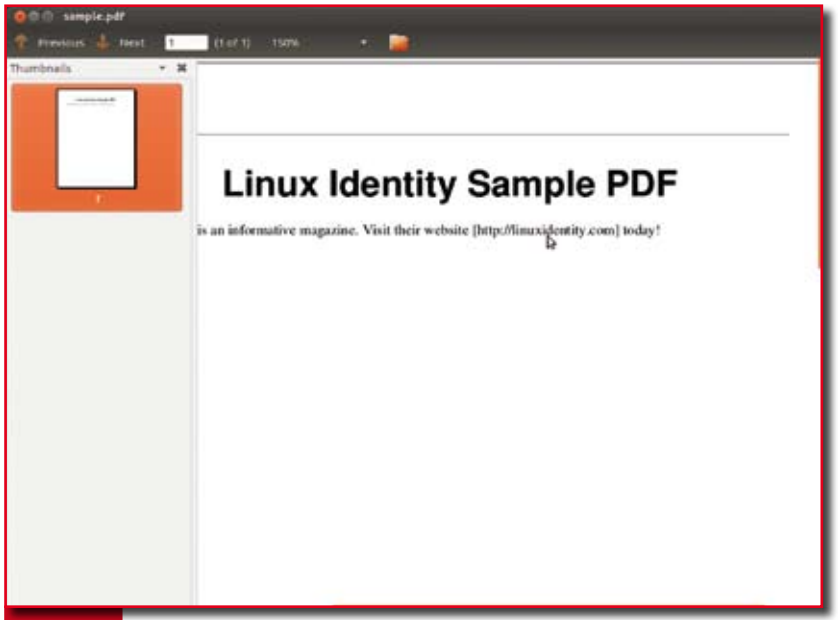

Figure 5. *Hyperlinked PDFs from free software tools*

Conclusion

PDFs are powerful tools for proofing and for delivering rich paperless documents. They can be over-used and mis-used, so think twice before you generate PDFs when you really mean to send text files, epubs, or .odt files. Whatever you use, you can be sure that Linux has plenty of tools to manipulate PDFs, all you need to do is explore them.