

Templates and PDFs in Scribus

Ready for Press

In this final part of our three-part series [1][2], we'll put the finishing touches on our Linux newspaper. You'll learn how to generate a PDF, and you'll find out what a real pre-press professional thinks about our project. **BY JASON WALSH**



www.scribus.hu

In last month's installment, we finished our newspaper front page and offered it as a guide for creating the remaining pages. Now we'll show how to add the finishing touches and assemble the newspaper for printing.

Templates

In any print design project (or web design project, for that matter), one key

issue is consistency. In order to be consistent, it is necessary to create pages that look relatively similar even if they hold different types of content. Basic page elements, such as the system for numbering pages (called the *folio* in publishing), should be identical, as should the fonts used throughout.

Our folios are set in Gill Sans Bold, but what typeface and style you choose is

entirely up to you. Whatever it is, keep it constant throughout your project.

In desktop publishing, a template is simply a file that contains all of the semi-static information, such as folios and page furniture. Take a look at the three illustrations shown in Figure 1, and you'll see how even relatively different pages can share similar items and a basic format.

THE AUTHOR

Jason Walsh was the art director on the *East Belfast Observer* from launch in January 2004, until July 2004, prior to which he was the art director for the Irish glossy magazines, *Gorgeous* and *CityCraic*. These days he works as a journalist and has contributed to *Linux Magazine*, *Variante*, *Mute*, the *Guardian* and many other newspapers and magazines on art, design and technology.



PDF Security

That *Security* tab that we will be ignoring later on is of no use in newspaper publishing, but that doesn't mean it's not interesting. You can use the *Security* tab to restrict access to the PDF. Options include not allowing printing, changing the text, or copying images without having a password.

The encryption routine used in Scribus is good: we tested it by making a password-protected file and attempting to print it from Acrobat Reader on both Linux and Mac

OS X. Neither system would print the file without permission, nor would Adobe Photoshop. This tactic is used for managing on-line subscriptions by several small newspapers, including the *Northern Pen*, a weekly title from Newfoundland. The *Extras* tab is mainly concerned with transitions, enabling you to make Powerpoint-style presentations from within Scribus. Even better, the presentations are platform independent.

What is most vital is that page measurements and style-sheets are identical across all pages. In order to ensure this, it is best to take the completed front page and make a template from it by removing all non-permanent items from it. This way, not only will the style-sheets be intact, but the page dimensions and measurements will also be correct.

Scribus [3] has a *Save as Template* option in its *File* menu, but simply using *Save as* will also work as long as you remember not to overwrite the template when saving later changes.

Preparing for Print

A few years ago, if you wanted to take a file to a printer, it was necessary to use files from industry-standard applications like Quark XPress or Macromedia Freehand. Today the industry-standard file format is still required, but the standard is no longer a propriety application-centric format: it's Adobe's Portable Document Format (PDF).

As a file format, the PDF has two major advantages: first, it frees the user from having to use specific applications, and second, native DTP file formats such as those of Quark XPress, InDesign, and even Scribus are, of course, editable. That is not a good thing at the print end of the workflow. If the printer is using fonts with slightly different metrics, the entire project can fail to print correctly: type will be in the wrong places, and the whole project will have to be re-done. The traditional workaround for this problem was to send your fonts to the printer, something that was actually forbidden by the font

licensing agreements! PDFs were explicitly created for document interchange, so this is no longer an issue.

Like all serious design applications, Scribus is capable of producing PDFs. PDFs can be as simple or as complex as the job demands. To make our newspaper PDFs, go to the Scribus main menu and choose *File > Export > Save page as PDF*. Have a look at the dialog box that appears (see Figure 2). Complex? Fear not; it's pretty easy to understand.

This menu contains plenty of options. First of all, in the *General* tab, select *Downsample Images* and set it to 200dpi. 200 dots per-inch is a high enough resolution for a newspaper press. This relatively low resolution is suitable for newspaper printing because the ink will spread out on the low-grade paper used to print

newspapers. For higher quality print jobs such as magazines, this setting would need to be 300dpi or higher. If memory and disk space are not issues, Scribus can handle almost anything you throw at it. Beware, however, that if your image files are at a lower resolution, say 72 dots per inch, as one might find on the worldwide web, neither Scribus nor any other application can magically increase the resolution. You might save at 800dpi, but



Figure 1: Using a template promotes consistency – even relatively different pages such as the three seen above are all created from the same basic Scribus template.

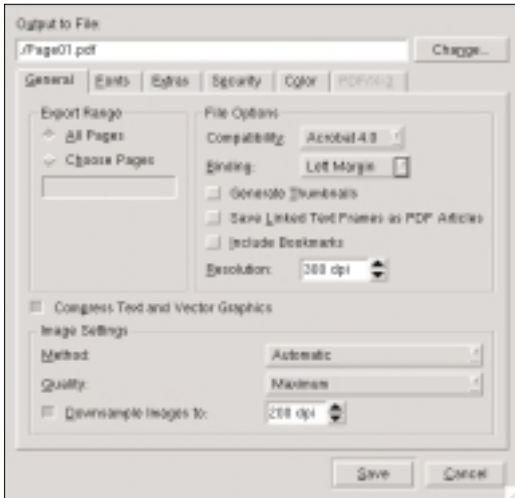


Figure 2: The Scribus PDF export menu features as many options as its commercial rivals.

if the images within it are low resolution, they will always be low resolution.

In the *Fonts* tab, select *Embed all Fonts*. This setting means that all typefaces used in the document will be included in the PDF – this feature not only means the document will print properly, it also avoids type licensing problems. (Remember, it’s actually against the law to supply fonts to a printer, though many will allow it, as they delete the font after one use.)

The *Extras* and *Security* tabs are of no interest to us here, so move directly to the *Color* tab (see Figure 3), which is vitally important for reasons discussed in Parts 1 and 2 of this series.

A quick recap of the “whats,” but not the “whys,” of working with color is that images destined for print are set in a CMYK colorspace, while screen images are set in the RGB colorspace. In the *General* drop-down menu, select *Output Intended for Print*.

Why Export Twice?

Some readers will doubtlessly be thinking that our scheme of importing PDFs and re-exporting them as spreads is needlessly complicated. Why not just work on spreads and export them in the right order? If you happen to be unlucky enough to be producing a newspaper on your own, by all means work directly on the spreads from within Scribus, but in a situation where many people are working on different pages, it simply makes life easier to see each page as a discrete object and pair them before going to press.

The *Color* tab also has some interesting advanced options. If you check *Use Custom Rendering Settings*, you can manually adjust the shape and frequency of halftone dots. This is part of the “screening” process, and if your printer gives you specific information, this menu is where you would enter it. In most cases, however, it’s safest just to ignore it, as the default settings are just fine.

This process should be repeated for all pages. Even double page spreads should be exported as single left and right facing pages for assembling later. Remember: just because

two pages sit beside one another in print doesn’t mean that’s how they’ll be printed; in fact, that will only happen with the very center pages.

An Imposing Figure

Assembling a newspaper is a tedious task – that’s why there are so many applications for automatically putting pages in the correct order – but these multi-thousand dollar applications are outside of the budget of this project, so how do we put our newspaper together?

Quite easily, though it can be a laborious experience. The first stage is to create a new page, large enough to accommodate two complete pages side by side.

Select *New* from the *File* menu. This allows us to create the new meta document that will contain our files (see Figure 4). In our case, the page depth is 386mm – exactly the same as a single page. However, the width is 594mm, which is exactly twice the width of a single page in our project. Scribus will automatically change the measurements from millimeters to points.

Next, select the image tool from the Scribus toolbar and draw two identical rectangles on the page that are the length of the page and half its width. Save this, as it is the basic double page spread template.

The next step is to import the pages, and this is where things

Sad News for Linux Publishers

Publishing is a volatile industry and, sadly, we have to report that the Twin Tier Times, which, as we mentioned in the first installment of this series, was the first newspaper to be produced in Linux, ceased publishing after only a few editions. What is important to note is that it wasn’t technical problems that sent the Twin Tier Times the way of Today and the Daily Herald; rather, it was the risky financial business that newspaper publishing is. Despite its short life, and we have no idea how well the newspaper read, it did prove that using Linux for professional publishing is now a real possibility. If you’re planning on using Linux and Scribus for a print project, publish with confidence.

can get tricky. The single page PDF files are re-imported back in – Scribus treats PDFs as ordinary images – and then re-exported as a PDF of the whole spread.

The difficulty here is the imposition, or the order in which pages go. Each page is paired with its opposite in the print run, so this order will change depending on the number of pages in the project. If you have a 48-page newspaper, page 48 and page 1 will sit together, with page 48 on the left of the leaf; page 1 on on the right. This is then folded in the middle and the right-facing page will become the front. Likewise, pages 2 and 47 will be paired, except, as this is on the reverse side of the leaf page, 2 will be on the left and page 47 will be on the right.

It is extremely easy to get confused at this point, so it’s best to work out the

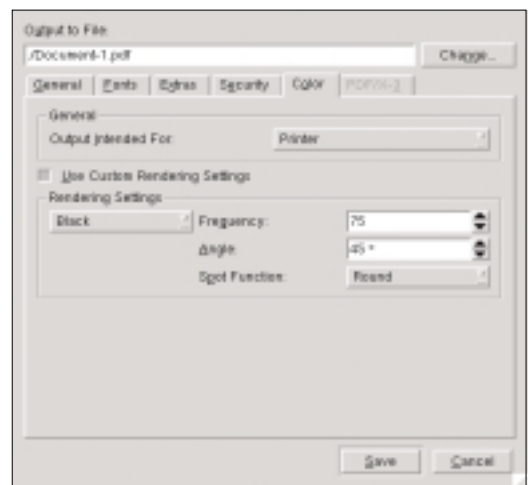


Figure 3: The Scribus PDF export menu’s Color tab has a range of sophisticated options.

Advertisement

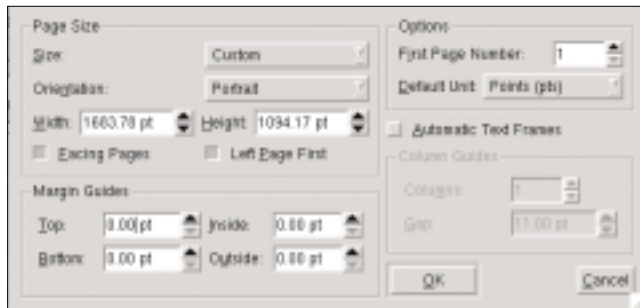


Figure 4: Creating a new meta page to contain the double page spreads.

page ordering on a piece of paper in advance of beginning. A single error will result in the pages being printed in the wrong places – something of a disaster for a newspaper. At the press, the pre-press team will assemble the double page spreads into a full newspaper, but they will be working using your spreads, so if you get it wrong, any incorrect spreads will have to be re-done, and this can be costly if the PDF pages have already been used to create a film from which to make the printed newspaper.

Each double page spread is then exported as a single large page and sent to the press for printing. Do that with all of the pages – in the correct order – and that's it, the newspaper is finished. Now simply repeat this every single day, and you're in the business of publishing a newspaper in Linux.

Taxing Exports?

PDFs, like HTML-exports, vary in quality from application to application. How does Scribus do for quality?

We sent our four page dummy newspaper to David Hunter, a newspaper

in Linux. If you hadn't told me, I wouldn't have known without checking the meta information. They were fine overall – no major problems,” said Hunter.

What about minor problems?

“Well, the images were a bit dark, but not terribly so. You did mention that your bitmap editor, GIMP, wasn't working in a CMYK colorspace, so I suspect that that's where the problem is located, rather than in Scribus itself. As I said, overall the pages were fine and would print without error.”

Conclusion

This series of articles has been an attempt to examine the complete process of publishing a newspaper in Linux using the Open Source publishing application Scribus. Part 1 [1] began with a description of some of the other desktop publishing tools, such as Quark XPress. I showed how to create a new document, how to manage color, and how to define page elements such as the nameplate. In Part 2 [2], I showed how to import a leading image and how to place text on the page. This final installment described

pre-press professional, to see how it compared to the Quark- and InDesign-produced PDFs he deals with on a regular basis.

What do you think?

“To be honest, I was surprised that they were produced

some final tasks, such as how to place the pages in order and how to export the finished document to a PDF. Of course, I couldn't include everything, but I hope I have given you a roadmap for how to get started on your own Scribus publishing project.

For those who have never tried Scribus, you must see it to believe it. Anyone who has struggled with the likes of Microsoft Publisher or Ragtime Solo will understand how inflexible non-industry standard DTP applications can be. Scribus isn't one of those applications. Instead, Scribus is a real challenger for Quark and Adobe — and you can bet Quark and Adobe are not ignoring the emergence of Scribus. But only time will tell whether Scribus and Linux gain some traction in the publishing industry. ■

What is a PDF?

Portable Document Format (PDF) is a file format developed by Adobe document display and printing. Based on Adobe's page description language, PostScript, a PDF file is a document containing any combination of text, graphics, and images in a device-independent and resolution-independent format. Crucially, PDFs are independent of the original application software, hardware, and operating system used to create them.

The PDF standard has been relatively popular for a number of years, but it really took off when Apple released the Mac OS X operating system. Mac OS X uses the PDF standard for all two-dimensional graphics, allowing any application on the Mac to make PDFs. The end-result has been a massive increase in the use of PDFs at the bureau and printer level. The upside of this is that PDF now is an accepted standard in the print industry. Any Linux application capable of producing PDFs can now be used to send material to press.

Common applications for viewing PDFs include Adobe's own Acrobat Reader and XPDF.

INFO

- [1] 1st part of the series: Jason Walsh: “A Linux Newspaper”, Linux Magazine, Issue #49 / December 2004, p53.
- [2] 2nd part of the series: Jason Walsh: “Typography and Text”, Linux Magazine, Issue #50 / January 2004, p50.
- [3] Scribus <http://www.scribus.org.uk/>

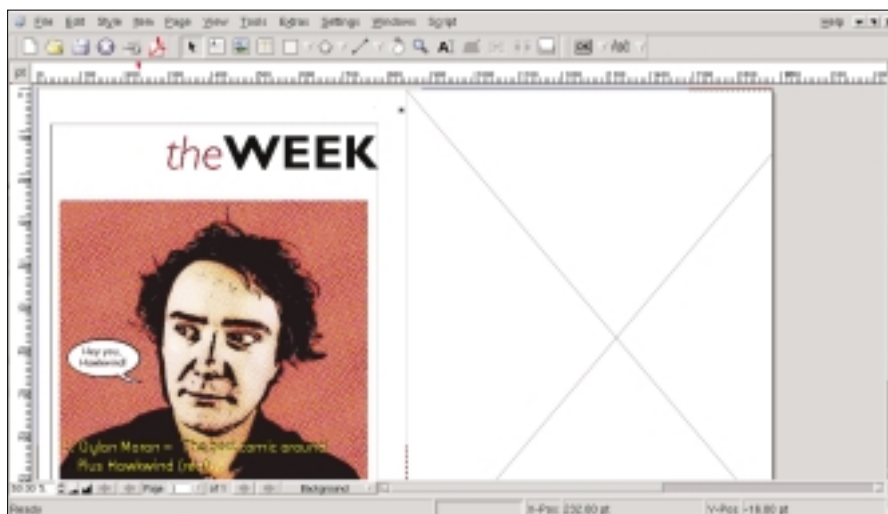


Figure 5: Importing two single pages onto a spread.