

Linux Layout with Scribus

A Linux Newspaper

In this first installment of a three-part series, former newspaper art director Jason Walsh looks at publishing in Linux. Jason will create a dummy newspaper using Scribus and other Linux applications, and have it tested in a real-world press situation.

BY JASON WALSH



The key to the increasing industrial penetration of Linux thus far has not been in the taking over of generalized desktop computer usage; instead it has been by systematically attacking niche areas of computing, notably web serving, scientific computing, and code development. It is indisputable that this has been a major success – Linux now effectively owns several key areas of computing.

One area where Linux has not been making major inroads is publishing, an industry that remains the almost exclusive preserve of the Apple Macintosh. After years of directing potential users to complex and unwieldy applications such as LayTeX, Linux finally has some excellent WYSIWYG desktop publishing tools.

In this series, we will produce four dummy newspaper pages under Linux and have them appraised by David Hunter at the pre-press department of the *Belfast News Letter*, the world's oldest continuously published daily English-language newspaper.

Programmers amongst you may wish to think of this mini-publication as the

“Hello World” of Linux desktop publishing.

A brief History of Publishing

Prior to the development of WYSIWYG desktop publishing systems, the production of newspapers and magazines was a complicated and arcane process involving specialist equipment such as linotype machines, process cameras, and bromide prints. The graphical user interface of the Apple Macintosh and its original killer app, Pagemaker, changed that forever. Finally layout was accessible to anyone, at least anyone who could afford a Mac and a copy of Pagemaker.

Pagemaker's dominance has long since dwindled to the point where it is a fringe product used largely by businesses to put together newsletters. The major application in the professional publishing world is Quark XPress. In the late-1990s a serious challenger to Quark's dominance arrived from the Adobe stable: InDesign. InDesign offers most of the features of Quark Xpress, as well as many new features not found in Quark, all with the Adobe user interface, which designers

are already familiar with from using Photoshop and Illustrator. Neither Quark nor InDesign is available for Linux, nor are they likely to be any time soon.

Introducing Scribus

All is not lost however. In typical open-source fashion, when an application was not available, a group of developers got together to produce a free alternative. That alternative is Scribus (see Figure 1).

Installing Scribus is reasonably straightforward – unfortunately, no universally compatible binary packages are available, but compiling is a simple enough matter:

```
./configure
make
make install
```

Then configure. Alternatively, RPMs are available for RedHat systems, while Gentoo and Debian are also supported.

Page Maker

For our project we are working with specific, predefined requirements. The end

result will be a number of pages from a tabloid newspaper. The first step is to define the shape and size of your page, along with some common parameters: margins and columns.

Figure 2 shows the Layout Properties dialog in Quark XPress 6 on Mac OS X. (*Layout > Layout Properties*). If you are converting to Scribus from Quark, step one is to copy down this basic data and enter it into Scribus. If you are creating a new layout, define equivalent layout properties as required.

Why not simply import the original Quark template into Scribus? Unfortunately, Scribus cannot import files from other layout applications. As the developers themselves say, "DTP file formats are very complex internally – probably the most complex on a PC. Creating import/export filters is a task far more complex than importing a spreadsheet or simpler word processing file formats."

This is an impediment, but it is not an uncommon problem. Adobe InDesign can open files from Quark 4 but not Quark 5 or 6. Quark cannot open any InDesign files whatsoever, nor can Quark Version 6 "save down" to Version 4. Showing a degree of forward thinking and openness that neither Quark nor Adobe display, Scribus uses XML standards for its native file format, meaning that it may be possible to open Scribus files in other layout applications in the future.

"Tabloid" in US-terms generally refers to a specific page size of 279.4mm x 431.8mm. In Europe, tabloid is a more general term referring to a roughly A3 newspaper page size, half the size of a broadsheet produced on the same press.

The New Document dialog in Scribus is where the basic page is initially created (see Figure 3). The page that we are creating will serve as a template for all subsequent front pages (and in fact, for all other pages, albeit in a slightly modified form). Our templates are very basic, simply defining the size and shape of the



Figure 1: Scribus running on Suse 9.1. Finally, professional-quality desktop publishing comes to Linux.

page and whether it will be left of right facing.

Our project is 297mm x 386mm; slightly squatter than most tabloids and roughly analogous with the size of the British *Independent* newspaper. This means we enter the page width as 297mm and height as 386mm.

Printing right to the edge of the page (called full-bleed in the industry) is common in magazines, but rare in newspapers as the presses are generally not capable of it. Even when it is possible, it is never called for on the front page. We need to define a printable area, and we do this by setting margin guides in the New Document dialog. These properties are defined by the press and differ from newspaper to newspaper. In our case, the margins are set up as: Top, 29mm; Bottom, 16mm; Inside 8mm; Outside 23mm.

The margins are set unevenly in order to create a rectangular box of 265mm x 340mm—this is the minimum printable area on each page—in the correct position.

On some pages, items can be outside this central rectangle; on others it would result in items not being printed. This is an issue of pagination and page imposition that is, again, set by the newspaper press rather than the designer, so we shall not be dealing with it in detail here.

Within this central rectangle, we will later define our main layout of five columns of text. Scribus handles columns in a slightly different manner from other DTP apps – more on that later – but the design is efficient and very usable.

The New Document dialog has a few other options: Automatic text frames fills the printable area with a text frame on every newly created page, handy for long document creation such as book design, but not relevant here.

The remaining options are related to imposition. Make sure you check the Facing Pages option, which calls for a layout in two-page spreads.

When the pages are printed, they are in pairs. The front and back pages are together and are folded in the middle and wrapped around the next set and so on. A complete newspaper is produced from several files with varying number of pages, rather than from a single file containing the entire newspaper. This design allows different pages to be produced at different times (and by different persons), and it stops the Scribus files from becoming unmanageably large.

Frame Based Layout

Some open-source commentators have complained about the Scribus interface, criticizing it as not suitable for the home user. (This is a strange criticism of a piece of software aimed at the professional user.) Both Quark XPress and Adobe InDesign doubtlessly seem

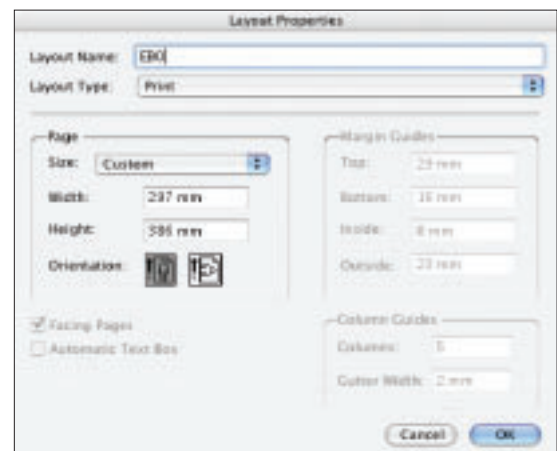


Figure 2: The Layout Properties menu in Quark XPress displaying the page defaults for the East Belfast Observer.

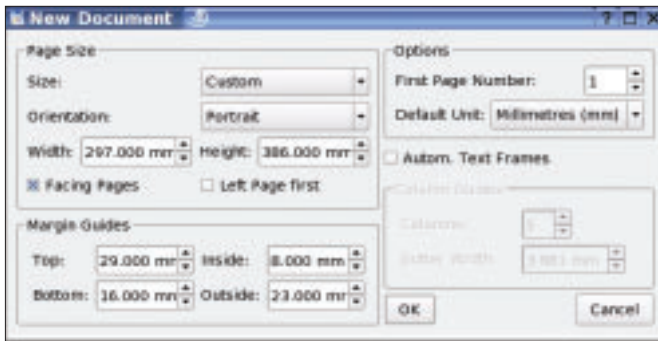


Figure 3: The New Document dialog in Scribus. In this dialog the basic format of the page is defined. Here we have used the EBO settings taken from the original Quark file.

counter-intuitive to amateurs, but publishing is a complex business – this is not word processing.

A greater problem facing Scribus is the fact that a stable, universally compatible binary is not available. Expecting designers and journalists to compile software from source is ridiculous and will be a serious impediment to Scribus' potential adoption in the industry. Believe it or not, there are hundreds of publications without IT departments; with a Mac-based system they simply don't need in-house IT support.

The principal objection to Scribus seems to be in its use of a dual modality: layout is frame based; that is, every object is contained within a frame and can be manipulated in two ways – as an object, or as content.

This method of working is common to both Quark and InDesign and is in fact a very efficient way of separating design from content. Furthermore, it is essential for Scribus to offer a comfortable environment to print professionals, be they designers, journalists or sub-editors, if it is to make serious inroads into publishing.

Once you become accustomed to it, this manner of working is quite productive: to place text on the page, you must create a text frame first, and then insert, or type, text into this frame. In order to place an image on the page, you create an image frame and then import the photo or graphic into this frame. The frames and the content that they contain can then be manipulated independently.

Color Management

Printing is as much of an art as a science, and it is perfectly possible to know everything about it and still face disaster

at the press. There are too many variables to be absolutely confident, and many of them cannot be controlled from the desktop. The grade and type of paper used, the quality and viscosity of ink, the type and age of the press, and the skill of the printers will

all have a massive effect on the final output.

One of the areas that is hardest to get right is reproduction of color, and although all of the issues listed above will have an impact, it is important to ensure that before you send files to press that they are of as high a quality as possible.

Color is a surprisingly complex subject. Built-in color management was one of the major advantages enjoyed by the Macintosh as a print production platform – and one of Linux's major disadvantages. Until now.

Color management is a technology used to calibrate the color of input devices, display devices, and output devices. The key principle is that of the "working space." This is a color reference that is not tied to any specific device, commonly "sRGB", a basic additive color profile suitable for most monitors. The problem with this is that the output device, be it the monitor or a printer, must make a transformation from sRGB to its native color space and the quality of this transformation will be dictated by the quality of the device driver.

A better method is the use of an International Color Consortium compatible color management system. The ICC defined an open standard for Color Matching Modules, preferably defined at OS level, and color profiles both for the devices and the working space.

Color management technology can be implemented in the operating system, in the API, or directly in an application. For years now, Mac OS has included Apple's own color management software, Color-Sync. Now Linux has a compatible

solution in the form of LittleCMS. Home users may not need to install LittleCMS, but for professional output it is a must.

LittleCMS includes standard ICC color profiles and thus with LittleCMS installed, PDF files produced in Scribus will be printed just as they appear on screen. Without it you'll be guessing how any given color will come out in print.

It's important to note that LittleCMS is not an application in its own right, rather it is a programmers' library that can be used in application development – Scribus is one of the applications that makes use of it. If you install it before installing Scribus, Scribus will automatically use it.

Sadly, in Linux, the only mainstream bitmap image editor that supports color management is Corel Photopaint. Hopefully the availability of LittleCMS will encourage GIMP developers to get on board sooner rather than later.

The most important aspect of using a color management system is remembering to activate it! When it comes to exporting our final PDFs, we'll go over this in detail, but for now have a look at *File > Export > Save Page as PDF*. In the dialog box that appears, choose the *color* tab. If you have LittleCMS installed, you will see a drop-down menu under PDF/X-3. This menu contains the press profiles – consult your printer and choose the relevant one.

In the case of this dummy, the correct profile is Euroscale Uncoated v2. This is because Euroscale is an attempt to pro-

Scribus in use

Despite Quark's dominance in the publishing industry some newspapers are moving to other solutions. The now defunct *Dublin Daily* used Adobe InDesign, as does Dublin's *Lucan Gazette* and the Condé Nast magazine, *Cosmo Girl*.

Until recently no publications were known to be using Scribus, however that has begun to change. The *Twin Tier Times*, a small weekly newspaper distributed in southern New York state and northern Pennsylvania has developed a workflow that not only incorporates Scribus, but is actually centered on it, using no other layout application. According to the Scribus team, two other newspapers are using Scribus for day-to-day production.



Figure 4: The East Belfast Observer's nameplate.

vide a generic press profile for Europe. North American and Japanese options are also available, Web/SWOP and Japan Color respectively, so if you live in America or Japan, choosing the relevant profile will result in improved results. *Uncoated* refers to the paper type; many magazines are printed on expensive coated paper, though newspapers are not.

In order to know why color management is essential, it is necessary to understand the basic principles of offset printing, as it differs significantly from inkjet printing.

Offset printing, or lithography, is the process by which newspapers are printed. Files are passed through a RIP, or Raster Image Processor, which converts the layout into a PostScript image. Each page becomes a single image, which is then output to film and transferred onto an aluminum plate.

This plate is fixed to a drum and rolled through water, which adheres to the negative areas of the image. Then the plate is rolled through ink, which adheres to

the positive areas; next, a drum covered with rubber is rolled over the plate, squeezing away the water and picking up the ink. This drum is then rolled over paper, transferring the ink onto the paper.

Color is printed by separating the colors into four single plates, one for each color used in the printing process. This process is called color separation. In order for this process to work correctly, color elements such as photographs or graphics must be in the CMYK colorspace. Screen images on a computer are usually in RGB format, Red, Green and Blue. These are the additive primary colors used to form secondary and tertiary colors with light.

With ink, it is necessary to use the subtractive primaries, Red, Yellow and Blue. In printing, these are substituted with

Magenta, Yellow and Cyan, which offers a wider range of printable colors. The order in which the colors are printed is Cyan first, then Magenta, and then Yellow – CMY. The “K” stands for Key, which is the registration color, Black. Without the final black plate, the result would be that “black” elements such as type, would be dark brown, purple, or green.

If you print an RGB file on an inkjet printer, some of the colors will be “out of gamut,” that is, outside of the range of printable colors. An annoyance, but not a disaster.

If you are outputting to a

Output

Before the RIP stage, we will be exporting our pages as PDF files. Adobe's PDF has become the standard format for exchanging print documents; if your PDF is properly created, it will print correctly on a wide range of output devices, including inkjet printers, digital presses, and four-color offset press.

CMYK press, it is therefore vital to appropriately map the RGB colors seen on-screen to colors that are printable in the CMYK gamut.

Moreover, if you send RGB files through the RIP, it will not recognize them as color images and will print them as a single film. The end result will be particularly murky greyscale images.

Using CMYK colors means you won't accidentally end up with a monochrome page; using color management means that the colors you see in your proofs will be just like the end result.

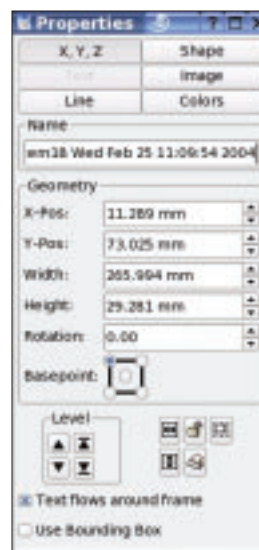


Figure 6: The Scribus Properties dialog, where you can adjust almost any aspect of an object.

Page Elements

Pages consist of three basic elements: text, graphics, and white space. The ratio of each will depend on what you

are designing; advertisements differ significantly from editorial content, and editorial pages will differ from publication to publication and even within publications.

With some notable exceptions, such as France's *Libération* and the UK's *Guardian*, newspaper design is uninspired. That is not to say that newspapers are necessarily badly designed, just that the criteria are different from other areas of graphic design.

The two key aspects of newspaper design are: ease of reading and ease of layout. The former is important because a newspaper must be digested quickly by a general readership, rather than a specialist one. Being consistent and conforming to certain basic rules will make the task of reading easier. The lat-

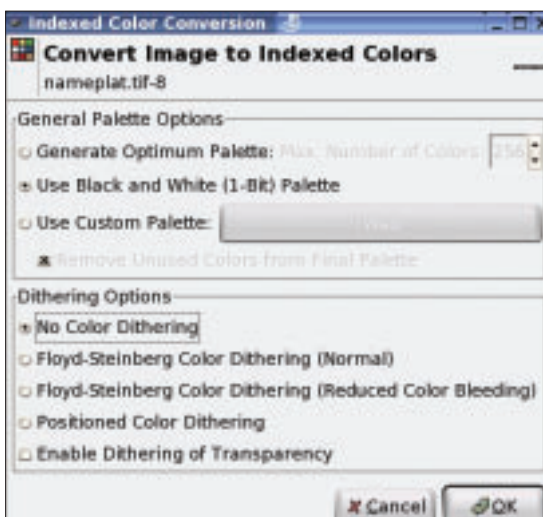


Figure 5: After exporting the nameplate from the drawing program – in our case Inkscape – open it in GIMP and change it into a 1-bit image by selecting *Image>Mode>Indexed* and choosing a 1-bit Palette with no dithering.

ter aspect is important because, unlike magazines, which generally have a monthly turnaround, newspapers tend to be daily and there is significantly increased pressure on the layout team.

For our dummy newspaper, the first element is the nameplate (see Figure 4). Often mistakenly referred to as the masthead, the nameplate is the name of the newspaper at the top of the front page. In this case, the typefaces used are Gill Sans Light Italic and Gill Sans Bold Condensed.

For this dummy newspaper, our nameplate is a high resolution TIFF file that was created in Inkscape, exported, and then converted into a 1-bit TIFF at a resolution of 600 dots per inch (see Figure 5). 600dpi is an incredibly high resolution, especially when you consider that the press we shall be using prints at around 180dpi, so why bother with such a massive resolution?

Files that use a 1-bit colorspace are black and white; not greyscale, but literally black and white. Generally speaking, 1-bit TIFFs are used for line art. The advantage in using a 1-bit colorspace is that the color of the lines (or in our case type) can be changed within the page layout application, so a single source file can be used for multiple variations of the same image. The downside of 1-bit images is that they degrade at low resolutions – even at 300dpi a 1-bit image will begin to display ragged edges.

To import the nameplate, select *Insert Picture*, the third icon in the toolbar, and draw a frame the entire width of the printable area between the two margins.

Next, select *File > Import > Get Picture*.

After selecting the appropriate file, select *Tools > Properties* to reach the Scribus Properties dialog (See Figure 6) and select *Image*. Select the *Scale to Frame Size* option, making sure that the proportional button is checked. The nameplate should now be the width of the newspaper front page, excluding the margins.

Above the nameplate are three boxes, which highlight content inside the newspaper. These boxes are empty image frames that have been colored using the *Tools > Properties > Colors* dialog. Directly below the nameplate are the date and issue number of the newspaper and its price. Both are created as text elements in Scribus in the normal manner. It is also possible to create the nameplate as a normal text frame within Scribus, but as it is an unchanging element, it is better to use a graphic file, lest a typo should creep in over time.

The next element to include is text. In terms of text entry, it is inefficient to type directly into Scribus. Even if you have a huge monitor, you'll spend a lot of time zooming in and out to read parts of the pages. Instead, prepare all text you wish to add to the file in a word processor like OpenOffice Writer, spell-check it, and then import it into Scribus.

At this point, you must save the text as plain text before importing it into Scribus. Scribus cannot import any form of styled text, be it Rich Text Format (RTF) or the native OpenOffice file format. This lack of support for formatted text may change soon – Scribus' developers are working on RTF importing. Support for RTF would give the user the ability to import text

styles, such as boldface, italics and so on, along with the text.

We'll be discussing how to work with text in detail next month, but for now let's just concentrate on getting some text into the layout.

Getting the text in is a simple business: select the *Insert Text Frame* button from the Scribus toolbar and draw a frame on the page, roughly where you want the text to go. Next, select *File > Import > Get Text*. The text from the chosen file will flow into the selected frame.

To edit the text, select *Edit Contents of Frame* from the toolbar and use the I-beam cursor to select the text as you would in any ordinary word processor. Alternatively, you can use the Story Editor (also on the toolbar) to edit the text.

Conclusion

This article introduced you to Scribus, a professional-quality Open Source layout application for Linux. In next month's issue, you'll learn more about managing text and graphics in the Scribus environment. ■

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*, *Variant*, *Mute*, the *Guardian* and many other newspapers and magazines on art, design and technology.



A Third, Fourth and Fifth-way

Initial impressions may suggest that there are relatively few desktop publishing packages on Linux, but that is not really the case. Perhaps they're buried under the mountain of command-based typesetting software such as TeX and the traditional UNIX utility, Troff, and its GNU derivative Groff, but good GUI DTP tools do exist for Linux.

The main commercial application is PageStream. Developed by Grasshopper LLC in the US, PageStream originated on the Amiga back in the 1980s; the Apple Macintosh's strong-arm lock on the publishing industry meant that it never gained currency in the professional arena. Instead PageStream found a niche making small

publications such as community group and church newsletters, nevertheless, it is a professional quality application and is available not only for the Amiga and Linux, but Windows, Mac OS X, and on two somewhat more exotic platforms: MorphOS and Amiga PowerPC.

In late 1999 Adobe released a beta-test version of FrameMaker for Linux. Never really popular on the Mac, FrameMaker is used for creating scientific and technical documents. Currently available only for Windows, Mac and Solaris/SPARC, but signifying Adobe's intention to confine FrameMaker to the technical publishing market, FrameMaker will no longer be developed for the Mac OS. What is

curious is why Adobe doesn't continue to develop FrameMaker for Linux – a market obsessed with technical documentation. If you're determined to use only open source software, two further options are Xclamation and Passepartout.

Xclamation, developed by Axene is part of the Axena office suite. Xclamation is a full-featured DTP app that uses the Motif toolkit and as such, should run on virtually any UNIX OS.

The final alternative is Passepartout. Where Scribus is a KDE application, Passepartout is a native GNOME app, offering those who prefer the GNOME GUI and toolkit a native layout application.