



Sudoku from PDF::API2

brian d foy
brian.d.foy@gmail.com

The PDF::API2 module is a hook into Adobe's Portable Document Format (PDF), and I have to know quite a bit about PostScript to actually use it. When Eric Maki sent me his program to generate Sudoku puzzles, I wanted to use PDF::API2 to turn them into something a bit nicer than plain text.

When I started to play with PDF::API2, I had to do a lot of digging to find the barest of examples. The documentation is probably sufficient for those who already know what they are doing, but they don't help the newcomer. Indeed, most of the stuff I found through Google complained about the lack of examples and docs.

PostScript

Is it odd to start the article with a postscript? The formal name, Adobe® PostScript®, should appear just as I have it in this sentence even though it's become a common word for programmers. Rather than type that out every time, I'll just use "PS" or "PDF".

Most people know about PS from printing documents, or creating documents to print. PS is actually another programmer language, and resembles FORTH. If you wanted to, you could use it to compute numbers such as π , although Damian Conway beat you to it.

Descendent from PS is its simpler cousin, PDF loses some of the programming goodness and focuses on the description of text, graphics, and embedded images in such a way that they display in the same way everywhere. There are many other differences, but they aren't important for this article. Just know that you have a crippled programming language that deals with putting things on the page.

The PDF::API2 module is the Perl interface to PDF, and its not much more than that. It's low-level, so the more you already know about PS or PDF, the easier time you'll have with it.

Creating a Page

Since PDF deals with documents, I need a document to do anything. I want to create a Sudoku grid, and then put some clues into it. As a shell, I create the object, set up some the options for the font to use and the size of the visible area that I want. I'll work with a normal page size. Finally, once I've done everything, I'll save the file.

```
#!/usr/bin/perl

use PDF::API2;

my $pdfer = PDF::API2->new;
my $font = $pdf->corefont('Helvetica-Bold');
$pdfer->mediabox( 595, 842 );
# ... stuff
$pdfer->saveas( $filename );
```

Now I have an object, `$pdfer`, that I can use to play with PDF. That's nice, but to do anything that people can see, I need a page too, That's going to be my canvas, really. Every time I call the page method, I get another page in my document.

```
my $page = $pdfer->page;
```

Once I have a page, I need a graphics object so I can play with lines and such. I'm going to need a square outline, some lines in the box to show the Sudoku grids, and then some lines to create the individual cells.

```
my $gfx = $page->gfx;
```

Now I'm ready to draw some lines, and this reminds me of my very first programming experiences when drawing a line on the screen was really high tech.

The Box

I want the outline of my puzzle grid to have a thick border. I can set the line width with my `$gfx` object. With a little experimenting, I found that 3 is a nice thickness, I don't have to use a whole number, either.