

work to make the tables look exactly like PDF::Writer's output, as Prawn's default output is very simple. However, even with this 'fancy' output, Prawn is still quite fast.

I'll leave running the [benchmarks](#) to individual users, but with a CSV of 1833 records, on my machine you can expect it to take a few seconds (3-4), which is about 25x faster than what I get for PDF::Writer (between 90-100s). Prawn scales linearly, so you can expect to process larger datasets and count the execution time in the seconds and minutes, rather than (as we unfortunately have seen with PDF::Writer) the hours. Some optimizations can be done to speed things up even more, but you will see orders of magnitude in improvement for table generation speed on Prawn.

Getting Fancy with Prawn Tables

Here's a quick sample of some other fun stuff you can do with Prawn's table drawing support that aren't necessarily tied to making things look similar to PDF::Writer:

id	name	phone	street	town	state
1	Inky	555-000-1234	Druary Lane	Union City	CT
2	Blinky	525-0529-123	Apple Street	Robot Town	NJ
3	Clyde	247-219-4820	Sandbox Hill	Alvin's Landing	PA
4	Pacman	283-102-8293	Rat Avenue	Southford	VT
5	Mrs. Pacman	214-892-1892	Conch Walk	New York	NY

Col A	Col B
This is Of tables	A Test Drawn 5 side
By side	and stuff

1	2	3
4	5	6
7	8	9

This free flowing textbox shows how you can use Prawn's cells outside of a table with ease. Think of a 'cell' as simply a limited purpose bounding box that is meant for laying out blocks of text and optionally placing a border around it.

This document demonstrates a number of Prawn's table features

Though 'fancy' may be a bit of an overstatement, it shows that Prawn's Table support has some flexibility to it. As we go on, I'm sure we'll figure out more ways to make it do the things we need, but for now we've covered the basics. If you're interested in how this all works, take a look at the [example code](#) that generated the above PDF.

Other New Stuff

I've also brought a few of Ruport's features into Prawn, to ease integration in the coming weeks. These features will be familiar to Ruport users, and worth taking a look at for new Prawn users:

- `horizontal_rule()`
- `horizontal_line(x1,x2)`
- `vertical_line_at(y,x1,x2)`
- `pad(y)`
- `pad_top(y)`
- `pad_bottom(y)`
- `move_up(y)`
- `move_down(y)`

These features aren't going to win Prawn any awards, but it cleans things up a little to use these helpers when they're appropriate.

In addition to these new features, a couple bugs were fixed in Permian, one to do with text height (which was mentioned in the last post here), and another to do with the way rectangles were being drawn.

On the Horizon

Even though I'm announcing Permian Prawn today, it was actually tagged, documented, and ready for use almost two weeks ago! Though not a ton of development happened on my end in the last week or so due to me taking some time off, ideas have been brewing and others have hacked together a bunch of different stuff.

Two different Rails plugins are under development, Steve Quinlan's [prawn_helper](#) and thorny_sun's [prawn_view](#). I haven't really had a chance to look into either project, but it's pretty exciting to think that we're already getting this kind of interest before an official release. Rails users will want to be sure to check these out. Between these two plugins and Ruport/Rails, Prawn will have much love on Rails apparently, and it'd be nice if we could find a way to combine efforts on these things...

In other news, our m17n support is slowly but surely improving thanks to the efforts of James Healy. I've been pulling these patches upstream as they come together, but it'd be great if we started getting some practical testing on these things from people who have a need for UTF-8 PDFs. Most of our testing so far has been somewhat trivial.

It seems like Prawn is in need of a 'real' release soon. I'm not promising an exact date, but I think we can look forward to a Prawn 0.1.0 gem before the end of July. All that really needs to be done is some work on text alignment and image embedding before I can package up a basic early beta version of Prawn for everyday use. With luck, that'll go smoothly and folks can start kicking the tires and let me know what they think.

Until then, there are still ways the adventurous can help out.

Get Involved!

As much as this self-promotion is starting to grind on me, I feel like the more eyes on this project, the better it'll be. Since my efforts on this project have been generously funded by donations from the Ruby community, I want to make sure that people have a voice in how the project turns out.

Pretty much all the relevant resources can be found through [Github](#), and if you have trouble finding anything or just want to get in touch, feel free to use the [mailing list](#).

If you happen to be near [Edmond, Oklahoma](#) tomorrow (2008.07.10), or near [Hartford, CT](#) on July 28th, I'll be doing Prawn talks. I'll try to get the materials out to the mailing list, but meeting some folks in person would be great, too.

That's pretty much all for now, I hope to see more patches, more discussion, and more PDF awesomeness between now and the first release.

Happy Hacking!

Comments

Name	Website	Save?
<input type="text"/>	<input type="text"/>	<input type="checkbox"/>
Email Address (required)		
<input type="text"/>		
Comments		
<input type="text"/>		
<input type="button" value="Preview"/> <input type="button" value="Post"/>		

Looking good sir, looking very good.

[Daniel Berger](#) | [July 9, 2008 08:11 PM](#)

Hey Greg, great job! Thanks for the update. Just a question. I looks like much of the text is evenly spaced and in some cases there appear to be extra large spaces between some characters, such as between the P and the r in Prawn. Are you doing any kerning to make the text closer in cases where it makes sense to do so? Does PDF have kerning features built-in, or do you have to do that yourself.

[Carl Youngblood](#) | [July 10, 2008 07:49 AM](#)

@Carl,

Check out the kerning stuff on mdaines/prawn, we'll bring it upstream when we can.

[Gregory Brown](#) | [July 10, 2008 07:54 AM](#)

Awesome work! Thanks.

[Steven Hansen](#) | [July 10, 2008 07:56 AM](#)

This is much awesomeness. I'm following with much interest and am hoping that I can pitch in and help out.

[Eugene Chan](#) | [July 18, 2008 12:35 PM](#)

The best way to help right now is to file bug reports. I just set a Permian-1 tag which is a set of incremental improvements and bug-fixes. Since this integrates code from other branches and also changes the way Prawn works, I really need help shaking out the issues.

Details here:

http://groups.google.com/group/prawn-ruby/browse_thread/thread/576e7b0a47bdea36

[Gregory Brown](#) | [July 18, 2008 12:42 PM](#)

Awesome work!

is there an easy way to use this library to print a header with an underline? I can print text but can't work out what position to draw the line at.

Also, is there a way to make a list of dot points? Similar to etc. in html?

Looks like this library is going to be the pdf/writer killer!

[Anko](#) | [July 19, 2008 11:33 PM](#)

@Anko. no underline support yet but expect that very soon. Not sure whether I'll support a ul style bullet list, though you could build one with circles and absolutely positioned text.

[Gregory Brown](#) | [July 20, 2008 10:08 AM](#)

[Contact Us](#) | [Advertise with Us](#) | [Privacy Policy](#) | [Press Center](#) | [Jobs](#)

Weblog authors are solely responsible for the content and accuracy of their weblogs, including opinions they express, and O'Reilly Media, Inc., disclaims any and all liability for that content, its accuracy, and opinions it may contain.



This work is licensed under a [Creative Commons License](#).

For problems or assistance with this site, email help@oreillynet.com | (707) 827-7000 / (800) 998-9938