

**Prawn: Fast, Nimble PDF Generation For Ruby**Installable via **RubyGems** : `gem install prawn`[Docs](#)[Bugs](#)[Community](#)[Source](#)

## Building printable documents doesn't have to be hard

If you've ever needed to produce PDF documents before, in Ruby or another language, you probably know how much it can suck. **Prawn takes the pain out of generating beautiful printable documents**, while still remaining fast, tiny and nimble. It is also named after a majestic sea creature, and that has to count for something.

## The features you need, without all the complexity

In addition to being the **fastest pure Ruby PDF generation library**, Prawn has features that might prevent you from hating your job. The samples below give a taste of what Prawn based programs looks like, **click the code to reveal the PDF it generates**.

### -- Built in support for UTF-8

Internationalized text in Prawn is as simple as providing UTF-8 strings for it to render, assuming you've got a Unicode aware TTF font handy. For those who are running on Ruby 1.9, any encoding that can be converted to UTF-8 will work out of the box!

```
Prawn::Document.generate("utf8.pdf") do
  font "#{Prawn::BASEDIR}/data/fonts/DejaVuSans.ttf"
  text "ύαλον φαγείν δύναμαλ· τουτο ού με βλάπτει." * 20
end
```

### -- Easy image embedding

Prawn makes embedding JPEG and PNG images a breeze. With support for alpha transparency, easy positioning and scaling of images, you'll have no problem including all the graphics you need in your documents.

```
Prawn::Document.generate("image.pdf", :page_layout => :landscape) do
  pigs = "#{Prawn::BASEDIR}/data/images/pigs.jpg"
  image pigs, :at => [50,450], :width => 450

  dice = "#{Prawn::BASEDIR}/data/images/dice.png"
  image dice, :at => [50, 450], :scale => 0.75
end
```

### -- Flexible table drawing

Prawn has built in support for rendering text in the form of tables, providing basic reporting functionality. This lets users focus on customizing their documents rather than forcing them to write a ton of low level graphics drawing code.

```
Prawn::Document.generate("fancy_table.pdf") do

  data = [[["Gregory Brown", "gregory.t.brown@gmail.com"],
           ["James Healy", "jimmy@deefa.com"],
           ["Ross Perot", "ross@mailinator.com"],
           ["Al Gore", "al@mailinator.com"],
           ["Ralph Nader", "ralph@mailinator.com"]]]

  table data,
    :position => :center,
    :headers => ["Name", "Email"],
    :row_colors => ["ffffff", "ffff00"],
    :vertical_padding => 5,
    :horizontal_padding => 3
end
```

## -- Simplified content positioning

Anyone who has done work with a low level graphics engine knows that doing coordinate math isn't fun. Prawn simplifies this by allowing you to box off a sub-section of the document and treat it as its own mini-coordinate space. This means that all positioning is relative, making it easy to move things around your document while keeping your code clean. Text can also be flowed within these sectioned off bounding boxes, so this makes it trivial to generate columns of text on the fly.

```
Prawn::Document.generate("bounding_boxes.pdf") do

  bounding_box [100,600], :width => 200 do
    text "The rain in spain falls mainly on the plains " * 5
    stroke do
      line bounds.top_left, bounds.top_right
      line bounds.bottom_left, bounds.bottom_right
    end
  end

  bounding_box [100,500], :width => 200, :height => 200 do
    stroke do
      circle_at [100,100], :radius => 100
      line bounds.top_left, bounds.bottom_right
      line bounds.top_right, bounds.bottom_left
    end
  end

  bounding_box [50,150], :width => 100, :height => 100 do
    stroke_rectangle bounds.top_left, bounds.width, bounds.height
  end
end
```

## -- And loads more to come

***Prawn is currently alpha-level software under active development.*** New features are cropping up every day, and we expect to see loads of cool things in the near future including integration with the popular **Ruby Reports** project. For now, you can keep an eye on the latest new features by checking out the **examples** distributed with the source, which include the code samples shown above and much, much more!

## Supported by the Ruby Community

The development on Prawn is in part made possible through donations from the community to Gregory Brown's **Ruby Mendicant** project, but quickly outgrew its humble beginnings.

Since the project began in April 2008, the project has seen contributions in the form of code, bug reports, and feature requests from a whole bunch of Ruby hackers. You can find many of their contributions by checking out the **Github network graph** for the project. Although several have contributed patches to Prawn, special thanks goes out to **James Healy** and **Michael Daines** for being instrumental to the forward development of the library.

Many people have expressed interest in using Prawn within their Rails applications, and for this purpose, you might be interested in checking out yet another community contribution, thorny\_sun's **Prawnto Rails** plugin. Although this is not officially part of Prawn, we'll do what we can to make sure this plugin continues to work as things move forward

***Please join us in the development of Prawn so that it can become the library of choice for PDF generation in Ruby.*** You can start by getting in touch with us on the **mailing list** or stopping by to chat in the **#prawn** channel on irc.freenode.net. There is lots left to be done, and we could use your help!

## What are you waiting for?

It's time to generate some PDFs. Get Prawn via RubyGems with `gem install prawn` or clone us at `git://github.com/sandal/prawn.git`