
Prawn: Fast, Nimble PDF Generation For Ruby



Prawn is a pure Ruby PDF generation library that provides a lot of great functionality while trying to remain simple and reasonably performant. Here are some of the important features we provide:

- Vector drawing support, including lines, polygons, curves, ellipses, etc.
- Extensive text rendering support, including flowing text and limited inline formatting options.
- Support for both PDF builtin fonts as well as embedded TrueType fonts
- A variety of low level tools for basic layout needs, including a simple grid system
- PNG and JPG image embedding, with flexible scaling options
- Security features including encryption and password protection
- Tools for rendering repeatable content (i.e headers, footers, and page numbers)
- Comprehensive internationalization features, including full support for UTF-8 based fonts, right-to-left text rendering, fallback font support, and extension points for customizable text wrapping.
- Support for PDF outlines for document navigation
- Low level PDF features, allowing users to create custom extensions by dropping down all the way to the PDF object tree layer. (Mostly useful to those with knowledge of the PDF specification)
- Lots of other stuff!

Should You Use Prawn?

If you are looking for a highly flexible PDF document generation system, Prawn might be the tool for you. It is not a reporting tool or a publishing toolchain, though it could be fairly easily used to build those things.

One thing Prawn is not, and will never be, is an HTML to PDF generator. For those needs, consider looking into FlyingSaucer via JRuby, or one of the WebKit based tools, like Wicked or PDFKit. We do have basic support for inline styling but it is limited to a very small subset of functionality and is not suitable for rendering rich HTML documents.

Supported Ruby Versions and Implementations

Because Prawn is pure Ruby and all of its runtime dependencies are maintained by us, it should work pretty much anywhere. We officially support all Ruby versions supported by Ruby Core Team and JRuby

versions of matching Ruby version. However we will accept patches to fix problems on other Ruby platforms if they aren't too invasive.

Installing Prawn

Prawn is distributed via RubyGems, and can be installed the usual way that you install gems: by simply typing `gem install prawn` on the command line.

You can also install from git if you'd like, the *master* branch contains the latest developments. We're trying to keep *master* branch in working order but you may encounter some rough edges and fresh bugs along with bugfixes. We encourage you to try *master* branch with your application.

Hello World!

If the following code runs and produces a working PDF file, you've successfully installed Prawn.

```
1 require "prawn"
2
3 Prawn::Document.generate("hello.pdf") do
4   text "Hello World!"
5 end
```

Of course, you'll probably want to do more interesting things than that...

Manual

The manual is a series of examples that demonstrate use of the wide range of features Prawn provides. You can get a generated version of the latest released Prawn version on the Prawn website. The examples themselves can be found in the `manual` directory in this repository.

Please note that while the manual is a great introduction and guide to Prawn it's not exhaustive. Please refer to API docs for more complete information on what Prawn provides and how to use it.

To build the manual, here's what you need to do:

1. Clone the repository
2. Run `gem install -g`
3. Run `rake manual`, which will generate *manual.pdf* in the project root

Release Policies

We're trying to not break things unnecessarily but we don't formally follow Semantic Versioning. The reason is that we release a number of experimental APIs. We don't make any promises on their stability. You can assume the stable portion of the API follows Semantic Versioning.

Also note that bug fixes can change behaviour. We don't consider that to be a breaking change for the purposes of versioning. Please test your applications after updating Prawn.

Be sure to read the release notes in CHANGELOG.md each time we cut a new release, and lock your gems accordingly.

Support

The easiest way to get help with Prawn is to post a message to our Discussions.

Feel free to post any Prawn related question there, our community is very responsive and will be happy to help you figure out how to use Prawn, or help you determine whether it's the right tool for the task you are working on.

Please make your posts as specific as possible, including code samples and output where relevant. Do not post any information that should not be shared publicly, and be sure to reduce your example code as much as possible so that those who are responding to your question can more easily see what the issue might be.

Code of Conduct

Prawn adheres to the Contributor Covenant. Unacceptable behavior can be reported to conduct@prawnpdf.org which is monitored by the core team.

Contributing

If you've found a bug or want to submit a patch, please enter a ticket into our GitHub tracker.

We strongly encourage bug reports to come with failing tests or at least a reduced example that demonstrates the problem. Similarly, patches should include tests, API documentation, and an update to the manual where relevant. Feel free to send a pull request early though, if you just want some feedback or a code review before preparing your code to be merged.

If you are unsure about whether or not you've found a bug, or want to check to see whether we'd be interested in the feature you want to add before you start working on it, feel free to post to our mailing list.

You can run our test suite in a few different ways:

1. Running `rake` will run the entire test suite excluding any unresolved issues
2. Running `rspec` will run the entire test suite including unresolved issues
3. Running `rspec -t unresolved` will run *only* unresolved issues
4. Running `rspec -t issue:NUMBER` will run the tests for a specific issue

These filters make it possible for us to add failing test cases for bugs that are currently being researched or worked on, without breaking the typical full suite run.

Maintenance team

Prawn has always been heavily dependent on community contributions, with dozens of people contributing code over the years. In that sense, the lines have blurred to the point where we no longer have a strong distinction between core developers and contributors.

That said, there are a few folks who have been responsible for cutting releases, merging important pull requests, and making major decisions about the overall direction of the project.

Current maintainers

These are the folks to contact if you have a maintenance-related issue with Prawn:

- Alexander Mankuta (PointlessOne)

Inactive maintainers

These folks have helped out in a maintenance role in the past, but are no longer actively involved in the project:

- Gregory Brown (practicngruby)
- Brad Ediger (bradediger)
- James Healy (yob)
- Daniel Nelson (Bluejade)
- Jonathan Greenberg (jonsgreen)
- Jamis Buck (jamis)
- Evan Sharp (PacketMonkey)

License

Prawn is released under a slightly modified form of the License of Ruby, allowing you to choose between Matz's terms, the GPLv2, or GPLv3. For details, please see the LICENSE, GPLv2, and GPLv3 files.

If you contribute to Prawn, you will retain your own copyright but must agree to license your code under the same terms as the project itself.

History

Prawn was originally developed by Gregory Brown, under the auspices of the Ruby Mendicant Project, a grassroots initiative in which the Ruby community collectively provided funding so that Gregory could take several months off from work to focus on this project.

Over the last several years, we've received code contributions from dozens of people, which is amazing considering the low-level nature of this project. You can find the full list of folks who have at least one patch accepted to Prawn on [GitHub Contributors page](#).

After a long period of inactivity, Prawn reached its 1.0 milestone in 2014 thanks to some modest funding provided to Gregory by Madrisk, Inc. (Brad Ediger's company).