This is a single string text.
With two paragraphs.

This is a *single string* **MD** text.
It should have two paragraphs.

This is a *single* ***string*** HTML text.
With two paragraphs.

This is an array.
Of single string texts. Two paragraphs.

This is an **array**
Of single *string* MD texts, two paragraphs.

This is an **array**
of single *string* HTML texts. Two paragraphs.

This is an array
of single string hashes.
With **some** *markup**!***

This is a single

string text. With

two paragra-phs.

This is a *single*

string **MD** text.

It should have

two paragra-phs.

This is a *single*

string HTML text.

Two paragra-phs.

This is an array

Of single string

texts. Two paragra-

phs.

This is an **array**

Of single *string*

MD texts. Two

paragra-phs.

This is an **array**

Of single *string*

HTML texts. Two

paragra-phs.

This is an array

Of single string

hashes. With **some**

markup ***!***

Paragraphs with default characteristics

Sed ut perspiciatis, unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam eaque ipsa, quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt, explicabo. Nemo enim ipsam voluptatem, quia voluptas sit, aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos, qui ratione dolor sit, voluptatem sequi nesciunt, neque porro quisquam est, qui dolorem ipsum, quia amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt, ut labore et dolore magnam aliquam quaerat voluptatem.

Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit, qui in ea voluptate velit esse, quam nihil molestiae consequatur, vel illum, qui dolorem eum fugiat, quo voluptas nulla pariatur?

At vero eos et accusamus et iusto odio dignissimos ducimus, qui blanditiis praesentium voluptatum deleniti atque corrupti, quos dolores et quas molestias excepturi sint, obcaecati cupiditate non provident, similique sunt in culpa, qui officia deserunt mollitia animi, id est laborum et dolorum fuga.

Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio, cumque nihil impedit, quo minus id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus.

Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat.

Paragraphs with block style (no indent, vertical space)

Sed ut perspiciatis, unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam eaque ipsa, quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt, explicabo. Nemo enim ipsam voluptatem, quia voluptas sit, aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos, qui ratione dolor sit, voluptatem sequi nesciunt, neque porro quisquam est, qui dolorem ipsum, quia amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt, ut labore et dolore magnam aliquam quaerat voluptatem.

Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit, qui in ea voluptate velit esse, quam nihil molestiae consequatur, vel illum, qui dolorem eum fugiat, quo voluptas nulla pariatur?

At vero eos et accusamus et iusto odio dignissimos ducimus, qui blanditiis praesentium voluptatum deleniti atque corrupti, quos dolores et quas molestias excepturi sint, obcaecati cupiditate non provident, similique sunt in culpa, qui officia deserunt mollitia animi, id est laborum et dolorum fuga.

Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio, cumque nihil impedit, quo minus id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus.

Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat.

Example of Markdown that needs to be supported in document text blocks. There is no need to support this within tables, although it would be a "nice" feature.

Firstly just some simple styling: *italics*, **bold** and ***both***.

There should also be support for *alternative italics*

Then a bulleted list:

- Unordered item
- Another unordered item

And a numbered list:

1. Item one
2. Item two

We will need a heading

And a subheading

Finally we'll need some external links.

Show that another link on the same page works.

Show some inserted text and underlined text that display underlines. Show some ~~deleted~~ text, ~~strike-out~~ text, and ~~s'd out~~ text that show line-throughs. More than ~~one at a time~~ are possible via style attribute, also via ~~nested tags~~ and ~~nested tags~~.

Then we need some styling features in tables as shown in the table below. There is no need to support this in text blocks, although it would be a nice feature (colored text is already available in text blocks using its options).

This is some red text	This is some green text
This is a red cross: ✘.	This is a green tick: ✔.
This is a red cross substitute: ✘.	This is a green tick substitute: ✔.
Plain old text	More plain text

PDF::Builder release 3.028

A Perl library to create and modify PDF (Portable Document Format) files

What is it?

PDF::Builder is a **fork** of the popular PDF::API2* Perl library. It provides a library of modules and functions so that a PDF file (document) may be built and maintained from Perl programs (it can also read in, modify, and write back out existing PDF files). It is not a WYSIWYG editor; nor is it a canned utility or converter. It does *not* have a GUI or command line interface -- it is driven by *your* Perl program. It is a set of **building blocks** (methods) with which you can perform a wide variety of operations, ranging from basic operations (such as selecting a font face), to defining an entire page at a time in the document (using a large subset of either Markdown or HTML markup languages). You can call it from arbitrary Perl programs, which may even create content on-the-fly (or read it in from other sources). Quite a few code examples are provided, to help you to get started with the process of creating a PDF document. Many enhancements are in the pipeline to make PDF::Builder even more powerful and versatile.

*Note that PDF::Builder is **not** built on PDF::API2, and does **not** require that it be installed. The two libraries are completely independent of each other and one will not interfere with the other if both are installed. However, you should *not* try mixing the two libraries within one running program, as they still share many routine names!

The Home Page includes Documentation and Examples.

This archive contains the distribution PDF::Builder. See **Changes** file for the version and list of changes from the previous release.

Obtaining and Installing the Package

The installable Perl package may be obtained from "https://metacpan.org/pod/PDF::Builder", or via a CPAN installer package. If you install this product, only the run-time modules will be installed. Download the full `.tar.gz` file and unpack it (uncompress, then extract directory -- hint: on Windows, **7-Zip File Manager** is an excellent tool) to get utilities, test buckets, example usage, etc.

Alternatively, you can obtain the full source files from "https://github.com/PhilterPaper/Perl-PDF-Builder", where the ticket list (bugs, enhancement requests, etc.) is also kept. Unlike the installable CPAN version, this will have to be manually installed (copy files; there are no XS compiles at this time).

Other than an installer for standard CPAN packages (such as 'cpan' on Strawberry Perl for Windows), no other tools or manually-installed prereqs are needed (worst case, you can unpack the `.tar.gz` file and copy files into place yourself!). Currently there are no compiles and links (Perl extensions) done during the install process, only copying of .pm Perl module files.

A package installer such as "cpan" (included with Strawberry Perl and some other systems) can retrieve the package, unpack and copy files, and run installation tests without manual intervention. Not only PDF::Builder itself, but any needed prerequisites, may be quickly installed in this manner. Some Perl distributions (e.g., ActiveState) may repackage PDF::Builder and prerequisites into their own install format, as may Linux distributions such as Red Hat or SUSE. Finally, it is possible to copy files directly from the GitHub repository to your system, and manually run the "t" (installation) tests, all without going through a `.tar.gz` CPAN package. There are many possibilities.

Note that there are several "optional" libraries (Perl modules) used to extend and improve PDF::Builder. Read about the list of optional libraries in PDF::Builder::Docs, and decide whether or not you want to install any of them. By default, **none** are installed.

Requirements

Perl

Perl 5.28 or higher. It will likely run on somewhat earlier versions, but the CPAN installer may refuse to install it. The reason this version was chosen was so that LTS (Long Term Support) versions of Perl going back about 6 years are officially supported (by PDF::Builder), and older versions are not supported. The intent is to not waste time and effort trying to fix bugs which are an artifact of old Perl releases. Frankly, if you're running a Perl version more than 6 years old, you should be thinking about upgrading.

Usually about once a year the minimum level is bumped up, but this depends on whether Strawberry releases the newest Perl level. As Strawberry Perl releases new Perl levels, usually on an annual basis, we intend to bump up our required minimum Perl level (even-numbered production releases), to keep support for the last 6 calendar years of Perl releases, dropping older ones.

Note that we have been informed that enough users want to run PDF::Builder at older levels that **for the time being** we will stay at **5.28**. If we make use of newer Perl constructs and operators in the code, however, and PDF::Builder no longer works with 5.28, we will have to raise the minimum level.

Older Perls

If you MUST install on an older (pre 5.28) Perl, you can try the following for Strawberry Perl (Windows). NO PROMISES! Something similar MAY work for other OS's and Perl installations:

1. Unpack installation file (`.tar.gz`, via a utility such as 7-Zip) into a directory, and cd to that directory
2. Edit META.json and change 5.028000 to 5.016000 or whatever level desired
3. Edit META.yml and change 5.028000 to 5.016000 or whatever level desired
4. Edit Makefile.PL and change `use 5.028000;` to `use 5.016000;`, change `$PERL_version` from `5.028000` to `5.016000`
5. `cpan .`

Note that some Perl installers MAY have a means to override or suppress the Perl version check. That may be easier to use. Or, you may have to repack the edited directory back into a `.tar.gz` installable. YMMV.

If all goes well, PDF::Builder will be installed on your system. Whether or not it will RUN is another matter. Please do NOT open a bug report (ticket) unless you're absolutely sure that the problem is not a result of using an old Perl release, e.g., PDF::Builder is using a feature introduced in Perl 5.022 and you're trying to run Perl 5.012!

Libraries used

These libraries are available from CPAN.

REQUIRED

These libraries should be automatically installed...

- Compress::Zlib
- Font::TTF
- Test::Exception (needed only for installation tests)
- Test::Memory::Cycle (needed only for installation tests)

OPTIONAL

These libraries are *recommended* for improved functionality and performance. The default behavior is

not to attempt to install them during PDF::Builder installation, in order to speed up the testing process and not clutter up matters, especially if an optional package fails to install. You can always manually install them later, if you desire to make use of their added functionality.

- Perl::Critic (1.150 or higher, need if running tools/1_pc.pl)
- Graphics::TIFF (19 or higher, recommended if using TIFF image functions)
- Image::PNG::Libpng (0.57 or higher, recommended for enhanced PNG image function processing)
- HarfBuzz::Shaper (0.024 or higher, needed for Latin script ligatures and kerning, as well as for any complex script such as Arabic, Indic scripts, or Khmer)
- Text::Markdown (1.000031 or higher, needed if using 'md1' markup)
- HTML::TreeBuilder (5.07 or higher, needed if using 'html' or 'md1' markup)
- Pod::Simple::XHTML (3.45 or higher, needed if using buildDoc.pl utility to create HTML documentation)
- SVGPDF (0.087 or higher, needed if using SVG image functions)
- Test::Kwalitee (1.28 or higher, needed if running xt/kwalitee.t)

The following external applications (programs) are needed for fully testing some TIFF-related PDF::Builder functionality. Install *before* installing PDF::Builder or Graphics::TIFF.

- ImageMagick (module 'magick' or 'convert')
- Ghostscript (module 'gs', 'gswin64c', or 'gswin32c')

If an optional package is needed for certain extended functionality, but not installed, sometimes PDF::Builder will be able to fall back to built-in partial functionality (TIFF and PNG images), but other times will fail. After installing the missing package, you may wish to then run the t-test suite for that library to confirm that it is properly running, as well as running the examples.

Note that some of these packages, in turn, make use of various open source libraries (DLLs/shared libs) that you may need to hunt around for, and install on your system before you can install a given package. That is, they may not necessarily come with your Operating System or Perl installation. Some such packages may try to use "Alien" to build such prereqs, if missing. Other packages are "pure Perl" and should install without trouble.

Fixes needed to OPTIONAL packages

Sometimes fixes or patches are needed for optional prerequisites. See the file **INFO/Prereq_fixes.md** for a list of known issues.

External utilities

t/tiff.t (install testing for TIFF support) makes use of GhostScript and ImageMagick (convert utility). You may need to install these in order to get full testing (tests that need them will be skipped if they are not installed). Note that it has been reported that some versions of Mac Perl systems have a 'convert' utility that is missing the default Arial font, and thus will fail (see ticket 223).

Manually building

As is the usual practice with building such a package (from the command line), the steps are:

1. perl Makefile.PL
2. make
3. make test
4. make install

If you have your system configured to run Perl for a .pl/.PL file, you may be able to omit "perl" from

the first command, which creates a Makefile. "make" is the generic command to run (it feeds on the Makefile produced by Makefile.PL), but your system may have it under a different name, such as dmake, gmake (e.g., Strawberry Perl on Windows), or nmake.

PDF::Builder does not currently compile and link anything, so `gcc`, `g++`, etc. will not be used. The build process merely copies .pm files into place, and runs the "t" tests to confirm the proper installation.

Copyright

We would like to acknowledge the efforts and contributions of a number of users of PDF::Builder (and its predecessor, PDF::API2), who have given their time to report issues, ask for new features, and have even contributed code. Generally, you will find their names listed in the Changes and/or issue tickets related to some particular item. See the **INFO/ACKNOWLEDGE.md** and **INFO/SPONSORS** files.

License

Carrying On...

PDF::Builder is Open Source software, built upon the efforts not only of the current maintainer, but also of many people before me. Therefore, it's perfectly fair to make use of the algorithms and even code (within the terms of the LICENSE). That's exactly how the State of the Art progresses! Just please be considerate and acknowledge the work of others that you are building on, as well as pointing back to this package. Drop us a note with news of your project (if based on the code and algorithms in PDF::Builder, or even just heavily inspired by it) and we'll be happy to make a pointer to *your* work. The more cross-pollination, the better!

See Also

- INFO/SUPPORT file for information on reporting bugs, etc. via GitHub Issues
- INFO/DEPRECATED file for information on deprecated features
- INFO/KNOWN_INCOMP file for known incompatibilities with PDF::API2
- INFO/Prereq_fixes.md possible patches for prerequisites
- CONTRIBUTING file for how to contribute to the project
- LICENSE file for more on the license term
- INFO/RoadMap file for the PDF::Builder road map
- INFO/ACKNOWLEDGE.md for "thank yous" to those who contributed to this product
- INFO/SPONSORS for "thank yous" to those who financially sponsored this product
- INFO/CONVERSION file for how to convert from PDF::API2 to PDF::Builder
- INFO/Changes* files for older change logs
- INFO/PATENTS file for information on patents

`INFO/old/` also has some build and test tool files that are not currently used.

Documentation

To build the full HTML documentation (all the POD), get the full installation and go to the `docs/` directory. Run `buildDoc.pl --all` to generate the full tree of documentation. There's a lot of additional information in the PDF::Builder::Docs module (it's all documentation).

You may find it more convenient to point your browser to our Home Page to see the full documentation build (as well as most of the example outputs).

We admit that the documentation is a bit light on "how to" task orientation. We hope to more fully address this in the future, but for now, get the full installation and look at the `examples/` and `contrib/` directories for sample code that may help you figure out how to do things. The installation tests in the `t/` and `xt/` directories might also be useful to you.

A Note on Paper Sizes

Per PDF specifications, the default paper (media) size set in PDF::Builder is 'US Letter' (8.5in x 11in, 216mm x 279mm). If you're in the civilized world, and don't measure things in "bananas", you may wish to specify 'A4' instead, in the `$pdf->mediabox('A4');` call. Note that A4 is narrower and taller than Letter. If you want to ensure that your output will be printable around the world, consider using the *universal* paper size: `$pdf->mediabox('universal');`. This will result in some wasted space at the top of a printed A4 page, or some wasted right margin on a printed Letter page, but it's better than having content cut off! It may also be satisfactory to leave sufficient *margins* around document content on standard US Letter *or* A4 media, with the following **minimum** margins (allowing 3mm/.125" for paper handling, plus extra for media size differences):

`bottom: 3mm = .125" = 9pt` (bottom same for all media)

`left: 3mm = .125" = 9pt` (left same for all media)

`A4 top: 21mm = .83" = 60pt` (A4 taller than Letter)

`Letter top: 3mm = .125" = 9pt`

`Letter right: 9mm = .375" = 27pt` (Letter wider than A4)

`A4 right: 3mm = .125" = 9pt`

If not using the *Universal* size, you can choose *Letter* size, with an extra-wide right margin of at least 27pt, or *A4* size media with an extra deep top margin of at least 60pt. In both cases, allow a minimum of 9pt for left and bottom margins, for paper-handling purposes. This should give you results printable on either US Letter or A4 media, anywhere in the world, without accidentally losing content.

Please see the discussion on `mediabox()` and other "box" calls, about how much of a page can actually be *printed* on, allowing for pinch rollers and other paper transport mechanisms. The above suggested margins assume, in addition to Letter paper being .25" wider and .7" shorter than A4, that 1/8" (9pt, 3mm) of paper is unprintable around the edges; your printer may vary.

Block Quote (left and right margins)

Sed ut perspiciatis, — unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam eaque ipsa, quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt, explicabo. Nemo enim ipsam voluptatem, quia voluptas sit, aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos, qui ratione dolor sit, voluptatem sequi nesciunt, neque porro quisquam est, qui dolorem ipsum, quia amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt, ut labore et dolore magnam aliquam quaerat voluptatem.

> Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit, qui in ea voluptate velit esse, quam nihil molestiae consequatur, vel illum, qui dolorem eum fugiat, quo voluptas nulla pariatur? At vero eos et accusamus et iusto odio dignissimos ducimus, qui blanditiis praesentium voluptatum deleniti atque corrupti, quos dolores et quas molestias excepturi sint, obcaecati cupiditate non provident, similique sunt in culpa, qui officia deserunt mollitia animi, id est laborum et dolorum fuga.

Sed ut perspiciatis, unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam eaque ipsa, quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt, explicabo. Nemo enim ipsam voluptatem, quia voluptas sit, aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos, qui ratione dolor sit, voluptatem sequi nesciunt, neque porro quisquam est, qui dolorem ipsum, quia amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt, ut labore et dolore magnam aliquam quaerat voluptatem.

Here is some text at 15 point size. We follow it *somewhere* down the line with much larger text, ~~and~~ follow it with some ginormous text. That should have moved the entirety of the baseline down by quite a bit, while maintaining an even baseline.

Markdown horizontal rules: 3 or more ---, *, or _. full width

---

Between two rules

Between two rules

Last commentary

HTML horizontal rules, with CSS

Between two rules, above is default settings

Between two rules, above is very thick and blue

Above rule is only 200pt long, and explicitly left aligned

Above rule is *very* thick orange and 300pt long, centered by default

This is within a 'glotz' tag. This is within another.

This paragraph has CSS 'snork' property and is green.

text-align: left

text-align: center

text-align: right

2.explicit LJ at 0% 1.text at center 3.RJ text at 100%

2.Center-72pt CJ text. 1.Center+72pt LJ text.