PDF Generation for Plan Documents

Using Pandoc, Eisvogel, and TikZ

Ethan Petuchowski

2025-12-31

Contents

Overview

This guide explains how to generate professional PDF documents from Markdown using Pandoc and the Eisvogel LaTeX template. The output uses Computer Modern fonts (standard in CS academic papers) with proper diagrams, syntax highlighting, and professional typography.

Stack:

- **Pandoc** — Markdown → PDF converter
- **Eisvogel** — Professional LaTeX template
- **pdflatex** — TeX engine (uses Computer Modern fonts)
- **TikZ** — Programmatic diagrams (flowcharts, architecture)
- **pgf-umlsd** — UML sequence diagrams

Setup

The build script auto-installs all dependencies on first run:

- **Homebrew** (if missing)
- **BasicTeX** (pdflatex)
- **LaTeX packages** (fonts, TikZ, etc.)
- **Eisvogel template** (if missing)

Just run the build script — it handles everything:

```
./plans/_templates/build-pdf.sh plans/viant/my-document.md
```

Note: First run may take a few minutes to install ~200MB of LaTeX packages. Interactive password input will be required for `tlmgr install`.

Document Structure

Minimal Frontmatter

Documents only need 4 fields. All styling comes from `pdf-defaults.yaml`:

```yaml
---
title: "Your Document Title"
subtitle: "Optional Subtitle"
author: "Author Name"
date: 2025-12-30
---
```

That's it! The build script automatically includes `pdf-defaults.yaml`.

Shared Configuration (pdf-defaults.yaml)

All common styling lives in `_templates/pdf-defaults.yaml`:

Element	Purpose
`titlepage: true`	Generates a styled title page
`toc: true`	Generates table of contents
`fontfamily: lmodern`	Computer Modern fonts (standard in CS papers)
`\usepackage{float}`	Enables `[H]` placement for figures
`\renewcommand\paragraph`	Makes H4 headers block-level (not inline)
`\renewcommand{\arraystretch}{1.4}`	Increases table row height
TikZ styles	Pre-defined shapes for architecture diagrams

To customize colors for a specific document, you can override in the frontmatter:

```yaml
---
title: "My Doc"
author: "Me"
date: 2025-01-01
titlepage-color: "1b4332"      # Override: Forest green
titlepage-rule-color: "95d5b2"
---
```

Diagrams with TikZ

Architecture Diagrams

Use the pre-defined styles for consistent diagrams. Wrap TikZ code in a raw LaTeX block:

~~~
```{=latex}
\begin{center}
\begin{tikzpicture}[node distance=1.5cm]
  \node (db) [database, fill=green!20] {Postgres};
  \node (api) [api, right of=db, xshift=3cm, fill=blue!20] {PowerSync};
  \node (client) [client, right of=api, xshift=3cm, fill=purple!20] {Flutter App};

  \draw [arrow, <->] (db) -- node[above, font=\footnotesize] {WAL} (api);
  \draw [arrow, <->] (api) -- node[above, font=\footnotesize] {sync} (client);
\end{tikzpicture}
\end{center}
```
~~~

Available TikZ Styles

Style	Shape	Use For
`database`	Cylinder	Databases (Postgres, SQLite)
`storage`	Cylinder (smaller)	File storage
`api`	Rounded rectangle	API services
`client`	Rectangle with shadow	Client apps
`box`	Rectangle	Generic containers
`bigbox`	Large rounded rectangle	Grouping boxes
`process`	Rectangle	Flowchart steps
`decision`	Diamond	Flowchart decisions
`startstop`	Rounded rectangle	Flowchart start/end

Stacked Elements (Multiple Clients)

~~~
```{=latex}
~~~

~~~~
\begin{center}
\begin{tikzpicture}
  \node (c3) [client, fill=purple!10] at (-0.12, 0) {};
  \node (c2) [client, fill=purple!15] at (-0.06, -0.06) {};
  \node (c1) [client, fill=purple!20] at (0, -0.12) {};
  \node at (c1.center) [font=\small] {Clients};
\end{tikzpicture}
\end{center}
```
~~~~

UML Sequence Diagrams

~~~~
```{=latex}
\begin{figure}[H]
\centering
\begin{sequencediagram}
  \newthread{client}{Client}
  \newinst[4]{server}{Server}

  \begin{call}{client}{request()}{server}{response}
  \end{call}

  \begin{sdblock}{Loop}{Repeat}
    \begin{call}{client}{poll()}{server}{}
    \end{call}
  \end{sdblock}
\end{sequencediagram}
\caption{Client-server interaction}
\end{figure}
```
~~~~

Code Blocks

Standard Syntax Highlighting

Just use fenced code blocks with language like this:

~~~
```dart
class MyService {
  final Database db;

  Future<List<Item>> getItems() async {
    return await db.query('SELECT * FROM items');
  }
}
```
~~~

which will render like this

```dart
class MyService {
  final Database db;

  Future<List<Item>> getItems() async {
    return await db.query('SELECT * FROM items');
  }
}
```

Supported languages: `dart`, `java`, `rust`, `typescript`, `python`, `sql`, `yaml`, `bash`, `json`, `http`, etc. Run `pandoc --list-highlight-languages` for the full list.

Tables

Basic Tables

```
| Column 1 | Column 2 | Column 3 |
|----------|----------|----------|
| Value A  | Value B  | Value C  |
| Value D  | Value E  | Value F  |
```

Important: Always leave a blank line before tables.

Wide Content

For tables with long content that might overflow columns, use definition-style formatting instead:

~~~
**Named volume** - Managed by Docker, survives container removal

```yaml
volumes:
  - postgres_data:/var/lib/postgresql/data
```

**Bind mount** - Maps a specific host file into the container

```yaml
volumes:
  - ./init.sql:/docker-entrypoint-initdb.d/init.sql
```
~~~

Markdown Best Practices for PDF

Lists Need Blank Lines

Always put a blank line before lists:

```
This is a *native Postgres API*. It's the same mechanism used for:

- Read replicas across data centers
- Real-time data pipelines
- Zero-downtime migrations
```

Wrong (renders inline):

```
It's the same mechanism used for:
- Read replicas
- Data pipelines
```

Use Proper Headers (Not Bold)

Use `###` and `####` for subsections, not `**bold text**`:

```
### Good: This is a proper header

**Bad: This looks like a header but isn't**
```

Internal Links

Link to other sections using Markdown anchors:

```
See [Database Migrations](#database-migrations) for details.
```

Avoid Unicode in Code Blocks

pdflatex cannot render Unicode box-drawing or special characters in code blocks. Two options:

Option 1: ASCII alternatives (in code blocks)

Instead of	Use
Box-drawing	`+--`, `\--`
Arrows	`->`, `<-`, `<->`
Checkmark	`[x]`, `(yes)`

Option 2: LaTeX math symbols (in regular text)

Use `$...$` for inline math with beautiful symbols:

Symbol	LaTeX	Renders as
Right arrow	`$\rightarrow$`	$\rightarrow$
Left arrow	`$\leftarrow$`	$\leftarrow$
Bidirectional	`$\leftrightarrow$`	$\leftrightarrow$
Checkmark	`$\checkmark$`	$\checkmark$
Maps to	`$\mapsto$`	$\mapsto$
Implies	`$\Rightarrow$`	$\Rightarrow$
Equivalent	`$\equiv$`	$\equiv$

Document Organization

Use \part{} for Major Sections

For long documents, group content into parts:

```
\part{Architecture}

# System Overview
...

\part{Docker Fundamentals}

# What is a Container?
...
```

Recommended Structure (Design Docs)

1. **Overview** — Problem statement, goals, non-goals
2. **Context** — Background info the reader needs
3. **Architecture** — Diagrams + component breakdown
4. **Implementation** — Key details, data models, APIs
5. **Alternatives Considered** — Why not other approaches
6. **Open Questions** — Unresolved decisions

Building a PDF

```
# From anywhere, with full path
path-to/build-pdf.sh plans/path-to/design-doc.md

# Output: plans/path-to-design-doc.pdf
```

Troubleshooting

"Missing package" Errors

The build script installs common packages automatically. If you still get a missing package error:

```
sudo tlmgr install <package-name>
```

Unicode Character Errors

If you see `Unicode character X not set up for use with LaTeX`:

1. Find the character: search the .md file
2. Replace with ASCII equivalent (see table above)
3. This only affects code blocks; regular text handles Unicode fine

H4 Headers Rendering Inline

This is fixed automatically by `pdf-defaults.yaml`. If you're not using the build script, add the paragraph fix to your frontmatter's `header-includes`.

Table Columns Overlapping

Option 1: Adjust column widths using dash proportions:

```
| Narrow | Wide Column                   |
|:-------|:-----------------------------|
| A      | This column gets more space   |
```

The ratio of dashes (7:29) sets the width ratio. More dashes = wider column.

Option 2 (usually simpler and more conclusive): Replace wide tables with definition-style sections (see Tables section above).

Recommended File Structure

```
plans-generator/
  _templates/
    Eisvogel-3.3.0/
      eisvogel.latex        # LaTeX template
    build-pdf.sh            # Build script
    pdf-defaults.yaml       # Shared styling (fonts, TikZ, colors)
  00-pdf-generation.md      # This guide
```

```
plans/
  viant/
    docker-compose-explained.md
    docker-compose-explained.pdf
  ...
```

Quick Reference

Minimal Frontmatter

```yaml
---
title: "Document Title"
subtitle: "Optional Subtitle"
author: "Author Name"
date: 2025-12-30
---
```

All styling (fonts, colors, TikZ) comes from `pdf-defaults.yaml` automatically.

Diagram Syntax

Element	Syntax
Raw LaTeX block	` ```{=latex} ... ``` `
Figure with caption	`\begin{figure}[H] ... \caption{text} \end{figure}`
Database node	`\node (name) [database, fill=green!20] {Label};`
API node	`\node (name) [api, fill=blue!20] {Label};`
Client node	`\node (name) [client, fill=purple!20] {Label};`
Bidirectional arrow	`\draw [arrow, <->] (from) -- node[above] {label} (to);`

Element	Syntax
Sequence diagram	`\begin{sequencediagram} ... \end{sequencediagram}`