

Table of Contents

- [1. Overview](#)
 - [1.1 Choose what you need](#)
 - [1.1.1 Authoring extensions](#)
 - [1.1.2 Publishing and project tools](#)
 - [1.2 Project status](#)
- [Index](#)

1. Overview

This section is for anyone new to prodockit. It explains what the package adds to Zensical, how to install it, and how to build a first local site before you choose authoring or publishing features.

prodockit adds the document features that professional and academic Zensical projects ([Zensical website](#)) commonly need: numbered sections, cross-references, citations, glossaries, richer tables, and a printable PDF. Install one Python package, then enable only the parts your project uses.

If you want to see it working before reading the reference pages, follow [Build your first site](#). It starts with a new Zensical project and ends at a live local preview.

1.1 Choose what you need

1.1.1 Authoring extensions

These are standard Python-Markdown extensions configured in `zensical.toml`:

Extension	Use it for
<code>prodockit.headings</code>	Numbered headings
<code>prodockit.refs</code>	Cross-references to headings, figures, and tables
<code>prodockit.citations</code>	A small, Markdown-defined reference list
<code>prodockit.glossary</code>	Acronyms and glossary terms
<code>prodockit.tables</code>	Widths, merged cells, dense tables, and richer headers
<code>prodockit.tree</code>	Readable directory trees
<code>prodockit.steps</code>	Procedures presented as numbered steps
<code>prodockit.bibliography</code>	BibTeX/BibLaTeX citations formatted with CSL
<code>prodockit.index</code>	A PDF-only back-of-book index

Every extension is independent. Start with one; add another when the document needs it.

1.1.2 Publishing and project tools

prodockit also provides commands and integrations rather than Markdown syntax:

Feature	Use it for
prodockit pdf	Build a standalone PDF from the same navigation as the site
prodockit source-bundle	Package the underlying Markdown and configuration as a PDF
prodockit.zensical_macros	Word counts, repository data, and document-wide numbering in templates
prodockit.testing	Check the built website and PDF with pytest
Maintain prodockit	Maintain the package repository, build pins, automation, and releases

Go to the [Authoring reference](#) when you want to add document features. Go to [Publish a document](#) when you are ready to use the template, build a PDF, or publish with continuous integration. The [command-line reference](#) says which commands change files and which only report.

1.2 Project status

prodockit is early but functional. All nine extensions, the PDF and source bundle builders, website macros, testing support, and project-management commands are implemented and tested. See [Support and compatibility](#) for maturity, PyMdown Blocks and other required versions, platform coverage, and known constraints. Read the [release notes](#) before upgrading.

Index

Z

Zensical, 2