

Table of Contents

| | |
|---|----------|
| 1 Introduction | 1 |
| 1.1 What is TerraPDF? | 1 |
| 1.2 Table of Contents Feature | 1 |
| 1.2.1 How it works | 1 |
| 1.2.1.1 Supported heading levels | 1 |
| 1.2.1.1.1 Example customisation | 1 |
| 1.2.1.1.1.1 Technical note | 1 |
| 2 Available Heading Levels | 3 |
| 2.1 Styling Customisation | 3 |
| 2.2 Sample Table | 3 |
| 2.2.1 Table note | 3 |
| 2.2.1.1 Further reading | 3 |
| 3 Linking Within Documents | 4 |
| 3.1 Using InternalLink() | 4 |

Introduction

This report demonstrates TerraPDF's Table of Contents generation feature. Headings are automatically collected from the document and used to build an interactive TOC page with clickable page numbers.

What is TerraPDF?

TerraPDF is a free, pure C# library designed for fast and reliable PDF generation. It provides a fluent, composable API that covers the full document-authoring lifecycle — from page layout and rich text to tables, images, hyperlinks, and multi-page pagination — with no native binaries, no third-party runtime packages, and no licensing restrictions.

Key features include:

- *Zero dependencies — pure C#*
- *Fluent, intuitive API*
- *Multi-page automatic pagination*
- *Internal hyperlinks and bookmarks*
- *Tables with repeating headers*

Table of Contents Feature

The Table of Contents feature allows you to automatically generate a navigable contents page from your document's headings. Simply call `.TableOfContents()` once at the desired location (typically at the beginning), then use `.H1()`, `.H2()`, etc. for section headings anywhere in the document.

How it works

TerraPDF performs a two-pass layout: first it measures the document and collects all heading positions, then it renders the final PDF with a fully populated TOC containing accurate page numbers and internal links.

Supported heading levels

Six levels are available: H1 (largest) through H6 (smallest). Each level has sensible default styling which can be overridden using the `TextDescriptor` fluent API.

Example customisation

You can customise any heading after creation, e.g. `.H2('Title').FontColor(Color.Blue).Underline();`

Technical note

Report with Table of Contents

Headings nested inside Tables or Rows are also collected. The TOC page itself is ignored so headings do not self-reference.

Available Heading Levels

TerraPDF provides six levels of headings with the following defaults:

H1 — Chapter title (24 pt, bold)

H2 — Section title (20 pt, bold)

H3 — Sub-section (16 pt, bold)

H4 — Minor heading (14 pt, italic)

H5 — Small heading (12 pt, bold)

H6 — Tiny heading (11 pt, regular)

Styling Customisation

You can customise any heading using the fluent TextDescriptor API after creation. For example:

```
container.H2('Custom Heading').FontColor(Color.Blue.Darken2).Underline();
```

Sample Table

Tables do not generate TOC entries themselves, but any heading inside a table cell is picked up by the scanner.

| Description | Qty | Amount |
|------------------|------|--------|
| TerraPDF License | 1 | Free |
| Support Hours | 24/7 | \$0.00 |

Table note

The table above is purely illustrative and does not affect the TOC.

Further reading

Refer to Chapter 3 for information about internal links and navigation.

Linking Within Documents

Internal links allow readers to click and jump to specific pages in the PDF. TerraPDF's TOC automatically creates these internal links for all headings, but you can also add custom internal-link anchors anywhere in your content.

Using `InternalLink()`

Wrap any content in `.InternalLink(targetPage, topY)` to make it navigable:

```
container.InternalLink(2, 120).Text('Go to Chapter 2');
```

The TOC entries themselves are created this way behind the scenes.