



uscribe User Guide

Contents

1	Introduction	1
1.1	What uscribe is for	1
1.2	A taste of Unicon highlighting	1
1.3	Relation to unidoc	2
1.4	How to read this guide	2
2	Installation	3
2.1	Requirements	3
2.2	TeX / PDF dependencies by platform	3
2.2.1	macOS (Homebrew)	3
2.2.2	Debian / Ubuntu	4
2.2.3	Fedora / RHEL-family	5
2.2.4	Arch Linux	5
2.2.5	Windows	5
2.2.6	What uscribe LaTeX output needs	5
2.3	Build uscribe	6
2.4	Verify	6
2.5	Build this manual	6
3	Quick Start	8
3.1	1. Create a manifest	8
3.2	2. Write chapters	8
3.3	3. Book settings	9
3.4	4. Build HTML	10
3.5	5. Build PDF (optional)	10
3.6	Where files go	10
4	Reports	12
4.1	Mark a file as a report	12

4.2	One report	13
4.3	Several reports and an index	14
4.4	Book vs report (summary)	15
5	Book Configuration	16
5.1	How the file is found	16
5.2	File format	16
5.3	title	17
5.4	copyright	17
5.5	logo	17
5.6	name	17
5.7	theme	17
5.8	themePath	18
5.9	docclass	18
5.10	frontmatter	18
5.11	toctree	18
5.12	sources	18
5.13	Not in book.conf	19
5.14	This guide	19
6	Markup Reference	20
6.1	Headings	20
6.2	Document fields	20
6.3	Paragraphs and lists	21
6.4	Definition lists	22
6.5	Tables	23
6.6	Labels and cross-references	25
6.7	Stable labels	26
6.8	Inline markup	26
6.9	Directives	26
6.10	Include / literalinclude	29
6.11	Unicon listings	30
7	Themes	32
7.1	Built-in themes	32
7.2	Build-time default	32
7.3	Theme layout on disk	33

8	Command Line	34
8.1	Synopsis	34
8.2	Options	34
8.3	Outputs	35
8.4	Exit status	36

Chapter 1

Introduction

uscribe is a prose/book documentation generator for Unicon. It reads a small RST-like markup, builds a doctree, resolves cross-chapter references, and writes themed HTML (and optionally LaTeX/PDF) for Unicon documentation and books.

This manual is the user guide for *uscribe* itself. Sources live under `doc/uscribe/` in the unicon tree and are built *with* *uscribe*.

1.1 What *uscribe* is for

- Multi-chapter books and manuals (ordered by a manifest file)
- Standalone reports (UTR-style): one HTML page per report, section navigation in the sidebar, and a catalog `index.html` of links
- HTML output with sidebar navigation (chapters, plus the current chapter's sections), search, and selectable themes
- Unicon code samples with syntax highlighting
- Images and simple admonitions

It is **not** a full RST/docutils toolchain. The markup subset is intentionally small; extend the parser as real documents need more.

1.2 A taste of Unicon highlighting

Mark listings with `.. code-block:: unicon` (also `icon` or `icn`). In HTML they are highlighted client-side:

```
procedure greet(who)
  write("Hello, ",
who | "world", "!")
end

procedure main()
  greet("uscribe")
  greet()
end
```

See [Markup Reference](#) for the full dialect, [Quick Start](#) to generate a tiny book, and [Reports](#) for a single paper or a catalog of papers.

1.3 Relation to unidoc

`uni/unidoc/` extracts API docs from Unicon source comments. *uscribe* is a separate tool for authored prose. They can coexist; a future `.. api::` bridge may connect them.

1.4 How to read this guide

- [Installation](#) — build and install the `uscribe` binary
- [Quick Start](#) — generate HTML from a tiny book
- [Reports](#) — one paper, or several papers plus a catalog index
- [Book Configuration](#) — `book.conf` keys, defaults, and CLI overrides
- [Markup Reference](#) — headings, lists, inline markup, directives
- [Themes](#) — basic, classic, dark, and the in-page switcher
- [Command Line](#) — CLI options

Chapter 2

Installation

2.1 Requirements

Required

- A built Unicon tree (`./configure && make` from the unicon sources)

Optional (PDF only)

- A TeX engine on your PATH: `pdflatex`, `xelatex`, or `lualatex`. `uscribe` picks the first one it finds. HTML and `--format=latex` (`.tex` only) do not need TeX.

There is no Homebrew or apt package literally named `pdflatex`; that name is a program shipped inside a TeX distribution. Install one of the packages below, then confirm:

```
which pdflatex || which xelatex || which lualatex
```

2.2 TeX / PDF dependencies by platform

2.2.1 macOS (Homebrew)

Compact (recommended for `uscribe`):

```
brew install --cask basictex
```

Full MacTeX (much larger):

```
brew install --cask mactex  
# or: brew install --cask mactex-no-gui
```

After BasicTeX or MacTeX, restart the terminal (or run `eval "$(/usr/libexec/path_helper)"`) so the new binaries are on `PATH`.

2.2.2 Debian / Ubuntu

Minimal (often enough):

```
sudo apt update  
sudo apt install texlive-latex-base
```

Recommended if `pdflatex` complains about missing packages (`hyperref`, `fancyvrb`, `graphicx` fonts, and similar):

```
sudo apt install texlive-latex-recommended texlive-fonts-recommended
```

Larger kitchen-sink options:

```
sudo apt install texlive-latex-extra  
# or everything: sudo apt install texlive-full
```

2.2.3 Fedora / RHEL-family

```
sudo dnf install texlive-latex
# fuller: sudo dnf install texlive-scheme-medium
# everything: sudo dnf install texlive-scheme-full
```

2.2.4 Arch Linux

```
sudo pacman -S texlive-basic texlive-latex
# or a broader set: sudo pacman -S texlive-meta
```

2.2.5 Windows

Install either MiKTeX (<https://miktex.org/download>) — packages on demand — or TeX Live for Windows (<https://tug.org/texlive/>).

Ensure the installer puts `pdflatex` on your `PATH`, then open a new Command Prompt or PowerShell window.

2.2.6 What uscribe LaTeX output needs

Generated `STEM.tex` uses a small set of packages (`graphicx`, `hyperref`, `xcolor`, `fancyvrb`, `booktabs`, `longtable`, `array`). `multirow` is loaded when the `.sty` is installed (needed for grid-table row spans). `caption` is loaded when present so Word-style **Figure N / Table N** captions can use `\\caption*`; without it `uscribe` falls back to a centered paragraph. `framed` is loaded when present so code samples get a shaded box; without it the listing is still full-width `Verbatim`. A latex recommended install is usually enough; GitHub Pages installs `texlive-latex-extra` so the user-guide PDF build has the full set. If `pdflatex` stops with `File ...sty not found`, install the matching TeX Live or MiKTeX package (or a larger scheme) and retry.

SVG figures are not embedded in PDF; convert them to PDF or PNG, or accept the boxed placeholder `uscribe` emits for `.svg` paths.

2.3 Build uscribe

From the unicon tree:

```
cd uni/uscribe
make
```

That compiles the `uscribe` package, links `./uscribe`, and installs a copy to `../..bin/uscribe`. `make install` in this directory does the same. A top-level `make install` also installs `uscribe` into `$(bindir)` alongside `unicon`, `udb`, and the other addon tools, and copies themes to `$(libdir)/unicon/uni/uscribe/themes`.

Regenerate Make dependencies after adding `.icn` files:

```
make deps
```

2.4 Verify

```
uscribe
# no book.manifest in this directory: prints usage and exits
```

2.5 Build this manual

HTML:

```
cd doc/uscribe
make
# open out/index.html
```

PDF (after installing a TeX engine as above):

```
cd doc/uscribe
make pdf
# open out/uscribe-userguide.pdf
```

`make latex` writes `out/uscribe-userguide.tex` without running a TeX engine. `make pdf` writes the `.tex` and runs the engine twice (TOC and cross-references).

Chapter 3

Quick Start

A uscribe project is a directory of chapter `.rst` files, a manifest that lists them in reading order, and usually a `book.conf` for book-wide title, logo, and PDF name.

3.1 1. Create a manifest

`book.manifest` (name is conventional; any path works):

```
# comments and blank lines are ignored
preface.rst
chapter1.rst
chapter2.rst
```

3.2 2. Write chapters

Underline-style headings, paragraphs, lists, and directives. Example `chapter1.rst`:

```
First Chapter
=====

Hello from uscribe. See :ref:`Second Chapter` later.
```

```

Second Section
-----

- alpha
- beta

.. note::

    Indented body of a note.

.. code-block:: unicon

    procedure main()
        write("hi")
    end

```

Underline rule: the underline must be at least as long as the title. Shorter underlines are not treated as headings.

Here is that Unicon listing as it appears when built (syntax highlighted in HTML):

```

procedure main()
    write("hi")
end

```

3.3 3. Book settings

`book.conf` next to the manifest (optional; CLI overrides). Full key list: [Book Configuration](#).

```

title: My Book
copyright: 2026, Your Name
logo: images/logo.png

```

```
name: my-book  
theme: basic
```

3.4 4. Build HTML

```
uscribe --targetDir=./out
```

Or from a project `Makefile` patterned on `doc/uscribe/Makefile`.

Open `out/index.html`. Use the sidebar for chapters (and the current chapter's sections), search, and theme switching.

3.5 5. Build PDF (optional)

Install a TeX engine first — see the *TeX / PDF dependencies by platform* section under [Installation](#). Then:

```
uscribe --format=pdf --targetDir=./out
```

Or `make pdf`. Output is `out/STEM.pdf` from the `name` key (default stem `book`). This user guide uses `name: uscribe-userguide`. SVG figures are not embedded; convert them to PDF or PNG for TeX.

3.6 Where files go

- `*.rst` — chapter sources
- `*.manifest` — chapter order
- `book.conf` — book title, logo, copyright, PDF name, theme
- `images/` — figures (copied into `out/images/`)

- `out/` — generated HTML and/or `STEM.tex` / `STEM.pdf`
- `themes/` — built-in skins shared across projects (HTML)

A report (one paper) or a catalog of reports uses the same tools with a different manifest. See [Reports](#).

Chapter 4

Reports

A **report** is one paper: one `.rst` source, one HTML page, one PDF. The left sidebar is that paper’s sections (Introduction, Motivation, ...). A **book** (this guide) is the other shape: many chapters, sidebar = chapter list, one combined PDF.

If every file in the manifest is a report, uscribe also writes a catalog `index.html` — a table of links. You can ignore it when you only have one report.

4.1 Mark a file as a report

Start the `.rst` with a field list. `:docclass: report` (or a `:trnumber:`) selects report layout: masthead, abstract, keywords, article PDF, section sidebar.

```
:title: Report Mode Example
:author: Uscribe Maintainers
:trnumber: 0
:date: 2026-08-07
:abstract: Minimal fixture for UTR-style report metadata, external
           links, sub/sup, and raw passthrough.
:docclass: report
```

```
Introduction
=====
```

Headings in the body become the sidebar TOC. Explicit labels (`.. _sec-intro:`) become `#sec-intro` jump targets.

4.2 One report

Put the paper in its own directory. Author, title, and abstract stay in the `.rst` field list (above). The manifest is one line. You do not need a per-report `.conf`.

```
my-paper/
  book.manifest      # one line: paper.rst
  paper.rst
  book.conf          # optional: themePath (same file books use)
```

```
book.manifest:
```

```
paper.rst
```

Build from that directory (defaults pick up `book.manifest` and, if present, `book.conf`):

```
uscribe --targetDir=./out
```

Open `paper.html`. The sidebar is that paper's sections. `index.html` is a one-row catalog; you can ignore it.

Optional `book.conf` is only *build* settings (theme, logo, `themePath`), not author or abstract. If you omit `title` there, the sidebar uses the paper's `:title:`.

This guide's live fixture lives next to the user-guide `book.conf`, so the Makefile passes `--config=report-example.conf` to avoid picking up the guide settings. The paper is [ex-report.html](#) (after `make -C doc/uscribe`).

Output:

```
out/
  paper.html      ← the report (section nav)
  paper.pdf       ← if you passed --format=pdf
  index.html
  search.html
  _static/
  _sources/
```

4.3 Several reports and an index

Same layout, more files in the manifest. Each paper is still one HTML page. `book.conf` title is the *series* name on the catalog (and the link back from each paper's sidebar).

```
my-reports/
  book.manifest
  book.conf      # title: My Technical Reports
  paper-a.rst
  paper-a.bib
  paper-b.rst
```

```
book.manifest:
```

```
paper-a.rst
paper-b.rst
```

```
uscribe --targetDir=./out
```

Open `index.html` for the table (number, title, author, date, pdf/txt). Open a row to read that paper; its sidebar is that paper's sections. Each paper's `:cite:` numbers start at 1, and `.. bibliography::` lists only that paper's cited works from its own `.bib`.

The Unicon Technical Reports tree is this pattern: `doc/utr/utr.manifest` and `doc/utr/book.conf`, via `make -C doc/utr html`.

Live fixtures in this guide (again using a separate `--config` because they share a directory with the user guide): [catalog index](#), [first report](#), [second report](#). The manifests are `report-example.manifest` (one file) and `reports-example.manifest` (two files).

4.4 Book vs report (summary)

Piece	Book	Report
Source	many chapters	one <code>.rst</code> per paper
Sidebar	other chapters	sections of this paper
<code>index.html</code>	ordered chapter list	catalog table of papers
PDF	one file for the whole book	one file per paper
When	manuals, this user guide	UTR, a standalone paper

A mixed manifest (some reports, some book chapters) is unusual. Each report page still gets a section sidebar; book chapters keep the chapter list. Prefer an all-report manifest or an all-book manifest.

See [Markup Reference](#) for the field-list keys and [Command Line](#) for `--format / --manifest`.

Chapter 5

Book Configuration

Book-wide settings live in `book.conf`. Chapter order lives in `book.manifest`. Per-chapter author, date, and copyright live in the RST field list (see [Markup Reference](#)). Build invocation (`--format`, `--targetDir`) stays on the command line.

5.1 How the file is found

1. `--config=FILE` if you pass it (the file must exist).
2. Otherwise `book.conf` in the same directory as the manifest.
3. If that file is missing, `uscribe` continues with CLI values and built-in defaults.

With the usual defaults (`book.manifest` in the current directory), a book directory only needs:

```
uscribe --targetDir=./out
```

5.2 File format

One `key: value` per line. Keys are case-insensitive. Blank lines and `#` comments are ignored. A leading colon on the key is allowed (`:title:`), so the file can look like an RST field list if you prefer.

Unknown keys print a warning on standard error and are ignored. Values do not wrap to the next line.

`logo` and `themePath` are paths. Relative paths are resolved against the directory that contains `book.conf`, not against the process current working directory. Absolute paths are left as-is.

Command-line options override the file. The Makefile for this guide passes `--theme` so `make classic` can change the initial skin without editing `book.conf`.

5.3 title

Book title: HTML `<title>`, sidebar heading, index heading, and the LaTeX `\title` / title page. Default if unset: `Untitled Book`. CLI: `--title`.

5.4 copyright

Book-wide footer text after © on HTML pages (for example 2026, Jafar Al-Gharaibeh). This is not a chapter byline; a chapter `:copyright:` field is separate (see [Markup Reference](#)). CLI: `--copyright`.

5.5 logo

Image file. HTML copies it to `targetDir/_static/` and shows it in the sidebar. LaTeX / PDF copies it next to the `.tex` and places it on the title page (page 1), not as a numbered figure. CLI: `--logo`.

5.6 name

Basename for the book `.tex` / `.pdf` (and the HTML footer `pdf` link). A trailing `.pdf` or `.tex` is stripped. Letters, digits, `_`, and `-` only. Default: `book`. Ignored for report collections (one PDF per report, named like the HTML stem). CLI: `--name`.

5.7 theme

Initial HTML theme: `basic`, `classic`, or `dark` (or a custom name under `themePath`). Default: `basic`. Readers can still switch skins in the sidebar. Ignored for LaTeX / PDF. CLI: `--theme`.

5.8 themePath

Directory that contains theme subdirectories (`DIR/basic/static,...`). Also accepted as `themepath` or `theme_path`. If omitted, uscribe searches `./themes, ../themes, themes`, then Unicon's `uni/uscribe/themes` via `&features` (Binaries at / Libraries at). CLI: `--themePath`.

5.9 docclass

`book` (default) or `report`. Report mode is usually selected per chapter with `:docclass: report` or `:trnumber::`; this key forces the class for every chapter. Also accepted as `doc_class` or `doc-class`. CLI: `--docclass`.

A report is one HTML page and one PDF. The sidebar on that page is the report's sections. `index.html` is a catalog of links. Workflow and live examples: [Reports](#).

5.10 frontmatter

How many leading chapters are unnumbered front matter (a preface before chapter 1). Default 0. When set, the HTML sidebar numbers sections as `1.1, 2.3.1, ...` matching the Word outline, and the PDF uses `\\frontmatter` for those chapters, then the table of contents, then `\\mainmatter`.

5.11 toctree

Sidebar tree lines (vertical spine and ticks) on the current chapter's section list. Default off: numbers and indentation only. Set `toctree: true` (also `yes / on / 1`; aliases `toc-tree, toc_tree`) to draw the lines.

5.12 sources

Copy each chapter's RST into `targetDir/_sources/` (as `.txt`) and add a `source` (or report `txt`) footer link. Default on. Set `sources: false` (also `no / off / 0`; alias `source`) so the published HTML tree does not include RST. A rebuild in the same `--targetDir` also deletes a leftover `_sources/` from a previous build. LaTeX/PDF builds still write `.tex` next to the PDF; they are a different `--format` from HTML. CLI: `--sources=no`.

5.13 Not in book.conf

These stay on the command line because they describe one *build*, not the book:

- `--format` — `html`, `latex`, or `pdf`
- `--targetDir` — output directory (default `./htmlBook` or `./latexBook`)
- `--manifest` — chapter list (default `book.manifest`)

5.14 This guide

The file that builds this manual:

```
# Book-level settings (not chapter order --- that is book.manifest).
# Paths are relative to this file. Command-line options override.
title: uscribe User Guide
copyright: 2026, Jafar Al-Gharaibeh
logo: images/uscribe-logo.png
name: uscribe-userguide
theme: basic
```

Chapter order is `book.manifest` in the same directory.

Chapter 6

Markup Reference

uscribe accepts a **restricted RST-like** dialect. Unknown constructs are either ignored, treated as plain text, or left as raw directives.

6.1 Headings

```
Chapter Title
=====

Section
-----

Subsection
~~~~~
```

Characters = - ~ ^ " are recognized as underline styles (chapter through heading 5). Heading level follows the underline character. The underline length must be greater than or equal to the title length.

6.2 Document fields

A chapter may start with an RST field list (Docutils bibliographic fields). Repeat **:author:** for several names. Book chapters show these as a byline under the first heading; omit the fields and nothing is added. Reports also

use `:title:`, `:trnumber:`, `:abstract:`, `:keywords:`, and `:docclass:` for the cover — see [Reports](#) for the one-paper and catalog workflows.

```
:author: Jane Doe
:author: Pat Lee
:date: 2026-08-15
:copyright: 2026, Jane Doe
```

```
Chapter Title
=====
```

Book-wide title, logo, theme, and PDF name belong in `book.conf` (see [Book Configuration](#)). They are not chapter fields.

6.3 Paragraphs and lists

Blank lines separate paragraphs. Bullet lists use `-` or `*` with a following space. Numbered lists use `1.` / `1)` or auto-number `#.` . Indented continuation lines belong to the current item. Nested lists, `.. note::`, and `.. code-block::` indented under an item are parsed as that item’s body (a blank line between items is allowed).

Source:

```
1. Install the ``uscribe`` binary (:doc:`install`)
2. Write a tiny book (:doc:`quickstart`)
#. Pick a theme (:doc:`themes`)
```

Rendered:

1. Install the `uscribe` binary ([Installation](#))
2. Write a tiny book ([Quick Start](#))
3. Pick a theme ([Themes](#))

A list item can also hold a nested listing:

```
- Default mode prompt:

.. code-block:: text

    admin@r0>

- Configure mode is a nested list of its own:

- enter with ``config``
- leave with ``end``
```

- Default mode prompt:

```
admin@r0>
```

- Configure mode is a nested list of its own:
 - enter with `config`
 - leave with `end`

6.4 Definition lists

A term on its own line followed by an indented body. Blank lines inside the body start a new paragraph.

Source:

```
format
  Selects the builder.

  Use ``html`` for a browsable book, or ``latex`` / ``pdf`` for
  ↪ print.

theme
  Names the HTML skin (``basic``, ``classic``, or ``dark``).
```

Rendered:

format Selects the builder.

Use `html` for a browsable book, or `latex` / `pdf` for print.

theme Names the HTML skin (`basic`, `classic`, or `dark`).

6.5 Tables

Simple tables use `=` column separators. The first row is the header. PDF output uses `longtable` with wrapping paragraph columns so long identifiers and prose stay within the page.

Source:

```
=====
Flag      Meaning
=====
html      HTML book
latex     ``book.tex``
pdf       run a TeX engine
=====
```

Rendered:

Flag	Meaning
html	HTML book
latex	“book.tex
pdf	run a TeX

Grid tables use `+---+` borders; a border with `=` marks the header. Omit an internal `|` to span columns:

Source:

```
+-----+-----+-----+
| Left  | Mid    | Right  |
+=====+=====+=====+
```

```
| A      | B spanning two |
+-----+-----+
| C spanning two | D      |
+-----+-----+
```

Rendered:

Left	Mid	Right
A	B spanning two	
C spanning two		D

A plain grid without spans:

Source:

```
+-----+-----+
| Flag   | Meaning   |
+=====+=====+
| html   | HTML book |
+-----+-----+
| latex  | book.tex  |
+-----+-----+
```

Rendered:

Flag	Meaning
html	HTML book
latex	book.tex

List tables use the `list-table` directive. The optional argument is the table caption (shown under the table in HTML, and as a LaTeX caption). Word-style labels such as **Figure 6:** ... are kept as written:

Source:

```

.. list-table::
   :header-rows: 1

   * - Flag
     - Meaning
   * - html
     - HTML book
   * - pdf
     - run a TeX engine

```

Rendered:

Flag	Meaning
html	HTML book
pdf	run a TeX engine

6.6 Labels and cross-references

Place `.. _name:` immediately before a heading, figure, or table. Pass 1 collects explicit labels and title slugs; pass 2 turns ref roles into **chapter**. `html#anchor` links.

Source:

```

.. _stable-labels:

Stable labels
-----

A preceding ``.. _name:`` attaches to the next heading. Link with a
ref role naming that label, optionally with display text in angle
brackets.

```

Rendered:

6.7 Stable labels

A preceding `.. _name:` attaches to the next heading. Link with a `ref` role naming that label, optionally with display text in angle brackets.

Live refs (explicit label and display-text form):

- plain: [Stable labels](#)
- with display text: [Jump here](#)
- chapter: [Installation](#)

Title-only refs (matching a heading exactly) still work. Full chapter sources are linked from each HTML footer as **source** (and **txt** in the report index Formats column).

6.8 Inline markup

- single asterisks for *emphasis*
- double asterisks for **strong**
- backticks for `inline literals`

6.9 Directives

Admonition bodies are re-parsed as nested blocks (lists, code, nested directives). Source then rendered:

```
.. note::

    Admonition bodies can hold nested blocks:

    - a bullet list
    - and even a nested tip

    .. tip::

        Nested directives are parsed, not flattened.
```

Note.

Admonition bodies can hold nested blocks:

- a bullet list
- and even a nested tip

Tip.

Nested directives are parsed, not flattened.

Also: `warning`, `tip`, `important`, `caution`, `attention`, `danger`, `error`, `hint`.

CLI command signatures use `.. clicmd::`, like FRR's command reference: the argument is the syntax line; an indented body hangs under the command as its description.

```
.. clicmd:: pki gen <key-name> <key-type> <key-length>
```

Generate a private key.

Figures take a caption. If the caption already starts with **Figure N** or **Table N**, HTML and LaTeX keep that label instead of adding another number. Uncaptioned `.. image::` is not numbered.

```
.. _logo-figure:
```

```
.. figure:: images/uscribe-logo.png
```

:alt: uscribe logo

The uscribe mark.

See The uscribe mark..

Other authoring forms (shown literally):



Figure 6.1: The uscribe mark.

```

.. code-block:: unicon

    procedure main()
        write("ok")
    end

.. image:: images/diagram.svg

    Optional alt text

.. include:: path/to/fragment.rst

.. literalinclude:: path/to/file.icn

    :language: unicon
    :lines: 1-10
    :start-after: marker
    :end-before: marker
    :dedent: 3

```

Paths are resolved next to the including chapter, then under the process cwd. Nested `include` is allowed; circular includes warn and are skipped.

6.10 Include / literalinclude

`literalinclude` shows a file as a listing; `include` parses it as markup. The tip below is the same fragment both ways.

Source (`includes/shared-tip.rst`):

```

.. tip::

    Snippets under ``includes/`` can be pulled into any chapter with
    ``.. include::`` so the same advice is not copy-pasted.

```

Rendered:

Tip.

Snippets under `includes/` can be pulled into any chapter with `.. include::` so the same advice is not copy-pasted.

Pull a real `.icn` file as a listing. Language defaults from the extension (`.icn` $\rightarrow$ `unicon`); override with `:language::`

```
# Shared sample for literalinclude demos
procedure main()
  write("hello from includes/hello.icn")
  write(&version)
end
```

The same file, only the body between markers:

```
write("hello from includes/hello.icn")
write(&version)
```

6.11 Unicon listings

code-block languages `unicon`, `icon`, and `icn` get Unicon syntax highlighting in HTML (via `highlight-unicon.js`). `json` is highlighted the same way (strings, numbers, `true` / `false` / `null`, and `//` comments in JSONC samples). Other languages are emitted as plain `<pre><code class="language-...">` without extra highlighting — use `sh` for shell, `rst` or `text` for markup samples. `literalinclude` of `.icn` files uses the same Unicon highlighting.

Example Unicon program as it appears in the built book:

```
# Hello from a uscribe listing
procedure main()
  every i := 1 to 3 do
    write("tick ", i)
  write(&version)
end
```

A line ending in `::` (RST literal-block introducer) followed by an indented block is also treated as a code listing. `Unicon::` becomes the label `Unicon:` plus the indented tree; a lone `::` introduces a block with no label.

Unknown `.. name::` directives become HTML comments so content is not silently dropped.

See also [Quick Start](#) and [Command Line](#).

Chapter 7

Themes

HTML chrome (sidebar, search, prev/next on book chapters) is fixed in the generator. A *theme* is a CSS skin plus shared JavaScript.

7.1 Built-in themes

- **basic** — light sidebar (default)
- **classic** — serif / warm-gray documentation look
- **dark** — dark background

All three are installed into `out/_static/` as `theme-NAME.css`. Readers pick a theme from the **Theme** dropdown in the sidebar; the choice is stored in the browser (`localStorage` key `uscribe-theme`).

7.2 Build-time default

```
uscribe ... --theme=classic
```

`theme` and `themePath` may also be set in `book.conf` (see [Book Configuration](#)).

Or with the sample Makefiles:

```
make          # THEME=basic
make classic
make dark
```

`--theme` only sets the *initial* stylesheet until the reader selects another in the UI.

7.3 Theme layout on disk

```
themes/
  basic/static/book.css
  classic/static/book.css
  dark/static/book.css
  _shared/search.js
  _shared/highlight.css
  _shared/highlight-unicorn.js
  _shared/theme-switcher.js
```

To add a custom theme, create `themes/mytheme/static/book.css` and pass `--theme=mytheme`. Keep the same HTML class names (`.sidebar`, `.main`, `.tok-keyword`, ...) so shared scripts keep working.

See `themes/README.md` in the source tree for layout details.

Chapter 8

Command Line

8.1 Synopsis

```
uscribe [options]
```

8.2 Options

- **--manifest=FILE**** Plain-text chapter list: one source path per line. Lines starting with # and blank lines are ignored. Paths are relative to the process current working directory. Default: `book.manifest` in the current directory.
- **--config=FILE**** Book-level settings file. If omitted, uscribe loads `book.conf` next to the manifest when that file exists. See [Book Configuration](#) for the key list, path rules, and defaults. Any command-line option overrides the file.
- **--title=TITLE**** Book title for <title>, the sidebar heading, the index page, and the LaTeX `\title`. Default: from `book.conf`, else `Untitled Book`.
- **--copyright=TEXT**** Optional footer copyright (text after ©), e.g. 2026, Jafar Al-Gharaibeh. HTML only. Every HTML page always ends with `Powered by uscribe`; chapter pages also link **pdf** (the

book PDF from `--name`, or a sibling PDF for reports) and `source` (`_sources/name.rst.txt`, viewable as plain text in the browser).

- **--name=STEM**** Basename for the book `.tex` / `.pdf` (default: `book`). HTML footers and the index link to `STEM.pdf`. A trailing `.pdf` or `.tex` is stripped. Letters, digits, `_`, and `-` only. Ignored for report collections (one file per report, named like the HTML stem).
- **--logo=FILE**** Optional logo. HTML copies it into `targetDir/_static/` and shows it in the sidebar. LaTeX / PDF copies it into `targetDir` and places it on the title page (page 1), not as a chapter figure.
- **--format=FMT**** Output format: `html` (default), `latex`, or `pdf`.
- `html` — one page per chapter plus index/search/themes. Report manifests get a catalog table on `index.html`. Each report page's sidebar lists that report's sections. Book chapter pages list every chapter and expand the current chapter with a numbered local TOC (`1.1`, `2.3.1`, ...). Tree lines are off unless `toctree: true` in `book.conf`. See [Reports](#).
 - `latex` — for books, a single `STEM.tex` (default `book.tex`); for report manifests (every chapter is a report), one `.tex` per report
 - `pdf` — like `latex`, then run `pdflatex`, `xelatex`, or `lualatex` (whichever is first in `PATH`) twice for TOC/xrefs
- **--targetDir=DIR**** Output directory. Default: `./htmlBook` for HTML, `./latexBook` for latex/pdf. Created if missing.
- **--theme=NAME**** Initial HTML theme: `basic`, `classic`, or `dark` (or a custom name under `--themePath`). Default: `basic`. Ignored for latex/pdf.
- **--themePath=DIR**** Directory that contains theme subdirectories (`DIR/basic/static`, ...). If omitted, uscribe searches `./themes`, `../themes`, `themes`, then Unicon's `uni/uscribe/themes` using the `Binaries` at `/Libraries` at strings in `&features`. HTML only.

8.3 Outputs

HTML (`--format=html`): for each chapter `path/name.rst`, writes `targetDir/name.html`, plus:

- `index.html` — contents landing page (chapter list, or a catalog table of reports with a **Formats** column of `pdf` / `txt`)

- `search.html + searchindex.js` — full-text search
- `_static/` — theme CSS, search JS, highlighter, theme switcher
- `_sources/` — chapter sources as `name.rst.txt` (browser-viewable text; linked as `txt / source`). Omitted when `sources` is off; a rebuild then deletes a leftover `_sources/`

LaTeX / PDF: book manifests write `targetDir/STEM.tex` (and `STEM.pdf` with `--format=pdf`; default stem `book`). Report manifests write one `.tex / .pdf` per report, named like the HTML stem (`ex-report.tex` beside `ex-report.html`). Local `.. image::` files are copied into `targetDir` automatically (relative paths preserved). SVG is not embedded in LaTeX (a boxed placeholder is emitted instead — convert to PDF/PNG for TeX).

8.4 Exit status

Usage errors and missing files call `stop()` (non-zero). A successful build prints `wrote ...` lines for each output file.