



margo site

Purpose

`site` builds linked HTML pages from a Markdown directory or a `.yaml` or `.yml` site config. Directory mode recursively discovers regular `.md` and `.markdown` files, rewrites valid Markdown links, and validates anchors and local images. It does not read stdin.

Input and output

Directory mode requires `--output-dir`. The destination must not already exist.

`--assets local` copies validated assets; `inline` embeds supported images.

On success, the new directory contains page artifacts, required runtime assets, and `margo-manifest.json`.

The report on stdout contains `margo-site-report/v1`, artifact count, manifest digest, page records, and an optional policy digest. Build and publication failures go to stderr.

Config mode takes source, output, assets, site identity, base URL or path, layout, navigation, locale, theme, and publication settings from the config. Without `--output-dir`, configured output is resolved beside that file. An explicit `--output-dir` changes only the publication destination.

Minimal configured site

`margo schema` currently emits only the `policy` and `document` JSON Schemas; there is no `site` schema command. A site config is a versioned, closed YAML contract with `version: 1`. The authoritative v1 field names and validation are

the `site.Config` definition; the repository's `showcase.yaml` is a complete, larger example.

The following is a complete minimal documentation site. It expects one Markdown source at `docs/index.md`, a local SVG at `assets/logo.svg` (the same file can be used for `icon`), and a local JPEG or PNG at `assets/social.jpg`. The social image must be exactly 1280×640; Margo validates that dimension, format, and the non-empty `alt` text before publication.

```
version: 1
source: docs
output: dist
assets: local
offline: true
base_path: /docs
site:
  name: Example Docs
  description: A small configured documentation site.
  base_url: https://docs.example.com
  home: index.md
  logo: assets/logo.svg
  icon: assets/logo.svg
  social_image:
    path: assets/social.jpg
    alt: Example Docs preview
layout:
  kind: docs
  default:
    families: [default]
    sidebar: true
    toc: true
    content:
      layout: article
  values:
    family: default
locales:
  default: en
  supported: [en]
navigation:
  mode: file-tree
theme:
  builtin: true
  name: modern
```

```
allow_switch_theme: false
color_mode: light
```

Create `docs/index.md`, put the local assets in the paths above, and build a fresh destination:

```
mkdir -p docs assets build
cat > docs/index.md <<'MARKDOWN'
---
title: Example Docs
description: A small configured documentation site.
language: en
---

# Example Docs

This page is generated by the configured site contract.
MARKDOWN

margo site ./site.yaml --diagnostics json > build/site-report.json
```

The command writes `dist/index.html`, `dist/sitemap.xml`, `dist/llms.txt`, and `dist/margo-manifest.json`; the report is `margo-site-report/v1`. `base_path: /docs` affects public URLs and canonical metadata, while `output: dist` remains a filesystem path beside `site.yaml`. Set `base_path: /` for a root deployment. `theme.name: modern` selects the built-in `modern` theme; `allow_switch_theme` and `color_mode` control its runtime controls and color family. The accepted `theme.builtin` boolean is shown in the example for the v1 config shape but is not a theme selector by itself. A custom theme instead adds a `themes` entry with a local `css_url` and `token_catalog`, then selects that name. `navigation.mode` is currently `file-tree`; `locales` must include the default locale in its supported list. `layout.kind: docs` owns search, sidebar, table of contents, pagination, and family navigation. Archive, tag, RSS, and Atom generation remain consumer-owned.

Page publication actions

Pages can retain their Markdown source and publish a pre-rendered PDF through frontmatter:

```
margo:
  actions:
    markdown: true
    pdf: true
```

The PDF action also accepts an object when a branded PDF must include the exact chart-data tables used by `margo pdf --print-chart-data`:

```
margo:
  actions:
    pdf:
      printChartData: true
```

The object form defaults to `pre-rendered`; existing boolean and string forms remain compatible. Use `pdf: client` for browser printing without publishing a PDF artifact. See the [margo pdf branding guide](#) for the complete branded publication workflow.

Article publication metadata

Site builds project the document metadata fields `authors`, `publishedAt`, `modifiedAt`, and `tags` into three stable surfaces: the generated article has an accessible publication-details header (`address`, labeled `time` elements, and a tag list). Published and modified timestamps are shown as separate `Published` / `Updated` values with a visible separator when both are present; the labels wrap with their values on narrow screens and localize with the page locale. The initial HTML head contains `article:*` metadata, and each page record in the site report and `margo-manifest.json` carries the same fields. This lets a blog or news consumer build archive and tag indexes from the route records without duplicating the values in Markdown prose.

For sources that use the common singular names, directory and configured site builds also accept `author: Name` and `date: YYYY-MM-DD` (or an RFC 3339 value) as aliases. The canonical names remain preferred when multiple authors or timestamps are needed:

```
---
title: A calm release
language: en
authors: [Ana Silva]
publishedAt: "2026-08-25T12:00:00Z"
tags: [operations, release]
---
```

Margo does not generate archive, tag, RSS, or Atom pages automatically. Route records are deterministic and expose enough metadata for a consumer-owned indexer; ordering, URLs, and publication policy remain the site's responsibility.

Examples

```
margo site ./docs \
  --output-dir ./build/site-2026-08-20 \
  --assets local \
  --diagnostics json > build/site-report.json
```

```
margo site ./site.yaml --diagnostics text > build/site-report.txt
```

Failures and diagnostics

Code ↕	Meaning ↕
site.output_required	Directory mode has no --output-dir
site.output_exists	Destination already exists
site.sources_empty	No public Markdown source was found

Code ↕	Meaning ↕
site.config_invalid	YAML or closed config fields are invalid
site.identity_required	Configured public identity is incomplete
site.link_missing	A Markdown or generated link target is absent
site.anchor_missing	A target fragment has no matching heading
site.asset_external	A site image is not a local site asset
site.asset_outside_root	An asset escapes the source root
site.output_collision	Sources map to the same output path
site.artifact_collision	Generated artifact paths conflict

Margo builds and validates the complete result in a sibling staging directory, then publishes by a no-replace rename. A failure exits **1** and leaves an existing destination untouched.

Limitations and care

Use a new versioned destination for every build; there is no **--force** for a site tree. A site build is filesystem publication only. It does not deploy, upload, tag, or release the generated directory.