

1 Site Architecture

Steam Association of Manitoba — Website Architecture Overview

This document explains how the SAM static website is organized on disk, how the generator interprets that structure, and how the final HTML pages relate to the directory tree. It is written for future maintainers so the system can be understood without reading the code.

1.1 Purpose of the Architecture

The SAM website is a **static site** generated entirely from:

- a directory tree
- `metadata.txt` files
- optional `preamble.html` and `postamble.html`
- the files and subdirectories themselves

There is **no database**, **no CMS**, and **no server-side logic**. Everything is derived from the filesystem.

This makes the site:

- transparent
 - easy to maintain
 - version-controlled
 - robust
 - portable
-

1.2 High-Level Structure

The site root contains:

```
/site
  index.html
  /public
  /members
  /css
  /images
```

The generator (`sitemgen.py`) walks this tree and produces an `index.html` for **every directory**.

Each directory becomes a “page” on the website.

1.3 Directory Anatomy

Every directory may contain:

<code>metadata.txt</code>	← controls ordering, labels, headings, dividers
<code>preamble.html</code>	← optional HTML inserted before the table
<code>postamble.html</code>	← optional HTML inserted after the table
<code>index.html</code>	← generated output (never edited manually)
<code><files></code>	← PDFs, images, documents, etc.
<code><subdirectories></code>	← become links in the table

1.3.1 The generator treats each directory as a self-contained unit.

- The directory’s own `metadata.txt` controls its own index page.
 - Subdirectories appear as rows in the table.
 - Files appear as rows in the table.
 - The directory’s own `preamble.html` and `postamble.html` wrap the table.
-

1.4 How Pages Are Generated

For each directory:

1. The generator reads `metadata.txt`
2. It updates it (adding new files, commenting out missing ones)
3. It loads `preamble.html` (if present)
4. It generates a two-column HTML table
5. It loads `postamble.html` (if present)
6. It writes `index.html`

This happens recursively for every directory under `/site`.

1.5 How Navigation Works

Navigation is **implicit**:

- A directory's index page links to its subdirectories.
- The parent directory links back to it.
- The root `index.html` is the homepage.

There is no global menu.

The site is navigated like a file explorer.

This is intentional:

the site mirrors the filesystem exactly.

1.6 Directory Naming Conventions

1.6.1 Public content

`/site/public`

Contains documents intended for general visitors.

1.6.2 Member-only content

`/site/members`

Contains documents intended for SAM members.

1.6.3 Assets

`/site/css`
`/site/images`

Contain static assets referenced by all pages.

1.6.4 Special directories

Directories beginning with `/` in metadata (e.g., `/Forms`) are treated as **directory links**, not files.

1.7 How the Generator Interprets the Tree

The generator does **not** infer meaning from directory names.

It only sees:

- files
- subdirectories
- metadata

This means:

- You can reorganize content by moving directories.
- You can create new sections by creating new directories.
- You can hide sections by marking them `hide` in metadata.

The architecture is deliberately simple.

1.8 Example Directory

```
/site/members/Articles
  metadata.txt
  preamble.html
  rivets.pdf
  boilers.pdf
  index.html ← generated
```

metadata.txt might contain:

```
*Articles:
rivets.pdf: Riveting techniques
boilers.pdf: Boiler safety notes
```

The generator produces a table with:

- a heading row
 - two file rows
-

1.9 Philosophy of the Architecture

The site is designed to be:

- **filesystem-driven**
- **metadata-controlled**
- **predictable**
- **easy to maintain**
- **easy to regenerate**

There is no hidden state.

Everything the generator needs is visible in the directory tree.

This makes the site maintainable by anyone comfortable with:

- folders
- files
- simple text editing

No web development experience is required.