1 Overview

Steam Association of Manitoba — Website Overview and Workflow

This document provides a simple, non-technical explanation of how the SAM website is organized, how it is maintained, and how updates are published.
It is intended for future maintainers who may have little or no background in software or web development.

The goal is clarity, stability, and confidence.

1.1 What the website is

The SAM website is a **static website**.
This means:

- every page is just a folder
- every page contains an `index.html` file
- there is no database
- there is no programming
- everything is built from files and folders

The entire website lives inside a directory called:

```
/site
```

This directory is stored in a GitHub repository and is also kept as a local copy on the maintainer's computer.

1.2 Public vs. Member-only content

The website is divided into two major sections:

1.2.1 Public content

Stored under:

```
/site/public
```

This material is visible to anyone who visits the website.

1.2.2 Member-only content

Stored under:

```
/site/members
```

This material is **restricted**.
Access is controlled by **Cloudflare**, which checks whether a visitor's email address is listed in:

```
members.txt
```

If the email is on the list, Cloudflare grants access to the `/members` area.
If not, access is denied.

The maintainer updates this list manually (see `membership-list.md`).

1.3 How the website is updated

The workflow has three parts:

1. **Local copy** — where changes are made
2. **GitHub** — where the updated files are uploaded
3. **Cloudflare** — where the live website is automatically deployed

The maintainer only interacts with the **local copy** and the **upload script**.
GitHub and Cloudflare handle the rest automatically.

1.4 The three batch files

The maintainer uses three simple batch files:

1.4.1 `generate-site.bat`

Rebuilds the website locally.
This updates all `index.html` files based on the contents of the `/site` directory.

1.4.2 `preview-site.bat`

Starts a temporary local web server so the updated site can be viewed in a browser before publishing.

1.4.3 `upload-site.bat`

Uploads the updated files to GitHub.
Cloudflare automatically deploys the new version of the site from GitHub.

No other tools or commands are required.

1.5 How pages work

Every page on the website corresponds to a **directory** inside `/site`.

Example:

```
/site/public/Forms
/site/members/Articles
/site/members/Policies
```

Each directory contains:

- `metadata.txt` — controls what appears on the page
- `index.html` — generated automatically (never edited manually)
- optional `preamble.html` and `postamble.html`
- documents such as PDFs

The directory structure *is* the site structure.

1.6 How changes are made

There are two kinds of changes:

1.6.1 Maintaining an existing page

This includes:

- adding a new PDF
- removing an old PDF
- changing a description
- adjusting headings or dividers

The process is:

1. Edit files or metadata
2. Run `generate-site.bat`
3. Preview (optional)
4. Upload using `upload-site.bat`

1.6.2 Creating a new page

A new page is created by adding a new directory inside `/site`.

The process is:

1. Create the directory
2. Run `generate-site.bat` (first pass)
    - This creates a new `metadata.txt` with empty descriptions
    - The parent page automatically gains a link to the new page
3. Edit the new `metadata.txt`
4. Run `generate-site.bat` again (second pass)
5. Preview (optional)
6. Upload using `upload-site.bat`

This two-pass process ensures metadata remains accurate and human-controlled.

1.7 Membership list

The membership list is a simple text file:

`members.txt`

It contains one email address per line.
Cloudflare uses this file to determine who may access the `/members` area.

Updating the list is straightforward:

- adding a line adds a member
- deleting a line removes a member

To deploy changes:

`upload-site.bat`

No site generation is required.

1.8 What GitHub does

When `upload-site.bat` is run:

- the local files are uploaded to GitHub
- GitHub stores the latest version of the website
- Cloudflare automatically detects the update

The maintainer does not need to log into GitHub manually.

1.9 What Cloudflare does

Cloudflare automatically deploys the updated website from GitHub.
It also protects the `/members` area by checking the email addresses in `members.txt`.

This means:

- no manual uploading to a server
- no special tools
- no configuration changes for each update

The live site updates itself based on the files uploaded by `upload-site.bat`.

1.10 Summary

The SAM website is designed to be simple and reliable:

- The entire site is just folders and files.
- Pages are generated automatically from `metadata.txt`.
- Three batch files handle generation, preview, and upload.
- GitHub stores the site.
- Cloudflare deploys it automatically.
- Cloudflare also protects the `/members` area.
- The membership list is a simple text file.

This system avoids complexity and ensures that future maintainers can keep the site running smoothly with minimal technical knowledge.