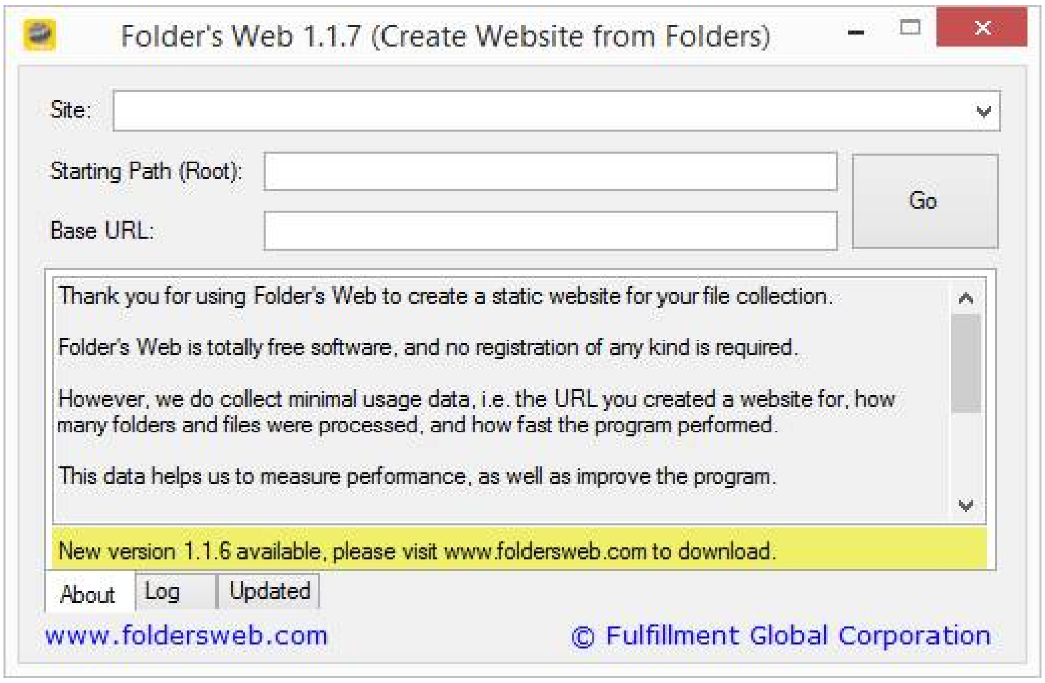
<u>Folder’s Web User Manual:</u>

Version 1.1.7
31 July 2022



Thank you for using Folder's Web to create a static website for your file collection.

Folder's Web is totally free software, and no registration of any kind is required.

However, we do collect minimal usage data, i.e. the URL you created a website for, how many folders and files were processed, and how fast the program performed.

This data helps us to measure performance, as well as improve the program.

New version 1.1.6 available, please visit www.foldersweb.com to download.

www.foldersweb.com © Fulfillment Global Corporation

Introduction:

Folder’s Web is a utility program to generate static index.html files for websites.

Download Folder’s Web here.

Folder’s Web is written in C#. .NET 6.0 is required to run Folder’s Web.

Folder’s Web walks a directory tree, given by a root folder that you specify in the program, and generates an **index.html** file in every non-empty folder. The root folder itself is also indexed. Files called **sitemap.xml** and **sitemap.xml.gz** will be placed into the root folder, which contain a sitemap of all the folders in which an **index.html** file was created by Folder’s Web.

When *Starting Path (Root)* and *Base URL* are entered, and *Go* is clicked, the *Log* tab will be populated with detailed information while the template engine processes and generates files. The *Updated* tab will be populated with the relative paths in which **index.html** and/or **sitemap** files have been generated (i.e. a list of relative paths that contain newly-generated **index.html** and/or **sitemap** files).

Multiple sites can be defined in Folder’s Web, each site having its own *Starting Path (Root)* and *Base URL*. The *Starting Path (Root)* folder must be unique for each site.

Each **index.html** file is generated based on a “master” template file in the root folder. A “header” and/or a “content” file can be placed in each folder, which will be included in the final **index.html** file, written to each folder. In case the current folder does not have a header and/or content file, then the generated header and/or content sections for the given **index.html** will be empty.

Folder’s Web can also be configured to ignore certain filenames, so that they will not be included inside the listings of the generated **index.html** files.

index.html files are not generated in empty folders. An empty folder is a folder that does not contain any listable files, either because all the ignored files in it have been filtered out, or because it does not contain any files or sub-folders on the file system. However, empty folders are still listed in their parent folders, but no anchor link is generated in the parent folder’s listing, to enter the empty folder, and neither is an **index.html** generated in the empty folder, as previously stated.

Folder’s Web’s settings file uses a JSON configuration format. The name of the settings file is hardcoded as **Foldersweb.json**, and the file should be placed in the same location as **Foldersweb.exe**. An example of **Foldersweb.json**:

```
{
    "sites": {
        "Site 1": {
            "rootFolder": "c:\\path\\to\\root\\folder1",
            "baseURL": "http://example1.com/",
            "ignoreFiles": [
                "desktop.ini",
                    "Thumbs.db"
               ]
          },
          "Site 2": {
               "rootFolder": "c:\\path\\to\\root\\folder2",
               "baseURL": "http://example2.com/",
               "ignoreFiles": [
                    "index.html"
               ]
          }
     }
}
```

Description of configuration settings:

- **sites**: a JSON object whereby each JSON property defines one site
- **rootFolder**: path to the root folder to be indexed
- **baseURL**: the base URL to use for generated listings
- **ignoreFiles**: an array of filenames to ignore as part of the generated listing in **index.html**. The following files are always ignored by default: **_content.html, _header.html, desktop.ini, e_sqlite3.dll, index.html, Foldersweb.db, Foldersweb.db-shm, Foldersweb.db-wal, Foldersweb.dll, Foldersweb.json, Foldersweb.runtimeconfig.json**

When you add a new site, change a setting, or start the indexing process, then the specified *Starting Path (Root)* folder and the specified *Base URL* for the website, is saved to **Foldersweb.json** in the same directory as **Foldersweb.exe**, and these settings are loaded back in, when Folder's Web is started again at a later time.

Folder's Web will extract and process metadata from out of certain types of files (such as PDF files), and store the processed results into **Foldersweb.db**, which is a compact SQLite database file. The database file will be automatically maintained each time Folder's Web is run, to remove old metadata as necessary.

After processing is complete, Folder's Web uploads anonymous usage information regarding the files that were generated, so that we can analyze the usage of the program, and improve the program's performance. If you do not wish to upload these usage statistics, you can add Folder's Web to your firewall, to prevent internet communication to our server. No functionality, except the automatic latest version check, will be affected if you disable internet connectivity for Folder's Web.

Template engine:

Folder’s Web uses a basic template engine to generate **index.html** files.

A single master template file is used by the engine to generate **index.html** files, and its name is hardcoded as **_template.html**. This file must be located in the root folder that is being indexed.

Inside **_template.html**, you can place four placeholder sections, which will automatically be substituted with the appropriate generated information, for the current folder being indexed. The template engine process generates and writes an **index.html** file to each folder that is indexed, as well as **sitemap** files at the end of the process.

The placeholder sections are:

- [HEADER]
  - The content of **_header.html** in the current folder, is substituted into this placeholder in the final **index.html** file. In its turn, compressed file information (see details in the ***Compressed Files*** section below), a **<link rel=”canonical”>** tag, and a **<meta name=”generator”>** tag are also inserted either just before the **</head>** tag, before the first **<meta>** tag, or otherwise prepended, into the generated header string, that is substituted into this placeholder.
  - If a **_header.html** file was found in the current folder, the generated header string substituted into this placeholder will be similar to: `<!-- Header: begin -->`*`contents of _header.html`*`<!-- Header: end -->`
  - If no **_header.html** file was found in a folder, the generated header string substituted into this placeholder will be: `<!-- Header: blank -->`
- [CONTENT]
  - If a **_content.html** file was found in the current folder, the generated content string substituted into this placeholder will be similar to: `<!-- Content: begin -->`*`contents of _content.html`*`<!-- Content: end -->`
  - If no **_content.html** file was found in the current folder, the generated content string substituted into this placeholder will be: `<!-- Content: blank -->`
- [NAVIGATION]
  - If a **_navigation.html** file was found in the current folder, and it is empty, the generated navigation string substituted into this placeholder will be: `<!-- Navigation: file blank -->`
  - If a **_navigation.html** file was found in the current folder, and it is non-empty, the generated navigation string substituted into this placeholder will be similar to: `<!-- Navigation: file begin -->`*`contents of _navigation.html`*`<!-- Navigation: file end -->`

    - If no **_navigation.html** file was found in the current folder, a breadcrumb trail to the folder being indexed will be generated, by splitting the path of the current folder into portions divided by the \ character, and writing each portion as a navigation link. For example,
        - **Root folder**: C:\My Websites\example.com\Webroot
        - **Sub-folder**: C:\My Websites\example.com\Webroot\Projects\Test Project\Source
        - **Navigation generated**:
            - `<li><a href="/Projects/">Projects</a></li>`
            - `<li><a href="/Projects/Test%20Project/">Test Project</a></li>`
            - `<li><a href="/Projects/Test%20Project/Source/">Source</a></li>`

    - If the navigation string was generated as explained above, and it is empty, the generated navigation string substituted into this placeholder will be: `<!-- Navigation: list empty -->`

    - If the navigation string was generated as explained above, and it is non-empty, the generated navigation string substituted into this placeholder will be: `<!-- Navigation: list begin -->`*`generated navigation list`*`<!-- Navigation: list end -->`

- [LINKS]

    - If a **_links.html** file was found in the current folder, and it is empty, the generated links string substituted into this placeholder will be: `<!-- Links: file blank -->`

    - If a **_ links.html** file was found in the current folder, and it is non-empty, the generated navigation string substituted into this placeholder will be similar to: `<!-- Links: file begin -->`*`contents of _links.html`*`<!-- Links: file end -->`

    - If no **_links.html** file was found in the current folder, a listing of the files and folders in the current folder being indexed will be generated. An example listing:

        - `<li class="`**`folder is-folder empty`**`">Empty folder</li>`

        - `<li class="`**`folder is-folder`**`"><a href="Non-empty%20folder/">Non-empty folder</a></li>`

        - `<li class="`**`folder compressed zip is-folder is-compressed is-zip`**`"><a href="Folder.zip">Folder</a></li>`

        - `<li class="`**`folder compressed rar is-folder is-compressed is-rar`**`"><a href="Folder.rar">Folder</a></li>`

        - `<li class="`**`file is-file is-txt is-file-txt`**`"><a href="file.txt">file.txt</a></li>`

    - Folders:
        - CSS classes: **folder / is-folder**
        - Conditional CSS classes: **compressed** / **empty / rar / zip / is-compressed / is-rar / is-zip**

    - Files:
        - CSS classes: **file / is-file** / **is-ext** / **is-file-ext** (where “ext” is the filename extension, e.g. txt or pdf, resulting in **file is-file is-txt is-file-txt**, **file is-file is-pdf is-file-pdf**, etc.).

    - If a links string was generated as explained above, and it is empty, the generated links string substituted into this placeholder will be: `<!-- Links: list empty -->`

    - If the links string was generated as explained above, and it is non-empty, the generated links string substituted into this placeholder will be: `<!-- Links: list begin -->`*`generated links list`*`<!-- Links: list end -->`

Compressed files:

If a .zip or .rar compressed file is found in the current folder, and the compressed file has the same name as a sub-folder within the current folder, then the compressed file is listed appropriately, and a <meta> tag is added to the <head> tag in the [HEADER] placeholder, for the sub-folder's generated **index.html** file. For example:

1. *C:\My Websites\example.com\Webroot\Projects\Test Project\**Folder**.zip* is found, and the folder *C:\My Websites\example.com\Webroot\Projects\Test Project\**Folder**\* exists.

2. Add **<li class="folder"><a href="Folder/">Folder</a></li><li class="folder compressed zip is-folder is-compressed is-zip"><a href="Folder.zip">Folder.zip</a></li>** to the [LINKS] placeholder for the **index.html** file in the current folder that contains **Folder.zip** and the **Folder** sub-folder.

3. Add **<meta name="download" content="{BaseURL}/Projects/Test%20Project/Folder.zip" />** to the [HEADER] placeholder for the **index.html** file in the **Folder** sub-folder (not the "current folder" in this context), where {BaseURL} is the Base URL configured on Folder's Web user interface.

A similar process is performed if **Folder.rar** was found instead of **Folder.zip**.

Sitemap:

A newly-generated **index.html** file is only written to disk, if:

- the new **index.html** file's contents will be different than the existing **index.html** file's contents; or
- the relevant **index.html** file, does not exist yet.

Sitemap files, namely **sitemap.xml** and **sitemap.xml.gz**, will be written to the root folder, after all **index.html** files have been generated and possibly written to disk, as explained above. An example **sitemap.xml**:

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
<url>
<loc>http://www.example.com/</loc>
<lastmod>2000-01-01T12:34:56+00:00</lastmod>
</url>
<url>
<loc>http://www.example.com/Projects/</loc>
<lastmod>2000-01-01T12:34:56+00:00</lastmod>
</url>
<url>
<loc>http://www.example.com/Projects/Test%20Project%20&amp;
%20Files/</loc>
<lastmod>2000-01-01T12:34:56+00:00</lastmod>
</url>
</urlset>
```

sitemap.xml.gz contains a gzip-compressed version of the contents written to **sitemap.xml**.

Links:

https://foldersweb.com
https://www.sitemaps.org/protocol.html