

Setup Guide

Copernicus Land Monitoring Service - Technical Library



Author: **European Environment Agency (EEA)**

Date: **2026-05-26**

Version: **1.0.0**

Table of contents

1. What This Sets Up	1
2. Prerequisites	1
3. Integration Steps	1
3.1 1. Add the base documentation repository as a remote	1
3.2 2. Pull in the <code>docs/</code> folder as a subtree	1
3.3 3. Configure the Git aliases	1
4. Verify the Setup	2
5. Handoff Checklist	2

Contact:

European Environment Agency (EEA)
Kongens Nytorv 6
1050 Copenhagen K
Denmark
<https://land.copernicus.eu/>

1. What This Sets Up

Running through this guide integrates an existing project repository with the CLMS Technical Library. The result is:

- a `DOCS/` directory in the project repo, linked to the central library as a git subtree
- a set of Git aliases (`git docs-publish`, `git docs-update`, `git docs-preview`, `git docs-restore`) for day-to-day documentation workflow

Once done, editors can follow the Editor Guide to create and publish documents. Your work here is a one-time setup.

2. Prerequisites

For this setup, you only need Git installed and access to the project repository.

Editors will also need the following on their machines - not your responsibility to install, but useful to pass along:

Tool	Purpose	Download
RStudio	Writing and previewing <code>.qmd</code> files	posit.co
Quarto	Document rendering engine	quarto.org

Pandoc is bundled with both RStudio and Quarto - editors don't need to install it separately.

3. Integration Steps

Run these commands from the root of the existing project repository.

3.1 1. Add the base documentation repository as a remote

```
git remote add clms-docs-base git@github.com:eea/CLMS_documents_base.git
```

3.2 2. Pull in the `DOCS/` folder as a subtree

```
git subtree add --prefix=DOCS clms-docs-base main --squash
```

This creates the `DOCS/` directory and commits it into the project repository.

3.3 3. Configure the Git aliases

Run the setup script for your operating system:

```
# macOS or Linux
./DOCS/_meta/scripts/linux/setup-docs-aliases.sh

# Windows (PowerShell)
./DOCS/_meta/scripts/win/setup-docs-aliases.ps1
```

This registers four Git aliases for the project:

Alias	What it does
<code>git docs-update</code>	Pulls updates from the central base repository into <code>DOCS/</code>
<code>git docs-publish</code>	Pushes local documentation changes to the Technical Library
<code>git docs-preview</code>	Renders the full project locally, including PDF output
<code>git docs-restore</code>	Fetches a published version of a document into the local working copy

4. Verify the Setup

i Note

TODO: A verification script is planned for this step. This section will be updated once the script is ready. In the meantime, use the manual checks below as a placeholder.

Confirm the following before handing off to editors:

- `DOCS/` exists at the project root with `templates/`, `_meta/`, `theme/`, and `includes/` inside
- `git remote -v` lists `clms-docs-base` as a remote
- `git docs-publish --help` (or equivalent) responds without an error

5. Handoff Checklist

Before passing to editors, confirm:

- All three integration steps completed without errors
- Verification checks above pass
- Editors have the Editor Guide link: [Editor Guide](#)
- Editors know they need RStudio and Quarto on their machines (see [Prerequisites](#))

The editors' starting point is the Editor Guide. They don't need to touch anything set up here.