

GitHub Initial Setup & Documentation Upload Guide v1.0.0 Simple, step-by-step guide for setting up GitHub and uploading initial project documentation. For CADMIES (Cosmium Angelo Digital Mycorrhizal Intelligence EcoSystem) development workflow.

## Table of Contents

1.0 Overview 2.0 Prerequisites 3.0 Step-by-Step Workflow 4.0 Document Management 5.0 Repository Settings 6.0 Troubleshooting (Simple Solutions) 7.0 Best Practices & Standards 8.0 Quick Reference Card Appendix A: GitHub Terminology Appendix B: Privacy Settings

## 1.0 Overview

This guide provides a complete, battle-tested workflow for setting up a GitHub repository and uploading initial project documentation. Created from our recent successful CADMIES repository setup.

## 2.0 Prerequisites

### 2.1 GitHub Account Setup

COMPLETE: GitHub account created (Hieros-CADMIES) COMPLETE: Verified email address COMPLETE: Logged into GitHub.com

### 2.2 Local System Requirements

#### Fedora Silverblue (Our Environment)

- Git already installed (git --version)
- Terminal access
- Project files ready for upload

#### Git Installation (If Needed)

```
bash # Fedora Silverblue rpm-ostree install git
```

## Verify installation

```
git --version
```

### 2.3 Project Files Prepared

COMPLETE: Documents organized in directory structure COMPLETE: Files named appropriately COMPLETE: No sensitive data in files to be uploaded 3.0 Step-by-Step Workflow 3.1 Step 1: Create GitHub Repository (Web Interface)

ACTION: Go to GitHub.com in browser

Click "+" icon in top-right then "New repository"

Fill repository details:

Repository name: CADMIES (or your project name)

Description: Brief project description (optional)

CRITICAL PRIVACY SETTING:

Private (selected) - Only you and invited collaborators can see

Public - Visible to everyone on internet

Initialization Options:

UNCHECK "Add a README file"

UNCHECK "Add .gitignore"

UNCHECK "Choose a license"

Click "Create repository"

EXPECTED OUTCOME: Empty repository created at <https://github.com/your-username/CADMIES> 3.2  
Step 2: Navigate to Project Directory (Terminal) bash

## Navigate to directory containing your documents

```
cd "/path/to/your/documents/"
```

3.3 Step 3: Initialize Local Git Repository bash

## Initialize Git in current directory

```
git init
```

## Expected output:

### Initialized empty Git repository in [path]/.git/

3.4 Step 4: Configure Git Identity (One-Time Setup) bash

## Set your email (use GitHub account email)

```
git config user.email "your-email@example.com"
```

## Set your name

```
git config user.name "Your Name"
```

3.5 Step 5: Connect to GitHub Repository bash

## Add GitHub as remote repository

```
git remote add origin https://github.com/your-username/your-repository.git
```

## Verify connection

```
git remote -v
```

3.6 Step 6: Add Specific Files to Upload bash

## Add individual files (use quotes for files with spaces)

```
git add "filename with spaces.pdf" git add "another-file.pdf"
```

## Or add all PDFs in directory

```
git add *.pdf
```

## Check what's staged for commit

```
git status
```

3.7 Step 7: Commit Files to Local History bash

## Create commit with descriptive message

```
git commit -m "Initial commit: Core project documentation PDFs"
```

## Expected output shows files committed

3.8 Step 8: Push to GitHub bash

## Push to GitHub (will ask for credentials)

```
git push -u origin master
```

## If authentication fails, use Personal Access Token

### (see Troubleshooting section 6.1)

3.9 Step 9: Create README.md (Optional but Recommended) bash

## Create README file

```
cat > README.md « 'EOF' # Project Name
```

### Overview

Brief project description.

### Documents

List of uploaded documents.

### Status

Project status information.

### Contact

For inquiries: your-email@example.com EOF

## Add and commit README

`git add README.md` `git commit -m "Add project README"`

## Push README to GitHub

`git push origin master`

### 4.0 Document Management 4.1 File Naming Standards

RECOMMENDED FORMAT: text

YYYY-MM-DD\_document-title\_vX.Y.Z.pdf YYYY-MM-DD\_document-title\_vX.Y.Z.md

### 4.2 What to Include in Initial Upload

CORE DOCUMENTS (Start with these):

Project overview/manifesto

Technical specifications

Architecture documentation

Development roadmap

SUPPORTING FILES:

README.md (project description)

.gitignore (privacy protection)

### 4.3 Privacy Checklist

CHECK: No API keys or passwords in files CHECK: No personal identifying information CHECK: No confidential business data CHECK: Repository set to Private (if needed) CHECK: .gitignore excludes sensitive files 5.0 Repository Settings 5.1 Privacy Management

TO CHANGE REPOSITORY VISIBILITY:

Go to repository on GitHub.com

Click Settings tab

Scroll to "Danger Zone"

Click "Change repository visibility"

Select Public or Private

Confirm changes

IMPORTANT: Making repository public makes ALL content public. 5.2 Branch Protection (For Teams)

For collaborative projects:

Settings then Branches

Add branch protection rule

Protect main/master branch

Require pull request reviews

Require status checks

### 5.3 GitHub Pages (For Project Websites)

SIMPLE STATIC SITE:

Settings then Pages

Source: Deploy from a branch

Branch: main or gh-pages

Folder: /root or /docs

Save

### 6.0 Troubleshooting (Simple Solutions) 6.1 Authentication Issues

PROBLEM: “Authentication failed” when pushing

SOLUTION: Use Personal Access Token instead of password:

Go to GitHub.com then Settings then Developer settings

Personal access tokens then Tokens (classic)

Generate new token then Generate new token (classic)

Select scopes: repo (full control)

Copy token (treat as password)

When prompted for password, paste token

### 6.2 File Not Found Errors

PROBLEM: “fatal: pathspec ‘file.pdf’ did not match any files”

SOLUTION: bash

## Check current directory

pwd

## List files

ls -la

## Verify file exists

ls “filename.pdf”

### 6.3 Wrong Directory Issues

PROBLEM: Git initialized in wrong directory

SOLUTION: bash

## Navigate to correct directory first

```
cd "/correct/path/"
```

## Remove .git from wrong directory (if needed)

```
rm -rf "/wrong/path/.git"
```

## Re-initialize in correct directory

```
git init
```

### 6.4 Connection Issues

PROBLEM: "remote origin already exists"

SOLUTION: bash

## Remove existing remote

```
git remote remove origin
```

## Add correct remote

```
git remote add origin https://github.com/username/repo.git
```

### 7.0 Best Practices & Standards 7.1 Commit Messages

USE DESCRIPTIVE MESSAGES:

```
git commit -m "Add initial project PDFs"
```

```
git commit -m "Update README with project description"
```

```
git commit -m "Fix: Correct filename in document list"
```

AVOID VAGUE MESSAGES:

```
git commit -m "update"
```

```
git commit -m "fix"
```

```
git commit -m "changes"
```

### 7.2 File Organization text

project-repo/ - README.md # Main project description - docs/ # Documentation folder - overview.pdf # Main documents - specs.pdf # Specifications - roadmap.pdf # Development plans - src/ # Source code (if applicable) - .gitignore # Privacy protection

### 7.3 Privacy First Approach

START PRIVATE: Always create repository as Private initially

REVIEW CONTENT: Ensure no sensitive data before making public

USE .GITIGNORE: Exclude configuration files, logs, secrets

REGULAR AUDITS: Periodically review repository contents

7.4 Documentation Standards

EVERY REPOSITORY SHOULD HAVE:

README.md with project overview

Clear document naming convention

Contact information

Status indicator (Active, Archived, etc.)

8.0 Quick Reference Card 8.1 Essential Commands bash

## Initialize repository

git init

## Connect to GitHub

git remote add origin <https://github.com/username/repo.git>

## Add files

git add "filename.pdf" git add \*.pdf

## Commit changes

git commit -m "Descriptive message"

## Push to GitHub

git push -u origin master

## Check status

git status

## View repository history

git log --oneline

8.2 GitHub URLs

YOUR PROFILE: <https://github.com/username>

YOUR REPOSITORY: <https://github.com/username/repo>

REPOSITORY SETTINGS: <https://github.com/username/repo/settings>

ACCESS TOKENS: <https://github.com/settings/tokens>

### 8.3 Quick Setup Sequence bash

```
cd /project/directory git init git config user.email "you@example.com" git config user.name "Your Name"
git remote add origin https://github.com/username/repo.git git add "document1.pdf" "document2.pdf" git
commit -m "Initial documents" git push -u origin master
```

## Appendix A: GitHub Terminology A.1 Core Concepts

REPOSITORY (REPO): Project folder containing all files and revision history

COMMIT: A saved change to files (like "Save As" with notes)

PUSH: Upload local commits to GitHub

PULL: Download changes from GitHub to local

BRANCH: Parallel version of repository for experimentation

MASTER/MAIN: Primary branch (default working version)

### A.2 Repository Types

PRIVATE: Only visible to you and collaborators you invite

PUBLIC: Visible to anyone on the internet

FORK: Your personal copy of someone else's repository

TEMPLATE: Pre-configured repository for starting new projects

### A.3 Account Types

PERSONAL ACCOUNT: Individual user (Hieros-CADMIES)

ORGANIZATION ACCOUNT: For teams/projects

ENTERPRISE ACCOUNT: For companies/institutions

## Appendix B: Privacy Settings B.1 Default Settings (Recommended)

### FOR NEW PROJECTS:

Repository: Private

Branch protection: Disabled (enable for teams)

Issues: Enabled

Wikis: Disabled (use README.md instead)

Projects: Disabled (enable if using project boards)

### B.2 Changing Privacy Levels

#### PRIVATE TO PUBLIC:

Makes all content publicly accessible



Can be reversed (Public to Private)

Historical commits remain public

PUBLIC TO PRIVATE:

Hides repository from public view

Existing clones remain accessible

GitHub Pages site goes offline

B.3 What's Always Public

Even with private repository:

Repository name

Description (if provided)

Your username

That the repository exists (but not contents)

Document Change Log

VERSION 1.0.0 (2025-12-31)

Initial release based on CADMIES repository setup

Step-by-step workflow from repository creation to upload

Includes troubleshooting from actual implementation

Privacy-first approach emphasized

Quick reference card for essential commands

CADMIES Integration Note

This guide represents the initial mycelial node of CADMIES documentation on GitHub. As a public repository, it demonstrates our commitment to transparent development while maintaining core proprietary work in protected environments. The guide itself follows the scientific documentation standards it describes, creating a recursive pattern of documentation about documentation.

End of GitHub Initial Setup & Documentation Upload Guide v1.0.0 CADMIES Framework - Cosmium Angelo Digital Mycorrhizal Intelligence EcoSystem For inquiries of any kind, email: [hieroscadmies@proton.me](mailto:hieroscadmies@proton.me) Let the mycelium grow!