

The background of the slide features a vibrant, abstract design. It consists of several diagonal streaks of light and color, primarily in shades of blue, green, and red, set against a dark blue background. The streaks have a soft, glowing quality, suggesting a digital or data-related theme.

USING GITHUB

A basic introduction

Matthew R. Rowles

Outline

- Background
- Basic navigation
 - Looking at dictionaries
 - Other features of the interface
- Issues
 - Creating issues
 - Looking at open and closed issues
- Forking to your own area
- PRs
 - Looking at PRs, open and closed
 - What they mean
- Branches
- Running automated checks
- Put all this (new) knowledge to work!

Github

- Founded in April 2008 by Tom Preston-Werner, Chris Wanstrath, PJ Hyett.
- Built on top of Git, providing a web-based interface for Git repositories.
- Enables easy code sharing, contribution, and tracking.
- Acquired by Microsoft in 2018, but operates independently.
- Online repositories for (text) files
- Issue tracking
- Continuous integration
- Code collaboration
- Free and paid access available
- Repositories don't need to be public

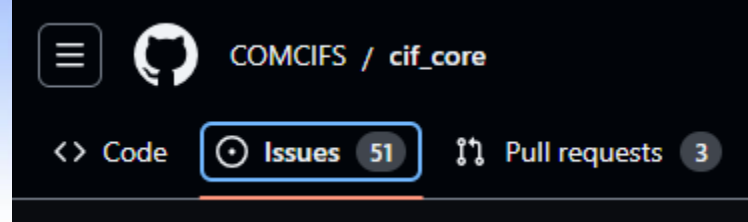
Git

- Distributed version control system
- Created by Linus Torvalds in 2005
- Developed for managing Linux kernel source code
- Manages code changes and history
- Tracks edits, additions, and deletions
- Enables collaboration among developers
- Supports branching and merging
- Very much a command-line interface
- education.github.com/git-cheat-sheet-education.pdf

Basic navigation

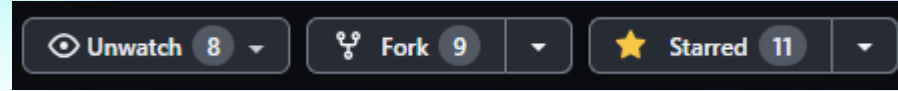
- https://github.com/COMCIFS/cif_core

Issues



- A place to document and discuss tasks or problems, or ask questions
- Issues may turn into PRs (more on that later)
- They can be labelled, assigned, opened and closed
- https://github.com/COMCIFS/cif_core/issues

Forking



- Also known as “cloning”
- Get your own copy of a repository
- You must do this if you want to contribute to a repository
- Just click the “Fork” button in the repository

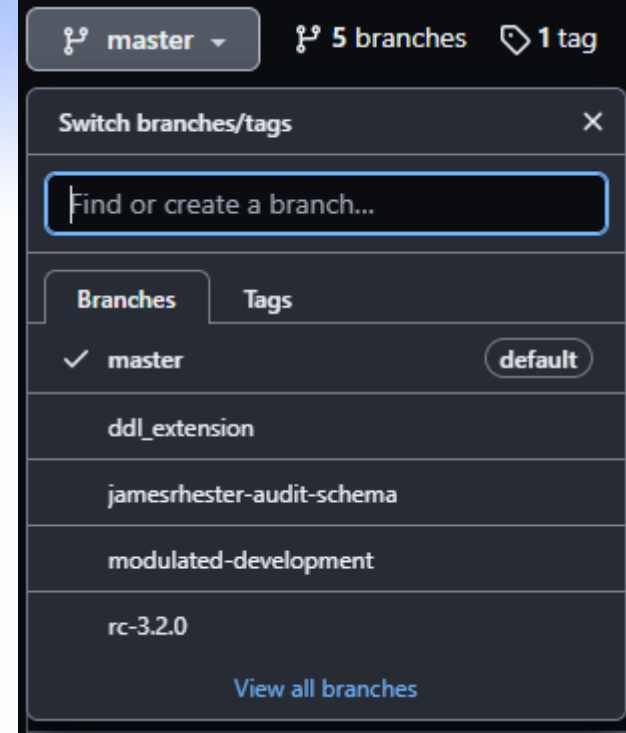
Pull requests



- “PRs”
- You have written some new code/addition and want it added to the main repository
- You first commit changes to your own fork of the repository, and then ask the original to pull across your changes
- Code can be reviewed, altered, accepted, rejected
- Finally ends up (hopefully) being merged into the main branch
- https://github.com/COMCIFS/cif_core/pulls

Branches

- Git branches are separate lines of development within a repository
- Enable multiple tasks or features to be worked on simultaneously
- Each branch maintains its own version of code changes
- You can create, switch, and merge branches
- Changes in one branch do not affect others until merged
- Useful for testing, experimenting, and implementing new features
- Helps prevent disruptions in the main code while developing



PRs and branches

- Create a new branch for each feature
- Main should stay stable and good to go
- Do all development in the branch
 - Regularly commit your changes
 - Use descriptive messages
- Open a PR on your branch
- Your addition is reviewed

Automated checks

- COMCIFS runs automated checks on each commit
- File formatting, syntax, validity, linting...
- Maintains consistency, helps identify bugs
- Helps focus reviews on content, rather than presentation

Commits

- Commits are snapshots of code changes within a repository
- Each commit represents a specific set of modifications, with a descriptive message
- Commits help track the history of code evolution
- Essential for understanding, reverting, and collaborating on changes
- Commits should be frequent and granular
- Can be associated with branches and merged into the main codebase
- Form the backbone of version control and project history

Github Desktop

- GitHub Desktop is a GUI application
- Designed to simplify interaction with Git repositories
- Offers a visual representation of code changes and history
- Supports easy creation, branching, and merging of code
- Simplifies tasks like pull requests and code reviews
- Available for Windows and macOS operating systems
- Can't do everything that Git can do, but is pretty good

Now you get to do some work!

- Each person/group gets a microsymposium dictionary data name
- Fork the repository, make a new branch, and add the data name
- Create a PR to the original repository