

Submodule Pain - or - How I Learned to Stop Worrying and Love the Subtree

Jackson Oursland



git Submodules

- git submodules are a means of pointing to an external repository.
- Submodules have a few components:
 - .gitmodules entry with the remote repo URL and path to check out
 - .git/modules/ directory which stores the .git of the module
 - reference in the index and objects which points to the SHA of the commit to be checked out
- Note that the files themselves are NOT tracked, only the commit reference which is typically a SHA.

Reference Repositories

- oursland/submodule-pain
 - Top-level repo
- oursland/submodule-pain-module
 - Submodule

PRs Take One: Branches

- ourslan/submodule-pain-module
 - PR#1: Branch 'a' adds the file 'a'
 - PR#2: Branch 'b' adds the file 'b'
 - There are no possible conflicts
- ourslan/submodule-pain
 - Branch 'master'
 - 81b3d609d3158822a6a37966436f5c64e1eb4ac6 module (heads/master)
 - PR#1: Branch 'a' updates the submodule at branch 'a'
 - c7c5d8c0d28bade1ba45706bf49d85df57771947 module (heads/a)
 - PR#2: Branch 'b' updates the submodule at branch 'b'
 - 75b743cf403b92c8f631d419490f04eef9832d1b module (heads/b)

Submodule repo PRs

The screenshot shows the GitHub interface for the repository 'oursland / submodule-pain-module'. The 'Pull requests' tab is selected, showing 2 open pull requests. A notification banner at the top encourages labeling issues and pull requests for new contributors with the 'good first issue' label. Below the banner, there are filters for 'is:pr is:open', 9 labels, and 0 milestones. A 'New pull request' button is also visible. The list of pull requests includes two items: 'Add b.' (opened 7 minutes ago) and 'Add a.' (opened 8 minutes ago). A 'ProTip!' at the bottom suggests excluding one's own issues with the filter '-author:oursland'.

oursland / submodule-pain-module

Code Issues **Pull requests 2** Actions Projects Wiki Security

Label issues and pull requests for new contributors [Dismiss](#)

Now, GitHub will help potential first-time contributors [discover issues](#) labeled with **good first issue**

Filters Labels 9 Milestones 0 [New pull request](#)

2 Open ✓ 0 Closed

<input type="checkbox"/>	Author	Label	Projects	Milestones	Reviews	Assignee	Sort
<input type="checkbox"/>	Add b.						
	#2 opened 7 minutes ago by oursland						
<input type="checkbox"/>	Add a.						
	#1 opened 8 minutes ago by oursland						

ProTip! Exclude your own issues with [-author:oursland](#).

Top-level repo PRs

The screenshot shows the GitHub interface for the repository 'oursland / submodule-pain'. The top navigation bar includes the repository name, search, and various icons. Below the navigation bar, the 'Pull requests' tab is selected, showing 2 open pull requests. A notification banner at the top of the pull request list encourages labeling issues and pull requests for new contributors. The pull request list includes two items: 'Use module at b.' and 'Use module at a.', both opened by 'oursland' 1 minute ago. A 'ProTip!' message at the bottom suggests updating the pull request with the date '2025-04-08'.

oursland / submodule-pain

<> Code Issues **Pull requests 2** Actions Projects Wiki Security

Label issues and pull requests for new contributors [Dismiss](#)

Now, GitHub will help potential first-time contributors [discover issues](#) labeled with [good first issue](#)

Filters Labels 9 Milestones 0 [New pull request](#)

2 Open 0 Closed

<input type="checkbox"/>	Author	Label	Projects	Milestones	Reviews	Assignee	Sort
<input type="checkbox"/>	Use module at b.						
	#2 opened 1 minute ago by oursland						
<input type="checkbox"/>	Use module at a.						
	#1 opened 1 minute ago by oursland						

ProTip! Updated in the last three days: [updated:>2025-04-08](#).

[Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact](#) [Manage cookies](#) [Do not share my personal information](#)

© 2025 GitHub, Inc.

Merge 'a' into Top-Level Repo

The screenshot shows a GitHub pull request interface for the repository 'oursland / submodule-pain'. The pull request is titled 'Use module at a. #1' and is in the 'Merged' state. The merge message indicates that 'oursland merged 1 commit into master from a' 1 minute ago. The pull request details show 0 conversations, 1 commit, 0 checks, and 1 file changed. A comment from 'oursland' 4 minutes ago states 'No description provided.' The commit history shows a merge of commit '19a94df' into 'master' 1 minute ago. A sidebar on the right lists various settings: Reviewers (No reviews), Assignees (No one—assign yourself), Labels (None yet), Projects (None yet), Milestone (No milestone), and Development. A bottom notification box states 'Pull request successfully merged and closed' and provides instructions on deleting the branch.

oursland / submodule-pain

<> Code Issues Pull requests 1 Actions Projects Wiki Security

Use module at a. #1

Edit <> Code

Merged oursland merged 1 commit into master from a 1 minute ago

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -1

oursland commented 4 minutes ago Owner

No description provided.

Use module at a. 138b875

oursland merged commit 19a94df into master 1 minute ago Revert

Pull request successfully merged and closed Delete branch

You're all set — the a branch can be safely deleted.

Reviewers
No reviews

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Development

Now the the 'b' PR cannot merge due to "conflicts"

The screenshot shows a GitHub pull request for the repository 'oursland / submodule-pain'. The pull request is titled 'Use module at b. #2' and is in the 'Open' state. It shows that 'oursland' wants to merge 1 commit into 'master' from branch 'b'. The pull request has 1 commit, 0 checks, and 1 file changed. A comment from 'oursland' is visible, stating 'No description provided.' Below the comment, a conflict error message is displayed, indicating that the branch has conflicts that must be resolved before merging. The error message includes instructions on how to resolve conflicts using the command line and a link to view command line instructions. The error message is highlighted with a red box.

oursland / submodule-pain

<> Code Issues Pull requests 1 Actions Projects Wiki Security

Use module at b. #2

Edit <> Code

Open oursland wants to merge 1 commit into master from b

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -1

oursland commented 3 minutes ago Owner

No description provided.

Use module at b. bade5e4

This branch has conflicts that must be resolved Resolve conflicts

Use the command line to resolve conflicts before continuing.

module

Merge pull request You can also merge this with the command line. [View command line instructions.](#)

Reviewers: No reviews, Still in progress? [Convert to draft](#)

Assignees: No one—[assign yourself](#)

Labels: None yet

Projects: None yet

Milestone: No milestone

Development: Successfully merging this pull request may

Recap

- No actual conflicts are present.
- One PR added the file 'a' and the other added the file 'b', but because they are submodules git has no knowledge of the files or their compatibility.
- Because each PR uses a different SHA, then the merge cannot take place because the conflict is in the SHA values.
 - PR#1: 81b3d609d3158822a6a37966436f5c64e1eb4ac6 -> c7c5d8c0d28bade1ba45706bf49d85df57771947
 - PR#2: 81b3d609d3158822a6a37966436f5c64e1eb4ac6 -> 75b743cf403b92c8f631d419490f04eef9832d1b

PRs Take Two: Default Branch Only

- Okay, normally we do not want people to have submodules at branches which may not have been reviewed, but rather to only accept changes from:
 - tags
 - commits on default branch
 - commits on a release tracking branch
- Let's fix that with two new PRs
 - Based upon the current `master`, so 'module' now points to 'a'
 - PR#3: adds the file 'a', but after being merged into 'master'
 - PR#4: adds the file 'b', but after being merged into 'master'

PRs Take Two: Default Branch Only

- ourslan/submodule-pain-module
 - Adds the file 'a' after merging PR#1 into submodule
 - Adds the file 'b' after merging PR#2 (both 'a' and 'b' are present)
- ourslan/submodule-pain
 - PR#3: Branch 'a2' updates the submodule at branch 'master' after PR#1
 - c78ac2db2eb6ca7df9877249135affe0411cf853 module (heads/master)
 - PR#4: Branch 'b2' updates the submodule at branch 'master' after PR#2
 - 5f015926eae1ba7100d6fb34c46b8e18b10ed5ec module (heads/master)

Merge PR#1 into Submodule

The screenshot displays a GitHub pull request interface for the repository 'oursland / submodule-pain-module'. The pull request, titled 'Add a. #1', has been successfully merged into the 'master' branch. The interface includes a top navigation bar with tabs for Code, Issues, Pull requests (selected), Actions, Projects, Wiki, and Security. Below the title, a status bar indicates the merge: 'oursland merged 1 commit into master from a 17 minutes ago'. A summary bar shows 'Conversation 0', 'Commits 1', 'Checks 0', and 'Files changed 1' with a net change of '+0 -0'. The main content area features a comment from 'oursland' stating 'No description provided.' and a commit history showing the merge of commit 'c78ac2d' into 'master'. A 'Revert' button is visible next to the merge commit. On the right, a sidebar lists settings for Reviewers, Assignees, Labels, Projects, Milestone, and Development. A bottom notification box confirms the pull request was successfully merged and closed, suggesting the branch can be safely deleted.

oursland / submodule-pain-module

<> Code Issues Pull requests Actions Projects Wiki Security

Add a. #1

Edit <> Code

Merged oursland merged 1 commit into master from a 17 minutes ago

Conversation 0 Commits 1 Checks 0 Files changed 1 +0 -0

oursland commented 53 minutes ago Owner

No description provided.

Add a. c7c5d8c

oursland merged commit c78ac2d into master 17 minutes ago Revert

Pull request successfully merged and closed Delete branch

You're all set — the a branch can be safely deleted.

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Development

Create PR#3 in Top-Level Project

- Submodule now has 'a' in the 'master' branch after code-review
- Submit PR#3 with the updated submodule

Merge PR#2 into Submodule

The screenshot displays a GitHub pull request interface for the repository 'oursland / submodule-pain-module'. The pull request, titled 'Add b. #2', has been successfully merged into the 'master' branch. The interface includes a top navigation bar with tabs for Code, Issues, Pull requests (active), Actions, Projects, Wiki, and Security. Below the title, a status bar indicates the merge: 'oursland merged 1 commit into master from b' 9 minutes ago. A summary bar shows 0 conversations, 1 commit, 0 checks, and 1 file changed. The main content area features a comment from 'oursland' 54 minutes ago with the text 'No description provided.' and a commit history showing the merge of commit 5f01592 into master 9 minutes ago. A 'Revert' button is visible next to the merge commit. On the right, a sidebar lists various settings: Reviewers (No reviews), Assignees (No one—assign yourself), Labels (None yet), Projects (None yet), Milestone (No milestone), and Development. A bottom notification box states: 'Pull request successfully merged and closed. You're all set — the b branch can be safely deleted.' with a 'Delete branch' button.

oursland / submodule-pain-module

<> Code Issues Pull requests Actions Projects Wiki Security

Add b. #2

Edit <> Code

Merged oursland merged 1 commit into master from b 9 minutes ago

Conversation 0 Commits 1 Checks 0 Files changed 1 +0 -0

oursland commented 54 minutes ago **Owner** ...

No description provided.

75b743c

oursland merged commit 5f01592 into master 9 minutes ago **Revert**

Pull request successfully merged and closed **Delete branch**

You're all set — the b branch can be safely deleted.

Reviewers
No reviews

Assignees
No one—[assign yourself](#)

Labels
None yet

Projects
None yet

Milestone
No milestone

Development

Create PR#4 in Top-Level Project

- Submodule now has 'b' in the 'master' branch after code-review
- Submodule now has both 'a' and 'b'
- Submit PR#4 with the updated submodule

Merge PR#3 into Top-Level

The screenshot displays a GitHub pull request interface for the repository 'oursland / submodule-pain'. The pull request title is 'Add module after merging PR#1, adding file 'a'. #3'. The status is 'Merged', indicating that the pull request has been successfully merged into the 'master' branch from the 'a2' branch. The commit hash is 'c440ceb'. The interface includes a navigation bar with tabs for Code, Issues, Pull requests (2), Actions, Projects, Wiki, and Security. A summary bar shows 1 commit, 0 checks, and 1 file changed. A comment from 'oursland' is visible, stating 'No description provided.' The right sidebar contains metadata sections for Reviewers, Assignees, Labels, Projects, Milestone, and Development. A notification box at the bottom left confirms the successful merge and provides instructions to delete the 'a2' branch.

oursland / submodule-pain

<> Code Issues Pull requests 2 Actions Projects Wiki Security

Add module after merging PR#1, adding file 'a'. #3

Edit <> Code

Merged oursland merged 1 commit into master from a2 now

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -1

oursland commented 4 minutes ago Owner

No description provided.

Add module after merging PR#1, adding file 'a'. c440ceb

oursland merged commit 8a94776 into master now Revert

Pull request successfully merged and closed Delete branch

You're all set — the a2 branch can be safely deleted.

Reviewers: No reviews

Assignees: No one—assign yourself

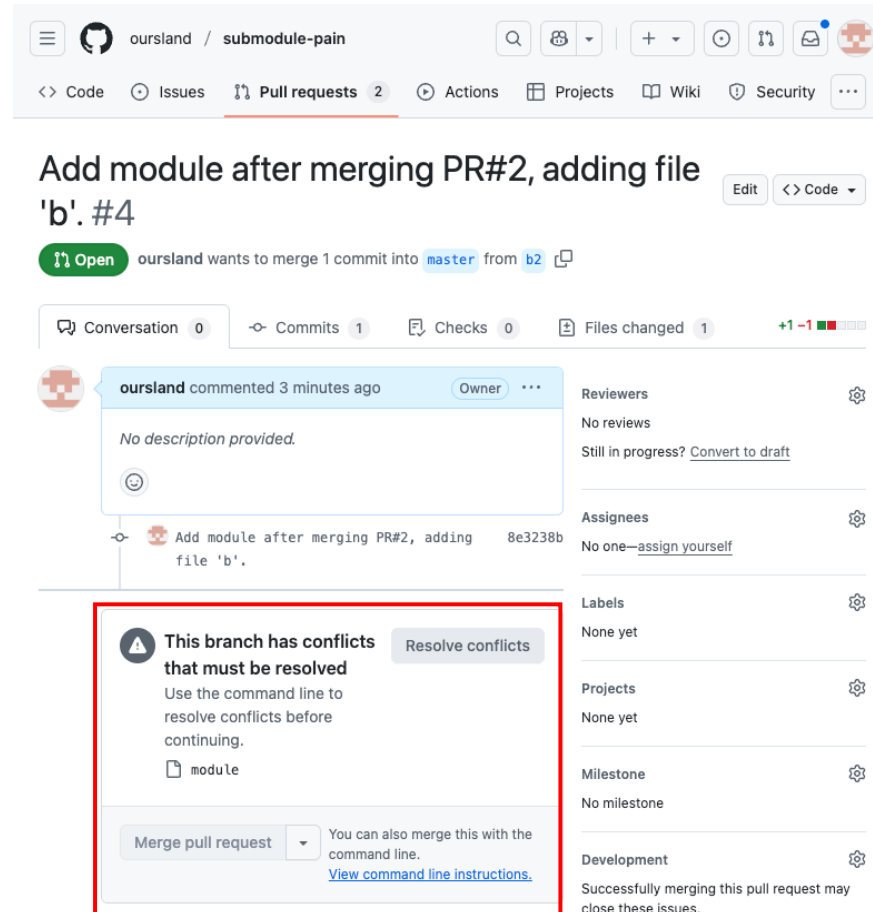
Labels: None yet

Projects: None yet

Milestone: No milestone

Development: Successfully merging this pull request may

Cannot merge PR#4 due to “conflicts”



The screenshot shows a GitHub pull request interface for a repository named 'oursland / submodule-pain'. The pull request title is 'Add module after merging PR#2, adding file 'b'. #4'. It is a pull request from branch 'b2' to 'master'. The interface includes tabs for Code, Issues, Pull requests (2), Actions, Projects, Wiki, and Security. Below the title, there is a comment from 'oursland' stating 'No description provided.' and a commit message 'Add module after merging PR#2, adding file 'b'.' with commit hash '8e3238b'. A red box highlights a conflict message: 'This branch has conflicts that must be resolved'. The message instructs the user to use the command line to resolve conflicts before continuing. It also provides a link to 'View command line instructions.' and a button to 'Merge pull request'. On the right side, there are sections for Reviewers, Assignees, Labels, Projects, Milestone, and Development, all of which are currently empty or show no activity.

oursland / submodule-pain

<> Code Issues Pull requests 2 Actions Projects Wiki Security

Add module after merging PR#2, adding file 'b'. #4

Open oursland wants to merge 1 commit into master from b2

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -1

oursland commented 3 minutes ago Owner

No description provided.

Add module after merging PR#2, adding file 'b'. 8e3238b

This branch has conflicts that must be resolved Resolve conflicts

Use the command line to resolve conflicts before continuing.

module

Merge pull request You can also merge this with the command line. [View command line instructions.](#)

Reviewers: No reviews. Still in progress? [Convert to draft](#)

Assignees: No one—[assign yourself](#)

Labels: None yet

Projects: None yet

Milestone: No milestone

Development: Successfully merging this pull request may close these issues.

Again Conflicts

- As before, git submodules do not track actual code changes and cannot perform merges
- The submodule's reference has changed in multiple PRs. After the first is merged, every following PR that touches that module will require manual resolution
- Increasing the number of submodules increases the likelihood of these conflicts

How this affects FreeCAD

- Proliferation of submodules increases the likelihood that a PR will cause a cascade of merge conflicts requiring manual resolution
- Even simple changes like mosfet80's frequent code cleanup PRs will invalidate every PR that takes place later in the chain
- Merge Request Mondays could be brought to a halt
- As davesrocketshop can attest to, weekly manual conflict resolution is no fun and does not push the project forward

How to Solve This?

- Let git do what it is meant to do
 - Track source code in a line-oriented way, not SHAs
 - Perform automatic merges when changes do not conflict
- Check in the source
 - Vendoring a release is an option, but it makes tracking local changes challenging and submitting patches upstream harder
 - Branches for vendored releases can be merged, which make managing the local repo easier, but still does not make submitting patches upstream any easier

How to Solve This? (cont.)

- Check in the source (cont.)
 - Vendoring a remote repo is another option
 - Checkout a clone into the host repo (you can have multiple roots in a git repo!)
 - Use `git read-tree` to check-out the source at a given reference to a subdirectory
 - Use `git filter-branch` to reconstruct patches for submission upstream
 - `git read-tree` and `git filter-branch` are both arcane magick that should not be used casually. `git filter-branch`'s man page even comes with a massive warning.
 - Use `git subtree`

git Subtree

- `git subtree add --prefix=path/to/subdirectory/ repo-url branch-name`
 - Will check out the repo at the desired branch and place it into the subdirectory.
 - This is done with a merge commit, preserving the history of the files.
 - Adding `--squash` will squash all commits to a single commit that is merged.
 - The merge commit's message contains references to path of subtree, current repo when the subtree is added, and the remote repo when it was merged

```
Add 'src/Mod/Assembly/OndselSolver/' from commit  
'09d6175a2ba69e7016fcecc4f384946a2f84f92d'
```

```
git-subtree-dir: src/Mod/Assembly/OndselSolver  
git-subtree-mainline: 5eb5af7068ac9e610ddb3c62f6a5a54c7c3f818  
git-subtree-split: 09d6175a2ba69e7016fcecc4f384946a2f84f92d
```

git Subtree Pulling in Upstream Changes

- `git subtree pull --prefix=path/to/subtree remote-repository branch-name`
 - Will pull in changes from the upstream repo
 - The references in the prior merge commit are used to select which changes to bring
- `git subtree push --prefix=path/to/subtree remote-repository branch-name`
 - Will push changes to the upstream repository
 - References in the prior merge commit are used to identify commits that are to be pushed upstream

Removing a git Subtree

- Just run `git rm -r path/to/subtree`
 - The subtree is just normal code. The metadata is in the merge commit message.
- Contrast that to the git submodule deletion

```
git submodule deinit -f -- path/to/submodule
rm -rf path/to/submodule
git rm -f path/to/submodule
```

 - Then commit
 - Note: switching branches which still have the submodule can cause problems