

Before merging your pull request, the source repository adds a conflicting commit. Now what?

"pr_sync() is a shortcut for pr_pull(), pr_pull_upstream(), and pr_push()" ([usetheis](#))

https://twitter.com/mauro_lepore

License: [CCO](#)

usetheis::pr_sync()

create_from_github()

```
Console Terminal x Jobs x
C:/Users/Mauro/Desktop/CONTRIBUTOR/abc/ ↗
> create_from_github("an-org/abc", fork = TRUE)
✓ Creating 'C:/Users/Mauro/Desktop/abc/'
✓ Forking 'an-org/abc'
✓ Cloning repo from 'git@github.com:maurolepore/abc.git' into 'C:/Users/Mauro/Desktop/abc'
✓ Setting active project to 'C:/Users/Mauro/Desktop/abc'
✓ Adding 'upstream' remote: 'git@github.com:an-org/abc.git'
✓ Pulling changes from GitHub source repo 'upstream/master'
✓ Setting remote tracking branch for local 'master' branch to 'upstream/master'
✓ Opening 'C:/Users/Mauro/Desktop/abc/' in new RStudio session
✓ Setting active project to '<no active project>'
> |
```

Fork on GitHub, then run from a terminal:

```
git clone git@github.com:maurolepore/abc.git
cd abc
```

Contrib@LAPTOP ~/abc (master)

```
git remote add upstream git@github.com:an-org/abc.git
git pull upstream master
git branch --set-upstream-to=upstream/master
```

git_sitrep()

```
Console Terminal x Jobs x
C:/Users/Mauro/git/abc-contributor/
> library(usethis)
> usethis::git_sitrep()
Git user
* Name: 'Mauro Lepore'
* Email: 'maurolepore@gmail.com'
* Vaccinated: FALSE
usethis + git2r
* Default usethis protocol: 'ssh'
* git2r supports SSH: TRUE
* Credentials: '<user-provided git2r credential object with class cred_ssh_key>'
GitHub
* Personal access token: '<found in env var>'
* User: 'maurolepore'
* Name: 'Mauro Lepore'
* Email(s): '<unknown>'
Repo
* Path: 'C:/Users/Mauro/git/abc-contributor/.git'
* Local branch → remote tracking branch: 'master' → 'upstream/master'
GitHub pull request readiness
* origin: maurolepore/abc, can push, forked from an-org/abc
* upstream: an-org/abc, can push
>
```

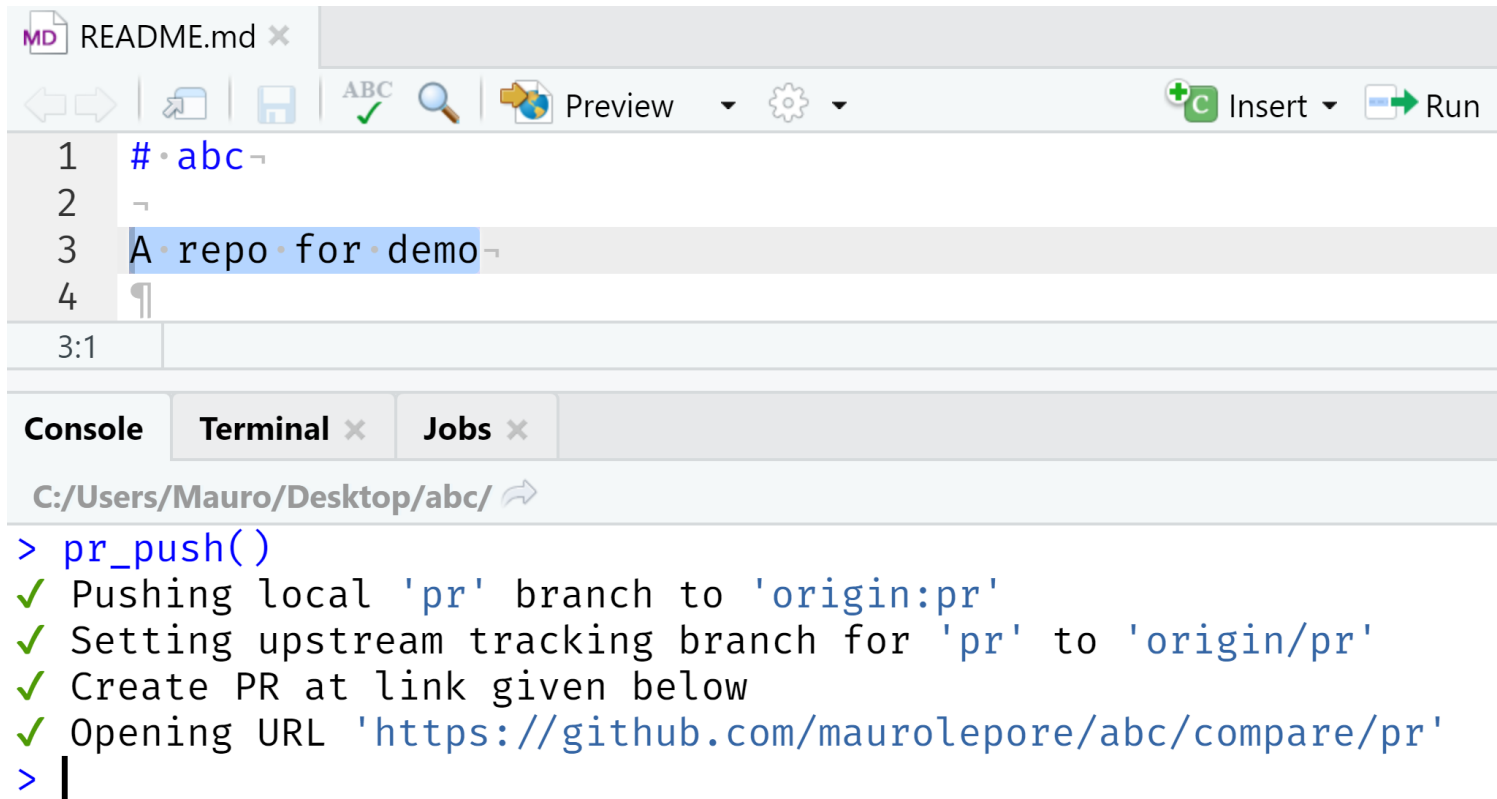
git remote -v

pr_init("pr")

```
Console Terminal x Jobs x
C:/Users/Mauro/Desktop/abc/ ↗
> pr_init("pr")
✓ Setting active project to 'C:/Users/Mauro/Desktop/abc'
✓ Checking that local branch 'master' has the changes in 'upstream/master'
✓ Creating local PR branch 'pr'
✓ Switching to branch 'pr'
● Use `pr_push()` to create PR
> |
```

```
# Contrib@LAPTOP ~/abc (master)
git pull upstream master
git branch pr
git checkout pr
```

pr_push()



The screenshot shows a code editor with a file named 'README.md'. The editor has a toolbar with icons for navigation, search, and a 'Preview' button. The code in the editor is as follows:


```
1 # abc
2
3 A repo for demo
4
```

Below the editor is a terminal window with tabs for 'Console', 'Terminal', and 'Jobs'. The terminal shows the execution of the `pr_push()` command and its output:

```
> pr_push()
✓ Pushing local 'pr' branch to 'origin:pr'
✓ Setting upstream tracking branch for 'pr' to 'origin/pr'
✓ Create PR at link given below
✓ Opening URL 'https://github.com/maurolepore/abc/compare/pr'
> |
```

```
# Contrib@LAPTOP ~/abc (pr)
git push origin
git branch --set-upstream-to=origin/pr
```

<https://github.com/maurolepore/abc/compare/pr>

 base repository: an-org/abc ▼


base: master ▼

←

head repository: maurolepore/abc ▼

compare: pr ▼

✓ **Able to merge.** These branches can be automatically merged.

 **Create pull request**

Discuss and review the changes in this comparison with others.

But before [pr] is merged, the maintainer adds a conflicting change to the source repository



This branch has conflicts that must be resolved

Use the [web editor](#) or the [command line](#) to resolve conflicts.

Resolve conflicts

Conflicting files

README.md

pr_sync()



The screenshot shows a code editor with a file named 'README.md'. The editor has a toolbar with icons for undo, redo, save, search, and a 'Preview' button. The code in the editor is as follows:

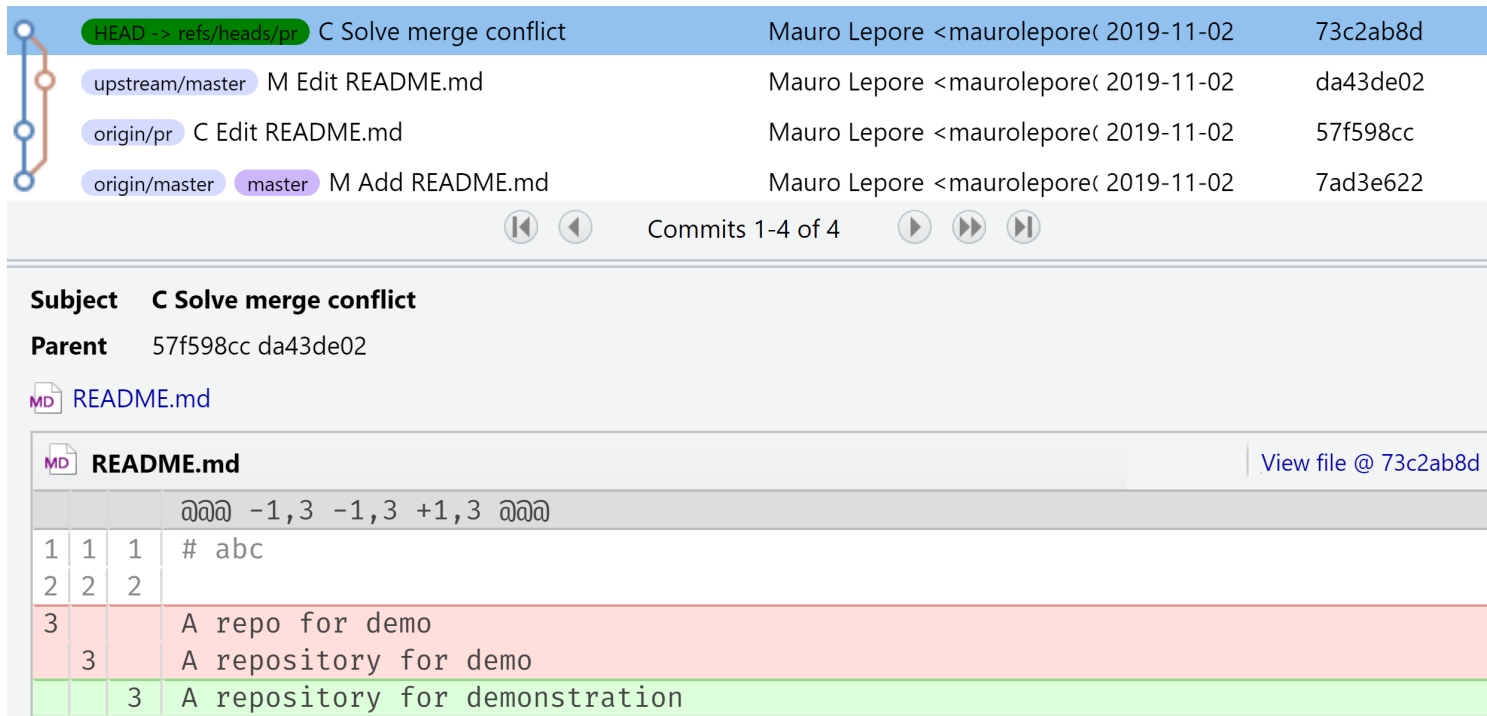
```
1 # abc
2
3 <<<<<<< HEAD
4 A repo for demo
5 =====
6 A repository for demo
7 >>>>>> master
8
```

Below the editor is a terminal window with the following output:

```
> pr_sync()
✓ Pulling changes from GitHub PR
✓ Pulling changes from GitHub source repo 'upstream/master'
Error: Merge conflict! Please resolve before continuing
>
```

```
# Contrib@LAPTOP ~/abc (pr)
# Using default remote "origin" and branch "pr"
# `git pull` is shorthand for `git fetch && git merge FETCH_HEAD`
git pull
git pull upstream master
git push
```

Solve merge conflicts



The image shows a Git commit history and a diff view for a file named README.md. The commit history at the top shows four commits by Mauro Lepore, dated 2019-11-02. The first commit (HEAD -> refs/heads/pr) is titled 'C Solve merge conflict' with hash 73c2ab8d. The second commit (upstream/master) is titled 'M Edit README.md' with hash da43de02. The third commit (origin/pr) is titled 'C Edit README.md' with hash 57f598cc. The fourth commit (origin/master) is titled 'M Add README.md' with hash 7ad3e622. Below the history, the diff view for README.md is shown, comparing the parent commit (57f598cc da43de02) with the current commit (73c2ab8d). The diff shows three lines of code: a header line, a line with '# abc', and a line with 'A repository for demonstration'. The first two lines are highlighted in red, and the third line is highlighted in green.

Commit	Author	Date	Hash
HEAD -> refs/heads/pr	Mauro Lepore <maurolepore@...>	2019-11-02	73c2ab8d
upstream/master	Mauro Lepore <maurolepore@...>	2019-11-02	da43de02
origin/pr	Mauro Lepore <maurolepore@...>	2019-11-02	57f598cc
origin/master	Mauro Lepore <maurolepore@...>	2019-11-02	7ad3e622

Commits 1-4 of 4

Subject C Solve merge conflict

Parent 57f598cc da43de02

MD README.md

MD README.md [View file @ 73c2ab8d](#)

Line	Parent	Current	Diff
1	1	1	# abc
2	2	2	
3	3	3	A repository for demonstration

pr_push()

```
> pr_push()  
✓ Checking that local branch 'pr' has the changes in 'origin/pr'  
✓ Pushing local 'pr' branch to 'origin:pr'  
✓ Create PR at link given below  
✓ Opening URL 'https://github.com/maurolepore/abc/compare/pr'  
> |
```

```
# Contrib@LAPTOP ~/abc (pr)
```

```
# Retry sync
```

```
git pull
```

```
git pull upstream master
```

```
git push
```

[pr] now has no conflicts with [upstream/master]



This branch has no conflicts with the base branch

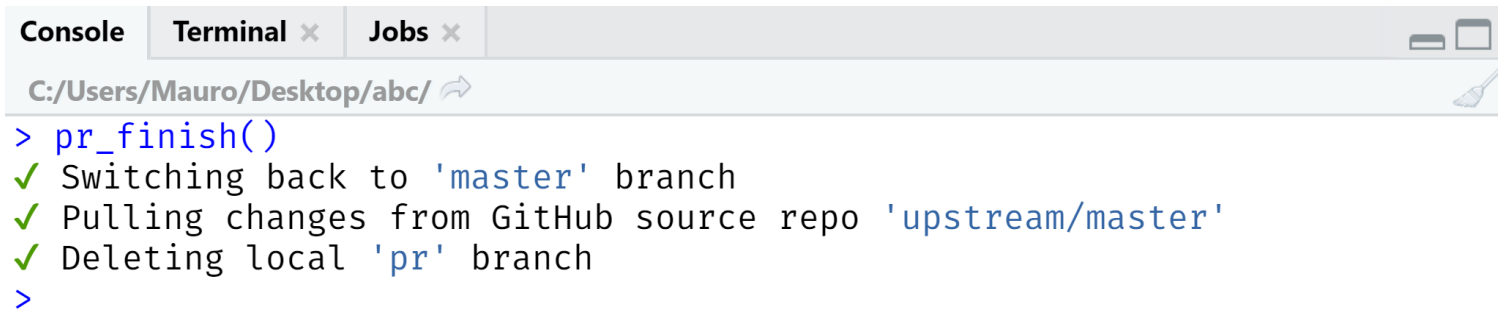
Merging can be performed automatically.

Squash and merge



You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

pr_finish()



A terminal window with tabs for 'Console', 'Terminal x', and 'Jobs x'. The active tab is 'Terminal'. The path 'C:/Users/Mauro/Desktop/abc/' is shown in the title bar. The command '> pr_finish()' has been executed, resulting in three green checkmark messages: 'Switching back to 'master' branch', 'Pulling changes from GitHub source repo 'upstream/master'', and 'Deleting local 'pr' branch'. The prompt '>' is visible at the bottom of the terminal output.

```
> pr_finish()  
✓ Switching back to 'master' branch  
✓ Pulling changes from GitHub source repo 'upstream/master'  
✓ Deleting local 'pr' branch  
>
```

```
# Contrib@LAPTOP ~/abc (pr)  
git checkout master
```

```
# Contrib@LAPTOP ~/abc (master)  
git pull upstream master  
git push -d origin pr  
git branch -d pr
```