# User

Trying to build fastparquet using msys2 and included:
[build]
compiler = mingw32
in setup.cfg

First I did git pull - I had to do git rebase. Then using pip install. - following errors:

$ pip install .
Processing d:\downloads\fastparquet
Installing build dependencies ... done
Getting requirements to build wheel ... done
Installing backend dependencies ... done
Preparing metadata (pyproject.toml) ... error
error: subprocess-exited-with-error

× Preparing metadata (pyproject.toml) did not run successfully.
│ exit code: 1
╰─> [33 lines of output]

Error compiling Cython file:

------------------------------------------------------------

...

<<<<<<< HEAD

^

------------------------------------------------------------

fastparquet\cencoding.pyx:1:0: Expected an identifier or literal

Compiling fastparquet/speedups.pyx because it changed.

Compiling fastparquet/cencoding.pyx because it changed.

[1/2] Cythonizing fastparquet/cencoding.pyx

Traceback (most recent call last):

File "D:\Programs\Python\Lib\site-packages\pip\\_vendor\pyproject\_hooks\\_in\_process\\_in\_process.py", line 353, in <module>

main()

File "D:\Programs\Python\Lib\site-packages\pip\\_vendor\pyproject\_hooks\\_in\_process\\_in\_process.py", line 335, in main

json\_out['return\_val'] = hook(\*\*hook\_input['kwargs'])

^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File "D:\Programs\Python\Lib\site-packages\pip\\_vendor\pyproject\_hooks\\_in\_process\\_in\_process.py", line 149, in prepare\_metadata\_for\_build\_wheel

return hook(metadata\_directory, config\_settings)

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File "D:\Programs\Msys2\tmp\pip-build-env-ly524vng\overlay\Lib\site-packages\setuptools\build\_meta.py", line 366, in prepare\_metadata\_for\_build\_wheel

self.run\_setup()

File "D:\Programs\Msys2\tmp\pip-build-env-ly524vng\overlay\Lib\site-packages\setuptools\build\_meta.py", line 480, in run\_setup

super().run\_setup(setup\_script=setup\_script)

File "D:\Programs\Msys2\tmp\pip-build-env-ly524vng\overlay\Lib\site-packages\setuptools\build\_meta.py", line 311, in run\_setup

exec(code, locals())

File "<string>", line 44, in <module>

File "D:\Programs\Msys2\tmp\pip-build-env-ly524vng\overlay\Lib\site-packages\Cython\Build\Dependencies.py", line 1154, in cythonize

cythonize\_one(\*args)

File "D:\Programs\Msys2\tmp\pip-build-env-ly524vng\overlay\Lib\site-packages\Cython\Build\Dependencies.py", line 1321, in cythonize\_one

raise CompileError(None, pyx\_file)

Cython.Compiler.Errors.CompileError: fastparquet/cencoding.pyx

[end of output]

note: This error originates from a subprocess, and is likely not a problem with pip.
error: metadata-generation-failed

× Encountered error while generating package metadata.
╰─> See above for output.

note: This is an issue with the package mentioned above, not pip.
hint: See above for details.

# Bing

The error you're encountering seems to be related to the fastparquet package itself, not pip. The error message indicates that there's a problem with the Cython file fastparquet/cencoding.pyx. The line <<<<<<< HEAD is a conflict marker that Git adds when it cannot automatically merge changes from different commits. This usually happens when you're trying to merge or rebase.

To resolve this issue, you need to manually edit the file to resolve the conflict. Here's what you can do:

1. Open the file fastparquet/cencoding.pyx in a text editor.
2. Look for the conflict markers. A conflict marked block will look like this:
3. Decide if you want to keep your changes, the changes from the other branch, or a mix of both. Edit the file to reflect this.
4. Once you've resolved all conflicts, remove the conflict markers.
5. Save the file.
6. Commit the changes with Git.
7. Try the build process again.

If you continue to encounter issues, it might be helpful to check if others have had similar problems. For example, there are discussions on Stack Overflow[**1**](https://stackoverflow.com/questions/64773160/error-installing-fastparquet-in-windows-10) and GitHub[**2**](https://github.com/dask/fastparquet/issues/492)[**3**](https://github.com/dask/fastparquet/issues/598) about issues with building fastparquet. These might provide some additional insights or solutions. If all else fails, consider reaching out to the fastparquet community or the package maintainer for help. Good luck! 👍

Source(s)

1. [Error installing fastparquet in windows 10 - Stack Overflow](https://stackoverflow.com/questions/64773160/error-installing-fastparquet-in-windows-10)

2. [building wheel for fastparquet ... error · Issue #492 · dask ... - GitHub](https://github.com/dask/fastparquet/issues/492)

3. [Version 0.6.0: gcc: error: fastparquet/speedups.c: No such ... - GitHub](https://github.com/dask/fastparquet/issues/598)

4. <https://developercommunity.visualstudio.com/content/problem/1207405/fmod-after-an-update-to-windows-2004-is-causing-a.html>

5. <https://visualstudio.microsoft.com/downloads/>

# User

I am often facing this issue with this fastparquet package. When I want to pull changes from github and do git pull, I am greeted with the message:

juhi123@juhi MINGW64 /d/Downloads/fastparquet
$ git pull .
error: cannot pull with rebase: You have unstaged changes.
error: Please commit or stash them.

Suppose I want to avoid this and do git pull smoothly to update local folder what should I do?

# Bing

The error message you're seeing is because you have changes in your local repository that haven't been committed. Git is cautious and doesn't want to overwrite these changes during the pull operation. Here are a few ways to handle this situation:

1. **Commit the changes**: If the changes are ready to be committed, you can add them to the staging area with git add . and then commit them with git commit -m "Your commit message". After that, you should be able to pull the latest changes with git pull.
2. **Stash the changes**: If you're not ready to commit the changes, you can stash them away temporarily with git stash. This takes your modifications (both staged and unstaged), saves them away for later use, and then reverts them from your working directory. You can pull the latest changes with git pull and when you're ready, you can get your stashed changes back with git stash pop.
3. **Discard the changes**: If you don't need the changes, you can discard them. To discard all unstaged changes, use git checkout -- .. To discard all changes (staged and unstaged), use git reset --hard.
4. **Use Autostash**: If you want to keep your working changes while performing a rebase, you can use --autostash[**1**](https://stackoverflow.com/questions/23517464/error-cannot-pull-with-rebase-you-have-unstaged-changes). This will automatically stash your changes before the pull operation and then reapply them after the pull. You can use it like this: git pull --rebase --autostash. If you want autostash to be the default behavior, you can set it in your Git configuration with git config --global rebase.autoStash true[**1**](https://stackoverflow.com/questions/23517464/error-cannot-pull-with-rebase-you-have-unstaged-changes).

Remember to always check the status of your repository using git status before performing operations like pull or rebase. This will give you information about any changes that have been made and whether they've been staged or not. It's a good practice to have a clean working state before pulling changes from a remote repository. I hope this helps! 👍

Source(s)

1. [Error: Cannot pull with rebase: You have unstaged changes](https://stackoverflow.com/questions/23517464/error-cannot-pull-with-rebase-you-have-unstaged-changes)

2. [git - cannot pull with rebase: You have unstaged changes - Stack Overflow](https://stackoverflow.com/questions/60838965/cannot-pull-with-rebase-you-have-unstaged-changes)

3. ["cannot pull with rebase: You have unstaged changes" on "git pull" - GitLab](https://gitlab.com/kicad/code/kicad/-/issues/5023)

4. [cannot pull with rebase: You have unstaged changes. #515 - GitHub](https://github.com/tophat/monodeploy/issues/515)

5. <https://devcenter.heroku.com/articles/sharing>

# User

Suppose I want to overwrite all changes from github during git pull to avoid, git commit, etc. can it be done?

# Bing

Yes, it can be done. If you want to overwrite your local changes during a git pull, you can use the following steps:

1. Fetch all branches from the remote repository:
2. Reset your current branch (let's say master) to match the origin/master branch:

This will discard all changes to tracked files in your working directory[**1**](https://stackoverflow.com/questions/1125968/how-do-i-force-git-pull-to-overwrite-local-files)[**2**](https://www.freecodecamp.org/news/git-pull-force-how-to-overwrite-local-changes-with-git/). Please be aware that any uncommitted changes, even if they are staged, will be lost[**1**](https://stackoverflow.com/questions/1125968/how-do-i-force-git-pull-to-overwrite-local-files). If you have any changes that you do not want to lose, consider committing them to another branch or stashing them before running these commands.

Remember, this is a destructive operation and any local changes will be lost. Always make sure you really don't need your uncommitted changes before you run these commands[**1**](https://stackoverflow.com/questions/1125968/how-do-i-force-git-pull-to-overwrite-local-files)[**2**](https://www.freecodecamp.org/news/git-pull-force-how-to-overwrite-local-changes-with-git/)[**3**](https://tecadmin.net/force-overwrite-local-files-git-pull/)[**4**](https://changedetect.com/how-to-pull-and-overwrite-local-changes-git/).

If you want to avoid this situation in the future, consider committing your changes more frequently or stashing your changes before pulling. This way, you can always retrieve your work if necessary. I hope this helps! 👍.

Source(s)

1. [How do I force "git pull" to overwrite local files?](https://stackoverflow.com/questions/1125968/how-do-i-force-git-pull-to-overwrite-local-files)

2. [Git Pull Force – How to Overwrite Local Changes With Git](https://www.freecodecamp.org/news/git-pull-force-how-to-overwrite-local-changes-with-git/)

3. [How to Force Overwrite Local Files on Git Pull – TecAdmin](https://tecadmin.net/force-overwrite-local-files-git-pull/)

4. [Mastering Git: How to Pull and Overwrite Local Changes Git Explained](https://changedetect.com/how-to-pull-and-overwrite-local-changes-git/)

5. [en.wikipedia.org](https://en.wikipedia.org/wiki/Git)