

# Wudu\_AI: Mastering the Shell with GitHub Copilot CLI

## AI: Mastering the Shell with GitHub Copilot CLI

### 1. Screenshot of Interaction

```

C:\> I committed locally but haven't pushed yet. I want to combine the last two commits into a single com...

Microsoft Windows [Version 10.0.19045.6466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\wud>node -v
v24.13.1

C:\Users\wud>npm -v
11.8.0

C:\Users\wud>npm install -g @github/copilot

changed 2 packages in 1m

C:\Users\wud>
C:\Users\wud>copilot login
To authenticate, visit https://github.com/login/device and enter code 7CFE-D505.
Waiting for authorization...
Signed in successfully as wudutesfa.

C:\Users\wud>
C:\Users\wud>copilot

  GitHub Copilot v0.0.410
  Describe a task to get started.
  Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create
  a PR. Copilot uses AI, so always check for mistakes.

• No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.
• Model changed to: gpt-5-mini (medium)

I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new
message while preserving all changes in the working directory. Generate the correct Git command using the git agent
and briefly explain why it works.

**Generating Git Command**

The user is asking for a Git command to combine the last two commits into a single one, while keeping all changes in
the working directory. Since we're not invoking any tools, I'll simply provide the command: `git rebase -i HEAD2`.
This works by allowing the user to squash the last two commits and create a new message, perfect for cleaning up

```

cross-platform!

• Run:

```
git reset --soft HEAD~2 && git commit -m "New combined commit message"
```

Explanation: `git reset --soft HEAD~2` moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent `git commit` creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

🔍 Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode



GitHub Copilot v0.0.410



Describe a task to get started.



Pick a model with `/model`. Send this session to GitHub with `/delegate`, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

• 🔍 No copilot instructions found. Run `/init` to generate a `copilot-instructions.md` file for this project.

• Model changed to: gpt-5-mini (medium)

🔍 I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

cross-platform!

• Run:

```
git reset --soft HEAD~2 && git commit -m "New combined commit message"
```

Explanation: `git reset --soft HEAD~2` moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent `git commit` creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

▢ Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode



GitHub Copilot v0.0.410



Describe a task to get started.



Pick a model with `/model`. Send this session to GitHub with `/delegate`, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

• ▢ No copilot instructions found. Run `/init` to generate a `copilot-instructions.md` file for this project.

• Model changed to: gpt-5-mini (medium)

▢ I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

their commit history before pushing.\*\*Evaluating Git Command Options\*\*

I have a couple of options for the user's request to combine the last two commits into one. The first option is using ``git reset --soft HEAD2 && git commit -m "New message"``. This integrates the last two commits into one with a new message while preserving the working directory, as the ``--soft`` option keeps the index intact. Alternatively, they could use ``git rebase -i HEAD2`` to squash commits, but that's a bit more complex.\*\*Providing the Git Command\*\*

I'll give the command: ``git reset --soft HEAD2 && git commit -m "New combined commit message"``. This effectively combines the last two commits into one while preserving all changes in the working directory, as the ``--soft`` option does not alter the working directory or staged changes. Since the user asked for clarity, I'll stick to this command and a clear explanation, keeping it concise and to the point. No tool calls are needed here!\*\*Generating the Command and Explanation\*\*

I'll provide the command like this:

```
`git reset --soft HEAD2 && git commit -m "Your new message"``
```

This command moves HEAD back two commits while keeping the changes staged, and then creates a new commit with your specified message. Since the user mentioned they haven't pushed yet, no need to worry about needing a force push. It's concise and should work perfectly on Windows as well since Git commands are cross-platform!

• Thinking (Esc to cancel · 1.8 KiB)

~

gpt-5-mini (medium) (1x)

▢ Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode · ctrl+q enqueue



GitHub Copilot v0.0.410

cross-platform!

• Run:

```
git reset --soft HEAD~2 && git commit -m "New combined commit message"
```

Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent git commit creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

🔍 Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode



GitHub Copilot v0.0.410



Describe a task to get started.



Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

🔍 No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.

• Model changed to: gpt-5-mini (medium)

🔍 I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

🔍 \*\*Generating Git Command\*\*

The user is asking for a Git command to combine the last two commits into a single one, while keeping all changes in the working directory. Since we're not invoking any tools, I'll simply provide the command: `git rebase -i HEAD2`. This works by allowing the user to squash the last two commits and create a new message, perfect for cleaning up their commit history before pushing. \*\*Evaluating Git Command Options\*\*

I have a couple of options for the user's request to combine the last two commits into one. The first option is using `git reset --soft HEAD2 && git commit -m "New message"`. This integrates the last two commits into one with a new message while preserving the working directory, as the `--soft` option keeps the index intact. Alternatively, they could use `git rebase -i HEAD2` to squash commits, but that's a bit more complex. \*\*Providing the Git Command\*\*

I'll give the command: `git reset --soft HEAD2 && git commit -m "New combined commit message"`. This effectively combines the last two commits into one while preserving all changes in the working directory, as the `--soft` option does not alter the working directory or staged changes. Since the user asked for clarity, I'll stick to this command and a clear explanation, keeping it concise and to the point. No tool calls are needed here! \*\*Generating the Command and Explanation\*\*

• Run:

```
git reset --soft HEAD~2 && git commit -m "New combined commit message"
```

Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent git commit creates a single new commit with the provided message, preserving

🔍 Thinking (Esc to cancel · 2.1 KiB)

~

gpt-5-mini (medium) (1x)

cross-platform!

• Run:

```
git reset --soft HEAD~2 && git commit -m "New combined commit message"
```

Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent git commit creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

🔍 Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode

🔍 GitHub Copilot v0.0.410

🔍 Describe a task to get started.

🔍

Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

• 🔍 No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.

• Model changed to: gpt-5-mini (medium)

🔍 I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

• Model changed to: gpt-5-mini (medium)

🔍 I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

🔍 \*\*Generating Git Command\*\*

The user is asking for a Git command to combine the last two commits into a single one, while keeping all changes in the working directory. Since we're not invoking any tools, I'll simply provide the command: `git rebase -i HEAD2`. This works by allowing the user to squash the last two commits and create a new message, perfect for cleaning up their commit history before pushing. \*\*Evaluating Git Command Options\*\*

I have a couple of options for the user's request to combine the last two commits into one. The first option is using `git reset --soft HEAD2 && git commit -m "New message"`. This integrates the last two commits into one with a new message while preserving the working directory, as the `--soft` option keeps the index intact. Alternatively, they could use `git rebase -i HEAD2` to squash commits, but that's a bit more complex. \*\*Providing the Git Command\*\*

I'll give the command: `git reset --soft HEAD2 && git commit -m "New combined commit message"`. This effectively combines the last two commits into one while preserving all changes in the working directory, as the `--soft` option does not alter the working directory or staged changes. Since the user asked for clarity, I'll stick to this command and a clear explanation, keeping it concise and to the point. No tool calls are needed here! \*\*Generating the Command and Explanation\*\*

• Run:

```
git reset --soft HEAD~2 && git commit -m "New combined commit message"
```

Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged) and the subsequent git commit creates a single new commit with the

cross-platform!

• Run:

```
git reset --soft HEAD~2 && git commit -m "New combined commit message"
```

Explanation: `git reset --soft HEAD~2` moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent `git commit` creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

📄 Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode



GitHub Copilot v0.0.410



Describe a task to get started.



Pick a model with `/model`. Send this session to GitHub with `/delegate`, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

📄 No copilot instructions found. Run `/init` to generate a `copilot-instructions.md` file for this project.

• Model changed to: gpt-5-mini (medium)

📄 I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

Pick a model with `/model`. Send this session to GitHub with `/delegate`, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

📄 No copilot instructions found. Run `/init` to generate a `copilot-instructions.md` file for this project.

• Model changed to: gpt-5-mini (medium)

📄 I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

📄 **\*\*Generating Git Command\*\***

The user is asking for a Git command to combine the last two commits into a single one, while keeping all changes in the working directory. Since we're not invoking any tools, I'll simply provide the command: `git rebase -i HEAD2`. This works by allowing the user to squash the last two commits and create a new message, perfect for cleaning up their commit history before pushing. **\*\*Evaluating Git Command Options\*\***

I have a couple of options for the user's request to combine the last two commits into one. The first option is using `git reset --soft HEAD2 && git commit -m "New message"`. This integrates the last two commits into one with a new message while preserving the working directory, as the `--soft` option keeps the index intact. Alternatively, they could use `git rebase -i HEAD2` to squash commits, but that's a bit more complex. **\*\*Providing the Git Command\*\***

I'll give the command: `git reset --soft HEAD2 && git commit -m "New combined commit message"`. This effectively combines the last two commits into one while preserving all changes in the working directory, as the `--soft` option does not alter the working directory or staged changes. Since the user asked for clarity, I'll stick to this command and a clear explanation, keeping it concise and to the point. No tool calls are needed here! **\*\*Generating the Command and Explanation\*\***

cross-platform!

- Run:  
git reset --soft HEAD~2 && git commit -m "New combined commit message"  
Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent git commit creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode

GitHub Copilot v0.0.410

Describe a task to get started.

Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

- No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.
- Model changed to: gpt-5-mini (medium)

I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

? for shortcuts

shift+tab switch mode · ctrl+q enqueue

GitHub Copilot v0.0.410

Describe a task to get started.

Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

- No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.
- Model changed to: gpt-5-mini (medium)

I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

\*\*Generating Git Command\*\*

The user is asking for a Git command to combine the last two commits into a single one, while keeping all changes in the working directory. Since we're not invoking any tools, I'll simply provide the command: `git rebase -i HEAD2`. This works by allowing the user to squash the last two commits and create a new message, perfect for cleaning up their commit history before pushing.

**Evaluating Git Command Options**

I have a couple of options for the user's request to combine the last two commits into one. The first option is using `git reset --soft HEAD2 && git commit -m "New message"`. This integrates the last two commits into one with a new message while preserving the working directory, as the `--soft` option keeps the index intact. Alternatively, they could use `git rebase -i HEAD2` to squash commits, but that's a bit more complex.

**Providing the Git Command**

cross-platform!

- Run:  
git reset --soft HEAD~2 && git commit -m "New combined commit message"  
Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent git commit creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode

GitHub Copilot v0.0.410

Describe a task to get started.

----

Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

- No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.
- Model changed to: gpt-5-mini (medium)

I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

Thinking (Esc to cancel · 2.1 KiB)

~

gpt-5-mini (medium) (1x)

Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode · ctrl+q enqueue

GitHub Copilot v0.0.410

Describe a task to get started.

----

Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

- No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.
- Model changed to: gpt-5-mini (medium)

I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

\*\*Generating Git Command\*\*

The user is asking for a Git command to combine the last two commits into a single one, while keeping all changes in the working directory. Since we're not invoking any tools, I'll simply provide the command: `git rebase -i HEAD~2`. This works by allowing the user to squash the last two commits and create a new message, perfect for cleaning up

cross-platform!

- Run:  
git reset --soft HEAD~2 && git commit -m "New combined commit message"  
Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent git commit creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode



GitHub Copilot v0.0.410



Describe a task to get started.



Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

- No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.
- Model changed to: gpt-5-mini (medium)
- I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

- Run:  
git reset --soft HEAD~2 && git commit -m "New combined commit message"  
Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent git commit creates a single new commit with the provided message, preserving all working-directory changes.

Thinking (Esc to cancel · 2.1 KiB)

~

gpt-5-mini (medium) (1x)

Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode · ctrl+q enqueue



GitHub Copilot v0.0.410



Describe a task to get started.



Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

- No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.
- Model changed to: gpt-5-mini (medium)
- I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

\*\*Generating Git Command\*\*

cross-platform!

- Run:  
git reset --soft HEAD~2 && git commit -m "New combined commit message"  
Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent git commit creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode

GitHub Copilot v0.0.410

Describe a task to get started.

----

Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

- No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.
- Model changed to: gpt-5-mini (medium)

I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

does not alter the working directory or staged changes. Since the user asked for clarity, I'll stick to this command and a clear explanation, keeping it concise and to the point. No tool calls are needed here!\*\*Generating the Command and Explanation\*\*

- Run:  
git reset --soft HEAD~2 && git commit -m "New combined commit message"  
Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so

~

gpt-5-mini (medium) (1x)

Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode

Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode · ctrl+q enqueue

GitHub Copilot v0.0.410

Describe a task to get started.

----

Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

- No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.
- Model changed to: gpt-5-mini (medium)

cross-platform!

- Run:  
git reset --soft HEAD~2 && git commit -m "New combined commit message"  
Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent git commit creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode



GitHub Copilot v0.0.410



Describe a task to get started.



Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

- No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.
- Model changed to: gpt-5-mini (medium)

I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

new message while preserving the working directory, as the '--soft' option keeps the index intact. Alternatively, they could use 'git rebase -i HEAD2' to squash commits, but that's a bit more complex.\*\*Providing the Git Command\*\*

I'll give the command: 'git reset --soft HEAD2 && git commit -m "New combined commit message"'. This effectively combines the last two commits into one while preserving all changes in the working directory, as the '--soft' option does not alter the working directory or staged changes. Since the user asked for clarity, I'll stick to this command and a clear explanation, keeping it concise and to the point. No tool calls are needed here!\*\*Generating the Command and Explanation\*\*

I'll provide the command like this:

'git reset --soft HEAD2 && git commit -m "Your new message"'

This command moves HEAD back two commits while keeping the changes staged, and then creates a new commit with your specified message. Since the user mentioned they haven't pushed yet, no need to worry about needing a force push. It's concise and should work perfectly on Windows as well since Git commands are cross-platform!

- Run:  
git reset --soft HEAD~2 && git commit -m "New combined commit message"  
Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent git commit creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

for shortcuts

Type @ to mention files, / for commands, or

shift+tab



cross-platform!

• Run:

```
git reset --soft HEAD~2 && git commit -m "New combined commit message"
```

Explanation: `git reset --soft HEAD~2` moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent `git commit` creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

▢ Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode

GitHub Copilot v0.0.410

Describe a task to get started.

----

Pick a model with `/model`. Send this session to GitHub with `/delegate`, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

• ▢ No copilot instructions found. Run `/init` to generate a `copilot-instructions.md` file for this project.

• Model changed to: gpt-5-mini (medium)

▢ I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

▢ **Generating Git Command**

The user is asking for a Git command to combine the last two commits into a single one, while keeping all changes in the working directory. Since we're not invoking any tools, I'll simply provide the command: ``git rebase -i HEAD2``. This works by allowing the user to squash the last two commits and create a new message, perfect for cleaning up their commit history before pushing.

**\*\*Evaluating Git Command Options\*\***

I have a couple of options for the user's request to combine the last two commits into one. The first option is using ``git reset --soft HEAD2 && git commit -m "New message"``. This integrates the last two commits into one with a new message while preserving the working directory, as the ``--soft`` option keeps the index intact. Alternatively, they could use ``git rebase -i HEAD2`` to squash commits, but that's a bit more complex.

**\*\*Providing the Git Command\*\***

I'll give the command: ``git reset --soft HEAD2 && git commit -m "New combined commit message"``. This effectively combines the last two commits into one while preserving all changes in the working directory, as the ``--soft`` option does not alter the working directory or staged changes. Since the user asked for clarity, I'll stick to this command and a clear explanation, keeping it concise and to the point. No tool calls are needed here!

**\*\*Generating the Command and Explanation\*\***

I'll provide the command like this:

```
`git reset --soft HEAD2 && git commit -m "Your new message"`
```

This command moves HEAD back two commits while keeping the changes staged, and then creates a new commit with your specified message. Since the user mentioned they haven't pushed yet, no need to worry about needing a force push. It's concise and should work perfectly on Windows as well since Git commands are cross-platform!

• Run:

```
git reset --soft HEAD~2 && git commit -m "New combined commit message"
```

Explanation: `git reset --soft HEAD~2` moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent `git commit` creates a single new commit with the provided message, preserving all working-directory changes.

cross-platform!

• Run:

```
git reset --soft HEAD~2 && git commit -m "New combined commit message"
```

Explanation: `git reset --soft HEAD~2` moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent `git commit` creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

▢ Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode



GitHub Copilot v0.0.410



Describe a task to get started.



Pick a model with `/model`. Send this session to GitHub with `/delegate`, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

• ▢ No copilot instructions found. Run `/init` to generate a `copilot-instructions.md` file for this project.

• Model changed to: gpt-5-mini (medium)

▢ I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

▢ **\*\*Generating Git Command\*\***

The user is asking for a Git command to combine the last two commits into a single one, while keeping all changes in the working directory. Since we're not invoking any tools, I'll simply provide the command: ``git rebase -i HEAD2``. This works by allowing the user to squash the last two commits and create a new message, perfect for cleaning up their commit history before pushing. **\*\*Evaluating Git Command Options\*\***

I have a couple of options for the user's request to combine the last two commits into one. The first option is using ``git reset --soft HEAD2 && git commit -m "New message"``. This integrates the last two commits into one with a new message while preserving the working directory, as the ``--soft`` option keeps the index intact. Alternatively, they could use ``git rebase -i HEAD2`` to squash commits, but that's a bit more complex. **\*\*Providing the Git Command\*\***

I'll give the command: ``git reset --soft HEAD2 && git commit -m "New combined commit message"``. This effectively combines the last two commits into one while preserving all changes in the working directory, as the ``--soft`` option does not alter the working directory or staged changes. Since the user asked for clarity, I'll stick to this command and a clear explanation, keeping it concise and to the point. No tool calls are needed here! **\*\*Generating the Command and Explanation\*\***

I'll provide the command like this:

```
`git reset --soft HEAD2 && git commit -m "Your new message"`
```

This command moves HEAD back two commits while keeping the changes staged, and then creates a new commit with your specified message. Since the user mentioned they haven't pushed yet, no need to worry about needing a force push. It's concise and should work perfectly on Windows as well since Git commands are cross-platform!

• Run:

```
git reset --soft HEAD~2 && git commit -m "New combined commit message"
```

Explanation: `git reset --soft HEAD~2` moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent `git commit` creates a single new commit with the provided message, preserving all working-directory changes.

cross-platform!

• Run:

```
git reset --soft HEAD~2 && git commit -m "New combined commit message"
```

Explanation: `git reset --soft HEAD~2` moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent `git commit` creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

▢ Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode



GitHub Copilot v0.0.410



Describe a task to get started.



Pick a model with `/model`. Send this session to GitHub with `/delegate`, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

• ▢ No copilot instructions found. Run `/init` to generate a `copilot-instructions.md` file for this project.

• Model changed to: gpt-5-mini (medium)

▢ I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

• ▢ No copilot instructions found. Run `/init` to generate a `copilot-instructions.md` file for this project.

• Model changed to: gpt-5-mini (medium)

▢ I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

▢ **\*\*Generating Git Command\*\***

The user is asking for a Git command to combine the last two commits into a single one, while keeping all changes in the working directory. Since we're not invoking any tools, I'll simply provide the command: ``git rebase -i HEAD2``. This works by allowing the user to squash the last two commits and create a new message, perfect for cleaning up their commit history before pushing.

**\*\*Evaluating Git Command Options\*\***

I have a couple of options for the user's request to combine the last two commits into one. The first option is using ``git reset --soft HEAD2 && git commit -m "New message"``. This integrates the last two commits into one with a new message while preserving the working directory, as the ``--soft`` option keeps the index intact. Alternatively, they could use ``git rebase -i HEAD2`` to squash commits, but that's a bit more complex.

**\*\*Providing the Git Command\*\***

I'll give the command: ``git reset --soft HEAD2 && git commit -m "New combined commit message"``. This effectively combines the last two commits into one while preserving all changes in the working directory, as the ``--soft`` option does not alter the working directory or staged changes. Since the user asked for clarity, I'll stick to this command and a clear explanation, keeping it concise and to the point. No tool calls are needed here!

**\*\*Generating the Command and Explanation\*\***

I'll provide the command like this:

```
`git reset --soft HEAD2 && git commit -m "Your new message"``
```

This command moves HEAD back two commits while keeping the changes staged, and then creates a new commit with your specified message. Since the

cross-platform!

- Run:  
`git reset --soft HEAD~2 && git commit -m "New combined commit message"`  
Explanation: `git reset --soft HEAD~2` moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent `git commit` creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

🔍 Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode

🔍 GitHub Copilot v0.0.410

🔍 Describe a task to get started.

🔍

Pick a model with `/model`. Send this session to GitHub with `/delegate`, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

- 🔍 No copilot instructions found. Run `/init` to generate a `copilot-instructions.md` file for this project.

- Model changed to: gpt-5-mini (medium)

🔍 I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

🔍 GitHub Copilot v0.0.410

🔍 Describe a task to get started.

🔍

Pick a model with `/model`. Send this session to GitHub with `/delegate`, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

- 🔍 No copilot instructions found. Run `/init` to generate a `copilot-instructions.md` file for this project.

- Model changed to: gpt-5-mini (medium)

🔍 I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

🔍 **\*\*Generating Git Command\*\***

The user is asking for a Git command to combine the last two commits into a single one, while keeping all changes in the working directory. Since we're not invoking any tools, I'll simply provide the command: ``git rebase -i HEAD2``. This works by allowing the user to squash the last two commits and create a new message, perfect for cleaning up their commit history before pushing.**\*\*Evaluating Git Command Options\*\***

I have a couple of options for the user's request to combine the last two commits into one. The first option is using ``git reset --soft HEAD2 && git commit -m "New message"``. This integrates the last two commits into one with a new message while preserving the working directory, as the ``--soft`` option keeps the index intact. Alternatively, they could use ``git rebase -i HEAD2`` to squash commits, but that's a bit more complex.**\*\*Providing the Git Command\*\***

I'll give the command: ``git reset --soft HEAD2 && git commit -m "New combined commit message"``. This effectively combines the last two commits into one while preserving all changes in the working directory, as the ``--soft`` option does not alter the working directory or staged changes. Since the

cross-platform!

- Run:  
git reset --soft HEAD~2 && git commit -m "New combined commit message"  
Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent git commit creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode

GitHub Copilot v0.0.410

Describe a task to get started.

----

Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

- No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.
- Model changed to: gpt-5-mini (medium)

I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

remain staged), and the subsequent git commit creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

Type @ to mention files, / for commands, or ? for shortcuts

shift+tab sw

itch mode

GitHub Copilot v0.0.410

Describe a task to get started.

----

Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

- No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.
- Model changed to: gpt-5-mini (medium)

I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

\*\*Generating Git Command\*\*

The user is asking for a Git command to combine the last two commits into a single one, while keeping all changes in the working directory. Since we're not invoking any tools, I'll simply provide the command: `git rebase -i HEAD2`. This works by allowing the user to squash the last two commits and create a new message, perfect for cleaning up their commit history before pushing.\*\*Evaluating Git Command Options\*\*

I have a couple of options for the user's request to combine the last two commits into one. The first option is using `git reset --soft HEAD2` followed by `git commit -m "New combined commit message"`. This approach moves the HEAD pointer back two commits, leaving the index and working tree intact. Then, a new commit is created with the provided message, effectively squashing the last two commits into one while preserving all changes in the working directory.

cross-platform!

- Run:  
git reset --soft HEAD~2 && git commit -m "New combined commit message"  
Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent git commit creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode

GitHub Copilot v0.0.410

Describe a task to get started.

----

Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

- No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.
- Model changed to: gpt-5-mini (medium)

I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

This command moves HEAD back two commits while keeping the changes staged, and then creates a new commit with your specified message. Since the user mentioned they haven't pushed yet, no need to worry about needing a force push. It's concise and should work perfectly on Windows as well since Git commands are cross-platform!

- Run:  
git reset --soft HEAD~2 && git commit -m "New combined commit message"  
Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent git commit creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

~

gpt-5-mini (medium) (1x)

Type @ to m

ention files, / for commands, or ? for shortcuts

shift+tab swi

tch mode

GitHub Copilot v0.0.410

Describe a task to get started.

----

Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

- No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.
- Model changed to: gpt-5-mini (medium)

I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all

cross-platform!

- Run:  
git reset --soft HEAD~2 && git commit -m "New combined commit message"  
Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent git commit creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode

GitHub Copilot v0.0.410

Describe a task to get started.

----

Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

- No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.
- Model changed to: gpt-5-mini (medium)

I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

I'll give the command: `git reset --soft HEAD2 && git commit -m "New combined commit message"`. This effectively combines the last two commits into one while preserving all changes in the working directory, as the `--soft` option does not alter the working directory or staged changes. Since the user asked for clarity, I'll stick to this command and a clear explanation, keeping it concise and to the point. No tool calls are needed here!\*\*Generating the Command and Explanation\*\*

I'll provide the command like this:

```
`git reset --soft HEAD2 && git commit -m "Your new message"``
```

This command moves HEAD back two commits while keeping the changes staged, and then creates a new commit with your specified message. Since the user mentioned they haven't pushed yet, no need to worry about needing a force push. It's concise and should work perfectly on Windows as well since Git commands are cross-platform!

- Run:  
git reset --soft HEAD~2 && git commit -m "New combined commit message"  
Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent git commit creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium ~

gpt-5-mini (medium) (1x)-

~

gpt-5-mini (medium) (1x)

Type @

to mention files, / for commands, or ? for shortcuts

switch mode

GitHub Copilot v0.0.410

Describe a task to get started.

----

cross-platform!

• Run:

```
git reset --soft HEAD~2 && git commit -m "New combined commit message"
```

Explanation: `git reset --soft HEAD~2` moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent `git commit` creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

▢ Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode



GitHub Copilot v0.0.410



Describe a task to get started.



Pick a model with `/model`. Send this session to GitHub with `/delegate`, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

▢ No copilot instructions found. Run `/init` to generate a `copilot-instructions.md` file for this project.

• Model changed to: gpt-5-mini (medium)

▢ I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

The user is asking for a Git command to combine the last two commits into a single one, while keeping all changes in the working directory. Since we're not invoking any tools, I'll simply provide the command: ``git rebase -i HEAD2``. This works by allowing the user to squash the last two commits and create a new message, perfect for cleaning up their commit history before pushing.\*\*Evaluating Git Command Options\*\*

I have a couple of options for the user's request to combine the last two commits into one. The first option is using ``git reset --soft HEAD2 && git commit -m "New message"``. This integrates the last two commits into one with a new message while preserving the working directory, as the ``--soft`` option keeps the index intact. Alternatively, they could use ``git rebase -i HEAD2`` to squash commits, but that's a bit more complex.\*\*Providing the Git Command\*\*

I'll give the command: ``git reset --soft HEAD2 && git commit -m "New combined commit message"``. This effectively combines the last two commits into one while preserving all changes in the working directory, as the ``--soft`` option does not alter the working directory or staged changes. Since the user asked for clarity, I'll stick to this command and a clear explanation, keeping it concise and to the point. No tool calls are needed here!\*\*Generating the Command and Explanation\*\*

I'll provide the command like this:

```
`git reset --soft HEAD2 && git commit -m "Your new message"``
```

This command moves HEAD back two commits while keeping the changes staged, and then creates a new commit with your specified message. Since the user mentioned they haven't pushed yet, no need to worry about needing a force push. It's concise and should work perfectly on Windows as well since Git commands are cross-platform!

• Run:

```
git reset --soft HEAD~2 && git commit -m "New combined commit message"
```

Explanation: `git reset --soft HEAD~2` moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent `git commit` creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

cross-platform!

• Run:

```
git reset --soft HEAD~2 && git commit -m "New combined commit message"
```

Explanation: `git reset --soft HEAD~2` moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent `git commit` creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

🔍 Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode



GitHub Copilot v0.0.410



Describe a task to get started.



Pick a model with `/model`. Send this session to GitHub with `/delegate`, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

• 🔍 No copilot instructions found. Run `/init` to generate a `copilot-instructions.md` file for this project.

• Model changed to: gpt-5-mini (medium)

🔍 I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

• Model changed to: gpt-5-mini (medium)

🔍 I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

🔍 **\*\*Generating Git Command\*\***

The user is asking for a Git command to combine the last two commits into a single one, while keeping all changes in the working directory. Since we're not invoking any tools, I'll simply provide the command: ``git rebase -i HEAD2``. This works by allowing the user to squash the last two commits and create a new message, perfect for cleaning up their commit history before pushing. **\*\*Evaluating Git Command Options\*\***

I have a couple of options for the user's request to combine the last two commits into one. The first option is using ``git reset --soft HEAD2 && git commit -m "New message"``. This integrates the last two commits into one with a new message while preserving the working directory, as the ``--soft`` option keeps the index intact. Alternatively, they could use ``git rebase -i HEAD2`` to squash commits, but that's a bit more complex. **\*\*Providing the Git Command\*\***

I'll give the command: ``git reset --soft HEAD2 && git commit -m "New combined commit message"``. This effectively combines the last two commits into one while preserving all changes in the working directory, as the ``--soft`` option does not alter the working directory or staged changes. Since the user asked for clarity, I'll stick to this command and a clear explanation, keeping it concise and to the point. No tool calls are needed here! **\*\*Generating the Command and Explanation\*\***

I'll provide the command like this:

```
`git reset --soft HEAD2 && git commit -m "Your new message"`
```

This command moves HEAD back two commits while keeping the changes staged, and then creates a new commit with your specified message. Since the user mentioned they haven't pushed yet, no need to worry about needing a force push. It's concise and should work perfectly on Windows as well since Git commands are cross-platform!

• Run:

```
git reset --soft HEAD~2 && git commit -m "New combined commit message"
```

cross-platform!

• Run:

```
git reset --soft HEAD~2 && git commit -m "New combined commit message"
```

Explanation: `git reset --soft HEAD~2` moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent `git commit` creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

📄 Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode



GitHub Copilot v0.0.410



Describe a task to get started.



Pick a model with `/model`. Send this session to GitHub with `/delegate`, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

📄 No copilot instructions found. Run `/init` to generate a `copilot-instructions.md` file for this project.

• Model changed to: gpt-5-mini (medium)

📄 I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

Pick a model with `/model`. Send this session to GitHub with `/delegate`, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

📄 No copilot instructions found. Run `/init` to generate a `copilot-instructions.md` file for this project.

• Model changed to: gpt-5-mini (medium)

📄 I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

📄 **\*\*Generating Git Command\*\***

The user is asking for a Git command to combine the last two commits into a single one, while keeping all changes in the working directory. Since we're not invoking any tools, I'll simply provide the command: ``git rebase -i HEAD2``. This works by allowing the user to squash the last two commits and create a new message, perfect for cleaning up their commit history before pushing. **\*\*Evaluating Git Command Options\*\***

I have a couple of options for the user's request to combine the last two commits into one. The first option is using ``git reset --soft HEAD2 && git commit -m "New message"``. This integrates the last two commits into one with a new message while preserving the working directory, as the ``--soft`` option keeps the index intact. Alternatively, they could use ``git rebase -i HEAD2`` to squash commits, but that's a bit more complex. **\*\*Providing the Git Command\*\***

I'll give the command: ``git reset --soft HEAD2 && git commit -m "New combined commit message"``. This effectively combines the last two commits into one while preserving all changes in the working directory, as the ``--soft`` option does not alter the working directory or staged changes. Since the user asked for clarity, I'll stick to this command and a clear explanation, keeping it concise and to the point. No tool calls are needed here! **\*\*Generating the Command and Explanation\*\***

I'll provide the command like this:

```
`git reset --soft HEAD2 && git commit -m "Your new message"``
```

cross-platform!

• Run:

```
git reset --soft HEAD~2 && git commit -m "New combined commit message"
```

Explanation: `git reset --soft HEAD~2` moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent `git commit` creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

🔍 Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode

🔍 GitHub Copilot v0.0.410

🔍 Describe a task to get started.

🔍

Pick a model with `/model`. Send this session to GitHub with `/delegate`, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

• 🔍 No copilot instructions found. Run `/init` to generate a `copilot-instructions.md` file for this project.

• Model changed to: gpt-5-mini (medium)

🔍 I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

shift+tab switch mode

🔍 GitHub Copilot v0.0.410

🔍 Describe a task to get started.

🔍

Pick a model with `/model`. Send this session to GitHub with `/delegate`, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

• 🔍 No copilot instructions found. Run `/init` to generate a `copilot-instructions.md` file for this project.

• Model changed to: gpt-5-mini (medium)

🔍 I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

🔍 **\*\*Generating Git Command\*\***

The user is asking for a Git command to combine the last two commits into a single one, while keeping all changes in the working directory. Since we're not invoking any tools, I'll simply provide the command: ``git rebase -i HEAD2``. This works by allowing the user to squash the last two commits and create a new message, perfect for cleaning up their commit history before pushing.**\*\*Evaluating Git Command Options\*\***

I have a couple of options for the user's request to combine the last two commits into one. The first option is using ``git reset --soft HEAD2 && git commit -m "New message"``. This integrates the last two commits into one with a new message while preserving the working directory, as the ``--soft`` option keeps the index intact. Alternatively, they could use ``git rebase -i HEAD2`` to squash commits, but that's a bit more complex.**\*\*Providing the Git Command\*\***

I'll give the command: ``git reset --soft HEAD2 && git commit -m "New combined commit message"``. This effectively combines the last two commits into one while preserving all changes in the working directory, as the ``--soft`` option does not alter the working directory or staged changes. Since the user asked for clarity, I'll stick to this command and a clear explanation, keeping it concise and to the point. No tool calls are needed here!**\*\*Generating the Command and Explanation\*\***

cross-platform!

- Run:  
git reset --soft HEAD~2 && git commit -m "New combined commit message"  
Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged), and the subsequent git commit creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode

GitHub Copilot v0.0.410

Describe a task to get started.

Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

- No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.
- Model changed to: gpt-5-mini (medium)

I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

and the subsequent git commit creates a single new commit with the provided message, preserving all working-directory changes.

~

gpt-5-mini (medium) (1x)

Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode

GitHub Copilot v0.0.410

Describe a task to get started.

Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.

- No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.
- Model changed to: gpt-5-mini (medium)

I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the working directory. Generate the correct Git command using the git agent and briefly explain why it works.

\*\*Generating Git Command\*\*

The user is asking for a Git command to combine the last two commits into a single one, while keeping all changes in the working directory. Since we're not invoking any tools, I'll simply provide the command: `git rebase -i HEAD~2`. This works by allowing the user to squash the last two commits and create a new message, perfect for cleaning up their commit history before pushing.

\*\*Evaluating Git Command Options\*\*

I have a couple of options for the user's request to combine the last two commits into one. The first option is using `git reset --soft HEAD~2 && git commit -m "New combined commit message"`. This command moves the HEAD pointer back two commits, leaving the changes from those commits staged. Then, a new commit is created with the provided message, effectively squashing the last two commits into one. This approach is useful when you want to keep the working directory clean and have a single commit representing a set of changes. The second option is using `git rebase -i HEAD~2`. This command opens an interactive rebase session for the last two commits. The user can choose to squash the commits, which would result in a single commit. However, this method is more complex and might be confusing for users who are not familiar with interactive rebases. Given the user's request, the first option is more straightforward and easier to understand. It directly addresses the need to combine the last two commits into one without requiring the user to understand the intricacies of rebasing. Therefore, I will recommend the first option to the user.

```
cross-platform!


• Run:
git reset --soft HEAD~2 && git commit -m "New combined commit message"
Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged),
and the subsequent git commit creates a single new commit with the provided message, preserving all working-directory changes.

~ gpt-5-mini (medium) (1x)

Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode




GitHub Copilot v0.0.410
Describe a task to get started.
Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Copilot uses AI, so always check for mistakes.



• No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.

• Model changed to: gpt-5-mini (medium)

I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message while preserving all changes in the
working directory. Generate the correct Git command using the git agent and briefly explain why it works.

`git reset --soft HEAD2 && git commit -m "Your new message"`

This command moves HEAD back two commits while keeping the changes staged, and then creates a new commit with your specified message. Since the user mentioned they
haven't pushed yet, no need to worry about needing a force push. It's concise and should work perfectly on Windows as well since Git commands are cross-platform!

• Run:
git reset --soft HEAD~2 && git commit -m "New combined commit message"
Explanation: git reset --soft HEAD~2 moves HEAD back two commits but leaves the index and working tree intact (so the changes from those commits remain staged),
and the subsequent git commit creates a single new commit with the provided message, preserving all working-directory changes.

~ gpt-5-mini (medium) (1x)

Type @ to mention files, / for commands, or ? for shortcuts

shift+tab switch mode
```

2. Accuracy Ranking and Justification

Accuracy Ranking: 9/10

I give the AI's response a perfect score. The last two commits are correctly combined while maintaining staged changes by using the recommended method, which involves running git reset --soft HEAD~2 and then creating a new commit. By effectively rewriting the local commit history prior to pushing and avoiding interactive rebasing complexity, this method lowers the possibility of user error.