

# GitButler CLI Cheat Sheet

Essential commands for working with GitButler

GitButler is a Git client that allows you to work on multiple branches simultaneously. This cheat sheet features the most important and commonly used GitButler CLI commands for easy reference.

## Setup & Installation

Getting started with GitButler CLI and configuring your environment

```
but setup
```

Initialize GitButler in your current repository

```
but teardown
```

Remove GitButler from the current repository

```
but config
```

Check or update your important config settings

## Working with Changes

Managing file changes across virtual branches

```
but status
```

Show the status of all virtual branches and uncommitted changes

```
but move [file] [branch]
```

Move a file's changes to a different virtual branch

```
but diff
```

Show changes in the current working directory

```
but commit -m "[message]"
```

Commit changes in a virtual branch

## Virtual Branches

Creating and managing virtual branches for parallel development

```
but branch
```

List all virtual branches

```
but branch create [name]
```

Create a new virtual branch

```
but branch apply [branch]
```

Apply (enable) a virtual branch to your working directory

```
but branch unapply [branch]
```

Unapply (disable) a virtual branch from your working directory

```
but branch delete [branch]
```

Delete a virtual branch

## Integration

Pushing changes and creating pull requests

```
but push [branch]
```

Push a virtual branch to the remote repository

```
but pr create [branch]
```

Create a pull request for a virtual branch

```
but pr list
```

List all pull requests associated with virtual branches

# GitButler CLI Cheat Sheet

## Workspace Management

Managing your GitButler workspace and state

```
but workspace
```

Show current workspace information and active branches

```
but workspace reset
```

Reset workspace to a clean state

## Advanced Operations

Advanced commands for power users

```
but integrate [branch]
```

Integrate a remote branch into your workspace

```
but snapshot create
```

Create a snapshot of your current workspace state

```
but snapshot restore [id]
```

Restore a previous workspace snapshot

```
but log
```

Show the history of all virtual branch operations