

Commit Commands

Project Skara provides committers with additional commit commands that enable additional functionality. A *commit command* is a comment made to a commit that starts with a slash ("/"), for example `/tag` or `/backport`. The command may appear on any line and with an arbitrary amount of whitespace in front of it, but it must be the first non whitespace characters that appear on a particular line.

Skara will only evaluate a given command once, so if you make a mistake and get an error message back, you need to enter a new command in a new comment. Editing the previous comment will not have any effect.

Note that if you are using Skara on Gitlab, there are built in commands that clash with some of the Skara commands. To enforce use of the Skara command and not the Gitlab variant, you can put some whitespace in front of the command in the comment. Also, you can prepend `/skara` as a prefix to the command. For instance, to execute the `/tag` command, you can use `/skara tag 1.1`, it will be interpreted as `/tag 1.1` by SKARA bot. The command that currently conflict with GitLab commands is `/tag`.

Commands

- [/backport](#)
 - [Syntax](#)
 - [Description](#)
 - [Examples](#)
- [/tag](#)
 - [Syntax](#)
 - [Description](#)
 - [Examples](#)
- [/branch](#)
 - [Syntax](#)
 - [Description](#)
 - [Examples](#)
- [/help](#)
 - [Syntax](#)
 - [Description](#)
 - [Examples](#)

`/backport`

Syntax

```
/backport <repo> [<branch> ]  
/backport [<repo>]:<branch>
```

Description

Applies the commit onto the given branch in the given repository and then shows to a link to create a pull request with the changes. In the first syntax, the branch is optional, if not given, the default for the repository is used. In the second syntax, repo is optional, if not given, the same repository where the command is issued is used. If the commit does not apply then a message is shown describing the files containing conflicts. See [Backports](#).

If the target repository is configured to support [dependent pull requests](#), it's possible to do this with the backport command by supplying the appropriate `pr /x` branch.

The pull request will be created from a branch in a shared fork of the target repository. On GitHub, this repository is owned by the `openjdk-bots` organization. The first time you issue the `/backport` command for a specific target repository, you will receive an invitation to collaborate in the fork repository. This invitation needs to be accepted to be able to further update the backport pull request with more changes.

Examples

- `/backport jdk16u`
- `/backport jfx jfx14`
- `/backport jdk17u-dev pr/4711`
- `/backport :jfx22`
- `/backport jdk17u-dev:pr/4711`

`/tag`

Syntax

```
/tag <name>
```

Description

Creates a tag for the commit with the given name. Note that only integrators in a repository are allowed to use this command. To become an integrator, file a Skara admin task issue explaining why.

Examples

- `/tag jdk-16+22`
- `/tag 15.0.1+1`

`/branch`

Syntax

`/branch <name>`

Description

Creates a new branch with the given name, starting at the commit. Note that only integrators in a repository are allowed to use this command. To become an integrator, file a Skara admin task issue explaining why.

Examples

- `/branch jdk22.0.1`

`/help`

Syntax

`/help`

Description

Shows help for all commit commands.

Examples

- `/help`