

# Command Pattern

[Project home](#)

## Description

"Encapsulate a request as an object, thereby letting you parameterize clients with different requests, queue or log requests, and support undoable operations."(Design Patterns: Elements of Reusable Object-Oriented Software (ISBN 0-201-63361-2))

- All specific commands are implementations of their superclass *Command*
- Each command type is build in a specific way and performs other tasks
- All commands have an *execute()* method in common which will be called to perform the action

## Use in this project

Since for every step type different actions need to be performed, this is an excellent pattern to use. Many different actions can be created and added to the step which will perform them. Each step type can execute a certain task or group of tasks. This way it is easy to add more actions to future versions of the workflow system. It is also a clean way to implement the required functionalities since one can define an interface to build a certain type of actions that each can perform the specific tasks needed once they get executed by the system.