# Introduction and prerequisites

Truffle uses lerna to manage multi-package repositories. Each Truffle module is defined in its own npm package in the `packages/` directory.

The entry point of these modules is `@truffle/core`. This is where the command line parser is setup.

Install lerna:

```shell

$ npm install -g lerna

$ npm install -g yarn

```

# The command flow

The heart of Truffle lies in the `@truffle/core` package. Whenever a command

is run, `packages/core/cli.js` gets run with everything following `truffle`

(on the command line) being passed in as arguments. In other words, if you run

`truffle migrate --network myNetwork`, then `packages/core/cli.js` gets run

with "migrate" and "--network myNetwork" as arguments.

Throughout the course of running `packages/core/cli.js`, Truffle parses out what

commands and options the user has provided, instantiates an instance of the

`Command` class, and calls the `run` method on that instance. The `run` method

is the interface that `cli.js` uses for ALL commands. You can find all of the

specific command files (one file for each command) at

`packages/core/lib/commands`. From the run method of each command you should be

able to trace the command lifecycle to libraries and modules in `@truffle/core`

as well as other packages in the monorepo.

# Add a new command in truffle

### Create a new lerna package

```shell

$ lerna create truffle-mycmd

```

### Add the package to `@truffle/core`

```shell

$ lerna add truffle-mycmd --scope=@truffle/core

```

### Create a new command in `@truffle/core`

Create a new file in `packages/core/lib/commands/`, let's call it `mycmd.js`.

```shell

$ cat << EOF > core/lib/commands/mycmd.js

const command = {

command: "mycmd",

description: "Run mycmd",

builder: {},

help: {

usage: "truffle mycmd",

options: []

},

run: function(options, done) {

const mycmd = require("truffle-mycmd");

// TODO: write the run command here, something like:

// mycmd(options, done)

}

};

module.exports = command;

EOF

```

### Link it from the commands/index.js file

```diff

--- packages/core/lib/commands/index.js

+++ packages/core/lib/commands/index.js

@@ -1,4 +1,5 @@

module.exports = {

+ mycmd: require("./mycmd"),

```

From there, you should see it in the help screen:

```shell

$ cd packages/core

$ node cli.js

Truffle v5.0.0-beta.1 - a development framework for Ethereum

Usage: truffle <command> [options]

Commands:

mycmd Run mycmd

build Execute build pipeline (if configuration present)

[...]

```

### Write your module/command

The setup is done, you can now write your command and organize your module as you want in: `packages/truffle-mycmd/`. You can have a look at `packages/box/` which is a good starting example to follow.