

# Changelog

All notable changes to this project will be documented in this file.

The format is based on [Keep a Changelog](#) and this project adheres to [Semantic Versioning](#). (Format adopted after v3.0.0.)

## 5.1.0 (2020-04-25)

### Added

- support for multiple command aliases, the first of which is shown in the auto-generated help ([#531](#), [#1236])
- configuration support in `addCommand()` for `hidden` and `isDefault` ([#1232])

### Fixed

- omit masked help flags from the displayed help ([#645](#), [#1247])
- remove old short help flag when change help flags using `helpOption` ([#1248])

## Changed

- remove use of arguments to improve auto-generated help in editors ([#1235])
- rename .command() configuration nohelp to hidden (but not remove old support) ([#1232])
- improvements to documentation
- update dependencies
- update tested versions of node
- eliminate lint errors in TypeScript ([#1208])

## 5.0.0 (2020-03-14)

## Added

- support for nested commands with action-handlers ([1](#) [#764](#))
- .addCommand() for adding a separately configured command ([#764](#) [\[#1149\]](#))
- allow a non-executable to be set as the default command ([#742](#) [\[#1149\]](#))
- implicit help command when there are subcommands (previously only if executables) ([#1149])
- customise implicit help command with .addHelpCommand() ([#1149])

- display error message for unknown subcommand, by default ([#432](#) [#1088] [#1149])
- display help for missing subcommand, by default ([#1088] [#1149])
- combined short options as single argument may include boolean flags and value flag and value (e.g. `-a -b -p 80` can be written as `-abp80`) ([#1145])
- `.parseOption()` includes short flag and long flag expansions ([#1145])
- `.helpInformation()` returns help text as a string, previously a private routine ([#1169])
- `.parse()` implicitly uses `process.argv` if arguments not specified ([#1172])
- optionally specify where `.parse()` arguments “from”, if not following node conventions ([#512](#) [#1172])
- suggest help option along with unknown command error ([#1179])
- TypeScript definition for `commands` property of `Command` ([#1184])
- `export program` property ([#1195])
- `createCommand` factory method to simplify subclassing ([#1191])

## Fixed

- preserve argument order in subcommands ([#508](#) [#962](#) [#1138])

- do not emit command : \* for executable subcommands ([#809](#))
- action handler called whether or not there are non-option arguments ([#1062] [#1149])
- combining option short flag and value in single argument now works for subcommands ([#1145])
- only add implicit help command when it will not conflict with other uses of argument ([#1153] [#1149])
- implicit help command works with command aliases ([#948](#))
- options are validated whether or not there is an action handler ([#1149])

## Changed

- *Breaking* . args contains command arguments with just recognised options removed ([#1032](#) [#1138])
- *Breaking* display error if required argument for command is missing ([#995](#) [#1149])
- tighten TypeScript definition of custom option processing function passed to . option () ([#1119])
- *Breaking* . allowUnknownOption () ([#802](#) [#1138])
- unknown options included in arguments passed to command action handler
- unknown options included in . args
- only recognised option short flags and long flags are expanded (e.g. -ab or --foo=bar) ([#1145])

[#1138]:	https://github.com/tj/commander.js/pull/1138
[#1145]:	https://github.com/tj/commander.js/pull/1145
[#1146]:	https://github.com/tj/commander.js/pull/1146
[#1149]:	https://github.com/tj/commander.js/pull/1149
[#1153]:	https://github.com/tj/commander.js/issues/1153
[#1157]:	https://github.com/tj/commander.js/pull/1157
[#1159]:	https://github.com/tj/commander.js/pull/1159
[#1165]:	https://github.com/tj/commander.js/pull/1165
[#1169]:	https://github.com/tj/commander.js/pull/1169
[#1172]:	https://github.com/tj/commander.js/pull/1172
[#1179]:	https://github.com/tj/commander.js/pull/1179
[#1180]:	https://github.com/tj/commander.js/pull/1180
[#1191]:	https://github.com/tj/commander.js/pull/1184
[#1195]:	https://github.com/tj/commander.js/pull/1191
[#1208]:	https://github.com/tj/commander.js/pull/1195
[#1232]:	https://github.com/tj/commander.js/pull/1208
[#1235]:	https://github.com/tj/commander.js/pull/1232
[#1236]:	https://github.com/tj/commander.js/pull/1235
[#1247]:	https://github.com/tj/commander.js/pull/1236
[#1248]:	https://github.com/tj/commander.js/pull/1247
	https://github.com/tj/commander.js/pull/1248

## 3.0.0-0 Prerelease / 2019-07-28

(Released as 3.0.0)

### Older versions

- [2.x](#)
- [1.x](#)
- [0.x](#)

[#1040]:	<a href="https://github.com/tj/commander.js/pull/1040">https://github.com/tj/commander.js/pull/1040</a>
[#1047]:	<a href="https://github.com/tj/commander.js/pull/1047">https://github.com/tj/commander.js/pull/1047</a>
[#1048]:	<a href="https://github.com/tj/commander.js/pull/1048">https://github.com/tj/commander.js/pull/1048</a>
[#1049]:	<a href="https://github.com/tj/commander.js/pull/1049">https://github.com/tj/commander.js/pull/1049</a>
[#1051]:	<a href="https://github.com/tj/commander.js/pull/1051">https://github.com/tj/commander.js/pull/1051</a>
[#1052]:	<a href="https://github.com/tj/commander.js/pull/1052">https://github.com/tj/commander.js/pull/1052</a>
[#1053]:	<a href="https://github.com/tj/commander.js/pull/1053">https://github.com/tj/commander.js/pull/1053</a>
[#1062]:	<a href="https://github.com/tj/commander.js/pull/1062">https://github.com/tj/commander.js/pull/1062</a>
[#1071]:	<a href="https://github.com/tj/commander.js/pull/1071">https://github.com/tj/commander.js/pull/1071</a>
[#1081]:	<a href="https://github.com/tj/commander.js/pull/1081">https://github.com/tj/commander.js/pull/1081</a>
[#1088]:	<a href="https://github.com/tj/commander.js/issues/1088">https://github.com/tj/commander.js/issues/1088</a>
[#1091]:	<a href="https://github.com/tj/commander.js/pull/1091">https://github.com/tj/commander.js/pull/1091</a>
[#1096]:	<a href="https://github.com/tj/commander.js/pull/1096">https://github.com/tj/commander.js/pull/1096</a>
[#1102]:	<a href="https://github.com/tj/commander.js/pull/1102">https://github.com/tj/commander.js/pull/1102</a>
[#1118]:	<a href="https://github.com/tj/commander.js/pull/1118">https://github.com/tj/commander.js/pull/1118</a>
[#1119]:	<a href="https://github.com/tj/commander.js/pull/1119">https://github.com/tj/commander.js/pull/1119</a>
[#1133]:	<a href="https://github.com/tj/commander.js/pull/1133">https://github.com/tj/commander.js/pull/1133</a>

- *Breaking* `.parseOptions()` ([#1138])
  - `args` in returned result renamed `operands` and does not include anything after first unknown option
  - `unknown` in returned result has arguments after first unknown option including operands, not just options and values
- *Breaking* `.on('command:*', callback)` and other command events passed (changed) results from `.parseOptions`, i.e. operands and unknown ([#1138])
- refactor `Option` from prototype to class ([#1133])
- refactor `Command` from prototype to class ([#1159])
- changes to error handling ([#1165])
  - throw for author error, not just display message
  - preflight for variadic error
  - add tips to missing subcommand executable
- TypeScript fluent return types changed to be more subclass friendly, return `this` rather than `Command` ([#1180])
- `.parseAsync` returns `Promise<this>` to be consistent with `.parse()` ([#1180])
- update dependencies

### Removed

- removed `EventEmitter` from TypeScript definition for `Command`, eliminating implicit peer dependency on `@types/node` ([#1146])
- removed private function `normalize` (the functionality has been integrated into `parseOptions`) ([#1145])

- parseExpectedArgs is now private ([#1149])

### Migration Tips

If you use `.on( 'command: * ' )` or more complicated tests to detect an unrecognised subcommand, you may be able to delete the code and rely on the default behaviour.

If you use `program.args` or more complicated tests to detect a missing subcommand, you may be able to delete the code and rely on the default behaviour.

If you use `.command( ' * ' )` to add a default command, you may be able to switch to `isDefault: true` with a named command.

## 5.0.0-4 (2020-03-03)

(Released in 5.0.0)

## 5.0.0-3 (2020-02-20)

(Released in 5.0.0)

- **Breaking** default value: defining `--no-foo` after defining `--foo` leaves the default value unchanged (previously set it to `false`)
- allow boolean default value, such as from environment ([#987](#))
- Increment inspector port for spawned subcommands ([#991](#))
- e.g. `node --inspect myCommand.js clone`

### Migration Tips

The custom event for a negated option like `--no-foo` is `option:no-foo` (previously `option:foo`).

```

program
  .option( '--no-foo' )
  .on( 'option:no-foo', ( ) => {
    console.log( 'removing foo' );
  } );

```

When using TypeScript, adding a command does not allow an explicit undefined for an unwanted executable description (e.g. for a command with an action handler).

```

program
  .command( 'action1', undefined, {
    noHelp: true } ) // No longer
    valid
  .command( 'action2', { noHelp:
    true } ) // Correct

```

- Change docs for `.command` to contrast action handler vs git-style executable. ([#938](#) [#990](#))
- **Breaking** Change TypeScript to use overloaded function for `.command`. ([#938](#) [#990](#))
- Change to use straight quotes around strings in error messages (like `'this'` instead of ``this``) ([#915](#))
- Add TypeScript "reference types" for node ([#974](#))
- Add support for hyphen as an option argument in subcommands ([#697](#))
- Add support for a short option flag and its value to be concatenated for action handler subcommands ([#599](#))
  - e.g. `-p 80` can also be supplied as `-p80`
- Add executable arguments to spawn in win32, for git-style executables ([#611](#))
  - e.g. `node --harmony myCommand.js clone`
- Add parent command as prefix of subcommand in help ([#980](#))
- Add optional custom description to `.version` ([#963](#))
  - e.g. `program.version('0.0.1', '-v, --vers', 'output the current version')`
- Add `.helpOption(flags, description)` routine to customise help flags and description ([#963](#))
  - e.g. `.helpOption('-e, --HELP', 'read more information')`
- Fix behavior of `--no-*` options ([#795](#))
  - can now define both `--foo` and `--no-foo`
- **Breaking** custom event listeners: `--no-foo` on cli now emits `option:no-foo` (previously `option:foo`)

## 5.0.0-2 (2020-02-10)

(Released in 5.0.0)

## 5.0.0-1 (2020-02-08)

(Released in 5.0.0)

## 5.0.0-0 (2020-02-02)

(Released in 5.0.0)

## 4.1.1 (2020-02-02)

### Fixed

- TypeScript definition for `.action()` should include Promise for async ([#1157](#))

## 4.1.0 (2020-01-06)

### Added

- two routines to change how option values are handled, and eliminate name clashes with command properties ([#933] [#1102])
  - see storeOptionsAsProperties and passCommandToAction in README
- `.parseAsync` to use instead of `.parse` if supply async action handlers ([#806](#) [#1118])

### Fixed

- Remove trailing blanks from wrapped help text ([#1096])

### Changed

- update dependencies
- extend security coverage for Commander 2.x to 2020-02-03
- improvements to README
- improvements to TypeScript definition documentation
- move old versions out of main CHANGELOG
- removed explicit use of `ts - node` in tests

8

## 3.0.1 (2019-08-30)

### Added

- `.name` and `.usage` to README ([#1010](#))
- Table of Contents to README ([#1010](#))
- TypeScript definition for `executableFile` in `CommandOptions` ([#1028](#))

### Changed

- consistently use `const` rather than `var` in README ([#1026](#))

### Fixed

- help for sub commands with custom `executableFile` ([#1018](#))

## 3.0.0 / 2019-08-08

- Add option to specify executable file name ([#999](#))
  - e.g. `command('clone', 'clone description', { executableFile: 'myClone' })`

13



## **2.20.1 (2019-09-29)**

### **Fixed**

- Improve tracking of executable subcommands.

### **Changed**

- update development dependencies

## **3.0.2 (2019-09-27)**

### **Fixed**

- Improve tracking of executable subcommands.

### **Changed**

- update development dependencies

## **4.0.1 (2019-11-12)**

### **Fixed**

- display help when requested, even if there are missing required options ([#1091])

## **4.0.0 (2019-11-02)**

### **Added**

- automatically wrap and indent help descriptions for options and commands ([#1051])
- `.exitOverride()` allows override of calls to `process.exit` for ad
- support for declaring required options with `.requiredOptions()` ([#1071])
- GitHub Actions support ([#1027](#))
- translation links in README

## Changed

- dev: switch tests from Sinon+Should to Jest with major rewrite of tests ([#1035](#))
- call default subcommand even when there are unknown options ([#1047](#))
- *Breaking* Commander is only officially supported on Node 8 and above, and requires Node 6 ([#1053](#))

## Fixed

- *Breaking* keep command object out of program.args when action handler called ([#1048](#))
- also, action handler now passed array of unknown arguments
- complain about unknown options when program argument supplied and action handler ([#1049](#))
- this changes parameters to command: \* event to include unknown arguments
- removed deprecated customFds option from call to `child_process.spawn` ([#1052](#))
- rework TypeScript declarations to bring all types into imported namespace ([#1081](#))

## Migration Tips

### Testing for no arguments

If you were previously using code like:

```
if (!program.args.length) ...
```

a partial replacement is:

```
if (program.rawArgs.length < 3) ...
```

## 4.0.0-1 Prelease (2019-10-08)

(Released in 4.0.0)

## 4.0.0-0 Prelease (2019-10-01)

(Released in 4.0.0)