
codot Documentation

Release 0.1

Garrett Powell

Feb 21, 2018

Contents:

1	codot.1	1
1.1	SYNOPSIS	1
1.2	DESCRIPTION	1
1.3	GLOBAL OPTIONS	2
1.4	COMMANDS	2
1.5	EXAMPLES	2
1.6	FILES	3
2	Indices and tables	5

1.1 SYNOPSIS

`codot [global_options] command [command_options] [command_args]`

1.2 DESCRIPTION

codot is a program for consolidating your dotfiles so that settings for multiple applications can be modified from one set of files.

1.2.1 Terminology

Source File A source file is a file created by an application to allow the user to configure that application (e.g. `~/.vimrc`). There is an example source file under *EXAMPLES*.

Template File A template file is a copy of a source file that has had certain values replaced with user-defined identifiers. There is an example template file under *EXAMPLES*.

Config File A config file is a file created by the user that acts as a centralized place to modify values from multiple source files. Config files have the following format:

- Lines starting with a hash symbol ‘#’ serve as comments.
- Options are key-value pairs separated by an equals sign ‘=’, where each key corresponds to the name of an identifier in one or more template files.

Config files go in the ‘config’ directory (see *FILES*) and each config file must have a ‘.conf’ extension. There is an example config file under *EXAMPLES*.

Identifier An identifier is a string used in a template file to signal where a value from a config file should be substituted in. Each identifier has a name, which corresponds to the name of an option in a config file. The default identifier format is ‘`{{%s}}`’, where ‘%s’ represents the name of the identifier. The same identifier name can be used more than once and in more than one template file.

Role A role is a way to easily swap out different sets of config values. Each set of values goes in a separate config file, one of which can fill the role at a time. To create a role, put one or more config files in a subdirectory of the 'config' directory (see *FILES*). The name of this subdirectory is the name of the role. The selected config file for a role can be switched using the **role** command.

Here are the steps for getting started with **codot**:

1. Identify which values from which source files you want to consolidate.
2. Use the **add-template** command to open the source files in your text editor. Then, replace the values you selected with identifiers. Name these identifiers whatever you want.
3. Create one or more config files that contain options named after the identifiers you created.
4. Populate those config files with values.
5. Start the daemon or run the **sync** command.

1.3 GLOBAL OPTIONS

-help Print a usage message and exit.

-version Print the version number and exit.

-debug Print a full stack trace instead of an error message if an error occurs.

-q, -quiet Suppress all non-error output.

1.4 COMMANDS

add-template [*options*] *files...* Open one or more source *files* in your editor and save them each as a template file.

-r, -revise If the template file already exists, edit it instead of creating a new one.

rm-template [*options*] *files...* Remove the template file for each of the source *files* specified. Remove from each config file any option that isn't being referenced in at least one template file.

-l, -leave-options Do not remove options from config files.

sync [*options*] Update source files with changes from config files. If those source files have been modified by the user since they were last synced, skip them. If the daemon is running, this command is run automatically whenever a config file or template file is modified.

-o, -overwrite Update source files even if they've been modified by the user since they were last synced.

list [*options*] List all identifiers and highlight the ones that aren't in any config file.

-g, -group Group identifiers by their template file.

role [*role_name* [*config_name*]] Make *config_name* the currently selected config file in the role named *role_name*. If *config_name* is not specified, print a list of config files available for that role. If *role_name* is not specified, print a table of all roles.

1.5 EXAMPLES

This is an example of a source file.

```

1 bar {
2   status_command i3status
3   position top
4   font pango:DejaVuSans 12
5
6   colors {
7     statusline #e0e0e0
8     separator #838383
9     background #212121
10  }
11 }

```

This is an example of a template file using the default identifier format.

```

1 bar {
2   status_command i3status
3   position top
4   font pango:{{Font}} {{FontSize}}
5
6   colors {
7     statusline {{ForegroundColor}}
8     separator {{AccentColor}}
9     background {{BackgroundColor}}
10  }
11 }

```

This is an example of a config file.

```

1 # These are colors for the cross-application color scheme.
2 ForegroundColor=#e0e0e0
3 AccentColor=#838383
4 BackgroundColor=#212121
5
6 # These are cross-application font settings.
7 Font=DejaVuSans
8 FontSize=12

```

1.6 FILES

~/config/codot/ This is the **codot** program directory. The program will respect `XDG_CONFIG_HOME` and, if it is set, put the directory there instead.

config/ This directory is where all config files and roles are stored. Config files must have a `.conf` extension.

templates/ This directory is where all template files are stored. The file structure under this directory mimics the file structure under the user's home directory.

settings.conf This file is for configuring the behavior of **codot**.

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`