
Config Suite TUI

Release 0.1.10-8-g7a52e62

Equinor ASA

May 04, 2021

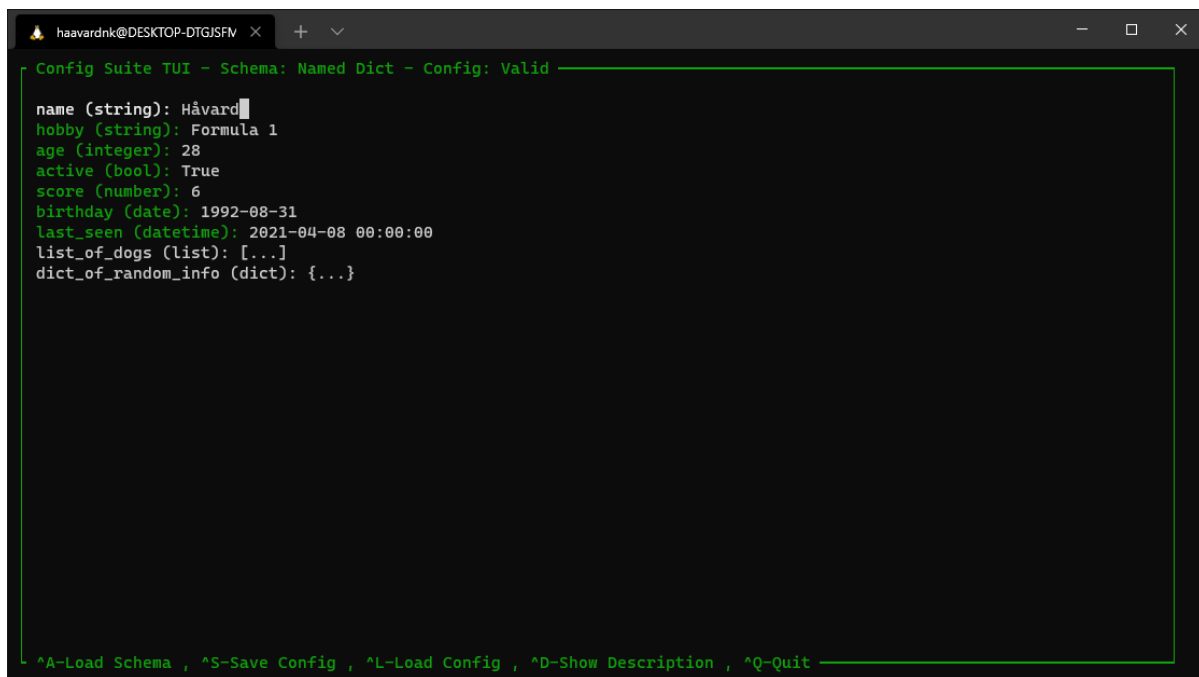
CONTENTS:

1	User Guide	1
1.1	Introduction	1
1.2	Usage	2
1.3	Schema Plugin	4

USER GUIDE

1.1 Introduction

Config Suite TUI is a text-based user interface extension for [Config Suite](#). It provides a simplified, visual way of creating and editing configuration files for a given schema.



```
haavardnk@DESKTOP-DTGJSFV x + v
Config Suite TUI - Schema: Named Dict - Config: Valid
name (string): Håvard
hobby (string): Formula 1
age (integer): 28
active (bool): True
score (number): 6
birthday (date): 1992-08-31
last_seen (datetime): 2021-04-08 00:00:00
list_of_dogs (list): [...]
dict_of_random_info (dict): {...}

^A-Load Schema , ^S-Save Config , ^L-Load Config , ^D-Show Description , ^Q-Quit
```

1.1.1 Features

- Simplified editing of configuration files
- Import and export YAML files
- Instant validation
- Plugin system to provide schemas from other python modules

1.1.2 Installation

Installation of Config Suite TUI can easily be done using pip:

```
$ pip install configsuite-tui
```

1.1.3 License

MIT License

Copyright (c) 2021 Equinor

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.2 Usage

1.2.1 Launching the TUI

The TUI can be launched from the CLI, the Python Shell, or inline in a function.

Launching from the CLI

```
usage: configsuite_tui [-h] [config] [-s SCHEMA]
```

Positional Arguments

config - Path to the configuration file (Optional)

A configuration file can be provided for the TUI to load on startup. A schema has to be specified if a configuration is provided, unless the configuration file has the schema name annotated at the top of the file.

Named Arguments

-s, --schema - Name of schema (Optional)

A schema can be provided to load chosen schema on startup. The schema has to be already registered with the TUI.

Launching from Python

```
from configsuite_tui.tui import tui
tui()
```

When using the TUI within Python, it can also return the configuration and validation boolean upon exit.

```
from configsuite_tui.tui import tui
config, valid = tui()
```

1.2.2 Keybindings

The TUI is operated using the keyboard and a set of keybindings, as will be informed in the TUI. The keybindings are as follows:

- Ctrl+A - Load Schema
- Ctrl+S - Save Config
- Ctrl+L - Load Config
- Ctrl+D - Show Description (of marked field)
- Ctrl+E - Edit List/Dict (only inside a List or Dict)
- Ctrl+Q - Quit

1.2.3 Schema

Config Suite TUI needs a [Config Suite](#) schema to work. A schema can be loaded either by specifying it at the CLI, or loading it from the menu using the keybind Ctrl+A.

All schemas for use with the TUI has to be registered with the schema plugin system of the TUI. This has to be done by the developer of the software in which the schema originates. How to register a schema with the TUI can be seen [here](#).

1.2.4 Configuration file

Config Suite TUI uses PyYAML [PyYAML](#) to work with configuration files. A configuration file can be loaded and saved after a schema has been initialized in the TUI.

Loading a configuration can be done from the menu using the keybind Ctrl+L.

Saving can be done from the menu using the keybind Ctrl+S. Quicksave is enabled if a configuration file has been loaded or saved earlier in the same session. This is done by double pressing Ctrl+S.

A configuration file saved by Config Suite TUI will have the used schema annotated in the header of the file. This makes it possible to load the configuration from the CLI without specifying a schema name.

An example configuration file with an annotated schema can be seen below.

```
!# configsuite-tui: schema=OwnedCars
name: Kjell
owned_cars:
- type: '911'
  brand: Porsche
```

(continues on next page)

(continued from previous page)

```
previous_owners:
- Thomas
- Bernt
```

1.3 Schema Plugin

Config Suite TUI has a plugin system powered by [pluggy](#). This means that any Python module can register its schemas for use with Config Suite TUI. All registered schemas will automatically appear in Config Suite TUI.

1.3.1 Registering a schema

A pluggy hook implementation has to be filled to register a schema with Config Suite TUI. The hook specification for the TUI is simple and is defined as follows:

```
import pluggy

hookspec = pluggy.HookspecMarker("configsuite_tui")

@hookspec
def configsuite_tui_schema():
    """Register Config Suite schema in Config Suite TUI
    Return dict: {name: schema}
    name = string
    schema = Config Suite schema object
    """
```

An example of a hook implementation can look like this:

```
# configsuite_tui_plugin.py
import configsuite_tui
from configsuite import types
from configsuite import MetaKeys as MK

@configsuite_tui.hookimpl
def configsuite_tui_schema():
    name = "Plugin_test"
    schema = {
        MK.Type: types.NamedDict,
        MK.Content: {
            "name": {MK.Type: types.String},
            "hobby": {MK.Type: types.String},
            "age": {MK.Type: types.Integer},
        },
    }

    return {name: schema}
```

The hook implementation will then have to be registered upon installation using [setuptools](#). This is done by adding an entry point for Config Suite TUI into the software's `setup.py` file. An example of this can be seen below:


```
from setuptools import setup

setup(
    name="module-name",
    install_requires="configsuite_tui",
    entry_points={"configsuite_tui": ["plugin = configsuite_tui_plugin"]},
    py_modules=["configsuite_tui_plugin"],
)
```

More in depth information about using the plugin system can be found in the [pluggy documentation](#).