

---

# **BODS Data Tool**

**Mar 05, 2020**



---

## Contents

---

<b>1</b>	<b>This tool is not currently maintained</b>	<b>1</b>
<b>2</b>	<b>Archived documentation</b>	<b>3</b>
2.1	Installing . . . . .	3
2.2	Command line tool . . . . .	3



# CHAPTER 1

---

This tool is not currently maintained

---

Command-line tooling for working with the Beneficial Ownership Data Standard (BODS) schema can be found in <https://github.com/open-contracting/ocdskit>. Tools for working with BODS data may be placed here in the future. Please contact us on [[support@openownership.org](mailto:support@openownership.org)](mailto:support@openownership.org) if you have any issues.



A suite of command-line tools for working with BODS data.

It is Python tool that can be used as a library from other Python programmes, or from the command line on your own server.

## 2.1 Installing

To install,

- 1) Check out the git repository ( <https://github.com/openownership/bodskit> ) onto your machine.
- 2) In that directory, create a new Python Virtual Environment (or similar, using the tool of your choice).
- 3) To install the tool and it's dependencies, run:

```
pip install -e .
```

If you will want to develop the tool, instead run

```
pip install -e .[test]
```

- 4) The tool should now be available! Run

```
bodskit --help
```

## 2.2 Command line tool

You can use the tool with the provided CLI script. There are various sub commands.

## 2.2.1 Command line tool - mapping-sheet option

This takes a schema and produces a spreadsheet with all field paths.

```
bodskit mapping-sheet ~/work/openownership-data-standard/schema/person-statement.json  
↳> out.csv
```

This produces output like (truncated):

```
section,path,title,description,type,range,values,links,deprecated,deprecationNotes  
,addresses,Addresses,One or more addresses for this entity.,array,0..n,,,,,  
,addresses,Address,"A free text address string, providing as much address data as is  
↳relevant, suitable for processing using address parsing algorithms. For some uses  
↳(for example, Place of Birth) only a town and country are required.",object,,,,,  
addresses,addresses/address,Address,"The address, with each line or component of the  
↳address separated by a line-break or comma. This field may also include the postal  
↳code. ",string,1..1,,,,,  
addresses,addresses/country,Country,"The ISO 2-Digit county code for this address.",  
↳string,1..1,,,,,  
addresses,addresses/postCode,Postcode,"The postal code for this address.",string,1..1,,  
↳,  
addresses,addresses/type,Type,"What type of address is this?",string,1..1,"Codelist:  
↳placeOfBirth, home, residence, registered, service, alternative",,,
```

## 2.2.2 Command line tool - schema-codelist-report option

Reports details of a JSON Schema (open and closed codelists).

```
bodskit schema-codelist-report ~/work/openownership-data-standard/schema/person-  
↳statement.json
```

This produces output like:

```
codelist,type  
addressType.csv,Closed  
annotationMotivation.csv,Open  
nameType.csv,Closed  
personType.csv,Closed  
sourceType.csv,Closed  
statementType.csv,Closed
```

## 2.2.3 Command line tool - all-codes option

This takes a directory of code list files and produces a spreadsheet with all code list options.

```
bodskit all-codes ~/work/openownership-data-standard/schema/codelists > out.csv
```

This produces output like (truncated):

```
codelist,code,title,description,technical note  
addressType,placeOfBirth,Place of birth,,  
addressType,home,Home address,,  
addressType,residence,Residential address,,  
addressType,registered,Registered address,,
```

(continues on next page)

(continued from previous page)

```
addressType,service,Service address,,  
addressType,alternative,Alternative address,,  
annotationMotivation,commenting,Commenting,"The description field provides contextual_  
↪comments for a field, object or statement.",
```