

# ReadMe - Tableau to ThoughtSpot Migration

[PS Tools](#) has been created by ThoughtSpot in order to help you move Tableau content such as a Tableau Workbooks (.twb) or Tableau Data Sources (.tds) to ThoughtSpot via importing TML (ThoughtSpot Modelling Language) files. This document has been prepared to help accelerate your migration but please note this tool is not a full migration and there are some manual steps/checks needed along the way.

This is also an ongoing project and so if you would like to help in improving the tool then please reach out to your ThoughtSpot CSM.

## Install Tableau Tools

**ps\_tools** requires at least Python 3.7, *preferably* Python 3.9 and above.

**Installation is as simple as:**

```
#Download the zip file from Github
#Create a virtual environment
python -m venv pyenv
source pyenv/bin/activate
pip install -e .
```

Change Python env

Need a clean environment without any dependencies. The tool will download all dependencies needed as part of the install.

Once the tool is installed then check this is working by typing in the below command and you should see the following screenshot:

**ps\_tools --help**

```
[(pyenv) thomas.day@Uk-Thomas ps_tools % ps_tools --help
```

```
Usage: ps_tools [OPTIONS] COMMAND [ARGS]...
```

```
Welcome to PS Tools!
```

```
Options
```

```
--help -h
```

```
Show this message and exit.
```

```
Commands
```

```
convert_files
```

```
create_spotapps
```

If you don't see the above then check the following:

- Are you in the correct folder - tableau\_to\_thoughtspot? If not then navigate there.
- Are you using the correct python virtual environment? If not then deactivate the current environment, navigate to the above folder and then

```
source pyenv/bin/activate
```

## Overview

Moving content from one BI system can be a tedious task. The following steps can help you to automate parts of the migration process. The following examples are showing the conversion of a Tableau (TDS) file to a Thoughtspot data model (table, worksheet objects). The steps and scripts can also be used independently from the source system as long as the metadata structure is similar. This article explains the required metadata structure that is needed to properly build Thoughtspot objects from scratch. Our approach starts with an identification step, where the most important dashboard and data sources are selected. After that we can extract the metadata either by downloading Tableau files manually or using the API. Once we have the data in the desired structure this content can be used to populate TML objects in order to generate SpotApp based on the tables, columns and relationships. Before importing the TML files to Thoughtspot we are programmatically validating the syntax of these objects.



## **Understand Current State of Tableau Server**

In most cases we don't want to migrate everything from a Tableau environment and so this step is an evaluation process where you can look at your Tableau environment and understand what are the most important data sources and dashboards that you'll be looking to migrate over. Use this opportunity to move the valuable content over such as the data model (tables, joins) and any business logic (formulas) that have been created rather than moving everything over.

The migration process can be done in phases. For example, phase 1 could be the top 5 data sources or dashboards that you want to move over. Or maybe you migrate content by departments, such as HR and Finance. Having smaller phases/go lives will help the overall migration process rather than a big bang approach. Once you know which content you want to move over then we need to download this from the Tableau environment.

## Export Tableau Content and Extract Metadata

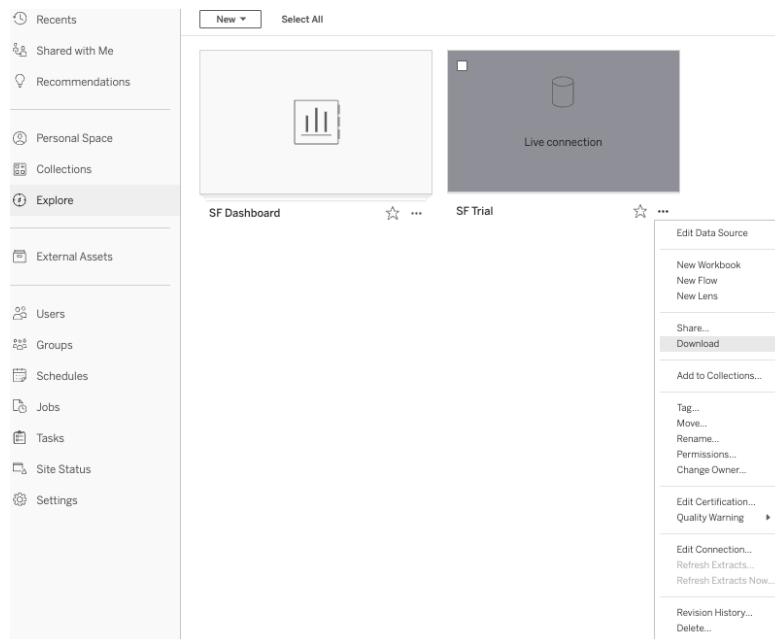
There are two types of files that you can download from a Tableau server, Tableau Data Source (.tds) or Tableau Workbook (.twb). When you download them from Tableau they will be in either .tdsx or .twbx which is the compressed file version, once you extract them then they become .tds or .twb file. The code language of these files is XML.

A .tds file contains information such as datasource connection, tables, columns and any calculations that have been used within that datasource.

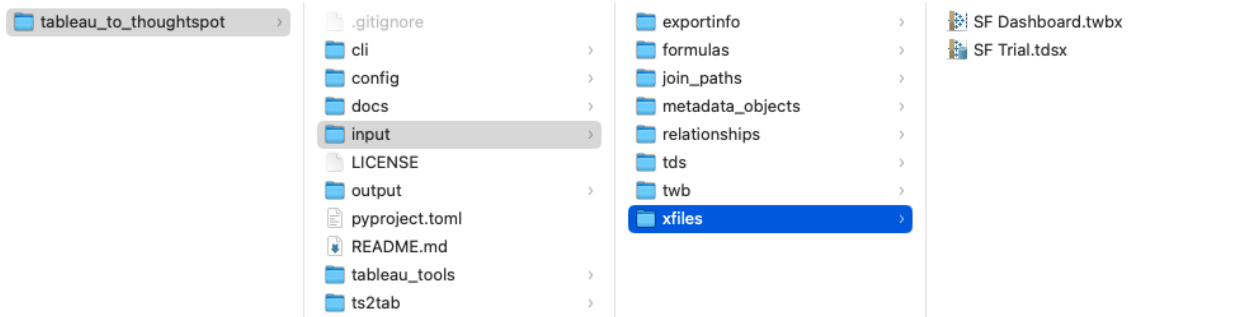
A .twb file contains information such as sheets, visualisations, dashboard properties and also the additional information from the .tds file that is used as the datasource to build the dashboard.

To Download either a .tdsx or .twbx file then follow the below steps:

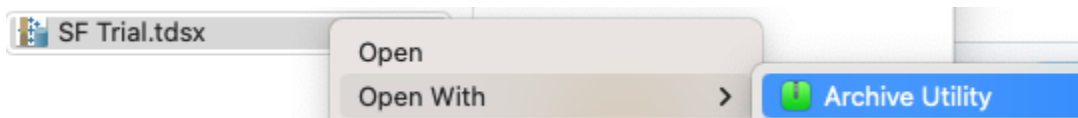
1. Navigate to the folder where your files are located.
2. Click on the “More” (3 dots)
3. Then click Download



4. You'll then have a number of .tdsx and .twbx files in your Downloads folder. Copy/Move these to the xfiles folder location of where the tool is installed.



5. Unzip these files using a convertor tool such as “Archive Utility”



6. Once unzipped then copy the .tds files to the tds folder and .twb to the twd folder. Similar to the screenshots below.



7. Now the files are ready to be converted into CSVs.

Note: The scripts will run through the folders for multiple files at once. For example, you have 20 files (twb files and 10 tds files) and the script will run through all of them and provide tml files for all 20 files.

Once the .tds or .twb files are in the respective folders (tds, twb) then the tool will look at each of the files and extract out the relevant information from the xml using Python. Below explains what each Python file is doing.

**\_tabimpclient.py** - A number of functions are defined here and each function looks at different parts of the xml and extracts the information required. Some functions include "connectiondetails", "tabledetails" and "formuladetails".

**\_tabimport.py** - Looks at all the .tds and .twb files in their folders and runs through each of them, calls on the functions from **\_tabimpclient.py** and exports the information into csv format in the ./input/... folders. We have 5 areas (exportinfo,

formulas, join\_paths, metadata\_objects, relationships). Each **csv** output will include the name of the .tds or .twb file name as part of the csv name.

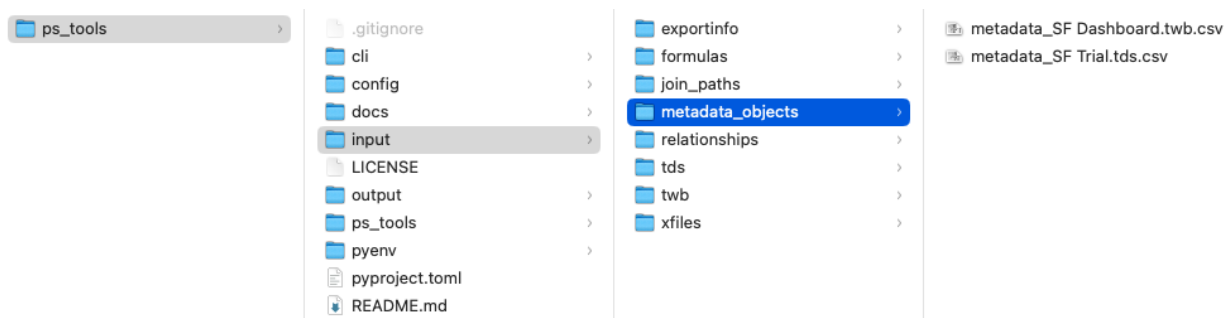
**\_client.py** - Takes all the csv outputs and converts them into ThoughtSpot TML files that we can then import into ThoughtSpot. It leverages the thoughtspot\_tml package then populates the tml templates so they can be imported. We have TML templates for Tables, Worksheets, Answers and Liveboards.

8. Run the below to extract the Tableau information out into CSV outputs

### **tableau\_tools convert\_files**

```
[(pyenv) thomas.day@Uk-Thomas ps_tools % ps_tools convert_files
./input/twb/SF Dashboard.twb
check twb: SF Dashboard.twb
./input/tds/SF Trial.tds
check tds: SF Trial.tds
check          filename filecount  fileid
0  SF Dashboard.twb           1  tdsid1
1    SF Trial.tds             2  tdsid2
```

You should now see a list of csvs in each of the input folders. The csvs will be prefixed with the name of the tds or twb file so you know where it has come from. Depending on how complex the tds/twb files are, you may get some outputs such as metadata\_objects but also outputs for formulas and relationships. Additionally, you'll see a summary csv in the summary info folder for when you run multiple files.



Example of csv output:

	A	B	C	D	E	F	G	H	I	K
1	col_name	db_column_name	aggregation	data_type	db_table	class	db	schema	warehouse	name
2	O_ORDERKEY	O_ORDERKEY	Sum	integer	ORDERS	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	ORDERS
3	O_CUSTKEY	O_CUSTKEY	Sum	integer	ORDERS	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	ORDERS
4	O_ORDERSTATUS	O_ORDERSTATUS	Count	string	ORDERS	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	ORDERS
5	O_TOTALPRICE	O_TOTALPRICE	Sum	real	ORDERS	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	ORDERS
6	O_ORDERDATE	O_ORDERDATE	Year	date	ORDERS	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	ORDERS
7	O_ORDERPRIORITY	O_ORDERPRIORITY	Count	string	ORDERS	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	ORDERS
8	O_CLERK	O_CLERK	Count	string	ORDERS	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	ORDERS
9	O_SHIPPRIORITY	O_SHIPPRIORITY	Sum	integer	ORDERS	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	ORDERS
10	O_COMMENT	O_COMMENT	Count	string	ORDERS	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	ORDERS
11	C_CUSTKEY	C_CUSTKEY	Sum	integer	CUSTOMER	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	CUSTOMER
12	C_NAME	C_NAME	Count	string	CUSTOMER	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	CUSTOMER
13	C_ADDRESS	C_ADDRESS	Count	string	CUSTOMER	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	CUSTOMER
14	C_NATIONKEY	C_NATIONKEY	Sum	integer	CUSTOMER	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	CUSTOMER
15	C_PHONE	C_PHONE	Count	string	CUSTOMER	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	CUSTOMER
16	C_ACCTBAL	C_ACCTBAL	Sum	real	CUSTOMER	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	CUSTOMER
17	C_MKTSEGMENT	C_MKTSEGMENT	Count	string	CUSTOMER	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	CUSTOMER
18	C_COMMENT	C_COMMENT	Count	string	CUSTOMER	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	CUSTOMER
19	N_NATIONKEY	N_NATIONKEY	Sum	integer	NATION	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	NATION
20	N_NAME	N_NAME	Count	string	NATION	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	NATION
21	N_REGIONKEY	N_REGIONKEY	Sum	integer	NATION	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	NATION
22	N_COMMENT	N_COMMENT	Count	string	NATION	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	NATION
23	R_REGIONKEY	R_REGIONKEY	Sum	integer	REGION	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	REGION
24	R_NAME	R_NAME	Count	string	REGION	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	REGION
25	R_COMMENT	R_COMMENT	Count	string	REGION	snowflake	SNOWFLAKE_SAMPLE_DATA	TPCH_SF1	MY_WH	REGION

9. Validate the CSVs. Please check that the output of CSVs looks sensible with special attention to the formula CSVs.

## Convert to TMLs

1. Now we have the CSVs we need to convert these to TML files that we can upload. Run the below command.

```
tableau_tools create_spotapps --type tds
```

or

```
tableau_tools create_spotapps --type twb
```

```
((pyenv) thomas.day@Uk-Thomas ps_tools % ps_tools create_spotapps --type tds
/Users/thomas.day/Documents/pythonenvs/ps_tools
Creating SpotApp for SF Trial.tds...
TABLE: ORDERS
Number of columns: 9
TABLE: CUSTOMER
Number of columns: 8
TABLE: NATION
Number of columns: 4
TABLE: REGION
Number of columns: 3
TABLE: LINEITEM
Number of columns: 16
TABLE: PART
Number of columns: 9
TABLE: PARTSUPP
Number of columns: 5
TABLE: SUPPLIER
Number of columns: 7
```

👍 successfully created SpotApp for SF Trial.tds

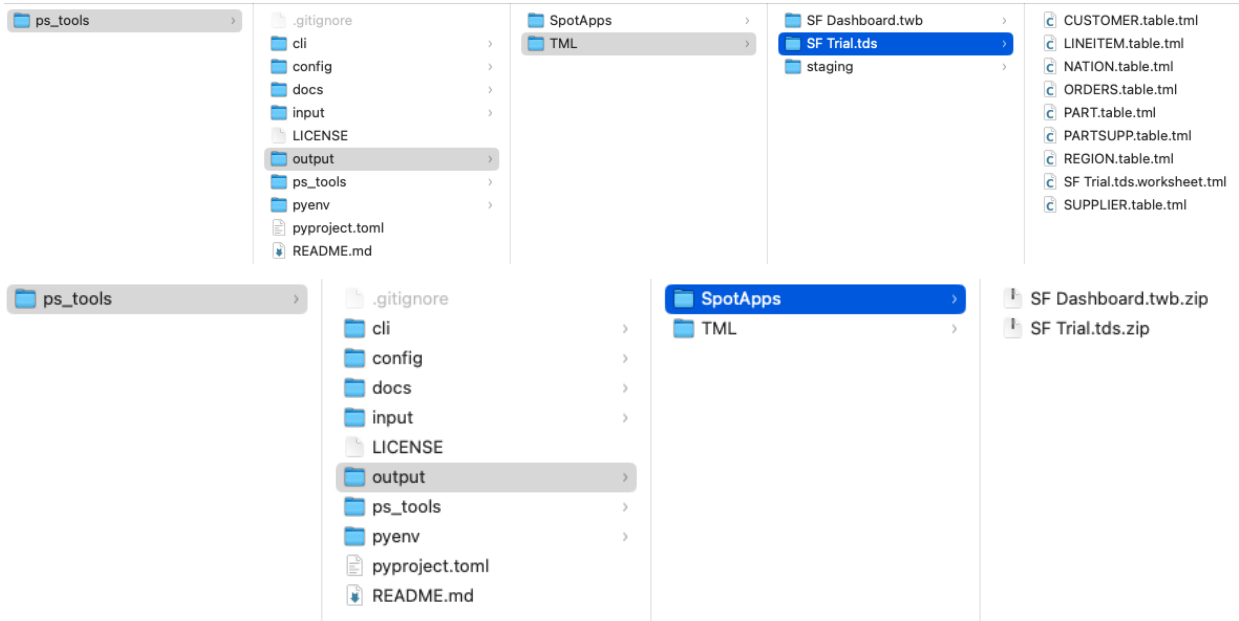
```
((pyenv) thomas.day@Uk-Thomas ps_tools % ps_tools create_spotapps --type twb
/Users/thomas.day/Documents/pythonenvs/ps_tools
Creating SpotApp for SF Dashboard.twb...
TABLE: CUSTOMER
Number of columns: 8
TABLE: REGION
Number of columns: 4
TABLE: ORDERS
Number of columns: 10
TABLE: LINEITEM
Number of columns: 16
TABLE: NATION
Number of columns: 4
TABLE: PARTSUPP
Number of columns: 5
TABLE: PART
Number of columns: 9
TABLE: SUPPLIER
Number of columns: 7
```

👍 successfully created SpotApp for SF Dashboard.twb

```
((pyenv) thomas.day@Uk-Thomas ps_tools %
```

2. Running the above command will then run through the csv files from the “input” directory and then create various .tml files and .zip files (Spotapps) in the “output” directory. A folder is created with the naming of the tds/twb file and the same is true for the Spotapps.





Below is a high level mapping of how a Tableau file (.tds/.twb) maps to a **ThoughtSpot TML** file:

Tableau (xml tag)	ThoughtSpot (tml)
relation connection	Connection
metadata-records	Table
metadata-records / remote-name	Column
relationships	Joins
calculation	Formulas
worksheet/view	Answer
dashboard	Liveboard

# Validation

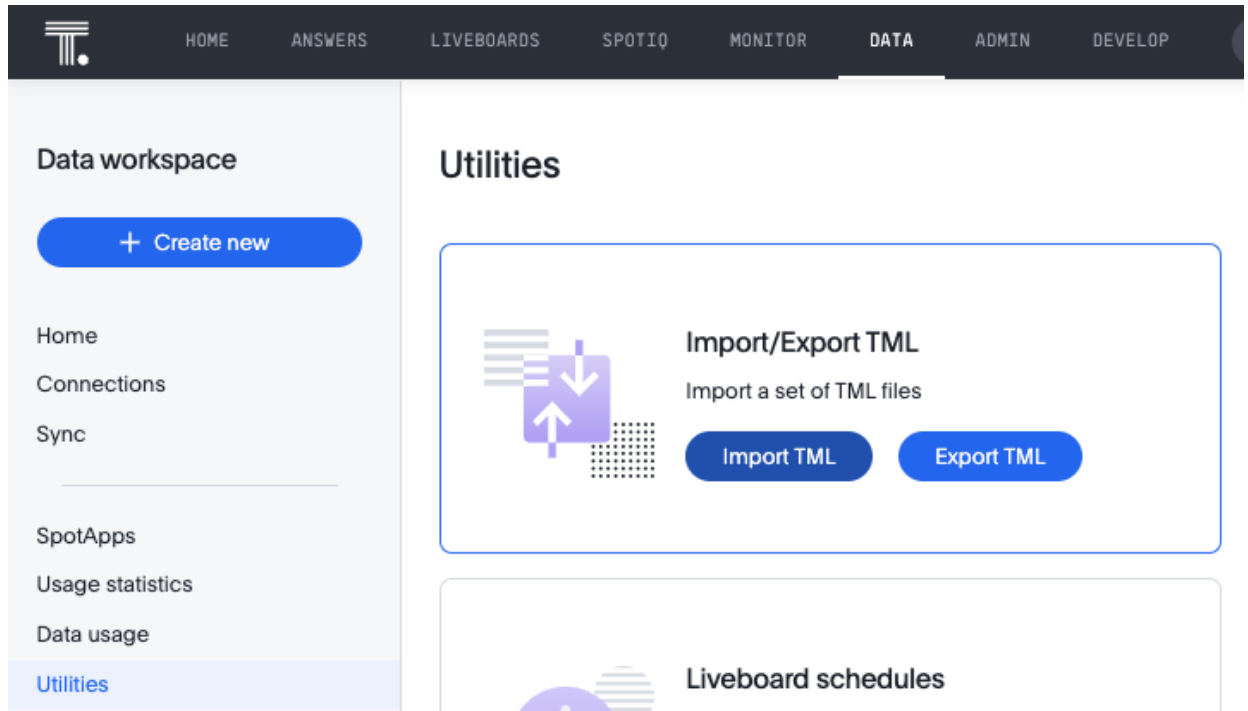
Check the TML files from the output and ensure they have populated correctly.

```
output > TML > SF Trial.tds > ≡ REGION.table.tml
1  guid: 55e01123-41cd-2fd5-998f-5cdedfa9f217
2  table:
3    name: REGION
4    db: SNOWFLAKE_SAMPLE_DATA
5    schema: TPCH_SF1
6    db_table: REGION
7    connection:
8      name: SF Trial
9    columns:
10     - name: R_REGIONKEY
11       db_column_name: R_REGIONKEY
12       properties:
13         column_type: MEASURE
14         aggregation: SUM
15         index_type: DONT_INDEX
16       db_column_properties:
17         data_type: INT64
18     - name: R_NAME
19       db_column_name: R_NAME
20       properties:
21         column_type: ATTRIBUTE
22         aggregation: COUNT
23         index_type: DONT_INDEX
24       db_column_properties:
25         data_type: VARCHAR
26     - name: R_COMMENT
27       db_column_name: R_COMMENT
28       properties:
29         column_type: ATTRIBUTE
30         aggregation: COUNT
31         index_type: DONT_INDEX
32       db_column_properties:
33         data_type: VARCHAR
34
```

```
output > TML > SF Trial.tds > ≡ SF Trial.tds.worksheet.tml
1  guid: 7ad60c3b-cfea-4dc4-bcad-447ff5eded15
2  worksheet:
3    name: SNOWFLAKE_SAMPLE_DATA
4    tables:
5      - name: CUSTOMER
6      - name: NATION
7      - name: REGION
8      - name: ORDERS
9      - name: LINEITEM
10     - name: PART
11     - name: PARTSUPP
12     - name: SUPPLIER
13    joins:
14      - name: CUSTOMER_NATION
15        source: CUSTOMER
16        destination: NATION
17        type: INNER
18        is_one_to_one: false
19      - name: CUSTOMER_ORDERS
20        source: CUSTOMER
```

# Import TMLs into ThoughtSpot

1. Log into ThoughtSpot and navigate to “Data” -> “Utilities” -> “Import TML”
2. Click on Import TML and select the file/files/spotapps that you would like to import.



3. Allow some time for the ThoughtSpot validation checks to run and then Import. If there are any errors then you may need to go into the tml files and validate why they have failed such as unable to find the connection or invalid join paths.
4. Check the various tables, worksheets and formulas have been imported correctly.
5. Happy searching and creating of Liveboards..

## **Appendix: What Can The Tool Do Now**

### **NOW:**

- ✓ Tables
- ✓ Columns
- ✓ Simple Joins
- ✓ Simple Formulas
- ✓ Worksheets
- ✓ This works in an automated way by running through multiple Tableau files that sit in a folder
- ✓ The output is then SpotApps and TMLs which we can import into a ThoughtSpot cluster
- ✓ We've tested this on a number of Tableau files that we've created and some files that we've downloaded from Tableau Public.

This code is ready now to use for a (pilot) customer.

## **What We Are Working On / Future Consideration**

### **WIP:**

- Connections
- Complex Joins
- Complex Formulas
- Visualisations -> Answers

### **For future Consideration:**

- Liveboards - Automate the process of pulling "views" and converting them into TS Liveboards
- Automate the process of pulling multiple Tableau files
- Automate the process of extracting Tableau Hyper files (data files) and placing the data into CDW tables such as Snowflake
- Automate the process of uploading tml files to a TS cluster
- Utilise the Tableau API to pull relevant metadata from so that we can use both Tableau API and tds/twb files approach