**Documentation for conversion of CSV/TSV to Apache Parquet**

As we have decided to use apache’s parquet instead of CSV/TSV which is slow and uses more storage compared to apache parquet which is an open source, column-oriented data file format designed for efficient data storage and retrieval. I have made a binary tool written in rust-lang to convert CSV/TSV files to parquet efficiently. This tool will allow us to convert our existing or old CSV/TSV data files to apache’s parquet with various other configurable parameters in a config file.

**Working**

It uses the native implementation of apache’s arrow and parquet in rust to read, decode and convert it. Using toml (Tom’s obvious minimal language) as a config file. It reads the users defined configuration from the *config.toml* file and parse it as parameters to the converter.

**Config.toml structure**

There can be multiple config files for multiple sensors or as per requirements, but the program will try to read from .configs/config.toml.

Sample config file:

[config]

# The name of the config file.

name = "bsp\_zone1\_snsr\_1-9"

# The description of the config file.

description = "This config is used for xyz purpose at BSP."

[arguments]

# Path to the CSV file

path\_to\_csv = '/sample\_data.csv'

# Path where the parquet file should be saved

path\_to\_pqt = '/sample.parquet'

[options]

# created\_by tag for the file

created\_by = "BSP\_BRM"

# The number of records to infer the schema from.

max\_read\_records = 100

# If the CSV file contains header

header = false

# Sets flag to enable/disable dictionary encoding for any column

dictionary = false

The square brackets [] are the data which is used by me to differentiate and categorised it.

* [config]: contains the basic information of the config file like name, description etc.
* [arguments]: contains the no null required values/arguments for the program like the path to CSV and where to save parquet etc.
* [options]: contains other optional parameters that can be defined by a user as per his/her needs.

**Fields**

* Name: The name of the config file. As there can be multiple config files, naming it will help us to differentiate.
  + String: A string value.
* Description: Along with the name there can be a small description for the file where the purpose and other things can be described.
  + String: A string value.
* Path\_to\_csv: The path where the CSV/TSV file is located, should be inside ‘ ‘.
  + String: A string value.
* Path\_to\_pqt: The path where you want to save the parquet file, should be inside ‘ ‘. Also, the name of the file along with the extension (.parquet) should be in the path.
  + String: A string value
* created\_by: Add created\_by tag for the parquet file.
  + String: A string value.
* max\_read\_records: The maximum number of records to be read from the CSV/TSV file.
  + usize: should be an Unsigned Integer of value, usize types depend on the architecture of the computer your program is running on, which is denoted in the table as “arch”: 64 bits if you’re on a 64-bit architecture and 32 bits if you’re on a 32-bit architecture.
* header: Whether the CSV file contains the header or not.
  + Bool: Either True or False.
* dictionary: Sets flag to enable/disable dictionary encoding for any column.
  + Bool: Either True or False.

Note: This is an initial version of the document. Please refer to the online version in git as I’ll keep updating the documentation(s) as per the sources and other requirements.

Link: <https://github.com/radcolor/iit_bsp_intern_files>

**References**

* <https://toml.io/en/v1.0.0> (Refer to this for more about toml syntax and structure)
* <https://docs.rs/arrow/latest/arrow/> (Rust Documentation for Apache’s Arrow)
* <https://docs.rs/crate/parquet/latest> (Rust Documentation for Apache’s Parquet)