Simple Table Converter

Franklin E. Powers, Jr.

February 25, 2021

1 Introduction

Simple Table Converter was created to allow for simple manipulation of tables in different formats. Additionally, it can remove different rows and columns.

At the moment, there are only two ways to use it:

1. As a library, which you can integrate into your own programs
2. As a command-line utility

This document will cover how to use it as a command-line utility.

Starting the tool depends on platform and version as shown in the table below:

C++	stc
C#	stc
Java	java -jar stc.jar
JScript Windows Shell	cscript stc-wsh.js

2 Converting Table Formats

STC always converts from one source table to one target table. Either can be specified on the command-line. When a source table isn't specified, stdin (standard input/comsole input) is used. When a target table isn't specified, stdout (standard output/console output) is used. By default, the table format is specified by the filename extension. If no format can be determined, then the default is CSV.

2.1 Supported File Formats

CSV	Comma Separated Values file
PSV	Pipe Separated Values file
TSV	Tab Separated Values file

3 Filters

The filtering commands will be listed in order of precedence.

INCLUDEROWS	This specifies a filter by which rows are included. The filter runs on the name (or first column) of the row.
EXCLUDEROWS	This specifies a filter by which rows are excluded. The filter runs on the name (or first column) of the row .
INCLUDECOLUMNS	This specifies a filter by which columns are included. The filter runs on the name (or first row) of the column.
EXCLUDECOLUMNS	This specifies a filter by which columns are excluded. The filter runs on the name (or first row) of the column.

Filtering comes in the following forms:

Counting List	Starts with a hash/pound (#) sign and has a comma (,) separated list.	Indicates that the filter is based on the index/count of the row/column/etc and should compare against the given list.
Counting Range	Starts with a hash/pound (#) sign and has a minus sign (-) between a lower and upper limit.	Indicates that the filter is based on the index/count of the row/column/etc and should compare against the given lower and upper limit (inclusive).
Counting Single	Starts with a hash/pound (#) sign and is neither a list or a range	Indicates that the filter is based on the index/count of the row/column/etc and should compare against the specified value.
Regular Expression	Starts and ends with a forward slash (/)	Indicates the filter is to run the given regular expression.
String Equals	Starts with an equals sign (=)	Indicates the filter should compare to the specified string and if it matches then the filter applies.
String With/Contains	Starts with no specific character	Indicates the filter should compare to the specified string and if it is found anywhere, then the filter applies.

Sorting can be done with:

ORDERBY	Orders the rows by the columns specified in the comma separated list.

Automated testing can be executed with the following parameters:

RUNTESTS	Executes the unit tests and determines if the resulting executable works as expected. Only failed tests are outputted by default.
VERBOSE	Outputs additional details about the test results.
RECORDPASS	Outputs the results for tests that pass.

4 Examples

Convert the CSV file in.csv to a TSV file out.tsv.

```
stc in.csv out.tsv
```

Remove the 3rd column from the CSV file in.csv and store in out.csv.

```
stc in.csv out.csv excludecolumns #3
```

Keep only the columns that have A in the header and store in out.csv.

```
stc in.csv out.csv includecolumns A
```

Remove the 3rd, 4th, and 5th columns from the CSV file in.csv and store in out.csv.

```
stc in.csv out.csv excludecolumns #3,4,5
stc in.csv out.csv excludecolumns #3-5
```