XSV Command Line

Installing

1. macOS Homebrew user

$ brew install xsv

2. macOS MacPorts user, then you can install xsv from the official ports:

$ sudo port install xsv

3. Windows

Steps

- Install - https://www.rust-lang.org/tools/install
- Cargo: cargo install xsv

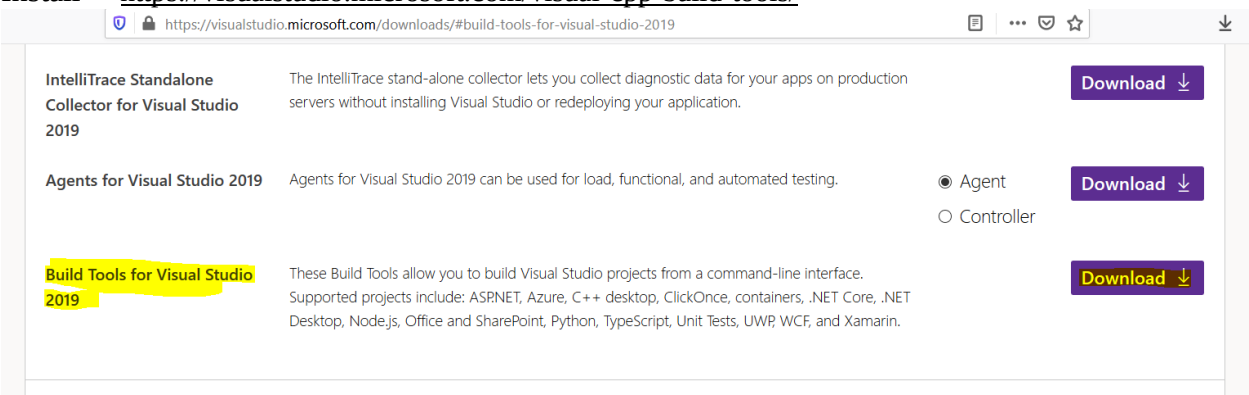
If above step did not work use

- git clone git://github.com/BurntSushi/xsv
- cd xsv
- cargo build --release

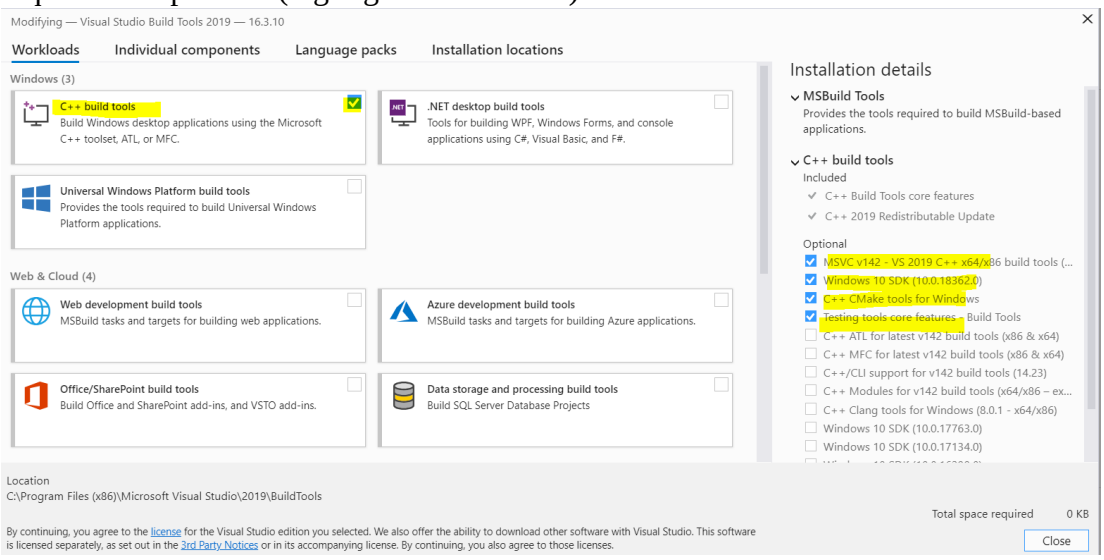
While compiling Rust program in a windows environment, you may encounter the error : linker `link.exe` not found. This is because of the absence of the C++ build tools in your machine. For compiling Rust programs successfully, one of the prerequisites is the installation of the Build Tools for Visual Studio 2019.

```
   Compiling untitled v0.1.0 (C:\Users\jithi\IdeaProjects\untitled)
error: linker `link.exe` not found
  |
  = note: The system cannot find the file specified. (os error 2)

note: the msvc targets depend on the msvc linker but `link.exe` was not found

note: please ensure that VS 2013, VS 2015, VS 2017 or VS 2019 was installed with the Visual C++ option

error: aborting due to previous error

error: could not compile `untitled`.

To learn more, run the command again with --verbose.

Process finished with exit code 101
```

- Install – https://visualstudio.microsoft.com/visual-cpp-build-tools/



- After the download, while installing the Build tools, make sure that you install the required components (highlighted in Yellow)



- This will download around 1.2GB of required files. Once everything is successfully installed, reboot and re-run your rust program and it will compile successfully.

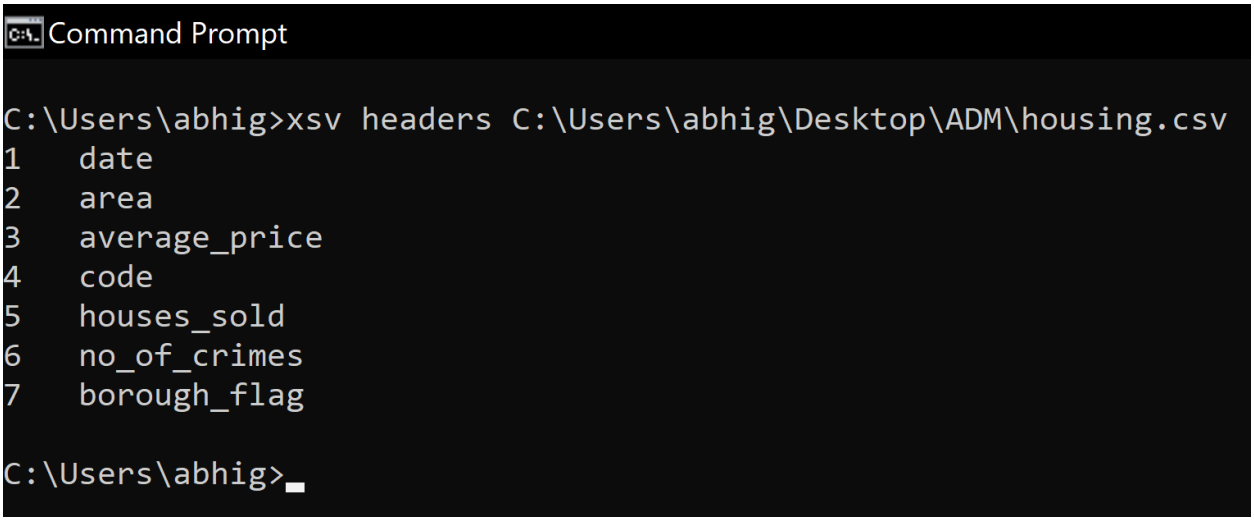
xsv is a command line program for indexing, slicing, analyzing, splitting and joining CSV files. Commands should be simple, fast and composable.

Available commands

- **cat** - Concatenate CSV files by row or by column.
- **count** - Count the rows in a CSV file. (Instantaneous with an index.)
- **fixlengths** - Force a CSV file to have same-length records by either padding or truncating them.
- **flatten** - A flattened view of CSV records. Useful for viewing one record at a time. e.g., xsv slice -i 5 data.csv | xsv flatten.
- **fmt** - Reformat CSV data with different delimiters, record terminators or quoting rules. (Supports ASCII delimited data.)

- **frequency** - Build frequency tables of each column in CSV data. (Uses parallelism to go faster if an index is present.)
- **headers** - Show the headers of CSV data. Or show the intersection of all headers between many CSV files.
- **index** - Create an index for a CSV file. This is very quick and provides constant time indexing into the CSV file.
- **input** - Read CSV data with exotic quoting/escaping rules.
- **join** - Inner, outer and cross joins. Uses a simple hash index to make it fast.
- **partition** - Partition CSV data based on a column value.
- **sample** - Randomly draw rows from CSV data using reservoir sampling (i.e., use memory proportional to the size of the sample).
- **reverse** - Reverse order of rows in CSV data.
- **search** - Run a regex over CSV data. Applies the regex to each field individually and shows only matching rows.
- **select** - Select or re-order columns from CSV data.
- **slice** - Slice rows from any part of a CSV file. When an index is present, this only has to parse the rows in the slice (instead of all rows leading up to the start of the slice).
- **sort** - Sort CSV data.
- **split** - Split one CSV file into many CSV files of N chunks.
- **stats** - Show basic types and statistics of each column in the CSV file. (i.e., mean, standard deviation, median, range, etc.)
- **Note**: To know commands syntax: type “xsv <command name> --help”

1) Header command is used to show the headers.



2) stats - Show basic types and statistics of each column.

```
C:\Users\abhig>xsv stats C:\Users\abhig\Desktop\ADM\housing.csv
field,type,sum,min,max,min_length,max_length,mean,stddev
date,Unicode,,1995-01-01,2020-01-01,10,10,,
area,Unicode,,barking and dagenham,yorks and the humber,5,22,,
average_price,Integer,3570428203,40722,1463378,5,7,263519.68433094764,187610.58111585828
code,Unicode,,E09000001,E92000001,9,9,,
houses_sold,Integer,52393691,2,132163,0,6,3893.9941285767372,12113.952285168458
no_of_crimes,Float,16055981,0,7461,0,6,2158.352063449405,902.0271075016559
borough_flag,Integer,9936,0,1,1,1,0.7333382537456639,0.44221404244881657
```

3) Count – Count the rows in a CSV file

```
Command Prompt

C:\Users\abhig>xsv count C:\Users\abhig\Desktop\ADM\Housing.csv
13549
```

4) Joins:

join options:

--no-case When set, joins are done case insensitively.

--left Do a 'left outer' join. This returns all rows in first CSV data set, including rows with no corresponding row in the second data set. When no corresponding row exists, it is padded out with empty fields.

--right Do a 'right outer' join. This returns all rows in second CSV data set, including rows with no corresponding row in the first data set. When no corresponding row exists, it is padded out with empty fields. (This is the reverse of 'outer left'.)

--full Do a 'full outer' join. This returns all rows in both data sets with matching records joined. If there is no match, the missing side will be padded out with empty fields. (This is the combination of

'outer left' and 'outer right'.)

--cross USE WITH CAUTION.
This returns the cartesian product of the CSV data sets given. The number of rows return is equal to N * M, where N and M correspond to the number of rows in the given data sets, respectively.

--nulls When set, joins will work on empty fields. Otherwise, empty fields are completely ignored. (In fact, any row that has an empty field in the key specified is ignored.)

To save the file after joining the dataset:

```
C:\Users\abhig>xsv join --no-case Order_ID C:\Users\abhig\Desktop\ADM\XSV\Order.csv Order_ID C:\Users\abhig\Desktop\ADM\XSV\SuperStore.csv > new.csv

C:\Users\abhig>xsv join --no-case Order_ID C:\Users\abhig\Desktop\ADM\XSV\Order.csv Order_ID C:\Users\abhig\Desktop\ADM\XSV\SuperStore.csv > C:\Users\abhig\Desktop\
ADM\XSV\new.csv

C:\Users\abhig>
```

References:

https://github.com/BurntSushi/xsv