# csv-value-counter

Sometimes it is hard looking for values when there are large amounts of data e.g. hundreds of different mutex names, named pipes, file names etc. csvvaluecounter is a very simple tool that takes a CSV, gets the field value (using the index of the field) from the CSV as specified from the file, keeps count of each unique value, sorts by number, and then value, and outputs a file, not rocket science and could be easily solved using the bash command line.

This is a **golang** rewrite of my .Net csvvaluecounter tool, cross compiled for Linux and Windows.

# gb

The project uses gb for building the project. **gb** allows for reproducible builds and vendoring so that all dependencies are kept with the project source.

To install **gb**, create a temporary directory and set the GOPATH environment variable to the new temporary directory.

```
$ export GOPATH=/home/bsmith/tempgb
```

Then download the source code for **gb**

```
go get github.com/constabulary/gb/...
```

Navigate to the **gb** sub-directory:

```
cd  /home/bsmith/tempgb/src/github.com/constabulary/gb
```

Build the project

```
go build
```

Copy the binaries to the local path

```
cp ../../../bin/* /usr/local/bin
```

The **gb** command maybe aliased with git, so check with:

```
alias gb
```

If the alias exists then you can unaliase by:

```
unalias gb
```

# Compile with gb

To compile the application use the following commands (assuming the same directory structure):

```
$ cd csv-value-counter
$ gb build all
```

# Compile for Windows under Linux

**gb** doesn't support cross compiling so you need to use the standard method for compiling e.g. set GOPATH etc:

```
export GOPATH=/path/to/csv-value-counter/vendor
cd /path/to/csv-value-counter/src/github.com/woanware/csv-val
GOOS=windows GOARCH=amd64 go build -o csv-value-counter.exe m
```