FUNCTIONAL and TECHNICAL REQUIREMENTS DOCUMENT of DATAGENERATOR

DEVELOPMENT & IMPLEMENTATION :
Uni Systems
Professional Services Technical Department
Continuous Innovation & Emerging Technologies Unit

DRAFT 22.9.2021

VERSION 3.0.4

Application Developed by:
George Bouras Senior Professional Services Engineer, Continuous Innovation & Emerging Technologies Unit

Documentation Created by:
Konstantinos Petrou Junior Professional Services Engineer, Continuous Innovation & Emerging Technologies Unit

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 Document Scope and Purpose

This document provides a description of the technical design for the Datagenerator Software System.This document's primary propose is to describe the technical vision for how data can be generated from libraries and matching as possible original data from any database system .This document also contains an example for all the above.

Please note that this is a baseline document and may be updated as development progresses.

1.2 Target Audience

This document is targeted (but not limited) to technical stakeholders :

- Development team

- Support Staff

1.3 Acronyms and/or Definitions

libraries	Plain text files containing data for input
gendata.pl	Perl file that contains the source code for the whole proccesses
options.conf	Any Configuration file with attributes for columns data munipulation
example.conf	Any Configuration file containing column names, data types and properties
hostname.txt	Plain text containing host names (library)
name_male.txt	Plain text containing male names (library)
name_female.txt	Plain text containing female names (library)
output	System's output stream (console, CSV file, XML file, JSON file)

1.4 System Environment

This application is running from a Linux terminal. Also needs the installation of Perl Programming Language version.

1.4.1 System Requirements

The Operating System that program has been tested successfully is Linux Ubuntu distribution 18.04 LTS. Also can run successfully from any Virtual Machine with Linux installed inside, at Windows Operating System or any other Operating System. The processor's architecture need to be x86-64 bit.

The encoded format for libraries text files, XML and JSON files should be UTF- 8 formatted.

Perl's version have to be 5.26 and later versions.

1.5 Summary of Functions

The Datagenerator application requires a technology based solution on data collection whose primary functions are :

- Data Accessibility

- Data Integrity

1.5.1 Functional Requirements

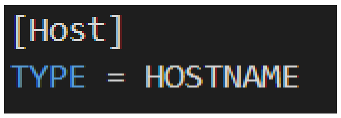
In order to accomplish the above articulated need, the datagenerator requires data collection and some basic knowledge for real database schema and values of the real data.

- ***Data Accessibility***
  - Data profile information is publicly available via a searchable web sites or web free text libraries.
- ***Data Integrity***
  - Database schema and/or some original database data from real data storage system.

1.5.2 Columns names, types and properties

The configuration file example.conf contains column names and columns properties

- Column names is the string inside the brackets [ ]:



```
[Host]
TYPE = HOSTNAME
```

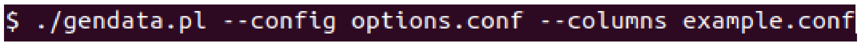
- Columns types and their properties are using in order to specialize what kind of data are going to represented inside the final columns after the process : typically under the column nane. Some of them are TYPE , FORMAT, LENGTH MINIMUM, LENGTH MAXIMUM, AUTO INCREMENT START NUMBER, AUTO INCREMENT RESET NUMBER, AUTO INCREMENT PADDED WITH 0.

The configuration file options.conf contains the representation of the final output .These are properties of the output such as : Columns separator , Number of lines to generate, Column names at fist line.

1.5.3 Running the application

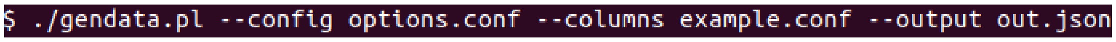
From the command line we pass the arguments that program wants to run in order to generate the success data.These commands involve the options.conf and example.conf files.The default output of program is the terminal.If the user wants other output stream for the last generated data then should add the options as shown the example below:

- Output stream : Console



```
$ ./gendata.pl --config options.conf --columns example.conf
```

- Output stream : JSON file



```
$ ./gendata.pl --config options.conf --columns example.conf --output out.json
```

1.5.4 Repositories

The repositories of the files are :

1. Configuration files and source code of the datagenerator are inside the same directory (Any).
2. Plain text files for library issues are inside a subdirectory where Configuration and Source Code files are.

2.0 ADDITIONAL SYSTEM REQUIREMENTS

2.1 Columns descriptions

Columns are generated from a configuration file.Inside that file there are columns names and the functionality of their data. Lines starting from # are comments.We can have multiple columns with the same name if is necessary. All columns properties are optional with reasonable defaults.The available column types have their correspondings such as :

- FIRST NAME MALE : English common male first name
- FIRST NAME FEMALE : English common female first name
- FIRST NAME : English common first name
- HOSTNAME : Computer name that looks real
- PERL CODE : Run arbitary perl code on any existing <column>. "PERL CODE" columns can be defined even before the columns they are using. FORMAT contains your code e.g.
  'Lest add ' .(<col1> + <col2>)
- TELEPHONE INTERNATIONAL : Phone number with international and country suffixes
- TELEPHONE LOCAL : phone number with country suffixes
- MASK : Text generated from the FORMAT
- CONSTANT : Fixed string
- AUTO INCREMENT : Auto increment integer
- DATE : Format date using the: <DD> <MM> <YYYY> <hh> <mm> <ss> <MONTHNAME> <DAYNAME> <D> <M> <YY> <h> <m> <s>
- DATE RANDOM : A random date between "DATE MINIMUM" and "DATE MAXIMUM" using the same format as DATE
- UUID : Universally unique identifier (39ef343e-78eb-418b-a618-36d896e974f6)
- NUMBER : Random integer from LENGTH MINIMUM to LENGTH MAXIMUM
- TEXT : Random text from LENGTH MINIMUM to LENGTH MAXIMUM with characters from PERMITTED CHARACTERS
- RANDOM : Random string from LENGTH MINIMUM to LENGTH MAXIMUM from PERMITTED CHARACTERS
- EMAIL : Random email address that looks believable
- SURNAME : English common surname
- IP ADDRESS : IP Address v4
- COUNTRY : Greece, New Zeland, etc
- STREET NAME : Street name

- FROM FILES : With this we can have for each column multiple files with separate lookup possibilities.

2.2 Columns properties

Some of the properties of columns are defined with the word FORMAT and after the keyword TYPE has been defined.The attribute FORMAT :

- When it used at CONSTAND is a fixed string e.g. hello world.
- When it used at MASK is a template e.g. foo_[a-t].[1-3].[A-Z]-[5-9]
- When when it used at PERL CODE is a template e.g foo <col1>@<col2>_<col3>
- When it used at DATE, DATE RANDOM contains any of the special variables :

<DAYNAME>	Name of day	Sun, Mon, ...
<MONTHNAME>	Name of month	Jan, Feb, ...
<DD>	Day	01 - 31
<MM>	Month	01 - 12
<YYYY>	Year	four digits
<YY>	Year	two digits
<hh>	Hour	00 - 23
<mm>	Minute	00 - 59
<ss>	Second	00 - 59
<D>	Day	1 - 31
<M>	Month	1 - 12
<h>	Hour	0 - 23
<m>	Minute	0 - 59
<s>	Second	0 – 59

Some other properties are :

- HIDDEN : If No, the field is not displayed. Usefull when you want to use fields as members of PERL CODE but not shown at the output
- DATE MINIMUM : Specify the minimum DATE like 07 Feb 1982, 00:30:00
- DATE MAXIMUM : Specify the maximum DATE like 17 Apr 1991, 23:30:00
- LENGTH MINIMUM : Minimum length of NUMBER, TEXT, RANDOM
- LENGTH MAXIMUM : Maximum length of NUMBER, TEXT, RANDOM
- PERMITTED CHARACTERS : Choose only from these characters
- AUTO INCREMENT START NUMBER : a)integer, start counting the "AUTO INCREMENT" from this number, b) EPOCH, start counting the "AUTO INCREMENT" from seconds from 1971
- AUTO INCREMENT RESET NUMBER : Reset "AUTO INCREMENT" to "AUTO INCREMENT START NUMBER" if it is equal or greater the specified number
- AUTO INCREMENT PREFIX: a string before the "AUTO INCREMENT" e.g. id_
- AUTO INCREMENT PADDED WITH 0 Adds as many 0 as PREFIX until the length of AUTO INCREMENT RESET

NUMBER. If you set something as AUTO INCREMENT PREFIX then it is disabled

2.3 Remarks

When using type of column/s as FROM FILES then, for all the FILE statements must sum to 100.For example :

```
FILE = FILEPATH: $ /dir1/dir2/important.txt  , POSSIBILITY: 80%
FILE = FILEPATH: $ /dir1/dir2/less often.txt , POSSIBILITY: 19%
FILE = FILEPATH: $ /dir1/dir2/obsolete.txt   , POSSIBILITY: 1%
```

Also , all the files that act as libraries must have their data as simple lines, for example :

```
line1
line2
line3
……
……
```

2.4 Examples of Columns definitions

Some examples of how columns should be represented inside the configuration file :

```
[compo1]
TYPE      = PERL CODE
FORMAT = "<first name>@<Host>"

[compo2]
TYPE      = PERL COD
FORMAT = "<Custom1> <Device id>"

[Custom1]
TYPE = FROM FILES
FILE = FILEPATH: $ /dir1/dir2/important.txt  , POSSIBILITY: 79%
FILE = FILEPATH: $ /dir1/dir2/less often.txt , POSSIBILITY: 20%
FILE = FILEPATH: $ /dir1/dir2/obsolete.txt   , POSSIBILITY: 1%

[RandomNumber]
TYPE                                = NUMBER
LENGTH MINIMUM              = 2
LENGTH MAXIMUM              = 3
PERMITTED CHARACTERS = 123

[RandomText]
TYPE                              = TEXT
LENGTH MINIMUM            = 2
LENGTH MAXIMUM            = 3
PERMITTED CHARACTERS = ADG

[Random]
TYPE                                 = RANDOM
LENGTH MINIMUM               = 2
LENGTH MAXIMUM               = 3
PERMITTED CHARACTERS  = ADG123
```

```
[Auto incr1]
TYPE = AUTO INCREMENT

[Auto incr2]
TYPE    = AUTO INCREMENT
AUTO INCREMENT PREFIX=

[Auto incr3]
TYPE = AUTO INCREMENT
AUTO INCREMENT START NUMBER = 0
AUTO INCREMENT RESET NUMBER = 100
AUTO INCREMENT PREFIX              = cell-

[Auto incr4]
TYPE                       = AUTO INCREMENT
AUTO INCREMENT START NUMBER = EPOCH
AUTO INCREMENT RESET NUMBER = 10000000
AUTO INCREMENT PREFIX      =

[Auto incr5]
TYPE                        = AUTO INCREMENT
AUTO INCREMENT START NUMBER  = EPOCH
AUTO INCREMENT RESET NUMBER  = 999999999999999
AUTO INCREMENT PREFIX       =
AUTO INCREMENT PADDED WITH 0 = YES

[SomeMask]
TYPE   = MASK
FORMAT = 0[0-3]-[A-Z]_[a-d]

[Pi]
TYPE   = CONSTANT

[Foo]
TYPE   = CONSTANT
FORMAT = I like Perl

[Date 1]
TYPE=DATE

[Date 2]
TYPE                 = DATE
DATE MINIMUM  = 01 Jan 1990, 00:00:00
DATE MAXIMUM = 08 Jan 2018, 23:59:59
FORMAT              = <DD> (<DAYNAME>) <MM> (<MONTHNAME>) <YYYY> - <hh>:<mm>:<ss>

[Date 3]
TYPE                 = DATE RANDOM
DATE MINIMUM  = 07 Feb 1982, 17:05:00
DATE MAXIMUM = 31 Dec 2011, 23:59:59
FORMAT              = <YYYY><MM><DD><hh><mm><ss>

[Host]
TYPE = HOSTNAME
```

```
[Device id]
TYPE = UUID

[e-mail address]
TYPE = EMAIL

[first name male]
TYPE = FIRST NAME MALE

[first name female]
TYPE = FIRST NAME FEMALE

[first name]
TYPE = FIRST NAME

[surname]
TYPE = SURNAME

[IP ADDRESS]
TYPE = IP ADDRESS

[World country]
TYPE = COUNTRY

[Street name]
TYPE = STREET

[phone a]
TYPE = TELEPHONE INTERNATIONAL

[phone b]
TYPE = TELEPHONE LOCAL
```

2.5 Command line usages

From the terminal we can run some basic but not limited commands for additional informations.Some of them are :

- ./gendata --version : shows the version of the program
- ./ gendata --help : help instructions
- ./gendata --config /tmp/opt.conf --columns example_perlcode.conf
- ./gendata --config options.conf   --columns example_perlcode.conf --format xml
- ./gendata --columns /tmp/col.conf --format json --output /tmp/out.json
- ./gendata -f xml -o output.xml
- ./gendata -f json -o output.json

Command line switches. All of them are optional:

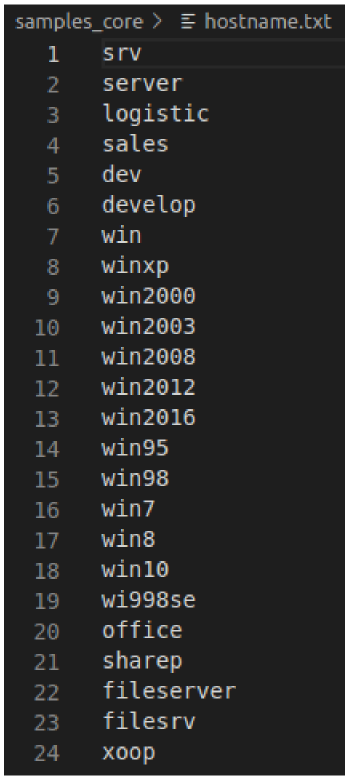
- --version Shows version

- --help   Shows help screen
- --config Default Options file. Default location is $script_dir/gendata.conf
- --columns Column definition file. Default location is $script_dir/gendata.columns
- --format csv, xml, json. Default is csv
- --output Output File. Default is screen

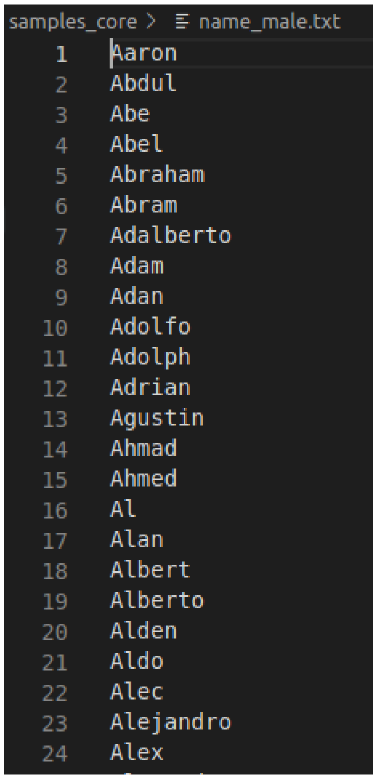
3.0 A COMPLETE EXAMPLE

It is appropriate to give a complete example of what datagenerator can do.This is a step by step example.Suppose that we already have defined the real data that software have to match.For this reason , we are going to generate 13 columns from a database column names at the first line.After the first line (columns names) , all of these columns are going to fill with 10 lines of data that datagenerator produces to fetch with real data.

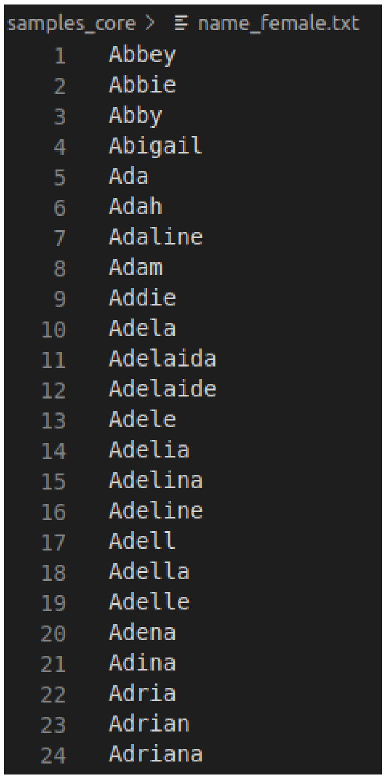
- Step 1 : Install the software in a directory and make a subdirecory for the libraries text files.This example is using 3 plain text files : hostname.txt , name_female.txt , name_male.txt


samples_core > hostname.txt

```
srv
server
logistic
sales
dev
develop
win
winxp
win2000
win2003
win2008
win2012
win2016
win95
win98
win7
win8
win10
wi998se
office
sharep
fileserver
filesrv
xoop
```

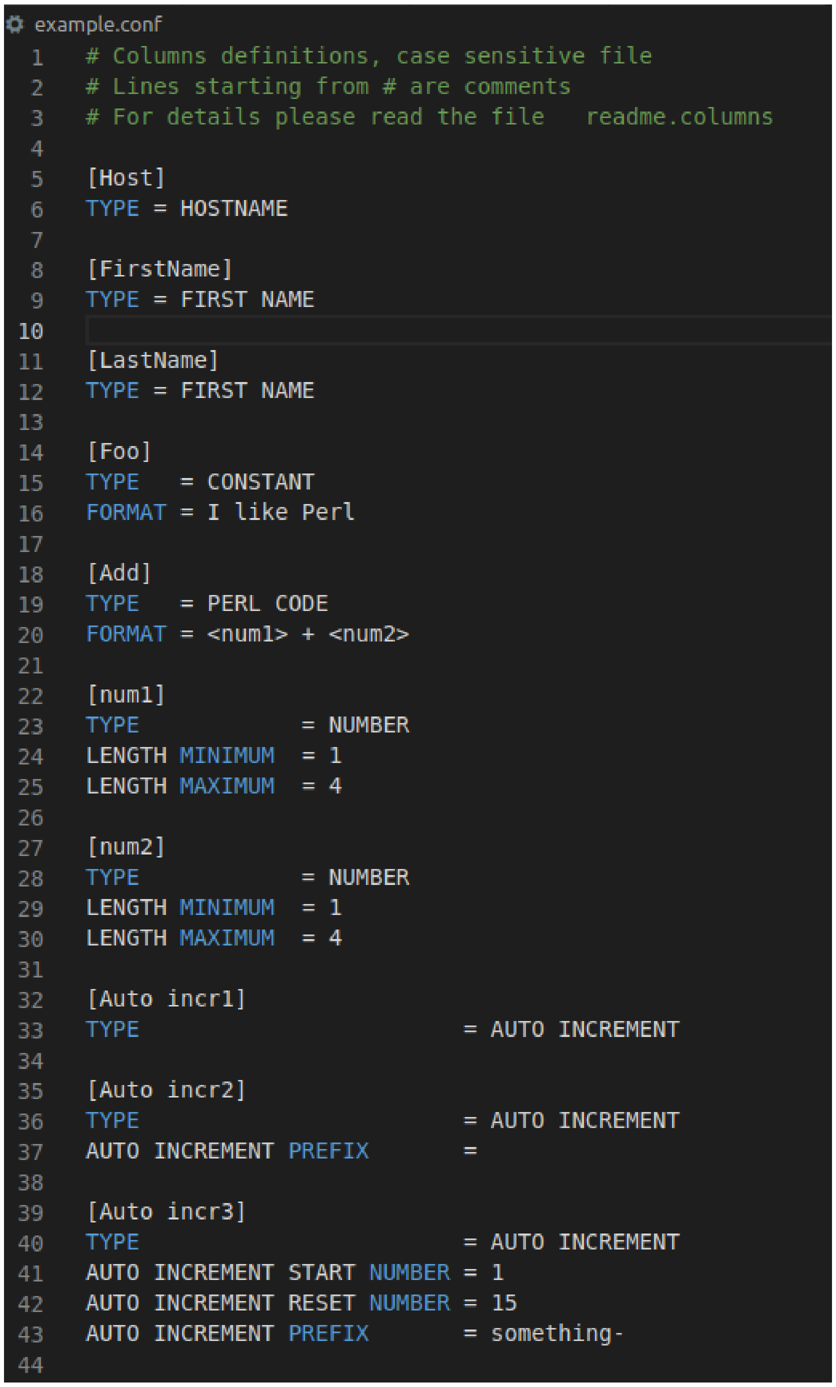

samples_core > name_male.txt

```
Aaron
Abdul
Abe
Abel
Abraham
Abram
Adalberto
Adam
Adan
Adolfo
Adolph
Adrian
Agustin
Ahmad
Ahmed
Al
Alan
Albert
Alberto
Alden
Aldo
Alec
Alejandro
Alex
```


samples_core > name_female.txt

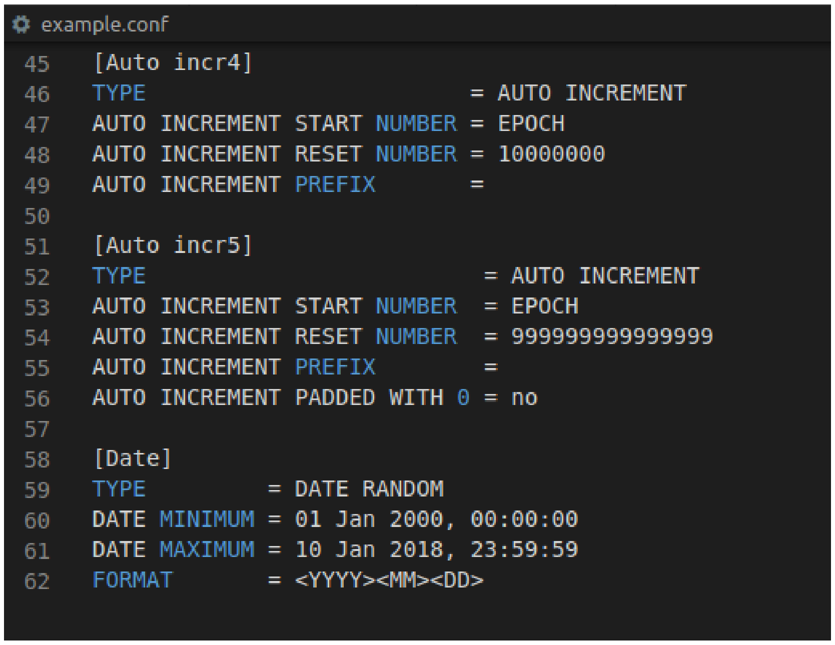
```
Abbey
Abbie
Abby
Abigail
Ada
Adah
Adaline
Adam
Addie
Adela
Adelaida
Adelaide
Adele
Adelia
Adelina
Adeline
Adell
Adella
Adelle
Adena
Adina
Adria
Adrian
Adriana
```

- Step 2 : Create the column configuration file : example.conf with columns names and the properties.



example.conf

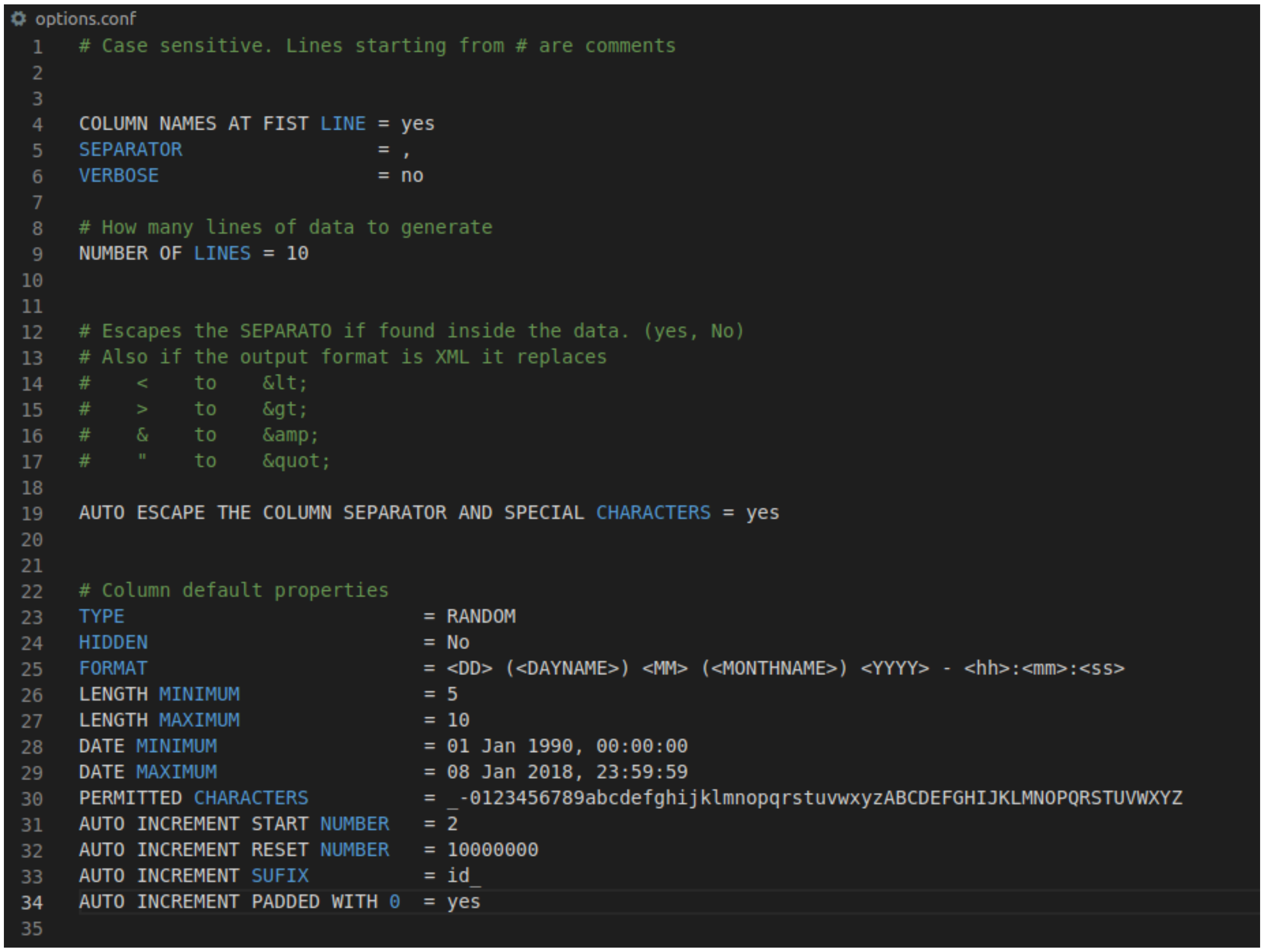
```
# Columns definitions, case sensitive file
# Lines starting from # are comments
# For details please read the file   readme.columns

[Host]
TYPE = HOSTNAME

[FirstName]
TYPE = FIRST NAME

[LastName]
TYPE = FIRST NAME

[Foo]
TYPE   = CONSTANT
FORMAT = I like Perl

[Add]
TYPE   = PERL CODE
FORMAT = <num1> + <num2>

[num1]
TYPE           = NUMBER
LENGTH MINIMUM  = 1
LENGTH MAXIMUM  = 4

[num2]
TYPE           = NUMBER
LENGTH MINIMUM  = 1
LENGTH MAXIMUM  = 4

[Auto incr1]
TYPE                        = AUTO INCREMENT

[Auto incr2]
TYPE                        = AUTO INCREMENT
AUTO INCREMENT PREFIX       =

[Auto incr3]
TYPE                        = AUTO INCREMENT
AUTO INCREMENT START NUMBER = 1
AUTO INCREMENT RESET NUMBER = 15
AUTO INCREMENT PREFIX       = something-

```



example.conf

```
[Auto incr4]
TYPE                        = AUTO INCREMENT
AUTO INCREMENT START NUMBER = EPOCH
AUTO INCREMENT RESET NUMBER = 10000000
AUTO INCREMENT PREFIX       =

[Auto incr5]
TYPE                         = AUTO INCREMENT
AUTO INCREMENT START NUMBER  = EPOCH
AUTO INCREMENT RESET NUMBER  = 9999999999999
AUTO INCREMENT PREFIX        =
AUTO INCREMENT PADDED WITH 0 = no

[Date]
TYPE         = DATE RANDOM
DATE MINIMUM = 01 Jan 2000, 00:00:00
DATE MAXIMUM = 10 Jan 2018, 23:59:59
FORMAT       = <YYYY><MM><DD>
```

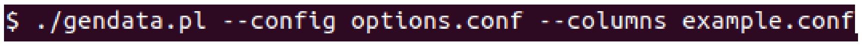
- Step 3 : Create the option configuration file : options.conf



options.conf

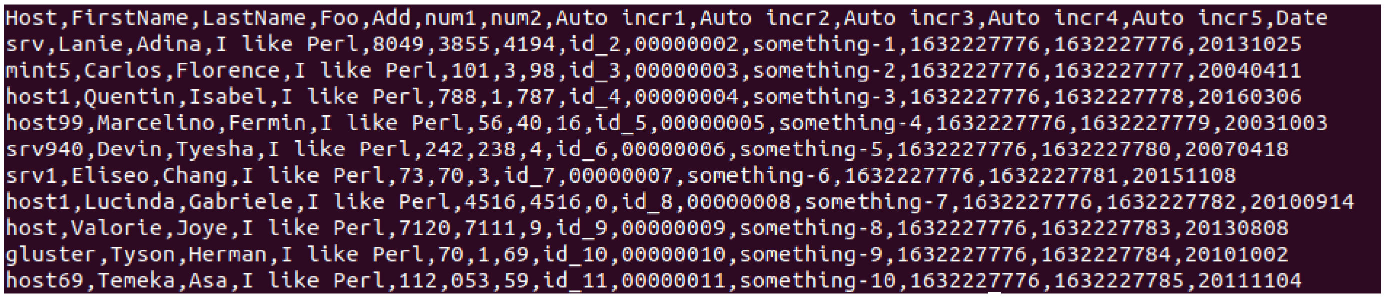
```
# Case sensitive. Lines starting from # are comments


COLUMN NAMES AT FIST LINE = yes
SEPARATOR                 = ,
VERBOSE                   = no

# How many lines of data to generate
NUMBER OF LINES = 10


# Escapes the SEPARATO if found inside the data. (yes, No)
# Also if the output format is XML it replaces
#    <    to    &lt;
#    >    to    &gt;
#    &    to    &amp;
#    "    to    &quot;

AUTO ESCAPE THE COLUMN SEPARATOR AND SPECIAL CHARACTERS = yes


# Column default properties
TYPE                          = RANDOM
HIDDEN                        = No
FORMAT                        = <DD> (<DAYNAME>) <MM> (<MONTHNAME>) <YYYY> - <hh>:<mm>:<ss>
LENGTH MINIMUM                = 5
LENGTH MAXIMUM                = 10
DATE MINIMUM                  = 01 Jan 1990, 00:00:00
DATE MAXIMUM                  = 08 Jan 2018, 23:59:59
PERMITTED CHARACTERS          = _-0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
AUTO INCREMENT START NUMBER   = 2
AUTO INCREMENT RESET NUMBER   = 10000000
AUTO INCREMENT SUFIX          = id_
AUTO INCREMENT PADDED WITH 0  = yes

```

- Step 4 : Run the command from terminal for the default output (console) :



```
$ ./gendata.pl --config options.conf --columns example.conf
```

- Step 5 : See the output from the program printed in terminal :



```
Host,FirstName,LastName,Foo,Add,num1,num2,Auto incr1,Auto incr2,Auto incr3,Auto incr4,Auto incr5,Date
srv,Lanie,Adina,I like Perl,8049,3855,4194,id_2,00000002,something-1,1632227776,1632227776,20131025
mint5,Carlos,Florence,I like Perl,101,3,98,id_3,00000003,something-2,1632227776,1632227777,20040411
host1,Quentin,Isabel,I like Perl,788,1,787,id_4,00000004,something-3,1632227776,1632227778,20160306
host99,Marcelino,Fermin,I like Perl,56,40,16,id_5,00000005,something-4,1632227776,1632227779,20031003
srv940,Devin,Tyesha,I like Perl,242,238,4,id_6,00000006,something-5,1632227776,1632227780,20070418
srv1,Eliseo,Chang,I like Perl,73,70,3,id_7,00000007,something-6,1632227776,1632227781,20151108
host1,Lucinda,Gabriele,I like Perl,4516,4516,0,id_8,00000008,something-7,1632227776,1632227782,20100914
host,Valorie,Joye,I like Perl,7120,7111,9,id_9,00000009,something-8,1632227776,1632227783,20130808
gluster,Tyson,Herman,I like Perl,70,1,69,id_10,00000010,something-9,1632227776,1632227784,20101002
host69,Temeka,Asa,I like Perl,112,053,59,id_11,00000011,something-10,1632227776,1632227785,20111104
```

__END__