

# TSL generator User Guide

## Download

A copy of TSL generator can be downloaded from JOOSE Moodle page. The current source code is maintained by Alex Orso from Georgia Institute of Technology, and can be downloaded from <https://github.com/alexorso/tslgenerator>.

The content of downloaded folder include:

- MacOSX, Linux and Windows versions of TSLgenerator
- A TSLgenerator manual  
(A good reference on the usage of comment, category and choice script syntax).
- A sample partition category testing for unix **grep** command in Specs folder.

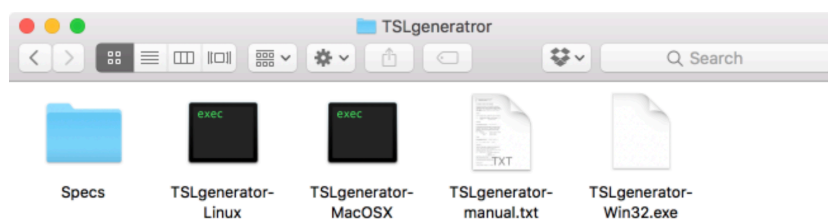
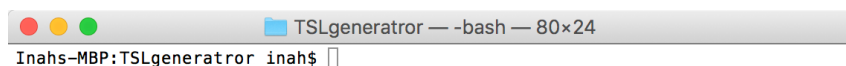


Figure 1: TSL generator

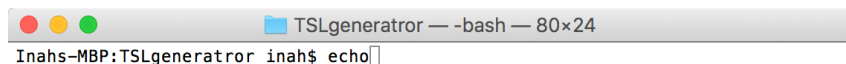
## Running TSL generator

You can run TSLgenerator on command line using linux shell, windows command prompt or mac terminal.

- (1) Navigate to the directory containing TSLgenerator.



- (2) Type **echo** on command line to write TSLgenerator arguments to standard output.



- (3a) To run TSLgenerator type **./TSLgenerator-[xx]**, where xx is the os version.

```

Inahs-MBP:TSLgenerator inah$ echo
Inahs-MBP:TSLgenerator inah$ ./TSLgenerator-MacOSX

-----
TSLgenerator

(C) University of California Irvine,
and Oregon State University, 2001
-----

USAGE: ./TSLgenerator-MacOSX [ --manpage ] [ -cs ] input_file [ -o output_file ]
Inahs-MBP:TSLgenerator inah$

```

(3b) To show TSLgenerator main page type `./TSLgenerator-[xx] --manpage`.

```

Inahs-MBP:TSLgenerator inah$ ./TSLgenerator-MacOSX --manpage

-----
TSLgenerator

(C) University of California Irvine,
and Oregon State University, 2001
-----

NAME
    tsl - generate test frames from a specification file

SYNOPSIS
    tsl [ --manpage ] [ -cs ] input_file [ -o output_file ]

DESCRIPTION
    The TSL utility generates test frames from a specification file
    written in the extended Test Specification Language. By default
    it writes the test frames to a new file created by appending a
    '.tsl' extension to the input_file's name. Options can be used
    to modify the output.

OPTIONS
    The following options are supported:

    --manpage
        Print this man page.

    -c
        Report the number of test frames generated, but don't
        write them to the output. After the number of frames is
        reported you will be given the option of writing them
        to the output.

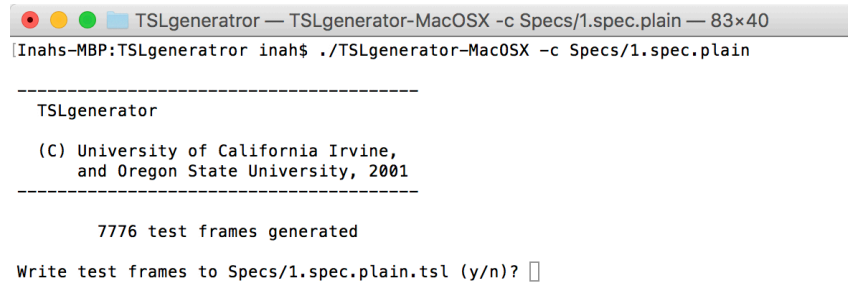
    -s
        Output is the standard output.

    -o output_file
        Output is the file output_file unless the -s option is used.

Inahs-MBP:TSLgenerator inah$

```

(3c) To report the number of test frames generated, but don't write them to the output. type `./TSLgenerator-MacOSX -c [yy]`, where yy is the TSLgenerator script file.



```

TSLgenerator

(C) University of California Irvine,
    and Oregon State University, 2001

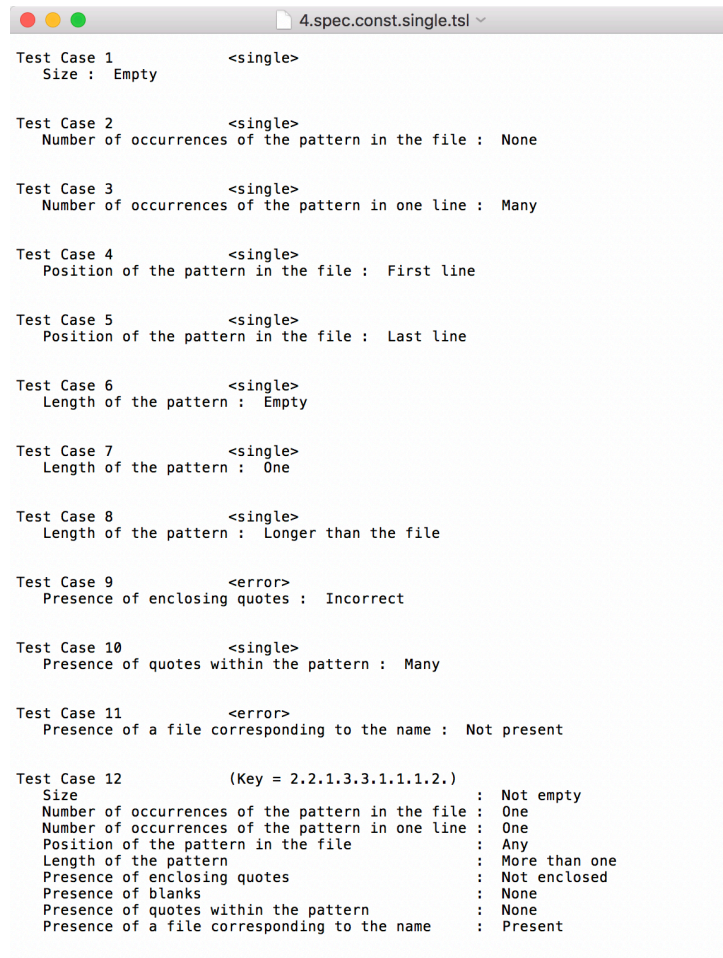
-----

7776 test frames generated

Write test frames to Specs/1.spec.plain.tsl (y/n)? 

```

(4) You can repeat (3c) with a different script file that defines additional constraint properties on the input domain. Once satisfied with the number of test frames generated, you can then write to file by choosing the **yes** option on command prompt. See sample outcome of generated testcases open with TextEdit below:



```

4.spec.const.single.tsl

Test Case 1          <single>
  Size : Empty

Test Case 2          <single>
  Number of occurrences of the pattern in the file : None

Test Case 3          <single>
  Number of occurrences of the pattern in one line : Many

Test Case 4          <single>
  Position of the pattern in the file : First line

Test Case 5          <single>
  Position of the pattern in the file : Last line

Test Case 6          <single>
  Length of the pattern : Empty

Test Case 7          <single>
  Length of the pattern : One

Test Case 8          <single>
  Length of the pattern : Longer than the file

Test Case 9          <error>
  Presence of enclosing quotes : Incorrect

Test Case 10         <single>
  Presence of quotes within the pattern : Many

Test Case 11         <error>
  Presence of a file corresponding to the name : Not present

Test Case 12         (Key = 2.2.1.3.3.1.1.1.2.)
  Size : Not empty
  Number of occurrences of the pattern in the file : One
  Number of occurrences of the pattern in one line : One
  Position of the pattern in the file : Any
  Length of the pattern : More than one
  Presence of enclosing quotes : Not enclosed
  Presence of blanks : None
  Presence of quotes within the pattern : None
  Presence of a file corresponding to the name : Present

```