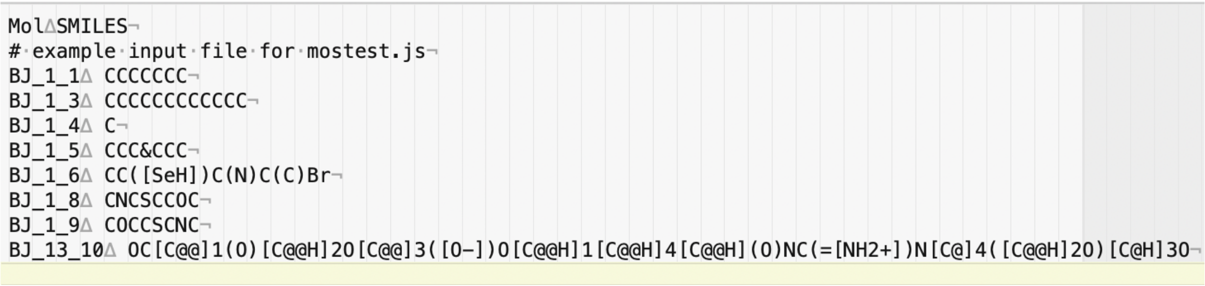
mostest.js
Test-Suite for the MOSFECCS Structural Formula Editor

mostest.js is an ES6 program for testing of the SMILES-generator and SMILES-parser of MOSFECCS under node.js.
In mostest.js, the code sections for calculating SMILES-codes from structural formulae (SMILES-generator; function getsmiles()) and for parsing SMILES-codes and reconstructing the structural formulae (SMILES-parser; function parse_m_Smiles() are identical with the corresponding sections of the MOSFECCS Editor. When the code of functions getsmiles() and/or parse_m_SMILES() in MOSFECCS has been changed, replace these functions in mostest.js accordingly to test the new version (if you use makeversion.pl, this is done automatically by the perl script).

mostest.js requires the name of a text file with SMILES-codes (one per line) as parameter. Each line of the input text file consists of an identifier (alphanumeric string) and a SMILES-code, separated by a tab character. Lines starting with # (comments) are ignored but copied to the SUMMARY file.

Example Input file *(tab separated text)*



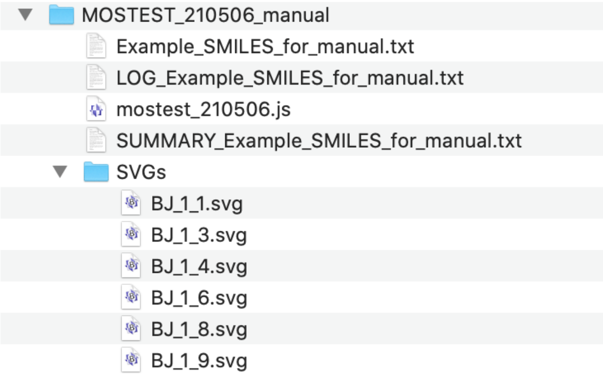
```
Mol△SMILES¬
#·example·input·file·for·mostest.js¬
BJ_1_1△ CCCCCCC¬
BJ_1_3△ CCCCCCCCCCCC¬
BJ_1_4△ C¬
BJ_1_5△ CCC&CCC¬
BJ_1_6△ CC([SeH])C(N)C(C)Br¬
BJ_1_8△ CNCSCCOC¬
BJ_1_9△ COCCSCNC¬
BJ_13_10△ OC[C@@]1(O)[C@@H]2O[C@@]3([O-])O[C@@H]1[C@@H]4[C@@H](O)NC(=[NH2+])N[C@]4([C@@H]2O)[C@H]3O¬
```

USAGE: the script mostest.js and the input text file must be in the same folder.
In terminal:
cd to this folder, then, at the terminal prompt, enter:
node mostest.js <inputFilename><return>.

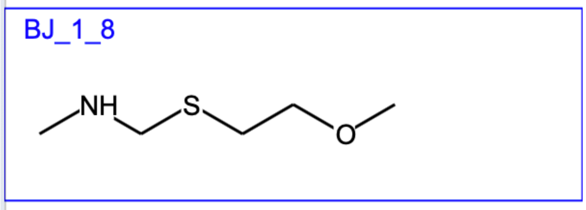
Example console output (from run with version 6 210506 *of mostest.js):*

```
bj@alanin MOSTEST_210506_manual % node mostest_210506.js Example_SMILES_for_manual.txt
09.05.2021 07:33:31 GMT+0200 (Central European Summer Time)
mostest.js run with txt file: Example_SMILES_for_manual.txt
mol:BJ_1_5 Parser ERROR for SMILES: CCC&CCC
      ERR: invalidSymbol in SMILES:"&"
mol:BJ_13_10 Parsing led to warnAtoms! SMILES: OC[C@@]1(O)[C@@H]2O[C@@]3([O-
])O[C@@H]1[C@@H]4[C@@H](O)NC(=[NH2+])N[C@]4([C@@H]2O)[C@H]3O
07:33:31 GMT+0200 (Central European Summer Time):
/Users/bj/MOSFECCS/MOSTEST/MOSTEST_210506_manual/mostest_210506.js run completed
Files SUMMARY_Example_SMILES_for_manual.txt and LOG_Example_SMILES_for_manual.txt written.
bj@alanin MOSTEST_210506_manual %
```

For each line in the input file, mostest.js parses the SMILES-code, reconstructs the structural formula as data objects (atoms, bonds etc.) in memory and generates an SVG graphic of the resulting structural formula as a file (with a 5-digit number constructed from the identifier as filename) in a subfolder "SVGs".



```
▼ MOSTEST_210506_manual
      Example_SMILES_for_manual.txt
      LOG_Example_SMILES_for_manual.txt
      mostest_210506.js
      SUMMARY_Example_SMILES_for_manual.txt
   ▼  SVGs
         BJ_1_1.svg
         BJ_1_3.svg
         BJ_1_4.svg
         BJ_1_6.svg
         BJ_1_8.svg
         BJ_1_9.svg
```



The SVG files can be visualized as shown above by most browsers.

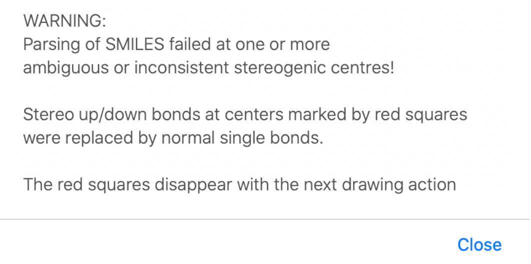
mostest.js then recalculates the SMILES-code from the molecular data objects and compares it with the SMILES-code in the input file. Errors and failures are recorded in a LOG file.

Example LOG file:

```
LOGFILE for run of /Users/bj/MOSFECCS/MOSTEST/MOSTEST_210506_manual/mostest_210506.js with file:
Example_SMILES_for_manual.txt
run by bj on node at 09.05.2021 07:33:31 GMT+0200 (Central European Summer Time)
========================================
BJ_1_1 OK
BJ_1_3 OK
BJ_1_4 OK
BJ_1_5 Parser ERROR for SMILES: CCC&CCC
        ERR: invalidSymbol in SMILES:"&"
BJ_1_6 OK
BJ_1_8 OK
BJ_1_9 parsing failed for SMILES: COCCSCNC checkSMILES: CNCSCCOC
BJ_13_10 Parsing led to warnAtoms! SMILES: OC[C@@]1(O)[C@@H]2O[C@@]3([O-
])O[C@@H]1[C@@H]4[C@@H](O)NC(=[NH2+])N[C@]4([C@@H]2O)[C@H]3O
```

An **ERROR** is recorded when the input SMILES cannot be parsed successfully (non-legal SMILES, illegal characters encountered, unpaired paranthesis etc.). No SVG graphics file is generated in this case.

WarnAtoms are generated when the parser was unable to draw the structure with the correct stereo-configuration at certain atoms. No SVG graphics file is generated in this case. If the parser would be used inside the Editor MOSFECCS, the warning would be displayed as an alert:



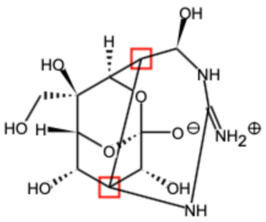
WARNING:
Parsing of SMILES failed at one or more
ambiguous or inconsistent stereogenic centres!

Stereo up/down bonds at centers marked by red squares
were replaced by normal single bonds.

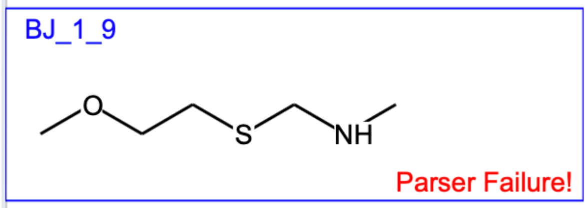
The red squares disappear with the next drawing action

and the problematic atoms would be highlighted with red squares in the structural formula draw by the parser:

Example for WarnAtoms after using "put Smiles" in MOSFECCS.:



A **FAILURE** is recorded in the LOG whenever the SMILES recalculated from the structure generated by parsing the input-SMILES is not identical to the latter (i.e. if the input was a legal SMILES code but not the canonical one generated by MOSFECCS for this structure. An SVG graphic file is generated but annotated with "Parser Failure!" in this case.



The statistics of the Test are summarized in the **SUMMARY_Example_SMILES_for_manual.txt** file:

Example SUMMARY file:

```
SUMMARY for run of /Users/bj/MOSFECCS/MOSTEST/MOSTEST_210506_manual/mostest_210506.js with file
Example_SMILES_for_manual.txt
Comments:
# example input file for mostest.js

5 of a total of 8 SMILES passed the test.

PARSER ERRORS:1
```

```
BJ_1_5: CCC&CCC Error:invalidSymbol in SMILES:"&"
ATOM WARNINGS:1
BJ_13_10: OC[C@@]1(O)[C@@H]2O[C@@]3([O-])O[C@@H]1[C@@H]4[C@@H](O)NC(=[NH2+])N[C@]4([C@@H]2O)[C@H]3O

PARSER FAILURES:1

BJ_1_9 SMILES entered:
COCCSCNC
CNCSCCOC
was returned by getsmiles() of parsed structure
```

B. Jaun

April 2021