

## TTS Substitution rules.

### Structure

The file should contain 2 sections: header and data. Each section is preceded by its title between square brackets. End of lines preceded by # are comments

Example:

---

```
# This is a comment and it is not interpreted
[header]
language = ENU                # ENU|FRF|SPM
charset = "windows-1252"

[data]
/VODKA 1 POLIAKOV 1x6/ --> "Vodka Poliakov six units"
```

---

### Header Section

It contains the charset and the language. Language can be FRF for French, ENU for English or SPM or Spanish.

Example:

```
# This is a comment and it is not interpreted
[header]
language = SPM                # ENU|FRC|SPM
charset = "windows-1252"
```

### Data Section

The "data" section contains zero or more "rules", a rule can only occupy one line.

Comments can also be inserted at the end of a rule and start with a '#' character and span till the end of the line.

A rule has the following syntax:

```
<search-spec> --> <replacement-spec> <comment>? \n
```

Note that special characters and characters with a special meaning need to be escaped.

Some examples are:

```
\      escape character
^      Match the beginning of the line
.      Match any character (except newline)
$      Match the end of the line (or before newline at the end)
|      Alternation
()     Grouping
[]     Bracketed Character class
*      0 or more times quantifier
+      1 or more times quantifier
?      1 or 0 times quantifier
"      double quotes in the replacement-spec
\s     matches any whitespace character,
\d     matches any digit
{n}    match exactly n times
{n,}   match at least n times
{n,m}  match at least n but not more than m times
```

### Search-spec

In general the format of the search-spec is:

```
/<regular-expression>/ <modifier>*
```

```
<modifier> := [imsx]
```

Optional modifiers are:

- **i**: search is case-insensitive
- **m**: let '^' and '\$' match even at embedded newline characters
- **s**: let the '.' pattern match even at embedded newline characters
- **x**: allows for regular expression extensions like inserting whitespace and comments

## Replacement-spec

The format of the replacement spec is a quoted ("...") string or a non-blank string in case the translation is a single word. It may contain back references of the form \$n (n: 1, 2, 3, ...) which refer to the actual match for the n-th capturing subpattern in <search-spec>.

E.g. \$1 denotes the first submatch. A back reference with a number exceeding the total number of submatches in <search-spec>, is translated into an empty string. A literal dollar sign (\$) must be escaped (\\$).

Everything following <replacement-spec> and on the same line is considered as comment when starting with '#', else it is just ignored.

## Rule examples

```
[data]
```

```
/MRT\*VODKA POLIAKOV20CL37\.5 1x6/ --> "Vodka Poliakov"
```

```
Replaces "MRT*VODKA POLIAKOV20CL37.5 1x6" by saying "Vodka Poliakov"
```

```
/SofToGo/ --> "Best Company of the Year"
```

```
Replaces each occurrence of the string "SofToGo " by "Best Company of the Year".
```

```
/(Sof)To(Go)/ --> "$1 2 $2"
```

```
Replaces "SofToGo" by "Sof 2 Go".
```

```
/(Sof)To(Go)/ i --> "$12$2"
```

```
Replaces "SofToGo", "SOFTOGO", softogo, etc. into "Sof2Go", "SOF2GO", "sof2go", etc.
```

```
/(Sof)To(Go)/ --> "$1 2 $2$3"
```

```
Replaces "SofToGo" by "Sof 2 Go".
```

```
/(\s):-\)(\s)/ --> "$1ha ha$2"
```

```
Replaces for instance " :-) " into " ha ha ".
```

```
/\x80 ?(\d+)\.(\d{2})\d*/ --> "$1 euro $2 cents"
```

```
Rewrites for instance "€9.751" into "9 euro 75 cents".
```