

# Managing terminology with term checker

Terminology management is essential for translation workflows.

**Jake Cahill** takes a look at the TechScribe term checker.

*If you consistently use one term to refer to the same concept, your text is easier to understand and cheaper to translate.*

To avoid unnecessary synonyms, it's important that you define a clear set of rules for how terms should be used.

After you've created your terminology rules, you can enforce those rules by using software such as the TechScribe term checker for ASD-STE100.

The term checker is a customised version of the LanguageTool (LT) platform that allows you to check text against the rules in Simplified Technical English (STE) and create your own rules.

The term checker has the following advantages over the standard LanguageTool:

- Better part-of-speech (POS) disambiguation than LT. The POS disambiguation in LT is basic. In LT, there is no POS disambiguation for multi-word terms.
- Thousands of rules for text simplification (words that are not in the STE specification, but which have STE equivalents). (Look at the category STE 7, rule 1.1. dictionary.) You can use these lexical rules to help you to write 'plain English'.
- Many STE grammar rules are applicable to technical texts. (And you can easily deselect a rule if it is not applicable to your documents.)

## How I used the term checker

In this article, I'm going to focus on how I used the term checker to check text against a list of unapproved terminology.

The first step was to create my own rules that the term checker would enforce. Each rule is written in XML.

To create my own rules, I added my non-approved terms to the `grammar-projectterms.xml` file.

The following rule is an example of how the term checker can detect if a term is used as a noun or a verb, allowing you to create complex decisions about how your own terms should be used, see Figure 1.

The `<token>` element tells the term checker what your non-approved term is. My rule uses the `regexp` attribute on the `<token>` element to tell the term checker to apply a regular expression when searching for a term. The question mark after 'screens' tells the term checker to find all instances of 'screen' and 'screens'.

The `<exception>` element tells the term checker to ignore certain instances of the term, depending on the value of the `postag` attribute. Here, the `postag` attribute is set to "IS VERB", which tells the term checker to ignore the term when it's used as a verb.

After I wrote all my rules, I wanted a way to tell the Language Tool to check my text only against my own terms and not the STE terms. After some research I found that LanguageTool has a command line interface, which includes the `--enablecategory` flag, which was exactly what I needed.

The term checker's `grammar-projectterms.xml` file contains all your own terms, and these terms are in a category called 'Terminology', so I passed that name to the `--enablecategory` flag.

The whole command looks like Figure 2.

## Going further

The term checker is an extremely simple, yet powerful tool.

```
<rule id="PROJECT_NOT_APPROVED_screen2" name="Project Not Approved noun: screen">
  <pattern>
    <token regexp="yes">screens?<exception postag="IS_VERB"/></token>
  </pattern>
  <message>The noun '<match no="1"/>' is not approved. Possible replacements: <suggestion><match no="1"
postag_regexp="yes" postag="(NNS?)" postag_replace="$1">page</match></suggestion></message>
  <short>Project Dictionary. Not approved noun: screen</short>
  <example correction="page" type="incorrect">This <marker>screen</marker> displays the results.</example>
  <example correction="pages" type="incorrect">If the <marker>screens</marker> do not show these
messages, stop the test.</example>
  <example type="correct">On this <marker>page</marker> you can enter a new name.</example>
  <example type="correct">When you <marker>screen</marker> the drugs for side-effects...</example>
  <example type="correct">Who <marker>screens</marker> the drugs for side-effects?</example>
  <example type="triggers_error">When the medical technicians <marker>screen</marker> the drugs for side-
effects...</example><!-- False positive -->
</rule>
```

Figure 1. Example of how the term checker can be detected if a term is used as a noun or a verb

```
java -jar languagetool-commandline.jar -l en-GB --enablecategories TERMINOLOGY -eo --json
C:\Users\Jake\Desktop\term-checker-test.txt
```

Figure 2. Checking text using Language Tool command line

Over the next few weeks I'm going to start writing an application in NodeJS that automatically executes this command and parses the results.

I hope that I'll be able to create an automated workflow that checks text before it's sent to translation, so we can improve the quality of our source text and our translations.

### Limitations

As is explained on the TechScribe website, "the term checker gives you an effective analysis of text. But, the term checker does not have features such as management reports and authoring memory that add to the cost of software".

Nevertheless, this tool has amazing possibilities for helping writers to keep their text and translation memories clean and consistent, helping to make content clearer and cheaper to translate. 

### References

Aerospace and Defence Industries Association of Europe (ASD) Simplified Technical English [www.asd-ste100.org](http://www.asd-ste100.org) (accessed August 2018)

LanguageTool – Spell and Grammar Checker <https://languagetool.org> (accessed August 2018)

LanguageTool Wiki' Software That Supports LanguageTool As A Plug-In Or Add-On' <http://wiki.languagetool.org/software-that-supports-languagetool-as-a-plug-in-or-add-on> (accessed August 2018)

Muegge, U (2013) 'TechScribe STE Term Checker' *Communicator*, Summer 2013: 28-29

Term checker for Simplified Technical English (ASD-STE100) [www.simplified-english.co.uk](http://www.simplified-english.co.uk) (accessed August 2018)



**Jake Cahill** is a Technical Author at dotmailer, based in Croydon, UK. He has a degree in Applied Languages from the University of Portsmouth and loves learning new languages,

including programming languages.

E: [jakecahill91@hotmail.co.uk](mailto:jakecahill91@hotmail.co.uk)

LI: [www.linkedin.com/in/jake-cahill](http://www.linkedin.com/in/jake-cahill)



## The UK's Leading Technical Communication Event



# TCUK Conference 25<sup>th</sup> – 27<sup>th</sup> September 2018

**De Vere Staverton Estate, Daventry, Northampton**



The UK's largest annual event for technical communicators, the Technical Communication UK Conference (TCUK), will take place at the De Vere Staverton Estate in Daventry, Northampton this year.

Our conference theme this year is '**The Pursuits of the Polymath**', and seeks to examine just how and in which directions our profession is developing in the short, medium, and long-term future. Many of us are already branching out into other communication fields, including video and animation, web and online delivery, illustration, translation, and delivery of training. Our clients, anxious to keep up with or to maintain a lead over their competitors, demand more and more from our skill set, and it's in our nature to respond to the challenges that their demand creates. Come along and network with others in our profession, to exchange ideas and experiences with them, and to help ourselves prepare for what lies ahead of us.

For further information on sponsorship or booking your place at the conference, please go to the website - [www.technicalcommunicationuk.com](http://www.technicalcommunicationuk.com) or email Claire Kelly at [claire.kelly@admin.co.uk](mailto:claire.kelly@admin.co.uk).