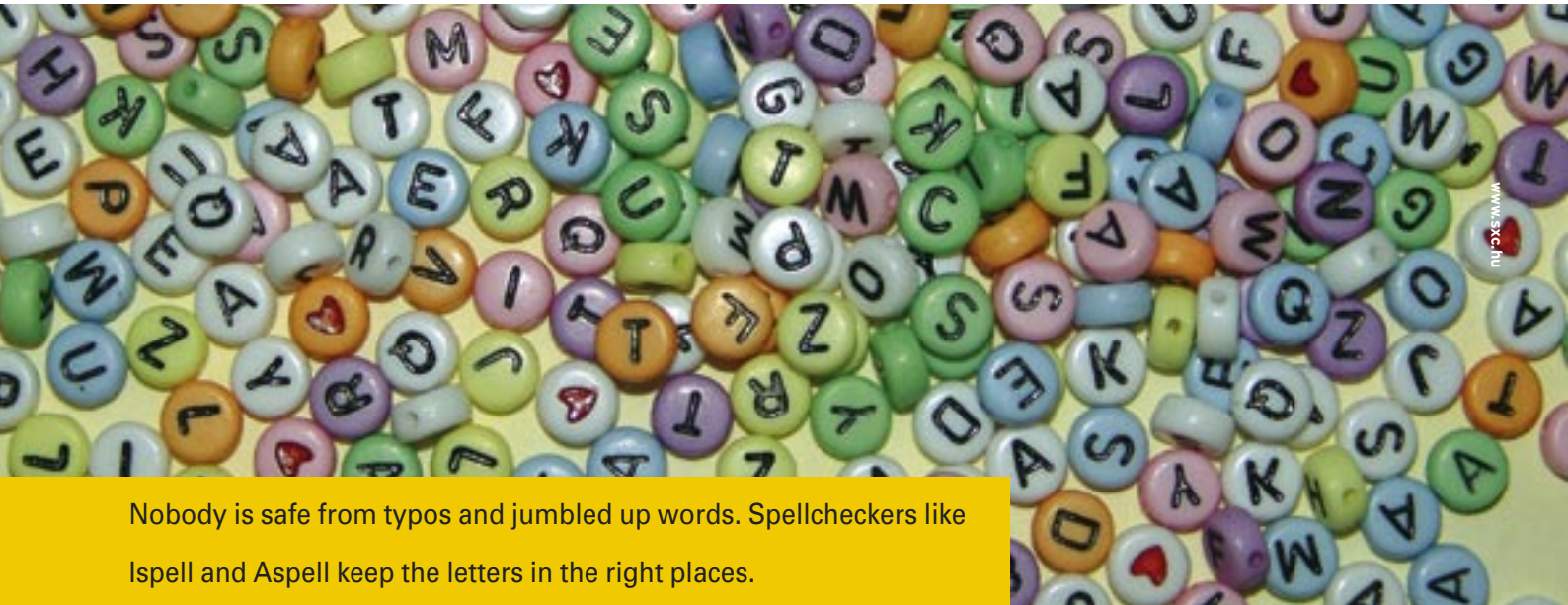


The Ispell and Aspell command line spellcheckers

SPELLING LESSON



Nobody is safe from typos and jumbled up words. Spellcheckers like Ispell and Aspell keep the letters in the right places.

BY HEIKE JURZIK

Spellcheckers in office packages and mail programs help find typos and suggest alternative spellings. Ispell and Aspell proofread your files at the command line, and, although they can't replace a good dictionary, they are extremely reliable helpers.

Ispell is a mature application that has been on various Unix derivatives for many years now. The GNU project introduced Aspell as a potential successor. Aspell comes with dictionaries for various languages and can handle UTF-8 (in contrast to Ispell).

This article shows you how to proofread at the command line. I'll describe some tips and tricks, and I'll demon-

Command Line

Although GUIs such as KDE or GNOME are useful for various tasks, if you intend to get the most out of your Linux machine, you will need to revert to the good old command line from time to time. Apart from that, you will probably be confronted with various scenarios where some working knowledge will be extremely useful in finding your way through the command line jungle.

strate how to call both Ispell and Aspell while working with the Vim and (X)Emacs editors.

Simply Ispell

Ispell runs interactively at the command line. If the spellchecker finds an error, it suggests an alternative spelling. If you decide not to accept the suggestion, you can add the word to your personal dictionary, or just ignore the term for the rest of the document.

The simple command syntax is

```
ispell filename
```

You may need to run Ispell with the *-d* (for "dictionary") option, depending on the dictionary you need. A quick glance at the */usr/lib/ispell* directory (Figure 1) tells you which dictionaries are available on your system. You should see a number of dictionaries with the *.aff* suffix – Ispell uses these dictionaries to manage hyphenation and accented characters in the various languages.

Some Linux distributions use different names for the dictionaries. To make sure that special characters are correctly en-

coded, define a character set using the *-T* option.

There is no need to specify the dictionary and character set options each time you run the program; instead you can add appropriate entries to the *~/.bashrc* Bash configuration file in your home directory:

```
export ?
DICTIONARY=dictionary name
export ?
CHARSET=character set
```

Then reparse the configuration file using the following command:

```
source ~/.bashrc
```

Interaction Required

Ispell uses the top line to display unknown words. If the spellchecker finds similar words in the dictionary, it gives you an enumerated list of the alternative words (Figure 2). To accept one of these suggestions, simply press the key for the number that appears with the word in the enumerated list to tell Ispell to replace the word.

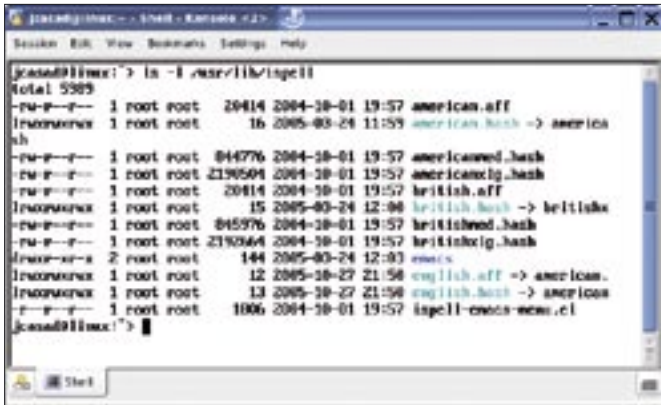


Figure 1: The Ispell on our lab machine understands British and US English.



Figure 2: Ispell offers similar words as alternatives.

If you do not want to replace the word that Ispell thinks is wrong, you can either press the space key to ignore the word once, press [A] to ignore the word for the rest of the Ispell session, or press [I] to store the word permanently. In the latter case, Ispell adds the term to your personal dictionary.

Your personal word lists are stored as hidden files in your home directory. The filename is made up of `.ispell_` and the name of the dictionary you used. For example, if you were working with the *ngerman* dictionary, your personal word list would be `.ispell_ngerman`.

If Ispell discovers an error but does not give you a (reasonable) alternative, you can press [R] (for “replace”) and type your changes at the prompt. Ispell replaces the word throughout the text.

The End of the Story

Ispell quits after spellchecking, but you can alternatively press [Q] to quit the spellchecker and discard any changes. Ispell will prompt you to determine if you really do want to discard the changes. You can then press either [Y] to quit the program or [N] to carry on spellchecking. The alternative is to press [X] to quit Ispell, but store any changes you have made.

Creating Backups

Ispell gives you a safe option if you specify the `-b` parameter when launching the program. This tells the spellchecker to create a backup for each file it checks. The backup copies are identifiable by a `.bak` or `~` suffix and contain the original text.

Many distributions have versions of Ispell that create backups without you

actually specifying the option for creating a backup copy. This is sometimes the default behavior as specified by the package maintainer. If prefer not to create backup copies, you can disable this feature by setting the `-x` flag.

Other Formats

Ispell has the ability to spellcheck the HTML and TeX/LaTeX formats in addition to text files. The spellchecker automatically identifies any file with the

`.html` or `.htm` suffix as an HTML file. In this case, the spellchecker ignores any HTML tags it discovers during the spellchecking session. The only exception is the *ALT* attribute, which you can use to define alternative text for an embedded image.

If the file suffix is missing or in capitals (`.HTM`), you can tell Ispell that the file is HTML when launching the program:

```
ispell -H file.HTM
```

The same thing applies to XML and SGML files: pass the `-H` option to the program when launching Ispell to tell the spellchecker to ignore tags when spellchecking. (Some distributions, such as Debian, use `-h` (with a lowercase “h”) for this option, whereas others, such as Suse, use a capital `-H`.)

Ispell identifies TeX-/LaTeX files by reference to the `.tex` file suffix and does not spellcheck formatting instructions. Ispell uses backslash, which normally introduces control sequences in the layout system, to identify formatting, however, text in curly brackets is spellchecked. The spellchecker identifies

Spellcheckers in Vim and Emacs

Ispell and Aspell run in the background in many KDE and Gnome applications, and will check your documents at a click. But it is quite easy to talk Vim and (X)Emacs into cooperating.

A single line in the Vim configuration file (`~/.vimrc`) gives you a macro. To map a function key [F10] to the spellchecker, add the following line to the file for Ispell:

```
map <F10> :w!<CR>:z
!ispell %<CR>:e! %<CR>
```

If you prefer Aspell, use the following entry instead:

```
map <F10> :w!<CR>:z
!aspell -c %<CR>:e! %<CR>
```

Place any command line options you need after the command.

If you use Emacs or Xemacs, add the following line to the configuration file (`~/.emacs` or `~/.xemacs/custom.el`):

```
(setq-default z
ispell-program-name "aspell")
or:
(setq-default z
ispell-program-name "ispell")
```

THE AUTHOR

Heike Jurzik studied German, Computer Science and English at the University of Cologne, Germany. She discovered Linux in 1996 and has been fascinated with the scope of the Linux command line ever since. In her leisure time you might find Heike hanging out at Irish folk sessions or visiting Ireland.



comments, which are escaped with the percent character (%) in TeX-/LaTeX files, and checks them for typos.

Aspell Alternative

Aspell is an alternative spellchecker for the command line. This program is the designated successor for Ispell, and it already offers a lot more. For example, the current versions (0.60.x) support the UTF-8 character set. The generic syntax is as follows:

```
aspell -c filename
```

It is not typically necessary to explicitly specify a dictionary, as Aspell evaluates your language settings. However, if you see an error telling you the correct dictionary could not be found, you might like to check if you really have the necessary dictionary installed on your system. Typing *aspell --help | less* at the command line tells you which languages Aspell “speaks” as a list of *Available Dictionaries*; dictionaries are located below */usr/lib/aspell-0.60/* by default.

If a required dictionary is missing, you

can use your package manager to install the missing resource. Just like in Ispell, you can use the *-d* option to specify a dictionary. If you have a UTF-8 encoded text, don’t forget to let Aspell know by specifying *--encoding=utf-8*.

Interaction Required

Aspell highlights unknown terms and gives you a numbered list of suggested replacements. To accept a suggestion, press the appropriate number and press [Enter].

If you need to replace a word with a completely different word, press [R] and type the replacement for the word you are replacing. In contrast to Ispell, this option in Aspell only replaces the current instance of the word. If you want Aspell to replace all instances of the word in the document, you need to press [Shift] + [R] instead.

You can press [I] to ignore a word exactly once, or press [Shift] + [I] to tell Aspell to ignore the word for the rest of the spellcheck session. To store a word permanently in your personal dictionary, press [A]. Again, Aspell stores private

dictionaries below your home directory, and – in a similar approach to Ispell – it uses a combination of the word *.aspell*. and the current dictionary to identify the file.

Aspell also quits automatically after completing the spellcheck. You can press [X] to quit the program and store any changes you have made up to that point. To quit the spellchecker and discard your changes, press [B] and confirm when prompted.

A Question of Format

Just like Ispell, Aspell supports various file formats. You can specify the *-H* (for HTML/SGML/XML) or *-t* (for TeX/LaTeX).

If you would like to migrate to the newer spellchecker, the Aspell package includes a Perl script titled *aspell-import* that imports your personal Ispell and Aspell dictionaries and adds them to the Aspell dictionary in your home directory.

Check the *Spellcheckers in Vim and Emacs* box for details on calling Ispell or Aspell while working in one of these editors. ■

ADVERTISEMENT