NAME
hunspell – spell checker, stemmer and morphological analyzer

SYNOPSIS
hunspell [−1aDGHhLlmnrstvw] [−−check−url] [−d dict[,dict2,...]] [−−help] [−i enc] [−p dict] [−vv] [−−version] [file(s)]

DESCRIPTION
Hunspell is fashioned after the *Ispell* program. The most common usage is "hunspell" or "hunspell filename". Without filename parameter, hunspell checks the standard input. Typing "cat" and "exsample" in two input lines, we got an asterisk (it means "cat" is a correct word) and a line with corrections:

> $ hunspell -d en_US
> Hunspell 1.2.3
> *
> & exsample 4 0: example, examples, ex sample, ex-sample

Correct words signed with an '*', '+' or '-', unrecognized words signed with '#' or '&' in output lines (see later). (Close the standard input with Ctrl-d on Unix/Linux and Ctrl-Z Enter or Ctrl-C on Windows.)

With filename parameters, *hunspell* will display each word of the files which does not appear in the dictionary at the top of the screen and allow you to change it. If there are "near misses" in the dictionary, then they are also displayed on following lines. Finally, the line containing the word and the previous line are printed at the bottom of the screen. If your terminal can display in reverse video, the word itself is highlighted. You have the option of replacing the word completely, or choosing one of the suggested words. Commands are single characters as follows (case is ignored):

R	Replace the misspelled word completely.
Space	Accept the word this time only.
A	Accept the word for the rest of this *hunspell* session.
I	Accept the word, capitalized as it is in the file, and update private dictionary.
U	Accept the word, and add an uncapitalized (actually, all lower-case) version to the private dictionary.
S	Ask a stem and a model word and store them in the private dictionary. The stem will be accepted also with the affixes of the model word.
0-*n*	Replace with one of the suggested words.
X	Write the rest of this file, ignoring misspellings, and start next file.
Q	Exit immediately and leave the file unchanged.
ˆZ	Suspend hunspell.
?	Give help screen.

OPTIONS
−1   Check only first field in lines (delimiter = tabulator).

−a   The **−a** option is intended to be used from other programs through a pipe. In this mode, *hunspell* prints a one-line version identification message, and then begins reading lines of input. For each input line, a single line is written to the standard output for each word checked for spelling on the line. If the word was found in the main dictionary, or your personal dictionary, then the line contains only a '*'. If the word was found through affix removal, then the line contains a '+', a space, and the root word. If the word was found through compound formation (concatenation of two words, then the line contains only a '−'.

If the word is not in the dictionary, but there are near misses, then the line contains an '&', a space, the misspelled word, a space, the number of near misses, the number of characters between the beginning of the line and the beginning of the misspelled word, a colon, another space, and a list of the near misses separated by commas and spaces.

Also, each near miss or guess is capitalized the same as the input word unless such capitalization is illegal; in the latter case each near miss is capitalized correctly according to the dictionary.

Finally, if the word does not appear in the dictionary, and there are no near misses, then the line contains a '#', a space, the misspelled word, a space, and the character offset from the beginning of the line. Each sentence of text input is terminated with an additional blank line, indicating that *hunspell* has completed processing the input line.

These output lines can be summarized as follows:

OK: *

Root: + <root>

Compound:
         −

Miss: & <original> <count> <offset>: <miss>, <miss>, ...

None: # <original> <offset>

For example, a dummy dictionary containing the words "fray", "Frey", "fry", and "refried" might produce the following response to the command "echo 'frqy refries | hunspell −a":
(#) Hunspell 0.4.1 (beta), 2005-05-26
& frqy 3 0: fray, Frey, fry
& refries 1 5: refried

This mode is also suitable for interactive use when you want to figure out the spelling of a single word (but this is the default behavior of hunspell without -a, too).

When in the **−a** mode, *hunspell* will also accept lines of single words prefixed with any of '*', '&', '@', '+', '−', '~', '#', '!', '%', '`', or '^'. A line starting with '*' tells *hunspell* to insert the word into the user's dictionary (similar to the I command). A line starting with '&' tells *hunspell* to insert an all-lowercase version of the word into the user's dictionary (similar to the U command). A line starting with '@' causes *hunspell* to accept this word in the future (similar to the A command). A line starting with '+', followed immediately by **tex** or **nroff** will cause *hunspell* to parse future input according the syntax of that formatter. A line consisting solely of a '+' will place *hunspell* in TeX/LaTeX mode (similar to the **−t** option) and '−' returns *hunspell* to nroff/troff mode (but these commands are obsolete). However, the string character type is *not* changed; the '~' command must be used to do this. A line starting with '~' causes *hunspell* to set internal parameters (in particular, the default string character type) based on the filename given in the rest of the line. (A file suffix is sufficient, but the period must be included. Instead of a file name or suffix, a unique name, as listed in the language affix file, may be specified.) However, the formatter parsing is *not* changed; the '+' command must be used to change the formatter. A line prefixed with '#' will cause the personal dictionary to be saved. A line prefixed with '!' will turn on *terse* mode (see below), and a line prefixed with '%' will return *hunspell* to normal (non-terse) mode. A line prefixed with '`' will turn on verbose-correction mode (see below); this mode can only be disabled by turning on terse mode with '%'.

Any input following the prefix characters '+', '−', '#', '!', '%', or '`' is ignored, as is any input following the filename on a '~' line. To allow spell-checking of lines beginning with these characters, a line starting with '^' has that character removed before it is passed to the spell-checking code. It is recommended that programmatic interfaces prefix every data line with an uparrow to protect themselves against future changes in *hunspell*.

To summarize these:

*   Add to personal dictionary

@   Accept word, but leave out of dictionary

#	Save current personal dictionary
˜	Set parameters based on filename
+	Enter TeX mode
–	Exit TeX mode
!	Enter terse mode
%	Exit terse mode
'	Enter verbose-correction mode
ˆ	Spell-check rest of line

In *terse* mode, *hunspell* will not print lines beginning with '*', '+', or '−', all of which indicate correct words. This significantly improves running speed when the driving program is going to ignore correct words anyway.

In *verbose-correction* mode, *hunspell* includes the original word immediately after the indicator character in output lines beginning with '*', '+', and '−', which simplifies interaction for some programs.

−−check−url
> Check URLs, e-mail addresses and directory paths.

−D   Show detected path of the loaded dictionary, and list of the search path and the available dictionaries.

−d dict,dict2,...
> Set dictionaries by their base names with or without paths. Example of the syntax:

−d en_US,en_geo,en_med,de_DE,de_med

en_US and de_DE are base dictionaries, they consist of aff and dic file pairs: en_US.aff, en_US.dic and de_DE.aff, de_DE.dic. En_geo, en_med, de_med are special dictionaries: dictionaries without affix file. Special dictionaries are optional extension of the base dictionaries usually with special (medical, law etc.) terms. There is no naming convention for special dictionaries, only the ".dic" extension: dictionaries without affix file will be an extension of the preceding base dictionary (right order of the parameter list needs for good suggestions). First item of −d parameter list must be a base dictionary.

−G   Print only correct words or lines.

−H   The input file is in SGML/HTML format.

−h, −−help
> Short help.

−i enc   Set input encoding.

−L   Print lines with misspelled words.

−l   The "list" option is used to produce a list of misspelled words from the standard input.

−m   Analyze the words of the input text (see also hunspell(4) about morphological analysis). Without dictionary morphological data, signs the flags of the affixes of the word forms for dictionary

developers.

−n   The input file is in nroff/troff format.

−P password
        Set password for encrypted dictionaries.

−p dict Set path of personal dictionary. The default dictionary depends on the locale settings. The following environment variables are searched: LC_ALL, LC_MESSAGES, and LANG. If none are set then the default personal dictionary is $HOME/.hunspell_default.

        Setting *−d* or the *DICTIONARY* environmental variable, personal dictionary will be **$HOME/.hunspell_dicname**

−r   Warn of the rare words, wich are also potential spelling mistakes.

−s   Stem the words of the input text (see also hunspell(4) about stemming). It depends from the dictionary data.

−t   The input file is in TeX or LaTeX format.

−v, −−version
        Print version number.

−vv   Print ispell(1) compatible version number.

−w   Print misspelled words (= lines) from one word/line input.

EXAMPLES
hunspell −d en_US english.html

hunspell −d en_US,en_US_med medical.txt

hunspell −d ˜/openoffice.org2.4/share/dict/ooo/de_DE

hunspell *.html

hunspell −l text.html

ENVIRONMENT
DICTIONARY
        Similar to *−d.*

DICPATH
        Dictionary path.

WORDLIST
        Equivalent to *−p.*

FILES
The default dictionary depends on the locale settings. The following environment variables are searched: LC_ALL, LC_MESSAGES, and LANG. If none are set then the following fallbacks are used:

/usr/share/myspell/default.aff Path of default affix file. See hunspell(4).

/usr/share/myspell/default.dic Path of default dictionary file. See hunspell(4).

       $HOME/.hunspell_default. Default path to personal dictionary.

SEE ALSO
       hunspell (3), hunspell(4)

AUTHOR
       Author of Hunspell executable is László Németh. For Hunspell library, see hunspell(3).

       This manual based on Ispell's manual. See ispell(1).

BUGS
       There are some layout problems with long lines.