
Proofreader

By: John Maxwell (Maxwell.pa@Xerox.com)

Use in conjunction with Analyzer, SpellingArray

INTRODUCTION

The Proofreader interactively looks for and corrects spelling errors in a given TEdit document. To use it, go to the TEdit menu (click the middle button while in the title bar) and invoke the menu item labeled "Proofread". This will simultaneously attach a special menu to the side of the document and start proofreading from the caret. The proofreader scans the document from the caret location and stops at the first misspelling it finds, highlighting it with a pending delete selection. Successive misspellings can be found by clicking the "Proofread" menu item in the TEdit menu or the "Proofread" menu item in the new side menu.

At this point, you can either correct the misspelled word or skip to the next misspelling. If you are not sure what the correct spelling is, you can get a menu of possible corrections by invoking the "Correct" menu item. Selecting a correction from this menu will cause it to be automatically inserted into the document in place of the misspelling. (Note: The Proofreader occasionally suggests some very bizarre spelling corrections for your misspelled word. Do not be alarmed; this is a known but unavoidable artifact of the heuristic used for checking for misspellings. (see notes at the end)) If the word was erroneously flagged as misspelled, then you can insert the word into your personal word list of acceptable words by invoking the "Insert" menu item. At the end of the editing session you can save the word list on a file with the "StoreWordList" command (see below).

If when the Proofreader is invoked the current selection has more than one word in it, then the Proofreader will only correct the words in that selection. Otherwise, the Proofreader always proofreads the text from the caret to the end of the document.

PROOFREADER SUB-COMMANDS

Under the TEdit Proofread menu item there are a number of sub-commands that the user can invoke. They are:

Proofread

The same as the top level Proofread command. Attaches a special menu to the side of the document and starts proofreading from the caret.

CountWords

Counts the number of words in the current selection. To count the number of words in the entire document, first click "All" in TEdit's expanded menu.

SetProofreader

Gives the user a menu of proofreaders to use for proofreading. If there is only one proofreader, no menu is generated. If the user selects a remote server, a second menu may be generated of proofreaders available on the server.

StoreWordList

Allows the user to save the words that he inserted into the proofreader onto a remote file. The words from an existing version of the file will be read in first and then a new file will be generated consisting of the old file plus the new words. After the file is written, the list of newly inserted words is set to NIL.

LoadWordList

The inverse of **StoreWordList**. If you have a file that you want to load every time you use the proofreader, you can add it to `Proofreader.AutoLoad`, and it will be loaded when the proofreader is first opened. `Proofreader.AutoLoad` can be either a file or a list of files.

AutoCorrect

Sets the variable `Proofreader.AutoCorrect` so that the proofreader *automatically* generates a list of corrections whenever it finds a misspelled word. In the AutoCorrect mode, the proofreader will also automatically scan for the next misspelled word whenever after a correction has been selected. If you want to stop the proofreading process, click outside of the correction menu. If you want to insert the flagged word into your word list, click the menu item labeled **"*INSERT*"**. If you want to continue proofreading without changing the flagged word, select the menu item labeled **"*SKIP*"**.

ManualCorrect

Sets the variable `Proofreader.AutoCorrect` so that the proofreader will not generate a list of corrections unless the user asks for it.

PROOFREADER VARIABLES

There are a couple of variables that the user can set in his init file to change how the proofreader works. They are:

`Proofreader.AutoLoad` [Variable]

A file or list of files to be loaded into the proofreader every time that the proofreader is initialized.

`Proofreader.AutoCorrect` [Variable]

A boolean that determines whether or not a list of corrections is automatically generated whenever the proofreader finds a misspelled word. The default value is NIL.

`Proofreader.AutoDelete` [Variable]

A boolean that determines whether or not to delete the old versions of a word list file when a new one is written out. The default value is T.

`Proofreader.MenuEdge` [Variable]

The side of the window that the proofreader menu appears on (can be either LEFT or RIGHT). The default value is LEFT.

`Proofreader.UserFns` [Variable]

A list of functions to be applied to misspelled words (as strings). If the function returns a non-NIL value, then the word is assumed to be correctly spelled. (If the function `ATOMHASH#PROBES` is added to `Proofreader.UserFns`, then any word defined as an atom becomes legal. `ATOMHASH#PROBES` tests whether or not a string is an atom without creating new atoms. If you want a more restricted test (i.e. "anything defined as a procedure is legal") first test that the string

exists at an atom before doing MKATOM. Otherwise, the atom space will fill up with misspelled words.)

NOTES

- The proofreader uses a heuristic to determine whether or not a word is in its word list that occasionally will produce false positives. This is most noticeable when the proofreader is generating corrections for misspelled words. I don't know of any way to eliminate this problem except to use a different algorithm, the fastest of which is at least twice as slow. Hopefully people will find it more of a nuisance than a real problem.
- There is no way to remove words once they have been inserted into the local dictionary. The only way to get rid of a bad word is to reload the dictionary. This can be done by reloading the SpellingArray file. If a bad word gets into one of the remote files, you can edit the file to get rid of it.

ACKNOWLEDGMENTS

The algorithm used in the Proofreader is based on the algorithm in the Cedar Spelling Tool by Bob Nix. For more information on the implementation, see the section "How it Works" in {Cyan}<CedarChest6.1>Documentation> SpellingToolDoc.tioga.