The Software Toolworks
Walt Bilofsky, Prop.

14478 GLORIETTA DRIVE
SHERMAN OAKS, CALIFORNIA 91423

TELEPHONE
(213) 986-4885

SPELL - Spelling Detection and Marking Program

Version 2.0 - October 1982
by Robert Wesson
Knowledge Engineering, Inc.

TABLE OF CONTENTS

1. INTRODUCTION

SPELL is a spelling error detection and marking program. It runs on 8080 or Z80 microcomputers under the CP/M or the Heath/Zenith HDOS operating systems. SPELL is compatible with documents produced by most text editors and word processors, including WordStar (tm), Magic Wand (tm), SpellBinder (tm), PIE, and TEXT.

In its normal mode of operation, SPELL reads a document from a disk file, and presents you with a list of the words it thinks are incorrect. For each word, you can decide whether to pass it, add it to the dictionary, or mark it as an error in the document. After marking the errors, you can quickly find and correct them using your text editor or word processor.

SPELL has other modes of operation as well. You can simply list the doubtful words from a document on your terminal, or in a file. Or you can enter a list of new words from a file into the dictionary.

SPELL uses a root dictionary and a table of common prefixes and suffixes. This allows it to recognize more than 50,000 English words.

SPELL requires a minimum of 48K of RAM memory and at least 30K of disk storage. It can be used on systems with only one 100K 5 1/4" disk, but most applications will require more disk storage than this.

Although most of this manual may eventually prove useful to you, in order to start making use of SPELL you only need to read through Section 3.1.

2. PREPARING A SPELL WORKING DISK

In order to use SPELL, you must first create a working disk containing the SPELL program and data files. Section 2.1 explains how to do this on a typical computer with 64K of RAM memory and at least two disk drives, running the CP/M operating system. If your computer does not fit this description, or if you do not want the SPELL program on your A: disk, see Section 2.2. If you are using HDOS, see Section 2.3. If you use the Magic Wand word processor, also see Section 2.4.

If you have a 5" disk system, you should first place a write protect label over the rectangular notch in your SPELL distribution disk. This will help prevent accidental destruction of data on the disk. Then make a working copy of your SPELL distribution disk. Consult your computer manual for instructions on copying a disk. (On the Osborne 1, use the COPY program, which is on your CP/M system disk.) Place your distribution diskette in a safe place and use the working disk you have just made.

2.1. On Computers with 64K and Two Disk Drives

If you are using single density 5" or other small capacity disks, you should just make your SPELL working disk into a bootable system disk. On most CP/M systems, you can do this using SYSGEN. (On the Osborne, SYSGEN can be found on your CP/M Utility Disk or CP/M system disk.) Put a bootable disk with SYSGEN in drive A, and your SPELL working disk in drive B. Then type the command

```
SYSGEN
```

SYSGEN will ask you to designate a source disk and a destination disk. The way it words its request will vary from system to system. Tell it to use A for the source disk and B for the destination disk. When it has finished writing on B, the disk in B will be a bootable system disk.

If this procedure does not work on your computer system, check your system manual; SYSGEN may work differently and you will have to follow the procedure described in your system manual.

Once you have a bootable working disk with the SPELL files on it, boot it up and type the following commands:

```
ERA DICTNARY.48?
REN DICTNARY.SPL=DICTNARY.64K
```

Your disk is now a completed SPELL working disk. Proceed to Section 3 for instructions on running SPELL.

If you are using larger capacity disks, you may want to have SPELL along with other programs on your system disk. In this case, just make sure there is at least 44K free on the system disk, boot up on it, insert the SPELL working disk in drive B:, and type the commands

```
PIP A:=B:SPELL.COM
PIP A:=B:AFFIXTAB.SPL
PIP A:DICTNARY.SPL=B:DICTNARY.64K
```

2.2. On Other Computers

SPELL can be used on computers with 48K or more of RAM memory and one or more disk drives.

If you have only one disk drive, perform the steps listed in the previous section. But instead of using the PIP commands, use the method described in your operating system manual for copying files on a single disk drive system. (On some single disk systems, such as Heath/Zenith CP/M, the operating system may allow you to follow the exact procedure from the previous section, asking you to switch diskettes by hand where necessary.)

If your system has less than 64K of RAM memory, or you get an "Out of Memory" message when you run SPELL, you will need to use one of the other dictionaries instead of DICTNARY.64K.

Your distribution disk contains the following files:

SPELL.COM (or .ABS)	The SPELL program.
PATCHES.SPL	A list of patchable locations.
DICTNARY.64K	The largest dictionary.
DICTNARY.48K	A smaller dictionary.
DICTNARY.48S	The smallest dictionary. (Not included on Xerox format disks.)
AFFIXTAB.SPL	The prefix/suffix table.

These files include three dictionaries, encoded into different sized files. You should try to use the largest dictionary which can run on your machine. If you have 60K or more of RAM, try using DICTNARY.64K. If you have a smaller system, you may need to use DICTNARY.48K (preferred) or DICTNARY.48S. To use the dictionary uou have chosen, copy it to the SPELL working disk, and rename it to DICTNARY.SPL.

More details on the contents of these dictionaries may be found in Section 5.6.

If you want to run the SPELL program from a disk other than your A: disk, use the commands in Section 2.1, but copy the SPELL files to the disk of your choice. Remember, though, that the DICTNARY.SPL and AFFIXTAB.SPL files must be on your logged in disk when you run SPELL.

2.3. On HDOS Systems

Using a new blank diskette, create a bootable system disk as described in the HDOS manual. Boot up on it. Place the SPELL distribution disk in SY1:. Type the following commands:

```
COPY *.*=SY1:SPELL.COM,AFFIXTAB.SPL
COPY DICTNARY.SPL=SY1:DICTNARY.64K
```

If you have a single drive system, use the ONECOPY program instead of the COPY command.

If you have only 48K of memory, do not use DICTNARY.64K in the above commands. Instead, if you have two or more disks on your 48K system, use DICTNARY.48K. If you have a single disk 48K system, and can not fit all the files you need on your system disk, use DICTNARY.48S and set HDOS to its stand-alone configuration (an undocumented but handy feature explained in Appendix A.) DICTNARY.48K will not work with HDOS in stand-alone mode unless you have more than 48K.

2.4. Using SPELL with Magic Wand

If you are using SPELL with documents generated by Magic Wand, SPELL needs some attention in order to operate correctly with "soft hyphens". If you are using WordStar, SpellBinder, or TEXT, SPELL will handle soft hyphens without patching.

Magic Wand users should run SPELL using the command

```
SPELL -I& filename
```

Or you can accomplish the same thing by patching the "ignore character" in the SPELL program. See Appendix B for instructions on how to patch SPELL. Type the file PATCHES.SPL from your distribution disk to see the address to patch, and the value to put there.

2.5. Installation Problems

To run SPELL, your working diskette must contain SPELL.COM (or SPELL.ABS), AFFIXTAB.SPL, and one of the DICTNARY files, copied as DICTNARY.SPL. If you get a FATAL error message such as "Not enough memory" when you first run SPELL, check the following:

1) Be sure you have the proper dictionary for your system configuration, and that your system meets the minimum 48K memory requirement. If your CP/M system does not print the correct memory size on booting, it must be reconfigured as described in your CP/M manual.

2) Minimize memory usage during SPELL's operation. You may not be able to run printer spoolers and other preloaded

programs because such programs take up memory space which is required for SPELL's dictionary.

With a "normal" 48K HDOS system, be sure you do not have the HDOS overlays loaded. Unload as many device drivers as you can. We have allowed for the presence of several device drivers (such as TT:, SY:, DK:, and LP:), but more may reduce available memory too much. Under HDOS 2.0 and later compatible versions, you can check overlay and device driver status using the STAT command. Sometimes rebooting can free up some memory and allow you to proceed.

3) As a last resort, use the next smaller dictionary.

It is possible that you will encounter disk errors in attempting to copy the SPELL distribution disk. This can occur because disk drives run at slightly different speeds, and may be quite usable but marginally out of adjustment. If you have this problem, you may return the disk to The Software Toolworks for replacement. However, before doing so, you may succeed in copying the disk by using another method.

You should try copying with the distribution disk on the other drive: from drive B: to drive A:, for example. Or, if you can not copy the distribution disk with a one step format and copy program (such as COPY or DUP), you can try formatting a blank disk first and using the PIP commands in Section 2.1 to copy the needed files to it. This avoids copying the remaining sectors on the distribution disk, which may be where the problem area is.

3. USING SPELL.

Although you may think of a document as a piece of paper, in this manual we will use the term "document" to refer to any disk file containing text. It may have been created by a text editor or a word processor program.

First we will tell you briefly how to use SPELL to check a document's spelling. Then we will explain in more detail what SPELL is doing as it runs. Special operations, such as creating a new dictionary from scratch, are described in Section 4.

3.1. Briefly.

To check a document using SPELL, boot up on your SPELL working diskette in drive A:, and place the diskette containing your document in drive B:.

(If the SPELL files are on another disk, make sure that disk is the logged in disk. For instance, if the SPELL working diskette is in drive B, give the command "B:". If you have a single disk system, all files must be on disk A:.)

(HDOS users only: Instead of A: and B:, substitute SY0: and SY1:. If the working diskette is not SY0:, you must specify it in the SPELL command. Example: SY1:SPELL filename.)

To check the spelling of a document called filename, enter the command

```
SPELL filename
```

The drive letter is part of the file name. For instance, if the document is on drive B:, you will need to type

```
SPELL B:filename
```

SPELL will check the document and ask you about each word it thinks might be misspelled. You can tell it:

M - to Mark the word as misspelled.
I - to Ignore it, because it is correct.
A - to Add it to your dictionary.
R - to add a Root of the word.
S - to Start over from the beginning of the list.

If you mark any words, SPELL will create a new copy of your document, placing the character # in the copy to indicate the marked words. You can then locate the marked words using the search command in your word processor or text editor. The old version of the document is saved using the original name and the extension .BAK.

If you add any words or roots to the dictionary, SPELL will rewrite the dictionary file.

If you make a mistake, the "Start over" choice lets you change your selections.

In order to avoid the inquiries from SPELL, and just print out a list on your terminal of the questionable words, enter the command

```
        SPELL -L filename
```

You can get the list of words on a file instead, by the command

```
        SPELL -L filename >listfile
```

Having read this far, you should know enough to begin using SPELL. If you run into problems, read Section 3.2. If you want a more detailed explanation of what SPELL is doing, see Section 3.3. Section 3.4 provides a step-by-step example of a SPELL session. Section 5.6 discusses the difference among the dictionaries provided with SPELL.

As you become more familiar with SPELL, you may wish to study the remainder of this manual.

3.2. Possible Problems.

If SPELL does not want to start up, make sure the three files SPELL.COM (or .ABS), DICTNARY.SPL and AFFIXTAB.SPL are located on your default disk (CP/M) or the same disk (HDOS). See Section 2 for more help in copying the proper files.

Remember that SPELL must be on your system disk (or the default disk on CP/M). If your document is on another disk, don't forget to include the device name (e.g., B: on CP/M, SY1: on HDOS) as part of the file name.

Don't forget that if you use the Mark option, SPELL will create a new copy of your document on disk, and will save the old copy as well. If there is not enough room for two copies of the document, SPELL will not be able to create the marked copy and will produce an error message. In this case, you will need to use the -L option and work from the list of misspelled words.

If you are adding to the dictionary, and there is not enough room to write the new copy of the dictionary file out before deleting the old one, SPELL will offer you the opportunity to delete the old file first. It warns you that this is risky; the danger is that if the computer or disk should fail to complete the write operation, the dictionary will be lost. This is not likely, so you can tell SPELL to proceed by typing "y" and a return. (Naturally, you always keep a backup copy of your important files on another disk, to limit the damage in case of computer or human failure.)

Remember that the Mark, Add and Root options all cause files to be written. If you get a BDOS R/O error, something is Read Only. This can be caused by a write protect label on the diskette. Or a particular file can be Read Only; use STAT *.* to display the file protection status of each file. Or a temporary write protection can result from changing disks in a drive. After changing disks, you can remove this temporary write protection by a warm boot: hold the CTRL shift key and type the letter C.

3.3. In More Detail.

When you run SPELL, it first takes a few seconds to read the large dictionary file from disk.

Then it enters what is usually the longest part of its operation: the dictionary lookup phase. During this time, SPELL reads your document file, comparing each word against its dictionary, and making an internal list of those words which it cannot recognize. Hyphenation at line ends and text processor commands are ignored.

As it reads your document, SPELL types out periodic messages to keep you informed of its progress through the file. It gives the total number of words processed and the number of words not recognized. SPELL can look up about 4000 words/minute on 4 MHz machines. It typically processes a page of single-spaced text in 7 seconds, and a twenty-page document in less than three minutes. Double these times if you are running a 2 MHz 8080 or Z80 (such as the Heath H/Z89 computers); reduce them if you are using a faster CPU such as a 6 MHz Z80.

Next, SPELL enters the word disposition phase. Each unrecognized word is displayed so that you can tell the program how to treat that word. You should type one of the following letters:

M Mark the word in the text for later correction. This answer should be used for each genuinely misspelled word.

I Ignore the word. Use this response for those words which are spelled correctly, but which you do not wish to enter permanently into the dictionary, because they are used so infrequently.

A Add the word to the dictionary. Use this response for words which are spelled correctly and which you use routinely.

R add the word's Root to the dictionary. If the word contains a prefix or suffix, you should use this response to enter its root into the dictionary. SPELL will then be able to recognize not only the word but many of its derivations as well. If you select this response, SPELL will ask you to type in the root, terminated by a return.

The root you type will be added to the dictionary. A fuller explanation of roots is below.

S start over with the word list. If you make a mistake or want to change a response, this allows you to go back to the beginning and start with the first word on the list. When you start over, the response you have already given to a word is shown following the word. To leave the response unchanged, just hit the RETURN key.

What is a root? Suppose the word "restored" occurs in your document. (SPELL knows this word, but say it did not.) "Restored" consists of the prefix "re-", the root word "store", and the suffix "-e+ed" (which means first remove a trailing e, and then add "ed"). Rather than adding "restored" to the dictionary, you should add the root word "store". Then SPELL will know not only "restored", but also words like "restore", "storing", and "stores".

When a word is added to the dictionary, rather than expanding the size of the file, SPELL uses a method which keeps the file the same size, but increases very slightly the chance that SPELL will miss an incorrect word. If you add a thousand words to the dictionary, you increase the chance of missing an error by roughly half. This means you should use some restraint in adding words to the dictionary. Section 5.4 discusses accuracy and other limitations of SPELL; Section 6 gives more details on SPELL's methods.

When all questionable words have been handled, SPELL asks if you want to proceed to mark the file and, if words have been added, update the dictionary. If you type the letter "n" (no), SPELL returns to the top of the word list and cycles through it again, in the same way as for the "S" response described above. Each word is now presented with the response you gave previously; hitting RETURN leaves the response unchanged.

If you answer the question with "y" (yes), SPELL enters its final phase: update. If any words were added to the dictionary, it writes your new, expanded dictionary on the disk. If any errors were marked in the text, SPELL will save your original document file with a ".BAK" extension, then create a new document identical to the old one except that each misspelled word will be terminated with a '#' character.

If necessary, you may abort SPELL's processing at any time and return to the system monitor by typing control-C under HDOS or control-B under CP/M.

3.4. An Example Session with SPELL.

Suppose I have just finished typing a file named LETTER.TXT which I now want to check for misspelled words. With SPELL on drive A: and LETTER.TXT on B:, the following is a transcript of my SPELL session. In this transcript, comments are added in braces {}, and everything typed by the computer is underlined. (This example was run on a CP/M system; for HDOS replace A> by > and B: by SY1:.)

```
A> spell b:letter.txt
Spell Version 2.0 (c) 1981 Knowledge Engineering, Inc.
574 words processed; 8 words not in dictionary.
cul? add                          {This is from the word cul-de-sac.}
dispite? mark                           {Should have been "despite."}
Harmon? ignore           {A person's name which I do not often use.}
Knollwood? ignore           {A local street: again, not often used.}
neighborhood? root = neighbor   {SPELL knows the suffix "+hood".}
Picayune? ignore             {Another local name I don't often use.}
Wesson? add                                     {I use my name a lot.}
Widespead? mark                       {Oops! A typo in "widespread."}

               {A:DICTNARY.SPL and B:LETTER.TXT are now updated.}

A> pie b:letter.txt

     {Using my editor, I search for '#' and correct my mistakes.}

A>era b:letter.bak                                {No longer needed.}
```

4. ADVANCED OPERATION

In running SPELL, you may select many options, using one or more switches in the command line. A switch consists of the character - or >, a letter, and perhaps a parameter. You must not put a space between the - and its parameter.

For example, the command

```
SPELL -L -DNEWDICT DOC.TXT
```

contains two switches: -L, which has no parameter, and -D, which has the parameter NEWDICT. DOC.TXT is the name of the document file, which must be the last thing in the command.

These are the switches you can use with SPELL:

-L LOOKUP MODE. The -L switch runs SPELL in a fast lookup mode. In this mode, SPELL scans the document file for unknown words, and simply lists these words on your terminal in alphabetical order, without any headings or other printout. SPELL doesn't ask you any questions or create any new files. The sample session above, replayed in lookup mode, would look like this:

```
A> spell -l b:letter.txt
cul
dispite
Harmon
Knollwood
neighborhood
Picayune
Wesson
widespread
A>
```

>filename

PRINT TO FILE. This switch makes SPELL type everything not on your terminal, but rather into the file called "filename". This feature is especially useful together with the -L switch, in order to place the list of misspelled words in a text file for later reference. For example, the command

```
A> spell -l >misspell b:letter.txt
```

will check file B:LETTER.TXT in lookup mode, print nothing on the terminal, but will save all misspelled words in a file called MISSPELL.

Remember that with this switch, nothing at all appears on your terminal, even the questions SPELL asks you. Everything goes to the file instead. So this switch has limited usefulness except with the -L switch.

-T TEXT FORMATTER COMMANDS. SPELL normally assumes that a line starting with a period is a text formatter command, and ignores such lines. (Many text formatters use such command lines, including the TEXT program available from The Software Toolworks). The -t switch changes this feature. If you simply say -t, SPELL will treat lines begun with '.' just like any other text line. However, if you say -t<character>, SPELL will substitute <character> as the text processor command line indicator, and will not check any line in the document which begins with that character. This is useful if your text processor program uses something other than a period to begin its command lines.

-M MARKER CHARACTER. When you mark errors which SPELL detects, each error is flagged by replacing its last character with a marker character. The default marker character is '#', but you can use a different character instead by using the -M switch followed by the single character you want to use.

-B BACKUP FILE. The default extension for the backup document file created by SPELL is ".BAK", but you can change that with -B. If you use the -B switch alone, with no characters following it, SPELL will not save the backup version of your text file. (This will not reduce the amount of disk space required, since SPELL marks by creating a backup file and copying the new document from it. Including -B on the command line simply makes SPELL delete the backup file when done.)

If you follow this switch with one to three characters, that extension will be used for the backup file. For example, combining this switch with the one above, you can make SPELL use the extension ".OLD" and the marker '@' with this command line:

```
A> spell -m@ -bold textfile
```

-D DICTIONARY. You can get SPELL to use a dictionary other than its default "DICTNARY.SPL" by using this switch followed by a filename. This is useful when you maintain separate dictionaries for different subject areas, or would like to keep the dictionary on a different disk from the spelling program itself.

-X AFFIX TABLE. This switch uses the filename following it instead of the default prefix/suffix table file AFFIXTAB.SPL. It works like the "D" switch does for dictionaries.

-I IGNORE CHARACTER. Most word processors have certain control characters which they insert within a word for various purposes, such as to mark a potential hyphenation point ("soft hyphen"). SPELL must ignore these characters in order to recognize words correctly. SPELL knows about the control characters used by WordStar, SpellBinder and TEXT. But in order to work with MagicWand, SPELL must be told to ignore the character '&' within words. This is accomplished by using the -I& switch, or by patching the default value of this switch. Other word processors may require a different value for this switch; you will have to determine the soft hyphen indicator from the manufacturer of the program.

-A ADD TO DICTIONARY. If you have a list of words to add to the dictionary, call SPELL with the -A switch. In this mode, SPELL expects the input file to contain a list of words which should simply be inserted into the dictionary; no lookup is performed. One use of this mode is to construct alternate dictionaries for different subject areas. For example, you might assemble a list of, say, 2000 medical terms which you want added into a special dictionary for your medical office. After typing up the file NEWWORDS of those 2000 words, your secretary might create that special dictionary as follows:

```
A> pip meddict.spl=dictnary.spl    {make a copy of the
main dictionary}

A> spell -a -dmeddict.spl newwords {add the medical terms
to it}
```

From then on, you could use the -D switch to specify that dictionary for your medical documents.

If the indicated dictionary does not exist, SPELL can create one from scratch, containing only those words from its normal input file. If you have your own complete dictionary, you may wish to use this feature of SPELL to encode it. In this case, you must follow the -A switch with an integer indicating the dictionary length in bytes. This length must be a multiple of 256, and should be as large as your system configuration will allow. The supplied dictionary lengths have been optimized for the various system configurations mentioned above. If a dictionary is too large, it will not fit in memory. One that takes up all the available memory will not leave much room for SPELL to keep its list of misspelled words, and one too small will yield poor accuracy. We found the following lengths to be a good compromise on the configurations they were designed for:

```
DICTNARY.64K = 29952 bytes
DICTNARY.48K = 21760 bytes
DICTNARY.48S = 17920 bytes
```

Remember, any or all of these switches may be used in combination with each other. As an exercise, try to understand what the following command line does:

```
A> spell -t/ -bxxx -dc:specdict -m$ document
```

5. OPERATING NOTES

5.1. Single-Disk Operation (HDOS users only)

If SPELL's full text/dictionary correction features are used, the single disk must accommodate not only the program, but also two copies of the dictionary and the textfile. (The original dictionary is not deleted from the disk until the copy is successfully written.) On a typical 368 sector disk, with HDOS in stand-alone mode and using the stand-alone 48K dictionary, your text file can occupy at most about 79 sectors, or about 6 1/2 single-spaced pages of text. If you do not change the dictionary, the file can be 114 sectors, and if you use lookup mode only (or do not request any text marking during the disposition phase), you can process a text file of about 228 sectors, or almost 20 pages of text. If you must use SPELL from a bootable disk, you will be limited to 67 sector text files even in lookup mode, so this type of operation is severely limited.

5.2. Changing the Defaults

Each of the switch default values discussed in Section 4 may be patched to accommodate your system and style of operation. For instance, you may prefer a marking character other than '#', or you may wish SPELL not to ignore lines beginning with a period. By using the HDOS utility PATCH or the CP/M utility DDT, you can alter the default values for every switch. You can even change the default dictionary and affix table filenames if you wish. See Appendix B for how to use these system programs to patch SPELL.

Switches which act as flags (that is, which activate or deactivate a feature) should be patched with either 0 or 1, depending on whether you wish the flag to be initially "on" or "off". Using that switch in a command line then toggles the flag to the opposite value. For example, ignoring text processor command lines is normally activated and its associated flag is normally 1. The switch "-t" turns it off and resets the flag to 0. If you patch SPELL to reset this flag to 0, SPELL will normally process lines beginning with '.', unless you reverse the flag by including "-t" in your command line.

5.3. Changing the Affix Table

You may add or delete prefixes or suffixes from AFFIXTAB.SPL to customize SPELL for your particular environment. You may notice that many of the words SPELL rejects end in a particular suffix, for example, or that you prefer British versions of words such as "civilised" instead of "civilized." You can use your text editor to inspect and modify AFFIXTAB.SPL. You may add up to about 40 suffixes and 20 prefixes to this file.

AFFIXTAB.SPL contains a list of prefixes, one per line, followed by a blank line, and a list of suffixes. Use lower case only, except for "C" as mentioned below. Order is not important, except that longer prefixes must follow shorter prefixes which contain them; e.g., "under" must follow "un."

Suffixes are represented as "-<characters deleted> + <characters added>," so that "mercy" becomes "merciless" via the application of the suffix "-y+iless,". An uppercase "C" indicates a final consonant, so that the rule "double the final consonant and add 'er'" is encoded as "+C+er." SPELL does not check AFFIXTAB.SPL for errors, so you should be careful in modifying it.

To cause SPELL to accept British spellings, you may add suffixes such as +re, -or+our, and +ise.

5.4. Limitations of SPELL

SPELL is not perfect, and can not detect every error you make. It has three major limitations. It can not check usage. It does not know every English word. And it can occasionally fail to catch an error.

Obviously, SPELL is not a complete text proofreading system. It cannot tell you whether your use of the word "affected" should really be "effected," or whether you typed "rather then" for "rather than." It just spots misspelled words.

Also, SPELL knows common prefixes and suffixes, but has little intelligence about where and how to use them. If you failed grammar school and use "buyed" instead of "bought," SPELL will happily accept it as past tense of the verb "to buy." If you say "bater" instead of "batter," SPELL will not complain, since its suffix rules allow for the simple addition of "er" as well as the final consonant doubling case. Thus, while SPELL will spot typographical errors easily, it is not a substitute for your English teacher.

SPELL does not know all the words in the language. You may need to add uncommon words that you use frequently, especially proper names. And we may have missed a few relatively common words. (If you run across one that you feel really ought to be in there, let us know. If we agree, we'll add it.)

Because of the method used to encode its dictionary, SPELL will occasionally let a nonexistent word slip through. Since this will generally happen much less frequently than errors in word or prefix/suffix usage, it does not raise SPELL's overall error rate substantially.

The encoding method, explained more fully in Section 6, was chosen in order to store the dictionary in a relatively small disk space, and to provide fast lookup of words. As a result, a few misspelled words in every thousand errors (which will occur in many tens of thousands of words of documents) will not be detected. Adding words to the dictionary increases the likelihood that errors will slip through. The following table shows the estimated frequency of undetected errors with the dictionaries provided, and the effect of adding 1000 words to each dictionary.

Dictionary	Nr. Missed Errors in 1000 Errors	Nr. if 1000 Words Added to Dict'n'ry
DICTNARY.64K	2.2	2.9
DICTNARY.48K	4.7	6.5
DICTNARY.48S	12.	16.

For larger dictionaries, the chance of an error is greater. This means that if you expect to add large numbers of words to your dictionary, you should maintain separate dictionaries for different subject areas, or else add more memory.

5.5. Other Considerations.

In addition to the limitations mentioned previously, SPELL can typically handle only a few hundred unrecognized words at once. (This is because SPELL keeps them all in memory, in alphabetical order.) In processing very large documents, SPELL may terminate the dictionary lookup phase early, announce that its wordlist is full, and enter the disposition phase. Presumably, during this phase you will add some words to the dictionary and correct others, so that another pass through SPELL will find the remaining misspelled words.

This concept of "graceful degradation" has been applied throughout SPELL in its error- and limitation-handling facilities. If SPELL encounters a problem writing the new files during its update phase, for example, it will let you know what happened and list the words on the terminal which you had marked as misspelled. Thus, although your text will not be marked (perhaps because you had it on a write-protected disk), your work with SPELL is not completely lost. Other errors which might occur are, for the most part, self-explanatory and fall into two classes. FATAL errors terminate SPELL completely and send you back to the operating system; WARNING errors indicate that SPELL detected something wrong but can continue operating, although perhaps in a reduced capacity. For example, problems reading the dictionary will cause a FATAL error, while an unrecognized switch on the command line will just generate a WARNING while SPELL continues.

5.6. SPELL's Dictionaries.

SPELL is provided with three dictionaries: DICTNARY.64K, for machines with a full 64K of memory, and DICTNARY.48K and DICTNARY.48S for smaller machines. The 64K dictionary contains about 18,500 root words. However, this number does not really indicate the dictionary's size, since it was created from a larger dictionary by removing many words which SPELL's prefix/suffix table allows it to recognize from the remaining root words. Also, the 64K dictionary contains several thousand common proper names, which are helpful when checking documents such as correspondence.

The 48K dictionary is the same as the 64K version, except that all but the most important proper names have been removed, as well as several hundred more words that are derivable using the prefix/suffix table. So you may find that this dictionary complains more often about place and street names. There are about 15,600 words in the 48K dictionary.

The 48S dictionary contains the same words as the 48K dictionary, compressed into a smaller space. Because this reduces accuracy, we recommend that the larger dictionaries be used whenever possible.

6. THEORY OF OPERATION

SPELL is based upon the concept of hash coding, a technique used in computer science for encoding information. Hash coding is applied in SPELL to compress dictionaries of up to 18,500 words into a file small enough to fit into the computer's memory while SPELL is running.

A hash function is simply a function that takes the string of characters making up a word, and computes a number based on those characters. Its usefulness comes from the fact that if the range of numbers is large enough, the likelihood is very small that two different words will have the same hash function number. So instead of storing and comparing words, we can compute a hash function for each word, and store and compare that.

It takes a lot of space to store words as character strings, and a lot of time to compare them to each other. Their hash functions are much easier to work with, and often something can be done with hash functions that simply is not possible with the original, less convenient data. Of course, it is still possible that two hash functions can be the same while the original words were different, so there is some chance of error. But that is the price of the increased efficiency.

To construct SPELL's dictionary, we consider the dictionary file as a long table of bits (where each byte holds 8 bits). Call the number of bits M, and the number of words in the dictionary N. Initially all the bits are set to 0.

A word is entered into the dictionary by hashing it using K different hashing functions, each of which gives the address of a specific bit location within the table. These K bits are all set to 1. In SPELL, we use seven hash functions; i.e., K = 7. After all N words are entered in the dictionary, many of the bits are 1. Some bits have been set to 1 several times.

Now, suppose we want to look up a new word in the dictionary. We simply compute the K hash functions for the new word, and look at the bits at those positions in the table. If any of the bits is a 0, we know that the word was never encoded into the dictionary. If all those bits are 1, we assume that the word is in the dictionary.

Of course, there is a possibility that a word which is really not in the dictionary will produce K hash function numbers that all happen to point to bits that were set to 1. When this happens, the word will be incorrectly reported as being in the dictionary. The probability of this happening can be computed; a close approximation is given by the formula

$$(1 - e-(KN/M))K$$

This number is the chance of an error, and can be reduced by reducing the number of words, and by increasing the size of the hash table. For a given number of words and hash table size, we can pick the number of hash functions K to minimize the chance of error.

Having the complete dictionary in memory allows SPELL to store only root words in the dictionary and strip suffixes and prefixes during word lookups. A given word may have to be looked up more than once depending on how many prefixes and suffixes it contains. SPELL will strip up to one prefix and two suffixes from a word while trying to find a root word that is in the dictionary.

We have checked SPELL against a 42,000 word dictionary and achieved a 96% recognition rate. It appears that SPELL has an effective vocabulary of greater than 50,000 words.

For more information about this hashing technique, see Knuth, Vol. 3 of The Art of Computer Programming, 561-562, and Bloom, Communications of the ACM, 13 (1970), 422-426.

APPENDIX A. HDOS Stand-Alone Mode

In a single-disk HDOS system, it is useful to run large programs like SPELL from non-bootable diskettes. You can execute from a non-bootable disk if your system disk has been placed in stand-alone mode.

To do this, make sure that SET.ABS is on your system disk, and type the command SET HDOS STAND-ALONE. HDOS will respond by recording this mode permanently on the system disk (so that the next boot-up from this disk will automatically leave HDOS in stand-alone mode) and will issue a mysterious message, which you should ignore.

The benefit of stand-alone mode is that you may use the RESET command, which is the equivalent of DISMOUNT followed by MOUNT. In particular, you may type RESET SY0: and replace the bootable diskette with another, which need not contain the full HDOS system. Such a "mini-system" diskette must contain two required files: SYSCMD.SYS and PIP.ABS. The rest of the disk - about 350 sectors - is available for your programs.

If you plan to use any device (such as LP:), you must load the driver with the command LOAD LP: after booting, but before you use the RESET command.

You can switch your system disk out of stand-alone mode by typing SET HDOS NOSTAND-ALONE with your system disk mounted. However, many single-disk users run stand-alone exclusively.

When you reset the system disk in stand-alone mode, HDOS loads its two system overlays into the top of user memory and leaves them there. Thus, the HDOS system is available at all times even though the bootable diskette is not. Unfortunately, the HDOS overlays take up quite a bit of space. So do any device drivers you may load. Therefore, the SPELLing dictionary used must be smaller than the one allowed under normal HDOS operation in order to leave these overlays intact. This is why we have provided a special DICTNARY.48S for stand-alone operation under 48K HDOS.

APPENDIX B. How to Patch SPELL

The file PATCHES.SPL on your SPELL distribution disk contains the addresses in SPELL.COM (or SPELL.ABS) which you may patch to change the program's defaults as described in Section 5.2. Determine what new values you wish to enter, and the locations at which they are to be entered. Then patch the SPELL program as follows:

CP/M Users:

Use the CP/M utility program DDT.COM to patch SPELL. After typing DDT SPELL.COM, you should get a "-" prompt in return. You want to set memory values, so type an "s" followed by the hex location that you wish to change. Be sure the old value matches that listed in PATCHES.SPL, and then type the new byte value followed by a return. A return alone leaves the value unchanged and advances to the next location, and a "." followed by a return exits the "set" mode. When you have finished patching, type a control-C to DDT to get back to CP/M, then type SAVE 50 NEWSPELL.COM to save the new version to disk. After testing NEWSPELL to make sure it works properly, you can delete the old version and rename the new one.

HDOS Users:

Use the HDOS utility program PATCH.ABS to patch SPELL. Copy it from your system disk to your working SPELL diskette temporarily, and type the command PATCH. At the filename prompt, type SPELL, and at the address prompt, type the (split octal) address you wish to change. PATCH will reply with the value stored in that location. Be sure the old value matches that specified in the PATCHES.SPL file. Type in the new value in octal, followed by a return, or just a return to leave that location unchanged. PATCH will then advance to the next location. Typing a control-D returns to the address prompt; typing another control-D closes the file. At this point the patches have been entered into the file. A third control-D exits to the HDOS monitor.