

MicroGenesis Spell Checker V2.0
"C C++ PASCAL and Visual Basic"
Copyright 1990, 1991 by Jeff Heaton
MicroGenesis Software
P.O. Box 25534
St. Louis, Mo 63123
BBS: (314) 638-5204

Table Of Contents

1. Introduction

- 1.1 Description of MGSPELL
- 1.2 What Is Shareware?
- 1.3 Registration
- 1.4 Warranty
- 1.5 Order Form

2. Using MGSPELL

- 2.1 Installing MGSPELL
- 2.2 How MGSPELL works
- 2.3 Using MGSPELL with C/C++
- 2.4 Using MGSPELL with Visual Basic
- 2.5 Function Reference

3. Misc./Support

- 3.1 Where Did All of These Words Come From
- 3.2 Contacting The Author/Trademarks/Version History
- 3.3 Version History
- 3.4 Trademarks

1. Introduction

1.1 PROGRAM DESCRIPTION

MGSPELL is a DLL that allows programmers to easily include spell checking to any Windows program that they are developing, without the need for them to take the time to create the necessary compression/searching routines necessary for spell checking. This program is not a stand alone spell checker, but rather a DLL for use with Turbo C, Borland C++, MicroSoft C, Visual Basic, Pascal, or any other Windows language which supports DLLs. MGSPELL comes with a modifiable dictionary containing either 59,000 or 119,000 words and offers such features as spelling suggestions and word list maintenance. The routines are built to be straight forward and easy to use in your own programs.

Some of the features of MGSPELL include:

- A 119,000+ word dictionary compressed to around 400k for use in programs that demand the biggest dictionary available.

- A 59,000+ word dictionary compressed to around 230k for use in a program that will fit on a 360k floppy drive.

- Windows DLL, for maximum Windows compatibility.

- Fast disk-based dictionary takes only around 20k of ram no matter what dictionary you choose.

- Routines to suggest words for incorrect spellings.

- Completely royalty free. Pay once and use in as many of your programs as you like.

- Marketed as Shareware, not cripple ware: Try everything before you buy! (except source code).

- Support for both registered and unregistered user by a variety of means.

1.2 A General Definition Of ShareWare

If you are unfamiliar with the shareware software marketing system this section describes it briefly.

Shareware distribution gives users a chance to try software before buying it. If you try a Shareware program and continue using it, you are expected to register. Individual programs differ on details -- some request registration while others require it, some specify a maximum trial period. With registration, you get anything from the simple right to continue using the software to an updated program with printed manual.

Copyright laws apply to both Shareware and commercial software, and the copyright holder retains all rights, with a few specific exceptions as stated below. Shareware authors are accomplished programmers, just like commercial authors, and the programs are of comparable quality. (In both cases, there are good programs and bad ones!) The main difference is in the method of distribution. The author specifically grants the right to copy and distribute the software, either to all and sundry or to a specific group. For example, some authors require written permission before a commercial disk vendor may copy their Shareware.

Shareware is a distribution method, not a type of software. You should find software that suits your needs and pocketbook, whether it's commercial or Shareware. The Shareware system makes fitting your needs easier, because you can try before you buy. And because the overhead is low, prices are low also. Shareware has the ultimate money-back guarantee -- if you don't use the product, you don't pay for it.

1.3 Licensing/Registration

If you going to be releasing programs with MGSPELL code in them you must register it with MicroGenesis Software prior to releasing your program.

By registering for \$50, you will receive:

Permission to release MGSPELL.DLL with your executable file.

You may NOT release MGSPELL as a stand alone DLL. MGSPELL must be used with your own program.

You also get the complete source code to MGSPELL. This allows you to recompile the libraries that make up MGSPELL. This source code is for your own use. You may not release the source code in any for other than as a compiled .DLL(and then only in conjunction with your own programs) or compiled as part of your programs .exe.

You do not have to credit MicroGenesis Software in the documentation or executable file if your program is copyrighted. If your program is public domain please include the line:

Portions Copyright (C) 1990,91,93 by MicroGenesis Software

1.4 Limited Warranty

Users of MGSPELL must accept this disclaimer of warranty: "MGSPELL is supplied as is. The author disclaims all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The author assumes no liability for damages, direct or consequential, which may result from the use of MGSPELL."

MGSPELL is a "shareware program" and is provided at no charge to the user for evaluation. Feel free to share it with your friends, but please do not give it away altered or as part of another system. The essence of "user-supported" software is to provide personal computer users with quality software without high prices, and yet to provide incentive for programmers to continue to develop new products. If you find this program useful and find that you are using MGSPELL and continue to use MGSPELL after a reasonable trial period, you must make a registration payment of \$20 to MicroGenesis Software.

You are encouraged to pass a copy of MGSPELL along to your friends for evaluation. Please encourage them to register their copy if they find that they can use it. All registered users will receive a copy of the latest version of the MGSPELL system.

You may not, under any circumstances, release a program containing MGSPELL code before you register. For all programs that you compile MGSPELL directly into, and only release the compiled version you need not contact me. No royalties are charged, and you are free to use it as you choose just so long as it is ONLY released in compiled form (unless you have a limited vendor license). With any questions please free to contact me.

ORDER FORM:MGSPELL V2.0 (C/C++ DLL)

NAME : _____
COMPANY: _____
ADDRESS: _____

(OPTIONAL)

PHONE : _____
WHERE YOU GOT THIS FILE: _____
NetWork Address : _____
Are you a member of ASP: Y N

Source Code MGSPELL V1.4 Registration : \$50 ___
Rush Order(send via COMPUSERVE) : +\$10 ___
NetWork Address to send to _____
Total Enclosed : _____

Disk Format: 5 1/4" ___ 3 1/2" ___

The product being registered is MGSPELL V1.4.
You will receive the latest version.

Send check or money order to:

MicroGenesis
P.O. Box 25534
St. Louis, Mo

Make check payable to: Jeff Heaton

NOTE: Rush orders will ONLY be EMAILED over Compuserve.

2. Using MGSPELL

2.1 Installing MGSPELL:

To conserve disk space the dictionaries have been compressed with SDAs(self-dissolving archives). This is mainly so that they will fit on the 360k/720k disks that I usually release them on. To bring the files out of the SDA simply change to the drive or sub directory that you want the files onto. For example to run the demo program, assuming your using A: to install MGSPELL to a hard drive C:.

```
C:  
MD \MGSPELL  
CD \MGSPELL  
COPY A:DEMO.EXE  
<INSERT LARGE dictionary disk>  
A:\BIGDICT  
DEMO
```

In order to use the functions included with the MGSPELL.DLL you must include MGSPELL.LIB into your project or MAKE file. Then include MGSPELL.H in all of your functions which use MGSPELL functions.

2.2 How MGSPELL Works

MGSPELL uses two dictionary files to hold the words. MGSPELL.DIC and MGSPELL.USE, these are respectively the main, and user dictionary. The main dictionary is stored in compressed form, and can be searched much faster than the user dictionary. All words that are added are put in the user dictionary. This has two advantages: #1 no changes to the dictionary are permanent, MGSPELL.USE may be deleted at any time to clear the user dictionary, and leave the main dictionary unaltered. #2 The dictionary need not be recomposed each time a word is added.

The only letters allowed by MGSPELL are all 26 capital letters, and the apostrophe ('). Also any word made up of only two letters is assumed to be a abbreviation and is counted as correct without any disc access. The limit on word length is 30 characters, which should be sufficient for the English language.

MGSPELL has been tested with the following compilers:

Turbo C++ V1.0

Turbo C V2.0

Borland C++ V2.0

Borland C++ V3.0

Borland C++ V3.1

MicroSoft Visual VC 1.0

MicroSoft Visual Basic 3.0

2.3 Using MGSPELL With C/C++

The MGSPELL DLL integrates perfectly with any C/C++ compiler that supports DLL's. In order to make use of the DLL you must give your compiler access to the DLL functions. To do this you should include the MGSPELL.LIB file into your project or make file. This small library contains the function links to the actual MGSPELL.DLL.

The LIB file included with MGSPELL was created with Borland C++. It is compatible with Borland C++ and Visual C++. It should be compatible with all others. If, however, your compiler does not accept the MGSPELL.LIB file you should rebuild it with IMPLIB. The following command will accomplish this:

```
IMPLIB MGSPELL.LIB MGSPELL.DLL
```

To actually use the spell checking commands you should include "MGSPELL.H". This file contains the function prototypes for all MGSPELL commands.

2.4 Using MGSPELL With Visual Basic

The MGSPELL DLL integrates perfectly with any Visual Basic compatible compiler that supports DLL's.

In order to access the functions in the DLL the appropriate Declare statements must be included in the declarations section of the general object. This will allow you to access the functions. The following is a list of the Declare statements that must be included:

```
Declare Sub reg Lib "MGSPELL.DLL" (ByVal code As String)
Declare Function spell_check Lib "MGSPELL.DLL" (ByVal word$) As Integer
Declare Function word_count Lib "MGSPELL.DLL" () As Long
Declare Sub suggest_word Lib "MGSPELL.DLL" (ByVal code As String)
Declare Sub suggestion Lib "MGSPELL.DLL" (ByVal s1 As String, ByVal s2
    As String, ByVal s3 As String, ByVal s4 As String, ByVal s5 As
    String, ByVal s6 As String, ByVal s7 As String, ByVal s8 As
    String, ByVal s9 As String, ByVal s10 As String)
Declare Function add_word Lib "MGSPELL.DLL" (ByVal word$) As Integer
Declare Function dump Lib "MGSPELL.DLL" (ByVal word$) As Integer
Declare Sub Import Lib "MGSPELL.DLL" ()
```

Sometimes Visual Basic will mistake a function all for an array. If this happens you should use the call keyword. You should place the Call keyword in front of your call. For example:

```
call Suggestion(s1,s2,s3,s4,s5,s6,s7,s8,s9,s10)
```

For even more information on how to use MGSPELL you should look over the demo which is included.

2.5 Function Reference

C: void reg(char *str);
VB: Sub Reg(ByVal str)

This function must be the first one called in the DLL. The str parameter holds your registration string(which is given to you when you register MGSPELL, it is located on the disk label). If you are using an unregistered version, then you may pass this function anything you wish as str.

C: int spell_check (char *word);
VB: Function spell_check(ByVal word\$) As Integer

This routine is the main access point to mgSPELL. It will scan the dictionary for "*word". If the word is located zero is returned, if the word is not in the dictionary a one is returned.

C : void add_word(char *word);
VB: Sub add_word(ByVal word\$)

This routine will add a word to the user dictionary. Disc space is really the only limit on how many words the dictionary can hold (the programs own "limit" is somewhere around four 2 trillion words). However the speed of MGSPELL is affected by how many words are in the user dictionary. If there is a large number "import" should be used to merge the user dictionary with the main dictionary.

C : void suggest_word(char *word)
VB: Sub suggest_word(ByVal word\$)

This routine will update the word suggest list (see below). It is usually called just after spell_check determines a word is not spelled correctly.

C : void suggestion(char *s1,char *s2,char *s3,char *s4,char *s5,char *s6,char *s7,char *s8,char *s9, char *s10);
VB: Sub suggestion(ByVal s1\$,ByVal s2\$,ByVal s3\$,ByVal s4\$,ByVal s5\$,ByVal s6\$,ByVal s7\$,ByVal s8\$,ByVal s9\$,ByVal s10\$)

This routine will retrieve the suggestion list after suggest word has been called. The parameters s1-s10 will be modified to contain up to ten suggestions. If fewer than

ten suggestions are possible then this function will blank the last strings that were not filled.

```
C : long word_count(void) ;  
VB: Function word_count() as Long
```

This routine returns, in long int form, the number of words in the dictionary.

```
C : void dump(char *name) ;  
VB: Sub dump(ByVal name$)
```

This routine dumps the whole dictionary to the path pointed to by "*name". The file will be in ASCII format with each line ended by a CR/LF. This routine will take awhile to execute(1 minute, ten seconds on my 10mghtz 80286).

```
C : void import(void) ;  
VB: Sub import()
```

This routine is used to merge the user and main dictionaries. Since this means repackaging the entire dictionary this is not the fastest routine in the world. It takes one minute and forty seconds on my 10mghtz 80286. This routine should be called to make added words (which go to the user dictionary) permanent additions to the dictionary. This routine will also accept ascii dictionaries to be merged into the main dictionary. However they MUST be in the form that DUMP issues files(SORTED and one word per line). You simply name your ascii text file MGSPELL.USE(or whatever you have named your user dictionary) and run import. I found this to be the fastest way to build the dictionary from the separate ascii files I received to create the dictionary.

1. Misc./Support

3.1 Where The Dictionary Came From

This is just a description of where the words in this dictionary came from, it is included as background information only. I have had a few registered users of previous versions of MGSPELL ask me if I was actually crazy enough to sit here and type out THAT many words, the answer is no. Like as many tedious tasks as I can I let my computer do it. Up to the 60,000 word version the words were typed by probably thousands of people. Whenever I logged onto a local BBS, or compuserve I would open the capture buffer and let it fill with whatever text came across my terminal. I even wrote a little utility that could go through disks and pull out every bit of textual data it could find. I then wrote another software program that would sort these huge text files(some over 1-2 meg long) and remove any duplicate letters. I had other programs that would run through the list and add extra words, basically adding a ED, ING, S, ES onto EVERY word in the text files. I then had a dictionary with countless mis-spelled words in it. So I located a shareware library, much like my own, only it was written in PASCAL and wrote a software program in pascal spell check my own dictionary against a known good dictionary. I did not do this to every file, some of the files I knew had no errors because they had been draw from news services and encyclopedias, which allowed my dictionary to surpass in size the dictionary of its "checker program."

This most recent addition of words that brought it up to 119,000 words came from a source of words that someone provided me with. So I now provide two dictionaries, the original MGSPELL dictionary that was created from words pulled from compuserve. And a 2nd one that has the much larger dictionary merged into it.

By the way if you are looking for a really good dictionary program for PASCAL the one that I used(and registered) is SPELL v1.5, by Acropolis Software 4620 Hazel Ave, Fair Oaks, Ca 95628. It was a great help in making sure no mis-spelled words entered this dictionary.

3.2 CONTACTING THE AUTHOR

Suggestions and questions regarding any MicroGenesis product are welcome by any of the following means:

CompuServe: 76476,1701

US POSTAL : MicroGenesis

P.O. Box 25534

St. Louis, Mo 63125

BBS : (314) 638-5204

3.3 Version History

V1.0/1.1 First release, 21,000+ words.

V1.2 Speed improvements, 59,000+ words.

V1.3 Bug Fix in Suggest Routine.

V1.4 Switch to a completely ShareWare version
Addition to 119,000+ words.

v2.0 Windows supported DLL for use by C Pascal and Visual Basic

3.4 trademarks

IBM

International Business Machines

MS-DOS,Quick C

MicroSoft Corporation

Borland C,Turbo C, Turbo C++ Borland International

CompuServe

CompuServe Information Service