



ISPF Edit Spelling Checker

By Lionel B. Dyck

Do you ever document your code or procedures using ISPF Edit (ok—the answer is that we all do and if we do not then we should)? After coding for hours or days or weeks in some programming language using strange abbreviations, contractions, and other variations of correct spelling are you able to do a mental reset and actually spell correctly? For some of us the answer to that last question is no. We need a little assistance in the spelling department, which is why spelling checkers were invented by programmers—because they needed to have their spelling checked.

For the mainframe user there have been spelling checkers available as ISPF Edit commands that used an external program, either Script (DCF) or SAS, to check the spelling and report the results. These spelling checkers are great if you have Script or SAS in your shop. The rest of us had to rely upon a tattered copy of Webster's that we have had for years (or in many cases non-tattered but in near mint condition because we did not use it).

Since May 1, 2004 there is now a free, open source (under the GPL License), spelling checker for ISPF Edit that comes with a list of over 10,000 words and works fairly well. It will never rival the integrated spelling checker that you have on your PC based word processor but it should keep you from looking foolish when you create text on the mainframe that others will be looking at.

The initial reaction from early users is that they like the tool. In my own use of it to check some of the documentation that I have included in some of the packages that I have posted (such as XMITIP) I have found a few corrections that had not previously been spotted.

This tool is called SPELLC, short for Spell Checker, and is written entirely in REXX using ISPF and ISPF Edit services to provide a user-friendly interface. This spelling checker was written to be used with two other packages that can be found on the web site where SPELLC can be found, specifically the News ISPF Dialog and the XMITIP package (the ISPF interface). Both of these packages enable the user to create a text file to either post in a news database or to be used as the text of an electronic mail message. In both cases, spelling mistakes can be very embarrassing.

FIGURE 1: SPELLC COMMAND

```
EDIT      SYSJOB.SPELLC.PDS(WGOOD) - 01.02      Columns 00001 00000
Command ==> spellc                               Scroll ==> CSR
***** Top of Data *****
000001 There are known knowns. These are things we know that we know.
000002 There are known unknowns. That is to say, there are things that we
000003 know we don't know. But there are also unknown unknowns. There are
000004 things we don't know we don't know.      quote from Donald Rumsfeld
***** Bottom of Data *****
```

SPELLC is invoked as an ISPF Edit command as it is designed as an ISPF Edit Macro. The user may provide up to two optional positional parameters for the starting and the ending records to be checked. There parameters can be either line numbers or line labels (see FIGURE 1).

Since this tool lets you specify the range of records for spell checking, it is easy to use for spell checking the comments within your programs.

Next, the user sees an ISPF popup panel that allows the user to enter the data set name of a personal dictionary, really a list of words, which will be added to the public dictionary. The format of the personal dictionary is that of a sequential data set, or member of a partitioned data set, consisting of a series of records with words separated by blanks.

On this popup panel the user may override the starting and ending line numbers by entering actual line numbers or line labels (see FIGURE 2).

At this point pressing enter will cause the spell checking process to begin. During the spell checking another popup panel will be displayed showing the user some information about the progress of the spell checking (see FIGURE 3). This panel includes a constantly changing field containing the word that is being checked. You have to look fast as this field changes quickly. It displays the current count of invalid words, the current count of words checked, the current record being checked and the record number of the last record to be checked. Also

When the spell checking is completed the user will be informed either that (see FIGURE 4 for an example where all words are valid and FIGURE 5 for an example with misspelled words):

- ▼ All words passed verification
- ▼ The number of misspelled words along with a list of the misspelled words.
 - ▶ All records with no errors will be hidden so that those records with misspelled words will be easier to find.
 - ▶ Under each record with a misspelled word will be an ISPF Edit message line with asterisks under the misspelled word.

Once the spelling check has completed, some words may be incorrectly flagged as misspelled words. To solve this problem for future spelling checks the user can use SPELLCA command to add the word to their personal dictionary. This command was written by Russell Nesbitt and works by entering SPELLCA on the ISPF Edit command line and then moving the cursor to the word to be added—then pressing the enter key.

If the user does not specify a personal dictionary in the SPELLC prompting panel, then SPELLCDF (the local customization REXX Exec) will be allocated on the first use of SPELLCA. In this case, SPELLCDF must have been updated to define the format for the user's personal dictionary data set name.

The following ‘rules’ are used during the spelling check:

1. Each line of text in the ISPF Edit (or View) data set is read from top to bottom
2. Each line will have the characters ;?()*=<>[]! translated to blanks
3. Each word (a string of characters without an intervening blank) is then converted to upper case for checking
4. Words that begin with numbers are not checked
5. The word is tested and if found then checking continues with the next word, otherwise:
 1. Words ending with S, 'S, and 'D are checked as if the S, 'S, and 'D are not there.
 2. Words with these other characters, 0123456789_@, are ignored.
6. A message is inserted after each line with an invalid word indicating the misspelling on the prior line
7. Upon completion the invalid words are displayed in sorted sequence at the top of the data set in ISPF message format.

The installation of this tool is easy. First, you need to download the package from <http://www.lbdsoftware.com> (look on the ISPF Tools page for the most current version). The package is distributed in zip format so you will need to unzip the file on your PC and then upload to your z/OS system the **spellc.xmit** file.

The upload should be performed in binary mode and into a data set with a DCB of RECFM=FB and LRECL=80. The resulting file needs to be processed using the TSO RECEIVE command (see the read.me file that is included in the package). This will create a partitioned data set on your z/OS system.

Within the distribution PDS are the following members of interest:

\$DOC brief documentation on how to install and use SPELLC

FIGURE 2: INITIAL PROMPTING PANEL

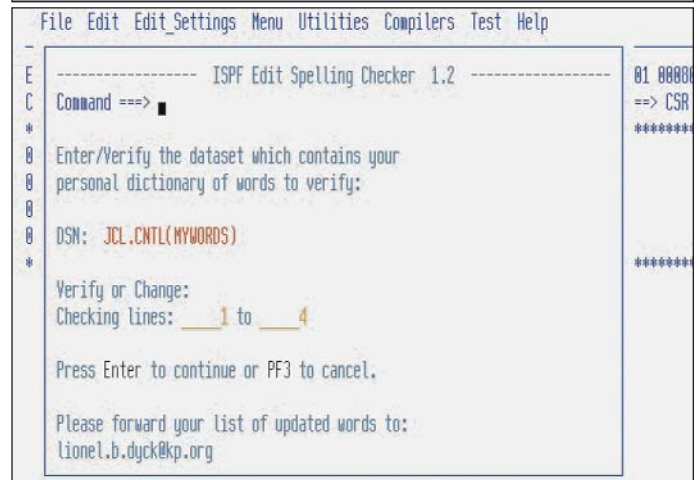


FIGURE 3: PROGRESS PANEL

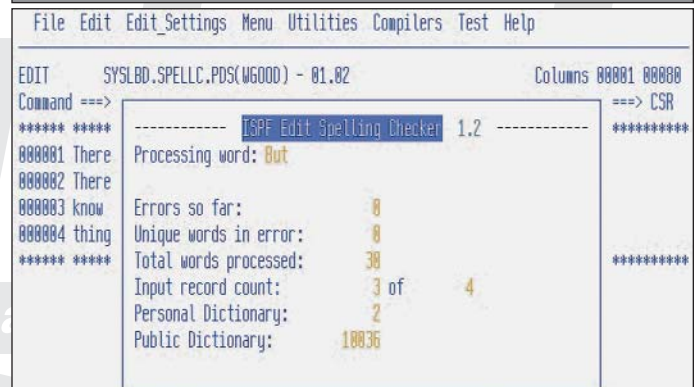
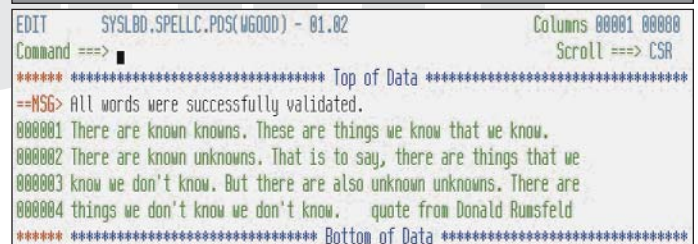


FIGURE 4: SAMPLE GOOD RESULTS



CHANGES	a change history of the package
EXEC	the REXX Exec library in TSO Transmit format
PANELS	the ISPF Panel library in TSO Transmit format
RECEIVE	a REXX exec that can be used to issue the TSO RECEIVE command on the EXEC and PANELS Members.
WBAD	a sample member with misspelled words
AGOOD	a sample member with all correctly spelled words
WORDS	a starting public dictionary of words

To use the RECEIVE member to rebuild the EXEC and PANELS members the simplest technique is to use ISPF 3.4 to list the distribution PDS and then open the data set using either Browse or Edit. Then enter EXEC next to the RECEIVE member and follow the prompts.

The next step is to create the public dictionary data set for your installation. The data set should be allocated using RECFM=VB and LRECL=84. Then copy the distribution PDS member WORDS into

this new data set. Make sure that this data set can be accessed in read mode by all of your intended users.

Once you have done this then you are ready to update the installation specific information in the SPELCDF REXX Exec. Within this member, you will define the e-mail address that you want your users to use to share correctly spelled words that should be added to the public dictionary. Then define the name of the public dictionary. You can also change the alphabet characters used to validate the first character of a word, the list of special characters that will be translated to blanks before a line is processed, and the list of special characters that will be translated to blanks within a word. The last thing you will define, or set to null, is a data set name to be used by SPELLCA to create a personal dictionary data set if the user does not specify one in the first SPELLC popup panel.

At this point you are ready to copy the EXEC and PANELS libraries created from running the RECEIVE member into libraries that your users have access to in the TSO LOGON Procs. The EXEC library should be copied into a library in either the SYSEXEC (preferred) or SYSPROC concatenation and the PANELS library into a library in the ISPLIB concatenation.

Now try it out. In the distribution PDS are members WBAD and WGOOD so you can see how SPELLC works with two small text files. Then try it on some of your data sets and when you are ready tell your users about it.

As the author of this package, I am interested in any suggestions to improve it as well as any bugs that you find. If you add words to the public dictionary, please also share those so that others will benefit from those additions. You can find SPELLC, and many other useful packages, at <http://www.lbdsoftware.com>. My e-mail is lionel.b.dyck@kp.org.

FIGURE 5: SAMPLE MISPELLED RESULTS

```
EDIT      SYSLBD.SPELLC.PDS(WBAD) - 81.00      Columns 00001 00000
Command ==>                                     Scroll ==> CSR
***** ***** Top of Data *****
==MSG> 5 invalid words found in 5 locations
==MSG> within SYSLBD.SPELLC.PDS(WBAD)
==MSG> A total of 34 words were checked.
==MSG> The following are the invalid words.
==MSG> > MISSPELLED MMBER SPELLC THA THS
000001 This mumber has a few misspelled words so tha this member can
==MSG> *** ***** ***
-- -- -- -- -- 1 Line(s) not Displayed
000003 invalid words using the spellc (which is an invalid word) command.
==MSG> *****
***** ***** Bottom of Data *****
```