spell/400 - spell checker for the System i

SPELL/400

Version 4 Release 2

www.spell400.com

Black and Blue Software Ltd.

Contents

1 Using Spell/400

1.1 Overview

1.1.1 Overview

Spell/400 is a spelling correction aid, comparable to those found in PC word processor packages. Given some text, it can check the spelling against several dictionaries and prompt the user to change the text, or select from the alternatives offered.

Spell/400 runs on any ’green screen’ or emulator, without requiring any PC component. There is also a ’graphical’ look which offers a ’point and click’ mouse and button interface.

Spell/400 V1 was written in 1997 and V3 was a rewrite that took Spell/400 beyond checking strings and allowed it to ’see’ and correct what was on the screen. V3 has developed with several UI and audit improvements.

1.1.1.1 Comparison to a Personal Computer

Most people are familiar with spell checkers found in PC word processors like Office. In many ways Spell/400 works and acts exactly the same...

Similarities

- ▷ Graphical pop-up window
- ▷ Each spelling mistake lists appropriate suggestions that can be selected
- ▷ Spelling mistakes (and corrections) are highlighted on screen to show the context
- ▷ Mouse control to click on suggestions or buttons
- ▷ Words can be retyped or simply ignored
- ▷ New words can be added
- ▷ Each user can use their own personal dictionary

...however because of the multi-user nature of the System i , it also offers a few features not generally found:

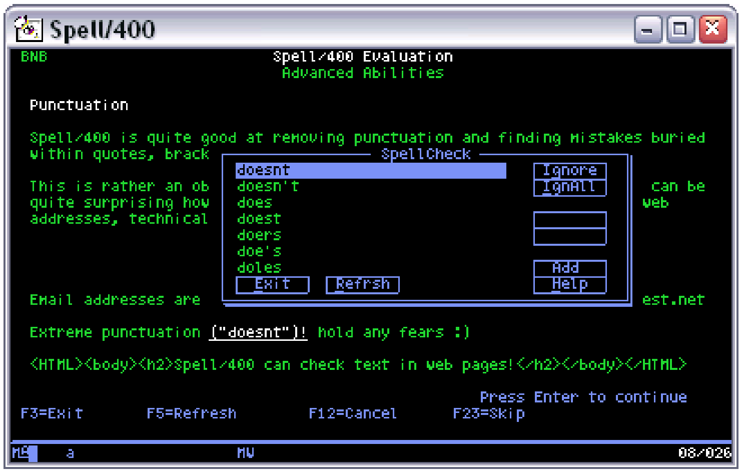
Improvements

- ▷ Standard Fn keys to add, change or ignore mistakes
- ▷ Several dictionaries can be used at once - for instance an English dictionary, an application dictionary and a personal dictionary.
- ▷ Dictionaries can be checked by administrators - all words can be held back before they are approved for general use.
- ▷ Dictionaries can be built from existing database fields - thus a dictionary of customer names, locations and addresses could be built to enable spell-checking of forms with customer addresses within.

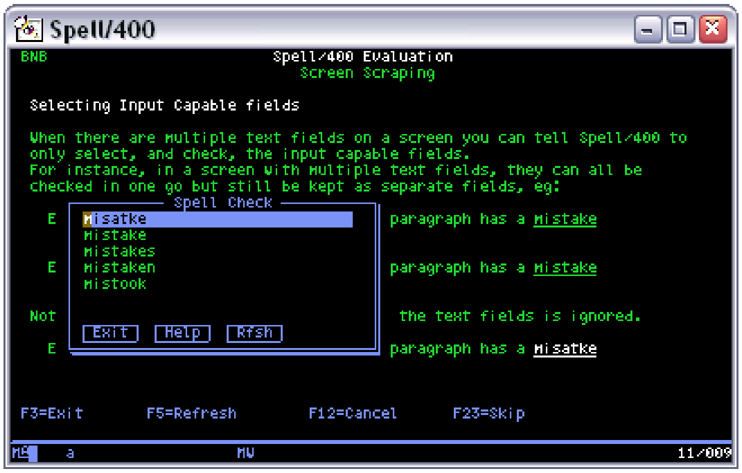
1.1.1.2 What the user sees

As an example, here are some screen prints from the evaluation program:

Client Access screen (using MS Windows)

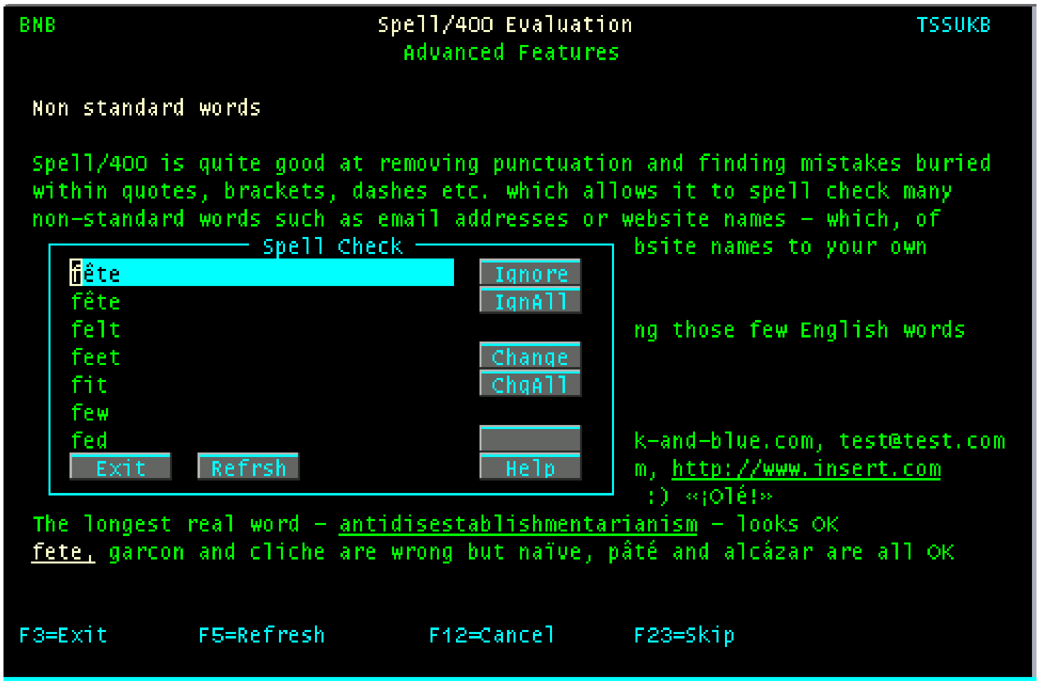


```
BNB                          Spell/400 Evaluation
                              Advanced Abilities

 Punctuation

 Spell/400 is quite good at removing punctuation and finding mistakes buried
 within quotes, brack ─────────────── SpellCheck ──────────────
                     │ doesnt                          Ignore │
 This is rather an ob│ doesn't                         IgnAll │  can be
 quite surprising how│ does                                   │ web
 addresses, technical│ doest                                  │
                     │ doers                                  │
                     │ doe's                                  │
                     │ doles                           Add    │
                     │  Exit    Refrsh                 Help   │
 Email addresses are └────────────────────────────────────────┘ est.net

 Extreme punctuation ("doesnt")! hold any fears :)

 <HTML><body><h2>Spell/400 can check text in web pages!</h2></body></HTML>

                                                 Press Enter to continue
F3=Exit        F5=Refresh         F12=Cancel      F23=Skip

MB     a                  MW                                        08/026
```



```
Spell/400
BNB                         Spell/400 Evaluation
                              Screen Scraping

 Selecting Input Capable fields

 When there are multiple text fields on a screen you can tell Spell/400 to
 only select, and check, the input capable fields.
 For instance, in a screen with multiple text fields, they can all be
 checked in one go but still be kept as separate fields, eg:
              Spell Check
   E   Misatke                          paragraph has a mistake
       mistake
       mistakes
   E   mistaken                         paragraph has a mistake
       mistook

 Not                                     the text fields is ignored.
       Exit   Help   Rfsh
   E                                    paragraph has a Misatke


F3=Exit       F5=Refresh        F12=Cancel       F23=Skip

MS     a             MW                                                 11/009
```

Client Access screen (using X-Window)



```
BNB                         Spell/400 Evaluation                          TSSUKB
                              Advanced Features

 Non standard words

 Spell/400 is quite good at removing punctuation and finding mistakes buried
 within quotes, brackets, dashes etc. which allows it to spell check many
 non-standard words such as email addresses or website names - which, of
   ┌──────────── Spell Check ──────────┐ bsite names to your own
   │ fête                      Ignore  │
   │ fête                      IgnAll  │
   │ felt                              │ ng those few English words
   │ feet                      Change  │
   │ fit                       ChgAll  │
   │ few                               │
   │ fed                               │ k-and-blue.com, test@test.com
   │ Exit    Refrsh            Help    │ m, http://www.insert.com
   └───────────────────────────────────┘  :) «¡Olé!»
 The longest real word - antidisestablishmentarianism - looks OK
 fete, garcon and cliche are wrong but naïve, pâté and alcázar are all OK


F3=Exit       F5=Refresh        F12=Cancel       F23=Skip
```

1.1.2 Evaluation

Evaluating software should be a quick process, with no searching through manuals or configurations. So we've designed the step-by-step evaluation program which takes you through the basic options immediately. Within a few minutes you'll have a good idea of what makes Spell/400 so unique and whether it fits into your applications.

Evaluation dictionary

Spell/400 has a set of dictionaries for different purposes, but the evaluation is only shipped with a small dictionary of English words.

1.1.2.1 Step by step evaluation

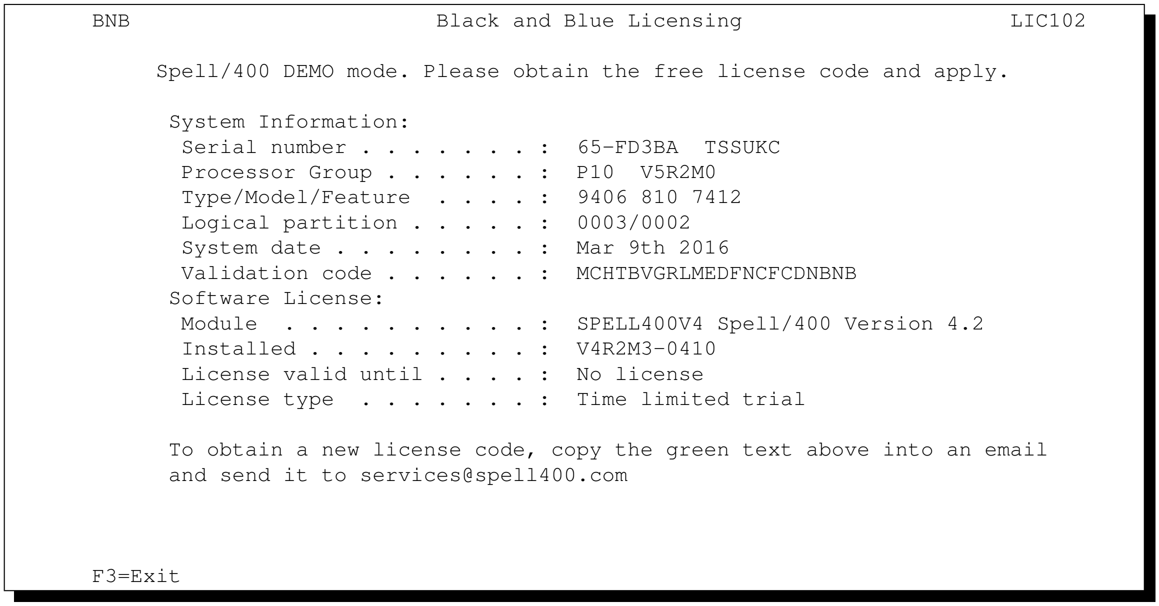
All Spell/400 commands can be accessed from the main menu, which is stored in QUSRSYS so it's accessible from any command line. To access the main menu:

GO SPELL400

If the menu isn't found try GO SPELL400/SPELL400 (note however that this could mean that the software did not complete installation sucuccessfully).

Demo license

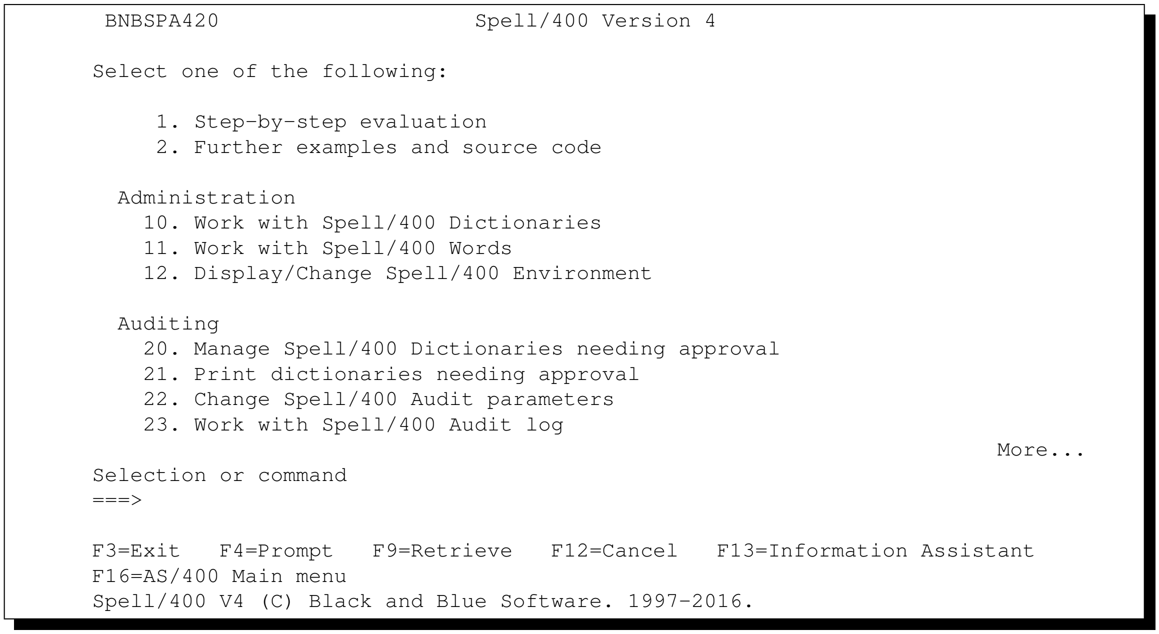
The first time Spell/400 is invoked the Demo license screen will be displayed:



```
BNB                        Black and Blue Licensing                     LIC102

     Spell/400 DEMO mode. Please obtain the free license code and apply.

      System Information:
       Serial number . . . . . . . :  65-FD3BA  TSSUKC
       Processor Group . . . . . . :  P10  V5R2M0
       Type/Model/Feature  . . . . :  9406 810 7412
       Logical partition . . . . . :  0003/0002
       System date . . . . . . . . :  Mar 9th 2016
       Validation code . . . . . . :  MCHTBVGRLMEDFNCFCDNBNB
      Software License:
       Module  . . . . . . . . . . :  SPELL400V4 Spell/400 Version 4.2
       Installed . . . . . . . . . :  V4R2M3-0410
       License valid until . . . . :  No license
       License type  . . . . . . . :  Time limited trial

      To obtain a new license code, copy the green text above into an email
      and send it to services@spell400.com



F3=Exit
```

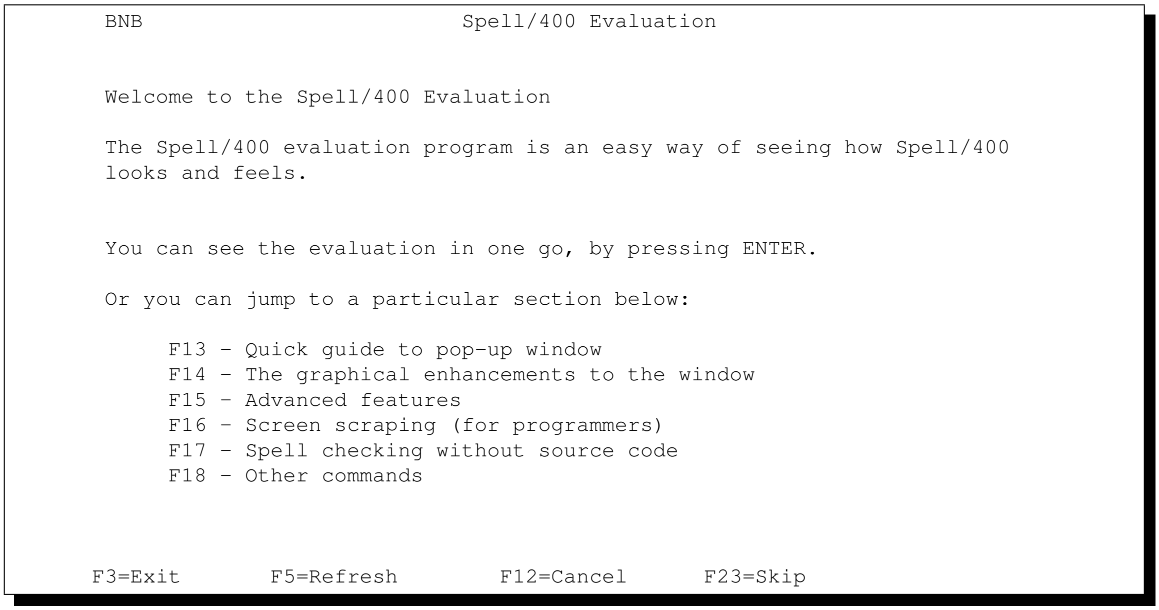
The demo license allows the spell checker to work identically to the licensed version, except the changes are never written back to your program. This can be easily rectified with an evaluation license key. Simply send a text copy/paste of the screen to services@spell400.com.

On pressing F3 to exit the license screen, the actual menu can be displayed.



```
 BNBSPA420                    Spell/400 Version 4

Select one of the following:

     1. Step-by-step evaluation
     2. Further examples and source code

  Administration
    10. Work with Spell/400 Dictionaries
    11. Work with Spell/400 Words
    12. Display/Change Spell/400 Environment

  Auditing
    20. Manage Spell/400 Dictionaries needing approval
    21. Print dictionaries needing approval
    22. Change Spell/400 Audit parameters
    23. Work with Spell/400 Audit log
                                                                   More...
Selection or command
===>

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=Information Assistant
F16=AS/400 Main menu
Spell/400 V4 (C) Black and Blue Software. 1997-2016.
```

The step-by-step evaluation is option 1:



```
 BNB                         Spell/400 Evaluation


 Welcome to the Spell/400 Evaluation

 The Spell/400 evaluation program is an easy way of seeing how Spell/400
 looks and feels.


 You can see the evaluation in one go, by pressing ENTER.

 Or you can jump to a particular section below:

      F13 - Quick guide to pop-up window
      F14 - The graphical enhancements to the window
      F15 - Advanced features
      F16 - Screen scraping (for programmers)
      F17 - Spell checking without source code
      F18 - Other commands



F3=Exit       F5=Refresh        F12=Cancel      F23=Skip
```

1.1.2.2 After the evaluation

After completing the step-by-step evaluation, you should have an idea of what to expect from Spell/400. There are a few more commands to be investigated, and you can play around with different dictionaries and create your own ones. Unfortunately you'll have to learn a bit of the administration and configuration options before going much further.

The evaluation copy allows you to use the software in any way for a few weeks, and then it gives another week with 'nag' screens. After that it stops and the evaluation is over, or at least until you contact us for a license code.

1.1.2.3 Extended Evaluations

If you have reason to extend the evaluation period, let us know and we can provide temporary license keys. Please allow up to 48 hours for receipt of a key.

To receive a new temporary code, follow the procedure in the licensing section on page 21.

1.2 User Interface

1.2.1 The suggestions window

There are three suggestion window formats - Graphical, Minimal and Keyboard. The Graphical format is designed for Client Access with mouse control, the Keyboard format is for dumb terminals and the Minimal format is somewhere inbetween.

They are all demonstrated in the evaluation.

Most of the keyboard and mouse controls work in all three formats.

1.2.1.1 Keyboard control

The current word

The current word, highlighted above the suggestions, can be edited at any time. If Enter is pressed after editing, a new list is generated. If F9 is pressed then the word is accepted without it being checked.

Suggestions

All suggestions windows show up to 50 suggestions (although it's unlikely to be that many). If the dictionaries are configured correctly, then the most obvious suggestions (based on word usage as well as word similarity) should appear near the top.

The lists can be paged up and down.

Function keys

All windows can use the traditional function keys:

F3 Exit

F5 Refresh the list of suggestions

F6 Add word to the dictionary

This is only available if the dictionary to add words to is specified.

F7 List the dictionaries where suggestions came from

F9 Quick replace

F9 is a 'quick-replace' option, which can make word selection quicker. Rather than the three-stage process of the ENTER key, F9 allows you to select and accept the word that the cursor is over, without further checks. Additionally, if the original word is replaced or altered, F9 accepts the alterations without further checks.

F10 Display spelling analysis

F11 Change window format

F12 Ignore word

F13	Select first suggestion
F14	Select second suggestion
F15	Select third suggestion
F16	Select fourth suggestion
F17	Select fifth suggestion
F18	Select sixth suggestion
F19	Select seventh suggestion
F21	Change all occurrence of the misspelled word with the current word.
F23	Change window format to/from *QUICK
F24	Ignore all occurrences of the word

Cursor and Enter Keys

The Enter key is context-sensitive - it usually means "accept" the word, or press the button, underneath the cursor. When the cursor is at other positions it has different meanings:

- Cursor one column left of a suggestion = F21

  Instead of selecting a word by positioning cursor over a suggestion and pressing Enter, if you move the cursor one space left and press Enter it will be the same as pressing F21=Change All.

- Cursor one row above word = F12

  If you move the cursor one row above the editable region (where the mistake can be typed over) and press Enter it will be the same as pressing F12=Ignore.

- Cursor one column left of word = F24

  If you move the cursor one column left of the editable region (where the mistake can be typed over) and press Enter it will be the same as pressing F24=Ignore All.

- Cursor one row above word and one column to the left = F6

  If you move the cursor one space left and up one row and press Enter it will be the same as pressing F6=Add.

- Cursor outside of window

  Whenever the cursor is outside the window the window will be re-positioned.

1.2.1.2 Mouse control

In graphical and minimal modes, you have the following mouse buttons:

Suggestions Up to 6 suggestions can all be clicked or double-clicked and the misspelled word will be replaced.

<Refrsh> Single click to refresh the list of suggestions (F5)

<Ignore> Single click to Ignore the current word (F12)

<IgnAll> Single click to Ignore all occurrences of the word (F24)

<Change> Single click to change the misspelled word to the current word (F9)

<ChgAll> Single click to change all occurrences of the misspelled word. (F21)

<Add> Single click to add the current word to the ADDWORDS dictionary (F6)

Mouse clicks

If your version of Client Access supports mouse clicks fully, you can also use the following:

Left-click Within the window, this selects the button or word under the mouse. Outside of the window, the window is moved.

Left-double-click Over one of the suggested words, this is the same as F9 or F21. Outside of the window, the window is moved.

Right-click The same as <Ignore>

Right-double-click The same as <IgnAll>

1.2.1.3 Moving the Window

Move the cursor to anywhere outside of the window - or left click anywhere outside the window - and the window will be redrawn using the cursor position as the top-left corner.

1.2.2 The User Guide

The user guide is designed to be a few pages that can be photocopied and distributed to new users. These users will want to know how to use Spell/400 but are not concerned with the administration or configuration. It is written with the view that they understand Spell checkers in general...

Using the Spell Checker

When the system detects you have a spelling mistake, it will pop up a box showing you the mistake, and offer a few alternatives. It's a standard Spell Checker similar to one you'd find on many PC programs.

The three types of pop-up window

There are three versions of the window. You can cycle through the different windows by pressing F11, and the title will change to *GRAPHICAL, *MINIMAL or *KEYBOARD. They all work in the same way except the 'ease of use' - the one which you prefer is very much down to your preferences.

***GRAPHICAL** works just like a PC Spell Checker - you have to select a word with a mouse click and then press a button like CHANGE or ADD.

***MINIMAL** is also designed for a mouse but it's more intuitive than the *GRAPHICAL interface. For instance, clicking on a suggestion automatically assumes you want to CHANGE the word and double clicking means you want to change all the occurrences.

***KEYBOARD** is designed for using with the TAB and ENTER keys (although it works fine with a mouse as well).

When a mistake is found

When a mistake is found it is underlined and highlighted. The mistake may be found in several places and they will all be highlighted. The pop-up window should be positioned somewhere near each mistake in turn.

Select one of the other suggestions.

If you can see the word correctly spelled in the list of suggestions, just move the cursor over it and press ENTER or point the mouse and left click.

If you're using the *GRAPHICAL window then the word will go to the top of the list and you need to press the CHANGE button. The other two pop-up windows will have already made the change.

The original mistake will be replaced and the changed word will be underlined.

When the cursor is on a suggestion, if you press F9 it will change the word (like pressing Enter), if you press Shift-F9 it will replace all occurrences.

TIP If you put the cursor one position to the left of the suggested word it will replace all occurrences.

The suggestion isn't in the list

First try pressing Page Up and Page Down and you might see some more suggestions. If you do, just move the cursor over it and press Enter as before.

If you're typing in a person's name, or a place name or a word that's not common English then you'll have to decide what to do. You can Add the word to the system dictionary or you can simply ignore it.

How do I ignore the word?

Just press F12 to ignore it this time, or press Shift F12 (i.e. F24) to ignore ’all occurrences’.

With a mouse you can right click anywhere on the screen. If you use the Shift key it’ll ignore all occurrences like Shift F12.

TIP If you put the cursor one position to the left of the spelling mistake it will ignore all occurrences.

How do I add the word?

Press F6 to add the word to the system dictionary (or click the ADD button if it’s shown). From this point on, it won’t complain about the word. Note that the word still has to be approved by the administrator before it becomes a ’proper’ dictionary word.

If the screen says ’Function key not allowed’ when you press F6 then you are not allowed to add the word.

How do I type in the real spelling?

You can simply type in the word at the top and press Enter. This new word will be checked and if it’s OK, you can press Enter again and the original word will be replaced.

I keep on typing a word wrong - can I change all of them in one go?

If you always want a word to be changed to another, press Shift F9. Then every occurrence will be changed until the end. Note that like Shift F12, this is forgotten as soon as you start spell checking something else.

Sometimes the pop-up box covers up something - can I move it?

You can move the pop-up box at any time by moving the cursor OUTSIDE of the box and pressing ENTER. Where ever you moved the cursor to will be the top-left corner of the box when it’s moved.

With a mouse, just click anywhere outside of the window.

Administration 2

2.1 Installation and Upgrades

2.1.1 Install the library

The install is now automated. When the restore is finished there will be a message in QSYSOPR, and the SPELL400 menu will be created in QUSRSYS.

User profile to install

Any user with the ability to create a library can be used to install Spell/400. It is preferable to use QSYSOPR or QSECOFR so that initialization tasks can be performed (such as checking for IBM dictionaries, creating the SPELL400 menu in QUSRSYS etc).

Object ownership

All objects will be owned by QDFTOWN, with PUBLIC *CHANGE. When a user creates a dictionary (i.e. when they add words to their *USER dictionary) the dictionary is owned by their user profile. When the administrator approves the dictionary it is owned by the administrator's user profile.

GO SPELL400

Once Spell/400 has been installed, you can access the main menu with GO SPELL400. If the attempt to initialize it during the install failed (normally due to a lack of authority) then it will attempt to initialize itself.

2.1.2 Upgrades/updates

Check the www.spell400.com website regularly for updates or upgrades.

Upgrades will replace programs and objects that might be in use by users, causing problems. These problems can be prevented by upgrading whilst Spell/400 is not in use or by requesting that users with problems sign off and on. In most cases a problem with a Spell/400 upgrade will be invisible to the user.

Upgrades may want to replace menu QUSRSYS/SPELL400 - anyone using that menu will prevent the update. However a newer menu should still be restored as SPELL400/SPELL400.

Be aware that upgrades may well conflict with existing APIs and care should be taken in identifying any changes before changing a production system. Any conflicts will be listed below (all changes since V3R4 should not cause a problem).

2.2 What's new?

2.2.1 V4R2 Apr 2012 - Mar 2016 (V4R2M3)

V4R2 is an evolutionary release update with bug fixes

2.2.1.1 Completely new

1. Auditing log rewritten
2. CSV list of spelling mistakes returned
3. Tutorial step-by-step examples

2.2.1.2 Improved/updated

1. Evaluation wizard improved
2. Car Warranty example programs improved
3. iAccess compatibility tested along with other emulators
4. Bug fixes, especially formatting of words onscreen

2.2.1.3 Upgrade considerations

1. None

2.2.2 V4R1 Nov 2010

V4R1 is an evolutionary release update with a new suggestions engine, but with several hidden improvements which will be activated gradually.

2.2.2.1 Completely new

1. i/OS V7 compatible

2.2.2.2 Improved/updated

1. Suggestions 'engine'

2.2.2.3 Upgrade considerations

1. None

2.2.3 V3R8 Aug 2008 - Jan 2010 (V3R8M4)

V3R8 has added a few new features but mainly it is for systems upgrading to i/OS V6R1+. V3R8M1 and M2 have concentrated on improving the recent features in diverse real world situations.

2.2.3.1 Completely new

1. V6 compatible (V5R2+). Systems on OS/400 V5R1 or lower cannot upgrade.

2.2.3.2 Improved/updated

1. Better integration with the SPELLCHECK VAR utility
2. Various improvements for LOD/SAVSPAENV
3. Multiple improvements to all ENV commands
4. Multiple improvements to Multi-Page Cache
5. Various bug fixes for uncommon or unique situations
6. Helptext improved
7. Evaluation and examples updated or added

2.2.3.3 Upgrade considerations

1. None

2.2.4 V3R7 Feb 2008

V3R7 has added a few new features, and improved handling of uncommon situations.

2.2.4.1 Completely new

1. Multi-Page Cache
2. Load/Save Spell environments for different applications or users

2.2.4.2 Improved/updated

1. QUICK UI and logging improved
2. Various bug fixes for uncommon or unique situations
3. Menus and helptext improved
4. Evaluation and examples updated or added

2.2.4.3 Upgrade considerations

1. None

2.2.5 V3R6 Oct 2006 - Nov2006 (V3R6M6)

V3R6 adds caching over multiple pages and spell checks to simulate a single spellcheck.

2.2.5.1 Completely new

1. Queued spellchecks
2. Multi-page cache

2.2.5.2 Improved/updated

1. Various bug fixes for unique situations
2. Installation/upgrade routines improved

2.2.5.3 Upgrade considerations

1. None

2.2.6 V3R5 May 2006

V3R5 was a beta release - introducing concepts which were retained in V3R6 or have been removed/downgraded.

2.2.6.1 Completely new

1. Queued spellchecks

2.2.6.2 Upgrade considerations

1. None

2.2.7 V3R4 Oct 2005

V3R4 has made changes to the audit control and logging.

2.2.7.1 Completely new

1. History log of actions taken
2. Function keys can be blocked to restrict options available to user
3. New *QUICK popup windows show all mistakes at once
4. New commands and reports making management of dictionaries easier.

2.2.7.2 Improved/updated

1. Simplified the AutoLearn/AutoChange confusions
2. ADDSPAWRDS speed improved
3. Improved handling of public dictionaries
4. Help text updated

2.2.7.3 Upgrade considerations

1. None

2.2.8 V3R3 May 2005

V3R3 allows you to prevent the confirmation window at the end of the spell check, and adds Auto-Learn. The accompanying API changes allow you to specify who can use AutoChange.

2.2.8.1 Completely new

1. AutoLearn allows common mistakes to be learned with the F21 key
2. Ask for Confirmation can prevent the confirmation window at the end of the spell check.
3. New set of dictionaries created

2.2.8.2 Improved/updated

1. Help text updated
2. Compatibility with webfacing

2.2.8.3 Upgrade considerations

1. OVRSPAENV API changed - new parameter added (minimal impact)
2. OVRSPAENV and CHGSPAENV commands changed (minimal impact)

2.2.9 V3R1 Jan 2005

V3R1 brings 'AutoChange' to Spell/400 - where common mistakes can be identified and changed automatically. The accompanying API changes allow you to specify who can use AutoChange.

2.2.9.1 Completely new

1. AutoChange allows common mistakes to be changed automatically
2. AUTOCHANGE dictionary containing 500 common mistakes and their replacements
3. Screen Overlay demonstration

2.2.9.2 Improved/updated

1. Screen overlay commands improved
2. Suggestions are slightly improved (the order is checked and altered)
3. Evaluation ’wizard’ updated to better show new (and old) features
4. Help text updated
5. More example programs
6. More extensive auditing of dictionary changes

2.2.9.3 Upgrade considerations

1. OVRSPAENV API changed - parameter positions changed to keep consistent with command layouts.
2. APIILE Member changed - reflects new positions of OVRSPAENV API
3. OVRSPAENV and CHGSPAENV commands changed (minimal impact)

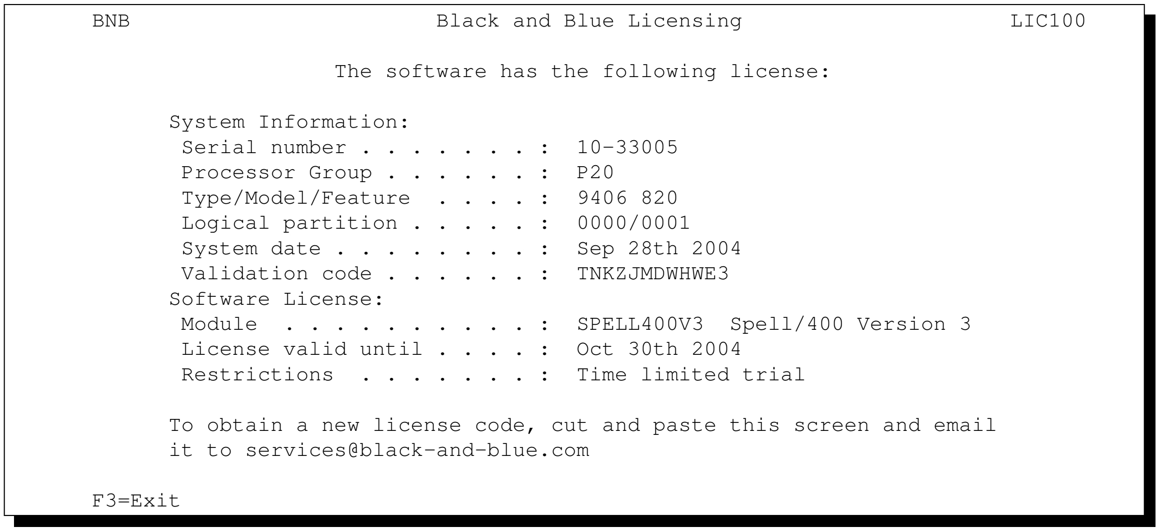
2.2.10 Previous

V3R4M1 - Aug 2005 (limited distribution)
V3R3 - May 2005
V3R2 - Apr 2005 (privately distributed)
V3R0M4 - Dec 2004
V3R0M3 - Nov 2004 (privately distributed)
V3R0M2 - Nov 2004
V3R0M1 - Oct 2004 (privately distributed)
V3R0M0 released in October 2004
V2 - 2000-2004
V1 - 1996-2000

2.3 License keys

Spell/400 needs a license key to work. The key is specific to your system and partition. To generate a key we need the text from the ’Display Spell/400 license’ screen (i.e. ’cut-and-paste’d into an email).

To see the license, take option 82 from the main SPELL400 menu, and you should see a screen similar to the following:



```
   BNB                        Black and Blue Licensing                      LIC100

                      The software has the following license:

         System Information:
          Serial number . . . . . . . :  10-33005
          Processor Group . . . . . . :  P20
          Type/Model/Feature  . . . . :  9406 820
          Logical partition . . . . . :  0000/0001
          System date . . . . . . . . :  Sep 28th 2004
          Validation code . . . . . . :  TNKZJMDWHWE3
         Software License:
          Module  . . . . . . . . . . :  SPELL400V3  Spell/400 Version 3
          License valid until . . . . :  Oct 30th 2004
          Restrictions  . . . . . . . :  Time limited trial

         To obtain a new license code, cut and paste this screen and email
         it to services@black-and-blue.com

   F3=Exit
```

2.3.1 To obtain a new key

Copy and paste the information above into a text-only email and send to services@spell400.com services@spell400.com. Please allow 48 hours for a new key.

2.3.2 Applying the license key

The license key returned needs to be entered from the command line or with option 83. A *SECOFR profile will be needed.

2.3.2.1 Add new Spell/400 Software License for V3R1+

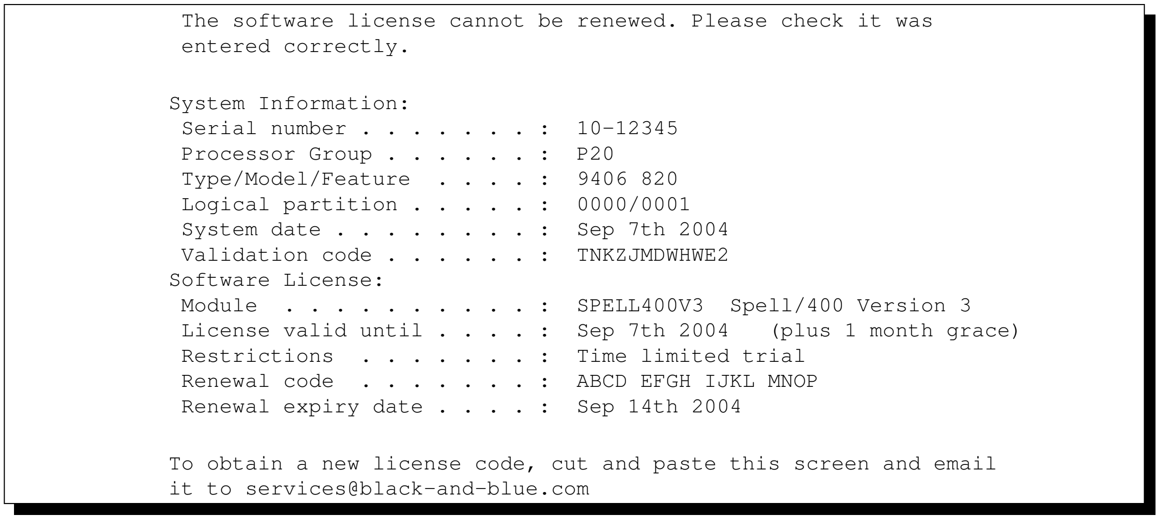
The key you will receive will consist of a command that can be cut and paste onto a System i terminal session. The entire key will show the serial number and some instructions:

```
----------------------------------------------------
System: 44-42054
----------------------------------------------------
--To apply the new code, cut and paste the ---------
--following command and run under a suitable -------
--user profile (e.g. QSECOFR) ----------------------
----------------------------------------------------
```

Followed by the command containing the key:

```
ADDSPALIC CODE(’ Feb 19 2006 ZZTD VCLV NLVG TDZJ ’)
```

When you press ENTER the code will be accepted and the new expiry date will be shown. However, if there was a problem, the code will be rejected; check the code was entered correctly and if you can’t find the problem then please send us a copy of the screen-print, eg:



```
 The software license cannot be renewed. Please check it was
 entered correctly.

System Information:
 Serial number . . . . . . . :  10-12345
 Processor Group . . . . . . :  P20
 Type/Model/Feature  . . . . :  9406 820
 Logical partition . . . . . :  0000/0001
 System date . . . . . . . . :  Sep 7th 2004
 Validation code . . . . . . :  TNKZJMDWHWE2
Software License:
 Module  . . . . . . . . . . :  SPELL400V3  Spell/400 Version 3
 License valid until . . . . :  Sep 7th 2004   (plus 1 month grace)
 Restrictions  . . . . . . . :  Time limited trial
 Renewal code  . . . . . . . :  ABCD EFGH IJKL MNOP
 Renewal expiry date . . . . :  Sep 14th 2004

To obtain a new license code, cut and paste this screen and email
it to services@black-and-blue.com
```

2.4 Environment

The spelling environment determines what settings should be used by the commands and APIs - either on a system wide basis or separate environments for individual jobs.

2.4.1 System v. Job

The System environment is the default configuration for all jobs. Any changes made take immediate effect.

The Job environment is an override over the System settings. Only those parameters specifically overridden are affected, and all others are taken from the System environment.

System Environment

DSPSPAENV	Display Environment	DSPSPAENV *SYSTEM
CHGSPAENV	Change System Environment	CHGSPAENV

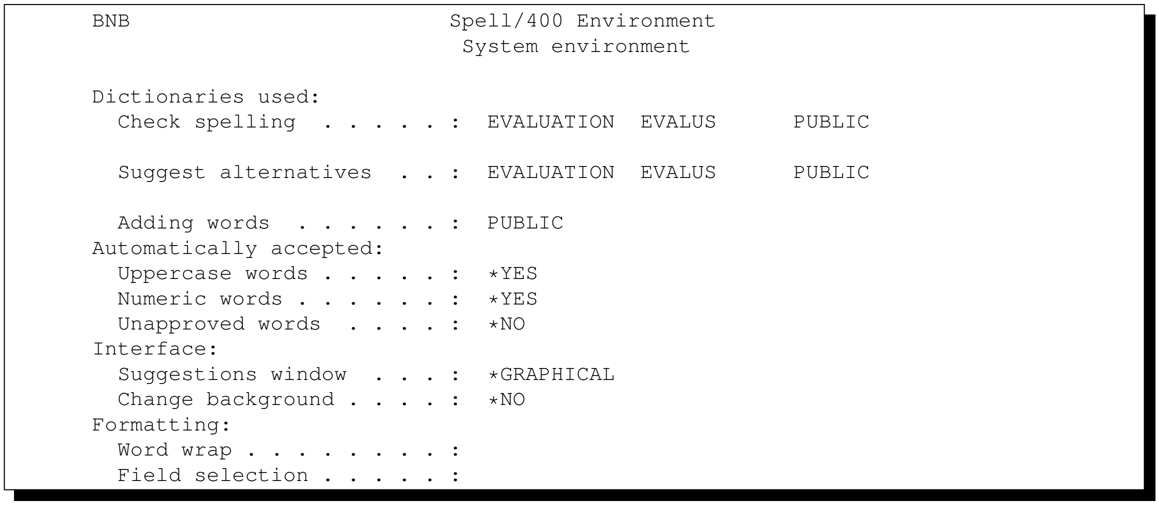
Job Environment

DSPSPAENV	Display Environment	DSPSPAENV *JOB
OVRSPAENV	Override System Environment	OVRSPAENV

2.4.2 Display Spell Environment

To display the System environment take option 12 from the main menu:

12. Display/Change Environment



```
      BNB                            Spell/400 Environment
                                      System environment

      Dictionaries used:
        Check spelling  . . . . . :  EVALUATION  EVALUS      PUBLIC

        Suggest alternatives  . . :  EVALUATION  EVALUS      PUBLIC

        Adding words  . . . . . . :  PUBLIC
      Automatically accepted:
        Uppercase words . . . . . :  *YES
        Numeric words . . . . . . :  *YES
        Unapproved words  . . . . :  *NO
      Interface:
        Suggestions window  . . . :  *GRAPHICAL
        Change background . . . . :  *NO
      Formatting:
        Word wrap . . . . . . . . :
        Field selection . . . . . :
```

For an explanation of the values, you can press F1 on the screen or see the descriptions below.

Note that when the environment has been overridden with OVRSPAENV, the new values are highlighted.

2.4.3 Change Spell/400 system environment (CHGSPAENV)

V3R1

The Change Spell/400 system environment (CHGSPAENV) command allows you to change the system-wide settings used by Spell/400 for spell checking.

The change in environment settings takes effect immediately and affects all jobs that don't have a specific override (OVRSPAENV).

Parameter list

V3R3

DICTIONARY	Spelling dictionaries	
SUGGESTION	Suggestion dictionaries	
ADDWORDS	Add words to dictionary	
AUTOLEARN	Add mistakes to dictionary	
UIM	User Interface Manager	
ACPUPR	Accept all words in upper case	_*YES_, *NO
ACPNUM	Accept all words with numerals	_*YES_, *NO
ACPUNA	Accept all unapproved words	*YES, _*NO_
ACPAUT	Accept all auto-learned words	*YES, _*NO_
BACKGROUND	Change background to blue	_*YES_, *NO
CONFIRM	Ask for confirmation	*ALWAYS, *NEVER, *INCOMPLETE

2.4.3.1 Parameter descriptions

DICTIONARY Spelling dictionaries

Specifies the list of dictionaries that are used for spell checking.

These dictionaries are only used for checking the validity of a word and do not provide suggestions.

AutoChange dictionaries cannot be used.

Up to 8 dictionaries can be specified for spell checking. The order does not matter.

The possible values are:

*SAME No change is made to the environment dictionaries.

*US A default set of dictionaries are used for US spelling. The actual dictionaries depend upon whether the software is in evaluation.

*UK A default set of dictionaries are used for UK spelling. The actual dictionaries depend upon whether the software is in evaluation.

*DFT A default set of dictionaries are used. The actual dictionaries depend upon whether the software is in evaluation and whether Spell/400 can determine whether UK or US spelling would be more appropriate.

*USER This value is translated to the job user profile.

*ADDWORDS This value is translated to the ADDWORDS dictionary at the beginning of a spell check - i.e. the actual dictionary used changes when the ADDWORDS dictionary changes.

dictionary-names Specify up to 8 dictionaries that can be found using the Work with Spell/400 Dictionary (WRKSPADCT) command.

SUGGESTION Spelling suggestion dictionaries

Specifies the list of dictionaries that are used for spell checking.

These dictionaries are used for suggesting alternatives for a spelling mistake.

Up to 8 dictionaries can be specified for spell checking.

The order they are listed provide the order of the suggestions, so dictionaries should be sorted in order of most commonly to least commonly used words.

R4

AutoChange dictionaries can be used, but the list will be resorted so that they are at the top of the list.

The possible values are:

*SAME No change is made to the environment dictionaries.

*SPELLING The spelling dictionaries parameter is used.

*US A default set of dictionaries are used for US spelling. The actual dictionaries depend upon whether the software is in evaluation.

*UK A default set of dictionaries are used for UK spelling. The actual dictionaries depend upon whether the software is in evaluation.

*DFT A default set of dictionaries are used. The actual dictionaries depend upon whether the software is in evaluation and whether Spell/400 can determine whether UK or US spelling would be more appropriate.

*USER This value is translated to the job user profile.

*ADDWORDS This value is translated to the ADDWORDS dictionary at the beginning of a spell check - i.e. the actual dictionary used changes when the ADDWORDS dictionary changes.

dictionary-names Specify up to 8 dictionaries that can be found using the Work with Spell/400 Dictionary (WRKSPADCT) command.

ADDWORDS Dictionary name to add new words

Specifies the name of the Spell/400 Dictionary where new words are added.

When the spell check window is shown to the user, the F6=Add or ADD button will only be available if a dictionary is specified here. If the dictionary is *NONE then users will not be able to add words.

You can specify different dictionaries for different departments or applications or users, or simply use a general dictionary like PUBLIC for all users.

The possible values are:

*SAME No change is made.

*USER The dictionary will be the same name as the job user profile. The dictionary does not have to exist until it is used.

*PUBLIC The PUBLIC dictionary will be used.

*NONE Words cannot be added by the user.

dictionary-name Specify a dictionary that can be found using the Work with Spell/400 Dictionary (WRKSPADCT) command.

AUTOLEARN Dictionary name to add mistakes/corrections

Specifies the name of the Spell/400 AutoChange Dictionary where new mistakes and replacement words are added (with the F21 or CHGALL button).

When a spelling mistake is found the user has the ability to select a suggestion and press F9 (CHANGE button). If they press F21 (CHGALL) to change all occurrences the mistake and replacement will be added to the AutoChange dictionary for future occurrences.

The AutoChange words will only be used if the AutoChange dictionary is in the SUGGESTIONS list and words have been approved and ACPAUT is set to *YES.

The possible values are:

*SAME No change is made.

*NONE Words cannot be added by the user.

dictionary-name Specify a dictionary that can be found using the Work with Spell/400 Dictionary (WRKSPADCT) command.

UIM User Interface Manager

Specifies the UIM to use when presenting the user with a choice of alternative word suggestions.

The graphical look may only work on graphically-enabled emulation products such as Client Access.

The possible values are:

*SAME No change is made.

*GRAPHICAL Use a graphical appearance similar to a PC spell checker.

*MINIMAL Use a graphical appearance with minimal screen ’real estate’.

*KEYBOARD Use a traditional appearance with emphasis on keyboard users.

*QUICKGRF Use a graphical appearance where all the words popup in a single window

*QUICKMIN Use a graphical appearance with minimal screen 'real estate' where all the words popup in a single window.

*QUICKKBD Use a traditional appearance with emphasis on keyboard users where all the words popup in a single window. .

ACPUPR Accept words in upper case

Specifies whether words entirely in upper case should be automatically accepted by spell checking. Many words are formed from acronyms or special job-related tasks which do not form part of a standard language, e.g. IBM, NASDAQ, ASAP, PTO. These words can be automatically accepted by the spell checker.

The possible values are:

*SAME	No change is made.
*YES	All words in upper case are automatically accepted.
*NO	Words in upper case are treated as normal words and will be spell checked as normal.

ACPNUM Accept all words with numerals

Specifies whether words containing numerical characters should be automatically accepted by spell checking. Many words are formed with numerical characters which do not form part of a standard language, e.g. a part number, or brand name such as System i . These words can be automatically accepted by the spell checker.

The possible values are:

*SAME	No change is made.
*YES	All words with numerals are automatically accepted.
*NO	Words with numerals are treated as normal words and will be spell checked as normal.

ACPUNA Accept unapproved words

Specifies whether words in the ADDWORDS dictionary should be automatically accepted by spell checking if they are pending approval.

Words that are added to the ADDWORDS dictionary may need to be approved by an administrator before being 'generally' acceptable.

The possible values are:

*SAME	No change is made.
*YES	All words in the ADDWORDS dictionary are automatically accepted.
*NO	Unapproved words are treated as normal words and will be spell checked as normal.

ACPAUT Accept AutoChange words

Overrides whether words in an AutoChange dictionary in the SUGGESTIONS list will be automatically accepted (replaced or ignored) by spell checking.

The possible values are:

*SAME	No change is made.

*YES All approved words in an AutoChange dictionary are automatically accepted.

*NO Unapproved words are treated as normal words and will be spell checked as normal.

BACKGROUND Change background to blue

Specifies if text outside of the screen-scrape area should be changed to blue with all other attributes removed. In some cases this makes the screen easier to read (and in others it doesn't), so it may be useful for certain application screens.

The possible values are:

*SAME No change is made.

*YES The area outside of the screen scrape area is changed to blue. All attributes are removed.

*NO The area outside of the screen scrape remains unchanged.

CONFIRM Ask for confirmation

Specifies if the spell check changes should be confirmed (i.e. accepted or rejected) at the end of a screen-scrape based spell check.

The possible values are:

*SAME No change is made.

*ALWAYS The confirmation window is always displayed (the situation pre V3R2)

*INCOMPLETE The confirmation window is only displayed if the spell check is incomplete (usually only if the user presses F3).

*NEVER The confirmation window is never displayed.

2.4.4 Applications and User environments

V3R8 allows programmers to Save and Reload environments (SAVSPAENV and LODSPAENV) so that environments customised to a particular application or user (department, system etc) can be standardised.

More information is given in the programming section.

2.5 Auditing

As well as controlling the environment, the administrator has control over the auditing abilities of Spell/400.

Auditing is split into 2 functions:

1. restriction of features available to users

2. history log of actions performed

2.5.1 Change Spell/400 auditing (CHGSPAAUD)

The Change Spell/400 Auditing (CHGSPAAUD) command allows you to restrict Spell/400 features,

The change in setting takes effect immediately and affects all jobs.

Note that there is no DSPSPAAUD command - to display the existing values, prompt CHGSPAAUD.

Parameter list

PVNF3	Prevent F3=Exit	*YES, *NO
PVNF6	Prevent F6=Add	*YES, *NO
PVNF9	Prevent F9=Change	*YES, *NO
PVNF10	Prevent F10=Analysis	*YES, *NO
PVNF12	Prevent F12=Ignore	*YES, *NO
PVNF13	Prevent F13=Select 1	*YES, *NO
PVNF14	Prevent F14=Select 2	*YES, *NO
PVNF15	Prevent F15=Select 3	*YES, *NO
PVNF16	Prevent F16=Select 4	*YES, *NO
PVNF17	Prevent F17=Select 5	*YES, *NO
PVNF18	Prevent F18=Select 6	*YES, *NO
PVNF21	Prevent F21=ChangeAll	*YES, *NO
PVNF24	Prevent F24=IgnoreAll	*YES, *NO
LOGALL	Log all actions	*YES, *NO
LOGSCR	Log screenshots	*YES, *NO
KEEPLOG	Number of days to keep	1-999, *NOMAX

2.5.1.1 Parameter descriptions

PVNF3 Prevent F3=Exit

Prevents exit from spell checking before all words have been checked by the user.

The possible values are:

*SAME	No change is made
*NO	Function key and buttons are allowed.
*YES	Function key and button are not shown, and are prevented.

PVNF6 Prevent F6=Add

Prevents adding words. Takes precedence over the environment ADDWORDS dictionary (which otherwise has the same effect of showing/allowing F6).

The possible values are:

*SAME	No change is made
*NO	Function key and buttons are allowed.
*YES	Function key and button are not shown, and are prevented.

PVNF9 Prevent F9=Change

Prevents user changing a mistake. Possibly useful for text that shouldn’t be changed (e.g historical data).

You may also want to apply the same rule to F21=Change All.

The possible values are:

*SAME	No change is made
*NO	Function key and buttons are allowed.

*YES	Function key and button are not shown, and are prevented.

PVNF10 Prevent F10=Analysis

Prevents user viewing the Spell/400 analysis (where the environment can be changed).
The possible values are:

*SAME	No change is made
*NO	Function key and buttons are allowed.
*YES	Function key and button are not shown, and are prevented.

PVNF12 Prevent F12=Ignore

Prevents user ignoring a mistake - forcing the mistake to be corrected.
The possible values are:

*SAME	No change is made
*NO	Function key and buttons are allowed.
*YES	Function key and button are not shown, and are prevented.

PVNF13 Prevent F13=Select 1

Prevents user changing the mistake to the first selection.
The possible values are:

*SAME	No change is made
*NO	Function key and buttons are allowed.
*YES	Function key and button are not shown, and are prevented.

PVNF21 Prevent F21=Change All

Prevents user changing all mistakes. Possibly useful to prevent
You may also want to apply the same rule to F9=Change.
The possible values are:

*SAME	No change is made
*NO	Function key and buttons are allowed.
*YES	Function key and button are not shown, and are prevented.

PVNF24 Prevent F24=Ignore All

Prevents user ignoring all mistakes.
You may also want to apply the same rule to F12=Ignore.
The possible values are:

*SAME	No change is made
*NO	Function key and buttons are allowed.
*YES	Function key and button are not shown, and are prevented.

LOGALL Log all actions

All spell actions can be logged for later review. This log is accessible during the spell check (if F10 is allowed), or afterwards. Useful when tight control is needed over shared dictionaries.

The possible values are:

*SAME	No change is made
*NO	No changes are logged.
*YES	All changes are logged.

LOGSCR Log screenshots

When a log entry is made, a screenshot is taken and stored for later review.

The possible values are:

*SAME	No change is made
*NO	No screenshots are logged.
*YES	All screens are logged.

KEEPLOG Number of days to keep logs

The log entries are removed with the CLRSPALOG command, however the minimum number of days to keep can be specified here.

The CLRSPALOG command will not override this value.

For example, setting a value of 10 means that when the CLRSPALOG command is run it cannot remove entries made in the the last 10 days.

The possible values are:

1-999	Specify the minimum number of days that data should be retained.
*SAME	No change is made
*NOMAX	Specifies that no log entries should be removed.

2.5.2 Clear Spell/400 auditing log (CLRSPALOG)

The Change Spell/400 Auditing (CHGSPAAUD) command allows you to restrict Spell/400 features,

The change in setting takes effect immediately and affects all jobs.

Note that there is no DSPSPAAUD command - to display the existing values, prompt CHGSPAAUD.

Parameter list

KEEPLOG	Number of days to keep	1-999, *NOMAX

2.5.2.1 Parameter descriptions

KEEPLOG Number of days to keep logs

The log entries are removed with the CLRSPALOG command, however the minimum number of days to keep can be specified here.

The CLRSPALOG command will not override this value.

For example, setting a value of 10 means that when the CLRSPALOG command is run it cannot remove entries made in the the last 10 days.

The possible values are:

1-999	Specify the minimum number of days that data should be retained.
*SAME	No change is made
*NOMAX	Specifies that no log entries should be removed.

2.6 Dictionaries

The multi-user aspect of Spell/400 requires that some administration be performed. The administration can be cut down to a minimalist single command (APVSPADCT) which can approve all the new words without further ado.

However, if you want to actively approve words before they are added, there are a few useful screens to allow administrators to do this.

2.6.1 Spelling Dictionaries

Spelling dictionaries are different from normal definition dictionaries in that they store many more words. For each ’root’ word they contain several entries; the plural, the different tenses, with an apostrophe etc. plus derivative words.

So Spelling Dictionaries can be huge, which, surprisingly, can actually worsen the likelihood of catching spelling mistakes. The more words a spelling dictionary has, the more likely a misspelled word is to be accepted (as another word). For instance, if someone were to mistype ’cough’ as ’chough’ then a complete (really, really complete!) English dictionary would accept ’chough’ as a valid word (in fact, a chough is a type of crow).

Spelling dictionaries thus have to provide two distinct functions - find spelling errors and suggest sensible alternatives. Whilst ’chough’ is a valid word, it would be surprising to most people if it was accepted as a typing mistake for ’cough’.

Spell/400 tries to overcome this by providing 3 different English dictionaries. The first 2 - COMMON and NORMAL - contain easily recognisable words. They provide the best suggestions for typing mistakes. The third and fourth - STANDARD and UNCOMMON - contain 225,000 additional words which are technically valid but least likely to be used. There is a another dictionary called COMPLETE with over 350,000 words, of which few are recognisable to the non-scholar! This dictionary can be obtained on request. See on page 36 for more details on the supplied dictionaries.

2.6.2 Multi-user dictionaries

One major difference between PC dictionaries and System i dictionaries is the multi-user aspect of the System i - where people can share (and add to) common dictionaries.

On both the PC and the System i, each user has the ability to change or ignore a word that is incorrect. However most times, the user would expect to be able to add a word. When they come across a proper name (e.g. ’London’), the obvious action is to add it so it’s never highlighted as a spelling mistake again.

If you want all users to be able to add words like this, then you have no problem and Spell/400 will cope easily.

However if words can be added by someone not gifted with a talent for spelling, you may find that incorrect words have been added to the common dictionaries, affecting everyone else. You might find that common mistakes such as 'febuary' or 'seperate' have been added and are no longer highlighted as mistakes.

So Spell/400 offers a system for approving new words, making adding words a two-stage process:

1. Users can add any word to the ADDWORDS dictionary.
2. Administrators can review and amend the additions before they are approved for general usage.

2.6.3 AutoChange dictionaries

All the normal Spell/400 dictionaries contain valid, correctly-spelled words. AutoChange dictionaries are the exact opposite - they contain invalid, badly spelled words.

When spell checking, words are compared to the normal dictionaries and if they're not found then they must be spelling mistakes - so the user is prompted to correct the spelling.

However, if the list of suggestion dictionaries contain an AutoChange dictionary, and the mistake is found in the AutoChange, it is automatically replaced with the correct spelling. i.e. AutoChange dictionaries search for mistakes but also track what the spelling should be.

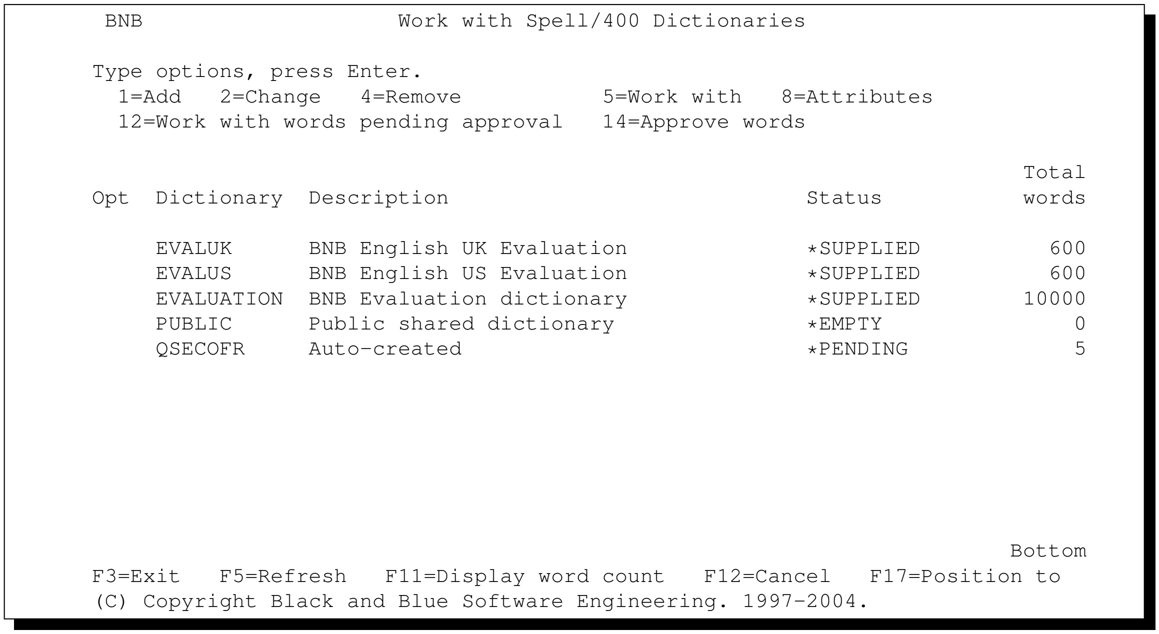
AutoChange dictionaries are typically maintained with the AutoLearn environment variable - where mistakes that are changed with F21=ChangeAll are added to the AutoChange dictionary.

2.6.4 Work with dictionaries command

To work with the Spell/400 dictionaries, take option 10 from the SPELL400 menu:

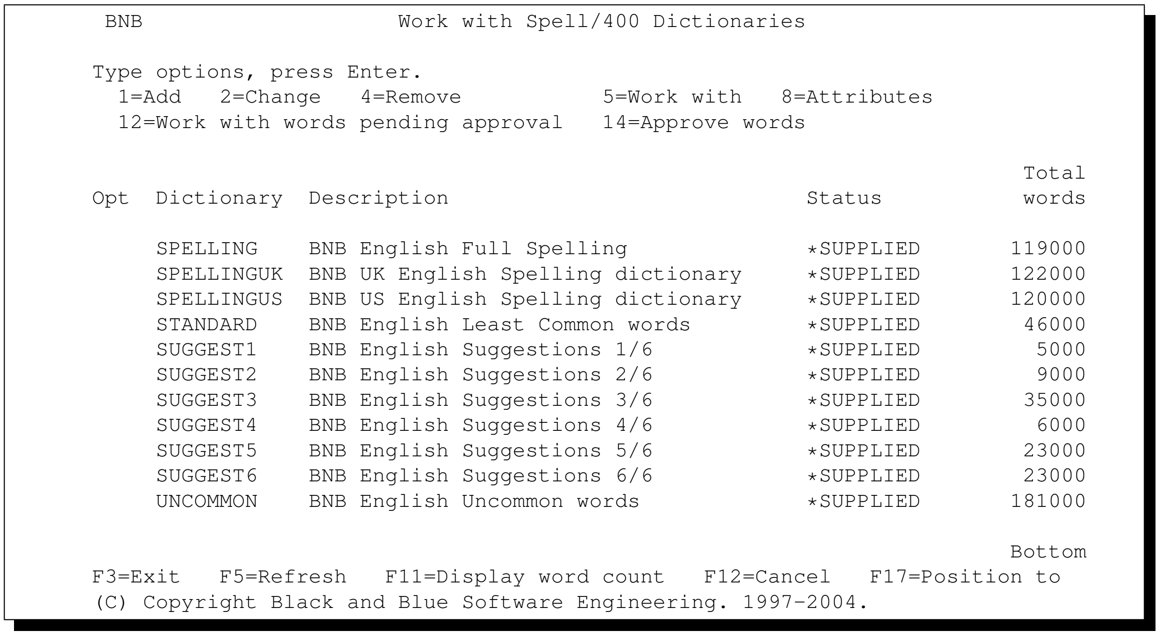
10. Work with Spell/400 Dictionaries

You should see a screen similar to the following:



```
 BNB                    Work with Spell/400 Dictionaries

Type options, press Enter.
  1=Add   2=Change   4=Remove           5=Work with   8=Attributes
  12=Work with words pending approval   14=Approve words

                                                                    Total
Opt  Dictionary  Description                            Status          words

     EVALUK      BNB English UK Evaluation              *SUPPLIED         600
     EVALUS      BNB English US Evaluation              *SUPPLIED         600
     EVALUATION  BNB Evaluation dictionary              *SUPPLIED       10000
     PUBLIC      Public shared dictionary               *EMPTY              0
     QSECOFR     Auto-created                           *PENDING            5








                                                                       Bottom
F3=Exit   F5=Refresh   F11=Display word count   F12=Cancel   F17=Position to
(C) Copyright Black and Blue Software Engineering. 1997-2004.
```

When you license Spell/400 you'll receive many more dictionaries:



```
 BNB                    Work with Spell/400 Dictionaries

Type options, press Enter.
  1=Add   2=Change   4=Remove           5=Work with   8=Attributes
  12=Work with words pending approval   14=Approve words

                                                                    Total
Opt  Dictionary  Description                            Status          words

     SPELLING    BNB English Full Spelling              *SUPPLIED      119000
     SPELLINGUK  BNB UK English Spelling dictionary     *SUPPLIED      122000
     SPELLINGUS  BNB US English Spelling dictionary     *SUPPLIED      120000
     STANDARD    BNB English Least Common words         *SUPPLIED       46000
     SUGGEST1    BNB English Suggestions 1/6            *SUPPLIED        5000
     SUGGEST2    BNB English Suggestions 2/6            *SUPPLIED        9000
     SUGGEST3    BNB English Suggestions 3/6            *SUPPLIED       35000
     SUGGEST4    BNB English Suggestions 4/6            *SUPPLIED        6000
     SUGGEST5    BNB English Suggestions 5/6            *SUPPLIED       23000
     SUGGEST6    BNB English Suggestions 6/6            *SUPPLIED       23000
     UNCOMMON    BNB English Uncommon words             *SUPPLIED      181000

                                                                       Bottom
F3=Exit   F5=Refresh   F11=Display word count   F12=Cancel   F17=Position to
(C) Copyright Black and Blue Software Engineering. 1997-2004.
```

2.6.5 Dictionary Status

There are two types of dictionary - supplied and local. The supplied dictionaries cannot be altered, whereas the local ones can be created and altered within Spell/400. Local dictionaries have a status of PENDING or APPROVED - which reflects the status of the words within them.

- *SUPPLIED

  These are generated by suppliers such as IBM and BNB. They cannot be viewed or altered. The 'Total words" (and most other attributes) are shown as *N/A meaning "Not Available". The Spell dictionaries do show the number of permanent words, although this cannot be deduced from IBM dictionaries.

- *APPROVED

  Local dictionaries with an *APPROVED status have been approved, and no new words have been added.

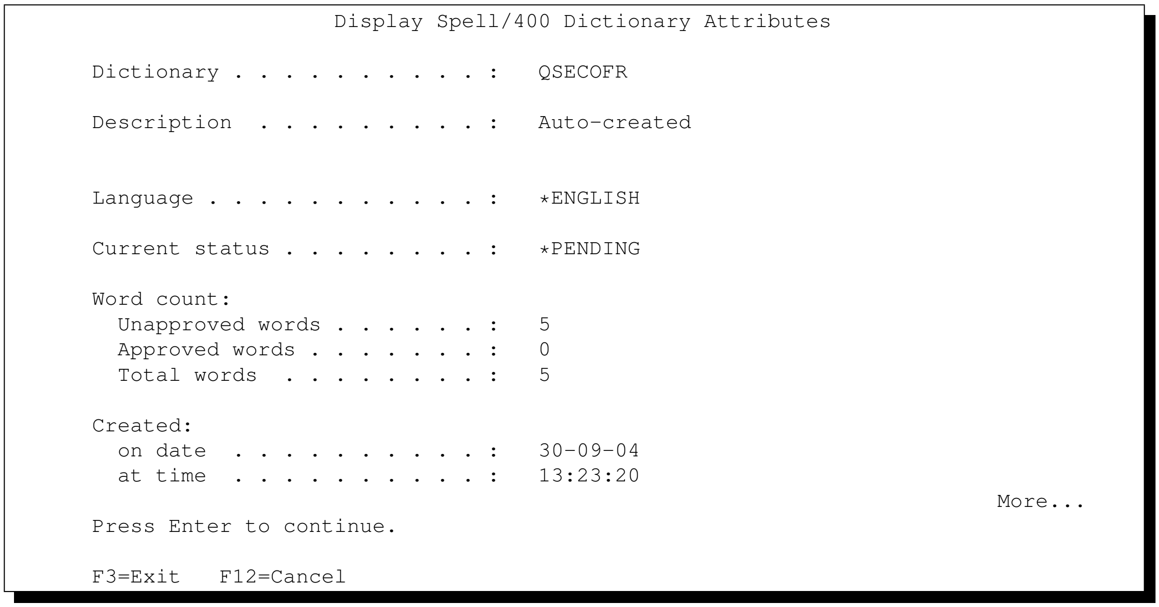
- *PENDING

  Local dictionaries with a *PENDING status contain new words that are pending approval. Option 12 will list all the new words and option 14 will approve all the words.

  Words that are Pending approval will not be offered as suggestions.

2.6.5.1 Viewing a dictionary status

Whilst working with the dictionaries, options 8 shows the status of a dictionary:



```
                          Display Spell/400 Dictionary Attributes

       Dictionary . . . . . . . . . . :   QSECOFR

       Description  . . . . . . . . . :   Auto-created


       Language . . . . . . . . . . . :   *ENGLISH

       Current status . . . . . . . . :   *PENDING

       Word count:
         Unapproved words . . . . . . :   5
         Approved words . . . . . . . :   0
         Total words  . . . . . . . . :   5

       Created:
         on date  . . . . . . . . . . :   30-09-04
         at time  . . . . . . . . . . :   13:23:20
                                                                    More...
       Press Enter to continue.

       F3=Exit   F12=Cancel
```

2.6.6 Approving dictionary words

All words added to a dictionary require approval before they are considered either valid spellings or valid suggestions.

Approving words

When you have checked all the words, you can apply them permanently to the dictionary by changing the dictionary to permanent status. you can do this with menu option 20 or the following command:

```
APVSPADCT DICTIONARY(dictionary-name)
```

Changing approved words

Occasionally you may need to remove words from a dictionary. If they are approved this cannot be done without changing the dictionary to *UNAPPROVED status. Change the dictionary with the following command:

```
UNASPADCT DICTIONARY(dictionary-name)
```

And then work with the dictionary words and make changes. Once you're satisfied you can change the dictionary back by APVSPADCT again.

2.6.7 Supplied dictionaries

Licensed copies of Spell/400 include several dictionaries, almost all of which are English. When combined, the typical total number of words is around 120,000. If required we can supply dictionaries up to 350,000 words.

2.6.7.1 All English words

Simple, complete lists of words suitable for the SPELLING dictionaries (as opposed to providing suggestions).

SPELLINGUS Complete US word dictionary.

SPELLINGUK Complete UK word dictionary.

SPELLING Complete word dictionary (without UK/US variations). Can be used with NORMALUK and/or NORMALUS.

2.6.7.2 All English words as SUGGESTIONS dictionaries

Note that these dictionaries are intended to be used together to build a full dictionary. For instance the STANDARD dictionary doesn't contain any of the words in the COMMON dictionary.

COMMON The most common English words (about 14,000) - and the most obvious choices for spelling errors suggestions. Conversational English uses approximately 5,000 - 10,000 words.

NORMAL An extension to the common words (about 40 ,000) all easily recognisable, and likely to be in normal written text.

STANDARD An extension to the normal words (about 46 ,000) mostly recognisable, but unlikely to be in normal business text.

UNCOMMON Rarely used words (about 181,000), with varying degrees of use. If 'everyday' text is to be checked, this dictionary might not be too useful (unless used for 'overloading' suggestions).

NORMALUK UK word supplement. Typically includes spellings such as 'colour' and 'cheque'. Also includes the *-ize* formations of words (such as initialize) which are to be preferred in standard English (according to the Oxford Guide to English Usage).

NORMALUS US word supplement. Typically includes the spellings such as 'color' and 'check'. It only has the *-ize* formations, and doesn't include *-ise* (such as initialise).

2.6.7.3 All English words as SUGGESTIONS dictionaries

These dictionaries are the same

SUGGEST1 Suggestions for most commonly used words.

SUGGEST6 Suggestions for least commonly used words.

SUGGEST2,3,4,5 Suggestions for the words inbetween!

AUTOCHANGE An AutoChange dictionary with about 500 common mistakes and their replacements. This is treated as a locally created dictionary since you can add and remove words.

2.6.7.4 Names and addresses

Names and addresses are often checked for spelling mistakes, and we have a number of dictionaries with slightly different characteristics.

To spell check a contact database (name and address contact information) we suggest using CONTACT and with UK or USA TOWNS:

NAMES A list of all first and second names (from Census)

UKTOWNS A 45,000 word list derived from UK counties, towns and villages

USATOWNS A 25,000 word list derived from USA towns and cities (from US Census)

Zip codes and other abbreviations can be ignored with the OVRSPAENV ALW* parameters.

For more general, or specific, spell checking these dictionaries can also be used:

PROPER A list of 18,000 Proper names - including names and well known Proper and Brand names.

FIRSTNAMES A list of 11,000 first names (from US Census)

SURNAMES A list of 178,000 surnames (from US Census)

2.6.7.5 Non-English dictionaries

Non-english text has different punctuation which can be difficult to plan for (and thus, for Spell/400 to cope with). Non-transient characters (the characters with accents) are handled very well.

FRENCH A list of 250,000 french words

DUTCH 3 dictionaries of varying complexity totalling 500,000 words

2.6.7.6 Supplied by IBM

To review what IBM dictionaries you have on your system, use the following command:

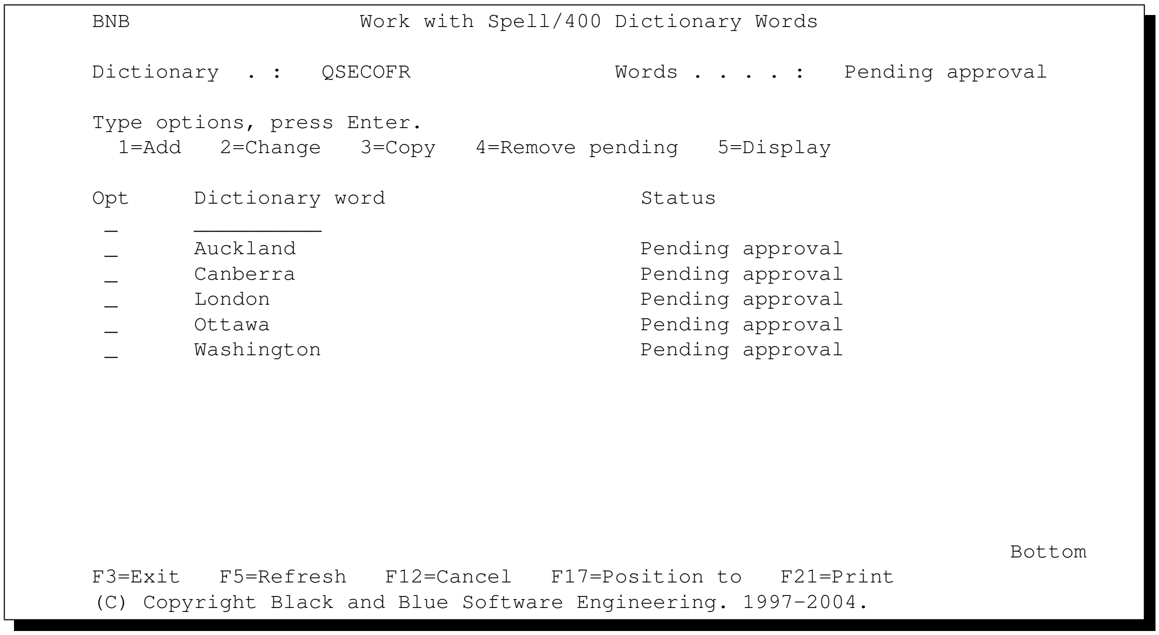
DSPSFWRSC

and page down to see if any licensed dictionaries are installed. If you have some dictionaries installed since installing Spell/400, you need to import them. See on page ??.

2.6.8 Work with Words command

From the list of dictionaries, take option 5 or 12 to work with the words. Option 5 lists all the words, whilst 12 just shows the words needing approval.

You should see a screen similar to the following:



```
BNB                  Work with Spell/400 Dictionary Words

Dictionary  . :   QSECOFR                Words . . . . :   Pending approval

Type options, press Enter.
  1=Add   2=Change   3=Copy   4=Remove pending   5=Display

Opt     Dictionary word                    Status
 _      __________
 _      Auckland                           Pending approval
 _      Canberra                           Pending approval
 _      London                             Pending approval
 _      Ottawa                             Pending approval
 _      Washington                         Pending approval






                                                                        Bottom
F3=Exit   F5=Refresh   F12=Cancel   F17=Position to   F21=Print
(C) Copyright Black and Blue Software Engineering. 1997-2004.
```

Changing a word

Whilst reviewing the new words, you can change or remove them before approval. You can also copy words and change the copy - to add a plural for example.

Note that all words should remain in lowercase.

Changing the status of a word

The status of each word is dependent upon the status of the dictionary. A word can only be approved by approving the entire dictionary.

2.6.9 Installing licensed dictionaries

When you get started with Spell/400 you'll probably install the licensed dictionaries. You can obtain them from the Internet and install them using the EXE upload procedure. Once installed you can see them with command WRKDCT, or 'Work with Spell/400 Dictionaries':

```
AUTOCHANGE  Auto change suggestions                *APPROVED         1312
COMMON      BNB English Full Spelling              *SUPPLIED        14000
EVALCHANGE  BNB Evaluation Auto-Learn              *EMPTY               0
EVALSG      BNB Evaluation overloading suggestion  *SUPPLIED            5
EVALUATION  BNB Evaluation dictionary              *SUPPLIED        10000
EVALUK      BNB English UK Evaluation              *SUPPLIED          600
EVALUS      BNB English US Evaluation              *SUPPLIED          600
NORMAL      BNB English Common words               *SUPPLIED        40000
NORMALUK    BNB English UK supplement              *SUPPLIED         3000
NORMALUS    BNB English US supplement              *SUPPLIED         1400
PROPER      BNB English Proper names               *SUPPLIED        30000
PUBLIC      Public shared dictionary               *EMPTY               0
SPELLING    BNB English Full Spelling              *SUPPLIED       119000
SPELLINGUK  BNB UK English Spelling dictionary     *SUPPLIED       121000
SPELLINGUS  BNB US English Spelling dictionary     *SUPPLIED       120000
STANDARD    BNB English Least Common words         *SUPPLIED        46000
```

If you prompt CHGSPAENV and you should see the current evaluation dictionaries are still in use:

```
Spelling dictionaries  . . . . .   EVALUATION    *SAME, *US, *UK, *SPELLING...
                                   EVALUS
               + for more values   *ADDWORDS
Suggestion dictionaries  . . . .   AUTOCHANGE    *SAME, *US, *UK, *SPELLING...
                                   EVALUATION
                                   EVALUS
               + for more values   *ADDWORDS
Add words to dictionary  . . . .   *USER         *SAME, *USER, *PUBLIC...
```

Replace the spelling and suggestion dictionaries with *US or *UK:

```
Spelling dictionaries  . . . . .   *US           *SAME, *US, *UK, *SPELLING...
               + for more values
Suggestion dictionaries  . . . .   *us           *SAME, *US, *UK, *SPELLING...
               + for more values
Add words to dictionary  . . . .   *USER         *SAME, *USER, *PUBLIC...
```

And if you then type in DSPSPAENV you should see it is now using the default essential dictionaries:

```
Check spelling  . . . . . :  SPELLINGUS  *ADDWORDS
Suggest alternatives  . . :  AUTOCHANGE  COMMON      NORMAL      NORMALUS
                             STANDARD    *ADDWORDS
```

2.6.10 Spelling v. Suggestions dictionaries

Although the default essential dictionaries are fine for normal usage you may find you need more or less variations of words or better suggestions, or less suggestions. You can download different dictionaries, but how you use them is probably more important.

A single SPELLING dictionary

The fewer dictionaries needed to spell check a block of text the better, so we have provided a single dictionary - SPELLING and it's UK and US variants - that can spell check all normal words.

And multiple suggestion dictionaries

However if you split up the SPELLING dictionary into several smaller dictionaries you may obtain more suggestions, in order of their common usage. To allow this we have split up SPELLING into 6 separate dictionaries:

- ▷ SUGGEST1, SUGGEST2, SUGGEST3 ... SUGGEST6

Both SPELLING and SUGGESTn have exactly the same list of words but using SUGGEST dictionaries will produce suggestions according to word frequency, and may produce more suggestions overall. As an example, 'seperate' produces the following suggestions with dictionary SPELLING:

> separate, separator, separated, separates, several,severity

And a few more with SUGGEST1, SUGGEST2, SUGGEST3;

> separate, separator, separated, separates, several, severity, seaport, serpent, speedboat, severest, spectra, seepage, superheat, stepper, sewerage, spewers

Many of these suggestions may be over the top - spectra & sewerage don't come to mind in most order processing notes! So you can remove SUGGEST3 and repalce it with SPELLING as a catch all in case you did indeed mispell spectra. The ideal combination of dictionaries is really down to the typical text in your application. When fine-tuning the suggestions you can press F7 (on the popup box of suggestions) to see the dictionaries that provided the suggestions.

2.7 FAQ

2.7.1 Dictionaries

Can Spell/400 use IBM dictionaries?

> Yes. Spell/400 can use any IBM dictionary once it's imported. The import utility doesn't alter the IBM format, but simply registers the dictionary within Spell/400.

What are Supplied dictionaries?

> Dictionaries are supplied direct by suppliers IBM and BNB, or created and distributed by other users of Spell/400. The dictionaries cannot be viewed or altered, and there is no interface for adding or removing words.

What's the advantage in approving a dictionary?

New words can be changed and removed before they are approved. They will be accepted by Spell as valid words (depending upon the Environment setting), however they will never be offered as alternative suggestions.

Once a dictionary is approved the words will be offered as suggestions.

AutoChange dictionary words will never be used until they are approved.

Does the order of the dictionaries matter?

Yes, which is why Spell/400 separates the various English dictionaries. The dictionaries are searched in the order specified, and the suggestions are specified in the order found. Thus the best way to organize the dictionary for US spellings could be:

1. COMMON
2. NORMAL
3. NORMALUS
4. STANDARD
5. *ADDWORDS

Actually this question now warrants its own section.

2.7.2 Languages

International spellings

English has several international spelling differences - the main two being US and UK spelling. The BNB dictionaries separate the words that are spelt differently into separate dictionaries:

NORMALUK	English (UK and International)
NORMALUS	English (US)

Note that the UK dictionary has both forms of *-ize* and *-ise* where appropriate (e.g. initialize), as determined by the Oxford Guide to the English Language - "where the ending is pronounced *eyes* (not *ice*) then -ize should be preferred over -ise, thus: precise, expertise, remise and authorize, initialize".

These dictionaries should be used in conjunction with the main English dictionaries (Common, Standard and Full). Alternatively, IBM supply both UK and US dictionaries.

Non-English letters within English words

There are a number of foreign words generally acceptable in English complete with their accent symbols. These words are handled with Spell/400 as you'd expect. Thus chateaux, attache and eclair would all be replaced with their accented equivalents.

Foreign languages

Spell/400 can accept many foreign language dictionaries and all features work. We have tested this with various dictionaries, all in English screens/codepages.

3 Programming Introduction

3.1 Using the APIs

Spell/400 has commands and APIs which can be used from your own applications. The commands function identically to the APIs.

3.1.1 Getting started

Consistency

Spell/400 APIs have been written for consistency with the commands. If you need a quick reference to the API parameters you can prompt the equivalent command for help.

API errors and crashes

If an API is supplied with incorrect or invalid parameters then it is designed to silently terminate, with an explanation in the joblog. This may cause a problem when trying out the APIs for the first time since you might expect more fuss! This silent termination is designed with the user application in mind.

There may be an error entry in the Audit log - it won’t give much detail but it should help in recreating a problem that a user reports.

Nothing happens!

If there is a problem - see the section above on API errors and crashes -”If an API is supplied with incorrect or invalid parameters then it is designed to silently terminate...”

Debug with STRSRVJOB

When debugging a program the Spell Checker will show the C debugger prompt:

```
  Start of terminal session.
   Start of debug session.
________________________________________________________
F3=End of File      F9=Retrieve      F21=Extend line
```

You can exit this by typing ’quit’ and Enter, followed by Enter again at the next prompt.

Another problem with debug is it may affect the screen scraping. You can prevent both the C debugger and the problems with screen scraping by using debug in a Serviced job. i.e. Run two sessions and STRSRVJOB/STRDBG in one session and run your applications in the other.

Omitting parameters

The APIs and commands are identical in many ways, and one of these is the omission of parameters. Some parameters (such as a text string to check) are required, and some (such as positioning the pop-up window) are optional. In either CL or RPG the optional parameters can be specified or omitted.

Screen scraping positions

When specifying the row and column that should be screen-scraped, you should specify the positions where the text begins and ends onscreen, i.e the row and column of the cursor when it enters the first field and exits the last field. This is, of course, obvious except to make the distinction between other tools where you specify the attribute byte in the preceding column.

Your application display files

For a Spell/400 window to appear over your own application display files, you might want to bear in mind the usage of DDS ASSUME, KEEP and CRTDSPF's RSTDSP(*YES) and DFRWRT(*NO). These might be necessary for your screen to re-appear normally when Spell/400's windows close.

As a rule of thumb none of them are needed.

DDS Wordwrap

Spell/400 works quite happily with word-wrap (the DDS WRDWRAP command), however it can be 'fun' exploring all the possible combinations of WRDWRAP and Spell/400 *WRAP! In short - if you already use WRDWRAP then Spell/400 should fit in nicely. If you want to start using WRDWRAP and Spell/400 then get used to the quirks and features of both in isolation :)

There is an example program - SCNCLPWW - which will show how Spell/400 works with word wrap.

Audit log

The audit log is a good place to track what happens with a spell check - each screen and prompt can be recorded and redisplayed within the log.

3.1.2 Putting into production

Environment

To simplify the parameters needed, Spell/400 takes many of its settings from the Spell 'Environment'. Such settings include the dictionaries used and which words are automatically accepted.

There is a default 'system' environment, which can be overridden by each job or application to create their own environment.

Library list

The SPELL400 library does not need to be in the library list. By calling OVRSPAENV or STRSPASCN the library will be placed in the second 'product library' slot - meaning it is in the library list but will not upset (or be upset by) the normal 'user' library list.

Number of Errors Remaining

Unlike previous versions, Spell/400 V3 has a more user-friendly approach to reporting the number of spelling mistakes that *remain*. Rather than report the ’real’ number of mistakes, it returns the number of mistakes that the user hasn’t checked.

For instance, if the user ignored a mistake (with F12 or the Ignore button) then V2 would say a mistake remained, but V3 would say the mistake had been accepted by the user, thus no mistake remained. In contrast, if the user pressed F3 to exit then the mistake would remain.

Disabling ADD

You can prevent the user adding words to the ADDWORDS dictionary by specifying it as *NONE in the OVRSPAENV command. The user is not presented with an ADD or F6=Add option and incorrect words have to be changed or ignored.

Audit controls disabling any key

The audit controls (2.5.1) can prevent individual F-keys and buttons from working.

3.2 “Hello World” tutorial

The following tutorials show spell checking programs can be developed, from the most basic usage to the point where you can take the concepts and apply them directly to your own programs.

3.2.1 Finding mistakes in text

The most basic usage of Spell/400 is simply to discover whether a text string has spelling mistakes or not, and then to return how many mistakes there are and finally to return a list of the mistakes.

3.2.1.1 Does the text have any spelling mistakes?

The simplest program simply reports whether a string has any spelling mistakes - it makes no attempt to correct them but returns the number of mistakes found.

1. First we need to pull in the definition for the ’SpellCheckString’ API. You can copy the definition from the APIILE member or just use a COPY directive:

   ```
    /COPY SPELL400/EXAMPLES,APIILE
   ```

2. Then we need some text to spellcheck, which we shall store in a field called TEXT

   ```
   D Text           S            500a    inz(’This stringhas two seperat misatkes’)
   ```

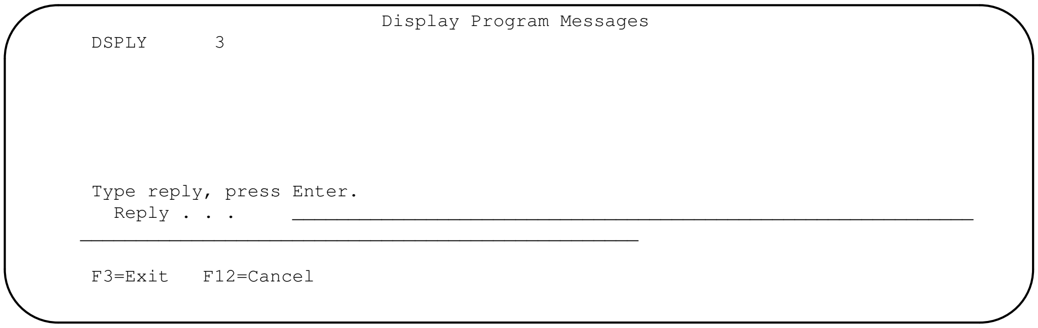
3. Lastly, we need to call the Spell API:

   ```
   c                   Callp     SpellCheckString
   c                               (TEXT:500:NbrRmn)
   ```

4. To show the number of mistakes we can use the DSPLY opcode

```
c                    dsply                  NbrRmn
```

5. You should have a program similar to the one called EXAMPLE1 (in SPELL400/EXAMPLES), and running it will produce a screen similar to this one



```
                           Display Program Messages
DSPLY      3






Type reply, press Enter.
  Reply . . .     _______________________________________________________________
________________________________________________________

F3=Exit   F12=Cancel
```

▷ Programming notes

- The Text field will always be overwritten by Spell/400 so ensure you allow for this.
- The full source is in the EXAMPLES sourcefile and at section **??** on page ??

3.2.1.2 What are the spelling mistakes?

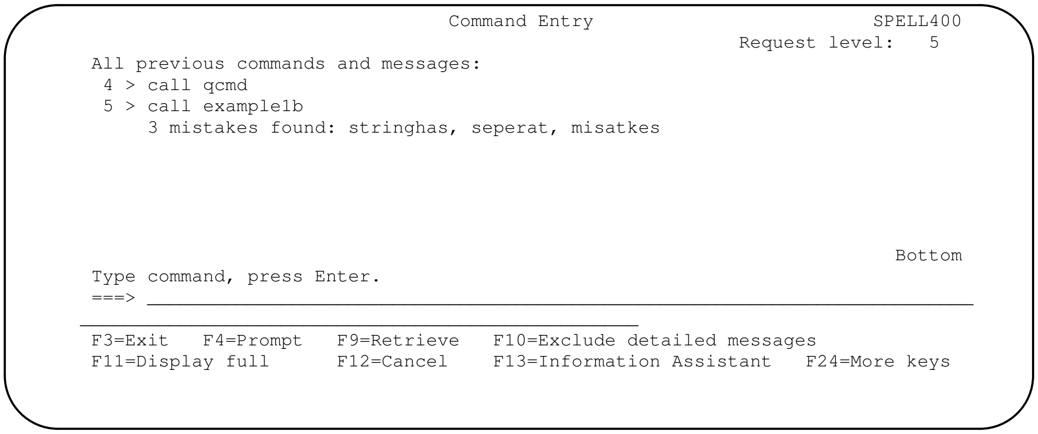
The next step is to show the spelling mistakes - no correcting is done, and the users is not prompted, but it's an easy way to check mistakes already existing in the database, especially if run within a batch job.

The example is stored as EXAMPLE1B and is identical except we have added a line to put the results in the joblog.

1. Instead of the DSPLY command we will display the results in the joblog using this API:

```
  * DSPJOBLOG to see the results of the spellcheck
c                    callp     SpellJoblog(
c                               + %triml(%editc(NbrRmn:'3'))
c                               + ' mistakes found: '
c                               + %subst(text:1:80) + CR)
```

2. When you run the program the results in the joblog should look something like this:



```
                                Command Entry                         SPELL400
                                                               Request level:   5
 All previous commands and messages:
  4 > call qcmd
  5 > call example1b
      3 mistakes found: stringhas, seperat, misatkes






                                                                          Bottom
 Type command, press Enter.
 ===> ___________________________________________________________________________
__________________________________________________
 F3=Exit   F4=Prompt   F9=Retrieve   F10=Exclude detailed messages
 F11=Display full      F12=Cancel    F13=Information Assistant   F24=More keys
```

▷ Programming notes

- The SpellJobLog is not a Spell/400 API but an IBM one stored in member TUTORIALIN
- The full source is in the EXAMPLES sourcefile and at section **??** on page ??

3.2.1.3 Using *CSV or *LST for a list of mistakes

The 4th parameter (which is called PROMPT) can specify how the mistakes are to be returned to your program. By specifying *CSV (or not specifying the parameter at all) you will receive a ’Comma Separated Variable’ list. If you enter *LST then the same mistakes are passed back but use a space to separate the words - using EXAMPLE1C you can enter *LST and see the response in your joblog:

```
4 > call tutorialc
    3 mistakes found: stringhas seperat misatkes
```

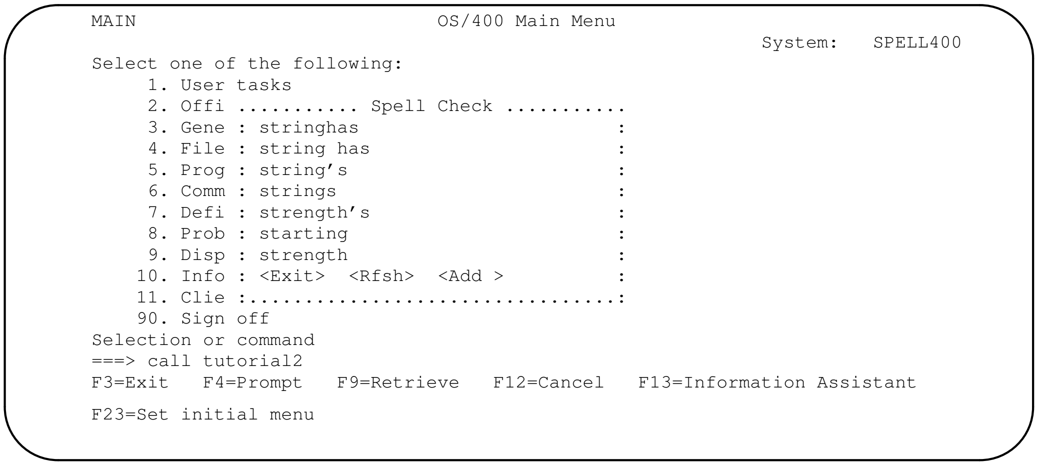
3.2.2 Prompt the user to correct the mistakes

The next stage after finding spelling mistakes is to prompt the user to correct them.

1. Using example TUTORIAL1C we simply change the PROMPT from *LST to *YES:

```
 * Find mistakes and prompt user to correct
c                    Callp     SpellCheckString
c                                (Text:500:NbrRmn:’*YES’)
```

2. Our copy is renamed to TUTORIAL2 and when run produces a popup window:



```
MAIN                           OS/400 Main Menu
                                                            System:   SPELL400
Select one of the following:
     1. User tasks
     2. Offi .......... Spell Check ..........
     3. Gene : stringhas                     :
     4. File : string has                    :
     5. Prog : string's                      :
     6. Comm : strings                       :
     7. Defi : strength's                    :
     8. Prob : starting                      :
     9. Disp : strength                      :
    10. Info : <Exit>  <Rfsh>  <Add >        :
    11. Clie :...............................:
    90. Sign off
Selection or command
===> call tutorial2
F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=Information Assistant
F23=Set initial menu
```

And the joblog will show the results

```
4 > call tutorial2
     0 mistakes remain: This string has separate mistakes
```

▷ Programming notes

- The NbrRmn parameter returns the number of mistakes remaining in the text
- However if the user presses F12 to ignore a mistake, it is considered to be accepted - i.e. it will not be returned as a mistake.
- Additionally, if the user presses F3 to exit then the NbrRmn will reflect the number of mistakes that still remain.
- In screen scraping examples a 'confirmation' window is presented which allows the user to discard the changes, however this is not available in this API (and so F3 doesn't discard changes, but accepts every change made).

3.2.2.1 Positioning the popup window

Now we are overwriting the screen the placement of the popup window is important

1. To change the popup window position, specify the Row and Column. Using TUTORIAL2B as the model, we change the call to row 5 and column 60:

```
 * Find mistakes and prompt user to correct
c                   Callp     SpellCheckString
c                               (Text:500:NbrRmn:'*YES':15:60)
```

▷ Programmers notes

- The Row and Column are specified as the position of the top/left corner of the window frame
- There is no need to work out whether the window will fit - positions will be adjusted if the window cannot fit
- There are several different windows, with different dimensions, and the Row and Col will be adjusted to ensure the window fits.
- The user can always reposition the window by moving the cursor anywhere on the screen (outside of the window) and pressing Enter

3.3 Highlighting mistakes onscreen

Spell/400 is at its most useful and user-friendly when it shows the mistakes on-screen, and puts the popup window near to, but not obscuring, the original text. It does this using the technique of 'screen scraping' where the text to be checked is already on the screen, and you just specify where to find it. Spell/400 can 'see' the errors onscreen, underline them, and position the window around them.

3.3.1 Spellchecking a field on screen

1. First we need a display file to show the text field. The important field is the Text field, which is defined as 500 characters long:

```
A            TEXT         500A  B  4 20CNTFLD(050)
A                                      CHECK(LC)
```

2. For the RPG program, we need to pull in the definition for the 'SpellCheckScreen' API. You can copy the definition from the APIILE member or just use a COPY directive:

```
 /COPY SPELL400/EXAMPLES,APIILE
```

3. Then we need some text to spellcheck, which we shall store in the field called TEXT

```
D Example         C                  'This stringhas two seperat misatkes'
c                   Eval      Text = Example
```

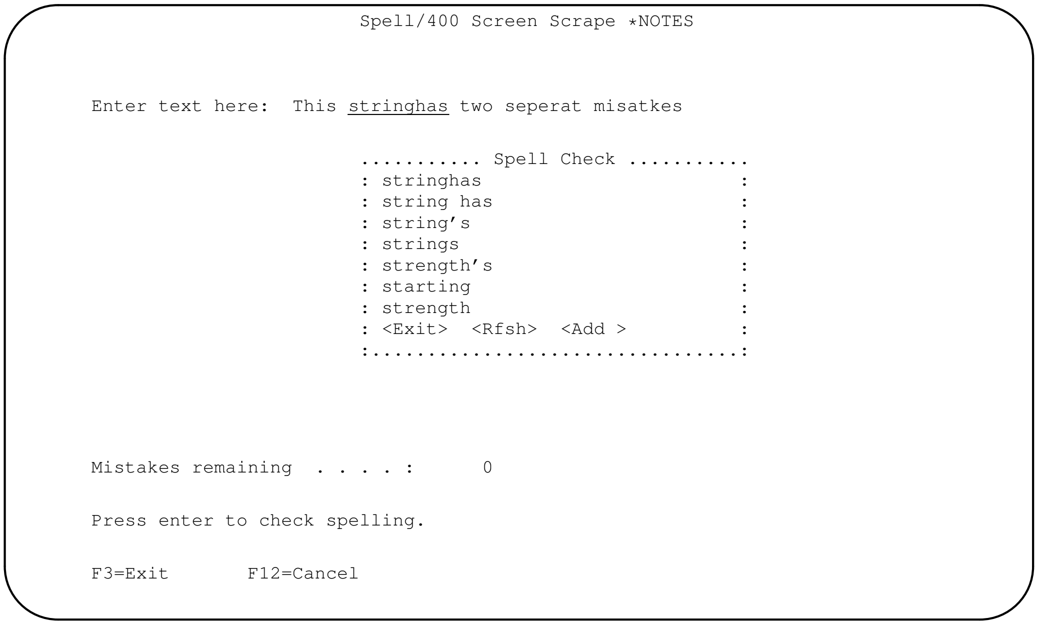
4. To display onscreen we need an EXFMT that loops until F3 is pressed

```
c                   Exfmt     screen
c                   Dow       *inkc=*off
```

5. Lastly, we need to call the Spell API. Here we know that the Text String is located in the display file on row 4, column 20 and is 500 characters long, wrapped over 10 lines using CNTFLD. This creates a block of text that starts at row 4 and ends on row 13, from columns 20 to 69.

```
c                   Callp     SpellCheckScreen
c                                (4:20:13:69:Text:NbrRmn)
** DEFN in SCREEN: A            TEXT         500A  B  4 20CNTFLD(050)
```

6. Once the variable definitions are declared you will have a program similar to the one called TUTORIAL3 (in SPELL400/EXAMPLES) and, using TUTORIALDF as the screen file you should see the following:



```
                              Spell/400 Screen Scrape *NOTES



      Enter text here:  This stringhas two seperat misatkes

                              .......... Spell Check ..........
                              : stringhas                     :
                              : string has                    :
                              : string's                      :
                              : strings                       :
                              : strength's                    :
                              : starting                      :
                              : strength                      :
                              : <Exit>  <Rfsh>  <Add >        :
                              :...............................:




      Mistakes remaining  . . . . :      0


      Press enter to check spelling.


      F3=Exit       F12=Cancel
```

▷ Programmers notes

- If the user presses F3 on the popup window they will get a chance to either accept the changes they have made so far, or discard them. If there are mistakes still pending the 'NbrRmn' parameter will be greater than 0.
- If the user presses F3 on the confirmation popup then the field returned will be blank

3.3.2 Real-world improvements to screen scraping

The problem with the TUTORIAL3 program is that there may be a number of situations where the Spell API returns blanks, which will overwrite the TEXT field. To counter that we should use 2 fields - one to hold the 'database' field that you want to check, and one to hold the 'screen-scrape' text that is returned.

1. Define the new field as identical to the text field:

```
D NewText          S                    Like(text)
```

2. We specify that SpellCheckScreen should return the text in this new field

```
c                    Callp     SpellCheckScreen
c                                 (4:20:13:69:NewText:NbrRmn)
```

3. And, lastly, we only use it if it is non-blank (otherwise we keep the database 'text' field as it was

```
c                    If        NewText <> *blanks
c                    Eval      Text = NewText
c                    Endif
```

▷ Programmers notes

- It's not obvious but you should remember that the screenscrape never returns your database field, but only what it finds onscreen so whilst you could use TEXT in both the display file and API, it's not a good idea
- The NewText field can return blanks if the user presses F3 and discards their changes - effectively it is blank because it has nothing useful to return, and the original text field should be used. The NbrRmn parameter will return the number of mistakes outstanding.
- The NewText field can also return blanks if there were no mistakes found, again it is better to use the original text field. In this situation the NbrRmn field will say 0.
- If the user press F3 but then 'Accepts' the changes made up to that point, the NewText field will be non-blank and the NbrRmn parameter will be positive.

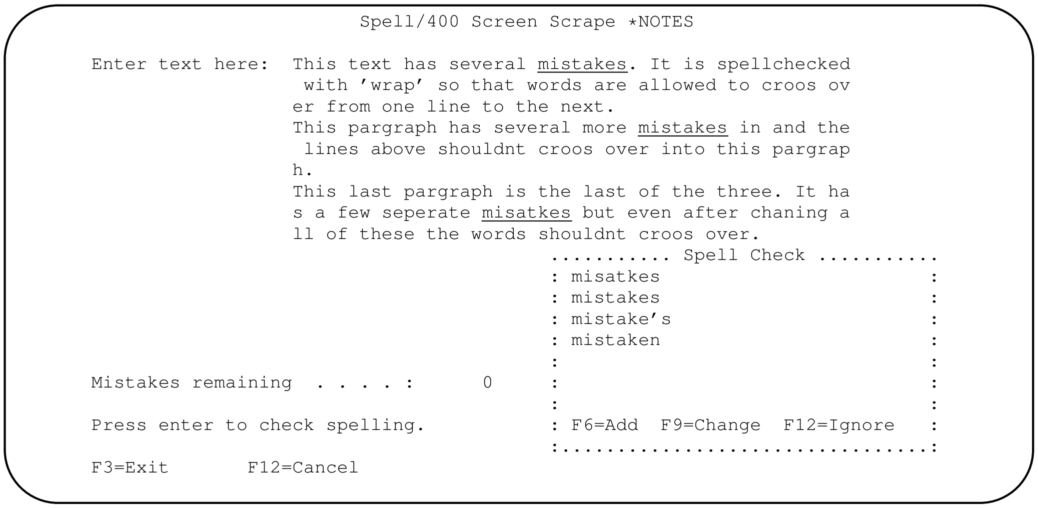
3.3.3 Screen scraping multiple rows as paragraphs

The API is capable of screen-scraping multiple lines. This concept is easily shown by using TUTORIAL3B and adding multiple lines to spellcheck, giving TUTORIAL3C.

1. For the example we have stored 10 lines of text at the bottom of the program, which we treat as a database field.

```
D ExampleArray    S             50    Dim(10) CtData
c                   Movea     ExampleArray  Text
```

2. Otherwise TUTORIAL3C is identical - text is assumed to flow from one line to another.



```
                        Spell/400 Screen Scrape *NOTES

Enter text here:  This text has several mistakes. It is spellchecked
                   with 'wrap' so that words are allowed to croos ov
                  er from one line to the next.
                  This pargraph has several more mistakes in and the
                   lines above shouldnt croos over into this pargrap
                  h.
                  This last pargraph is the last of the three. It ha
                  s a few seperate misatkes but even after chaning a
                  ll of these the words shouldnt croos over.
                                        .......... Spell Check ..........
                                        : misatkes                       :
                                        : mistakes                       :
                                        : mistake's                      :
                                        : mistaken                       :
                                        :                                :
Mistakes remaining  . . . . :      0    :                                :
                                        :                                :
Press enter to check spelling.          : F6=Add  F9=Change  F12=Ignore  :
                                        :................................:
F3=Exit      F12=Cancel
```

▷ Programmers notes

- Note that the words span across lines, so this is only suitable for 'paragraphs' of text.
- Note that Spell/400 has a concept of paragraphs - so that one paragraph increasing in length does not affect a paragraph below it, provided there is enough room for both.

3.4 Some advanced concepts

3.4.1 Multi-page cache

The multi page cache is used to simulate a single spell check over multiple screens, using a cache to store and re-use the user actions.

The problem

When calling the spell checking APIs consecutively - for instance, over multiple screens - the user may press 'Ignore All' or 'Change All' and expect the 'All' to apply to all the spell checks on all the screens.

The solution

Spell/400 can store the 'Ignore All' and 'Change All' requests in a cache, and apply them to every API call until the cache is closed.

In essence, it allows the F21/F24 or ChgAll/IgnAll buttons to be carried over onto another 5250 screen. As an example, if there are several screens of information with the same 'mistake' that requires an F24=Ignore All, then the F24 will be carried from one screen to the next. It doesn't matter if the screens are in different programs.

An example

The Multi Page cache can easily be demonstrated with the following example program;

```
CALL SCNMPCR
```

The example program uses a customer name and address, which is spell checked. On the next page the same name and address is used for billing and shipping and can be spell checked. The multi-page cache 'remembers' F24=Ignore All.

Subfiles with Multi-page cache

More advanced demonstrations are supplied with the NOTPAD command. This command calls one of several programs to allow typical free-format notepad-like text entry. Two of the programs use subfiles to page through the text, and two use large continuation fields (ie large fields with DDS CNTFLD specified). There are various differences programmed in but essentially they all work the same way to demonstrate real-world uses of the multi-page cache.

3.4.2 Screen overlays

Screen overlays are used by Spell/400 trigger programs.

The problem

When you can't access source code of an application you are relying on the database trigger to kick off the spellchecker. However by this time the screen may have changed and prompting the user to correct text on-screen doesn't work.

The solution

Screen overlays take a screenshot of your data-entry screen - the one that needs spell checking - and allows you to edit it. Then, when the spell check is triggered it can display this overlay and fill it with the text to be checked. It can then proceed with highlighting and correcting the text on-screen.

An example - where screen overlays are not needed

An order shipping application doesn't have any source code, so cannot be altered to use Spell/400 APIs. However the dispatch notes often have spelling mistakes which need to be corrected. When the dispatch notes are entered, the application writes them to a file which has a trigger, which invokes Spell/400 to check the notes. Spell/400 asks the user to correct the words whilst they are still onscreen, and then writes the corrected text to the file. No screen overlays are needed to achieve this.

An example - where screen overlays can help

The preceding example above assumes that the screen is still showing the dispatch notes on screen, when in fact there might be an 'order confirmed' or blank screen. Peforming the spell checking whilst the 'order confirmed' screen is active is confusing to the user, but we have no control over the application.

A screen overlay is a screen shot of the 'dispatch notes' screen which has been pre-prepared for this situation. When the dispatch notes are spell checked, Spell/400 writes the overlay onscreen before showing the text that needs to be corrected. Once the spell checking is complete the overlay is removed and the application regains control - showing the 'order confirmed' screen or whatever it would do before Spell/400 was introduced.

Displaying and editing screen overlays

To enable this, you need to capture the dispatch notes screen and store it as an overlay, then the trigger can redisplay it along with the dispatch notes.

For more information, and an example see 4.4 on page 63.

3.4.3 Highlighting unusual words

'Unusual' or 'suspect' words can be highlighted. An example might be to have UK-spellings highlighted in USA, or vice-versa. Or to always highlight the very-commonly misused word 'loosing' in place of 'losing'. This is achieved by having more words in the suggestions list than the spelling list - if the word is not found in the SPELLING dictionaries, but is found in the SUGGESTIONS dictionaries it is shown as a possible problem.

For instance, if you wanted to accept but highlight US spellings at a UK site you would put NORMALUS in the suggestions list - a US spelling would be found, highlighted and accepted - ensuring the user knows the spelling is acceptable but non-standard.

4 Programming Reference

4.1 Spell Check Screen Text

The Start Spelling Screen (STRSPASCN) command allows you to check the spelling of a field on screen and prompt the user to correct any mistakes. Any mistakes found are highlighted on screen and the user is aware of the context of the mistake.

4.1.1 STRSPASCN Command and StartSpellScreen API

The command requires an area of screen where text can be ’scraped’ and checked. The command then receives the corrected text.

Common uses

Spell checking text on a user screen

Spell checking text without changing your application source code

4.1.1.1 Parameter list

1	STRROW	Start row	Input	*	Decimal(3)	Range 1-27
2	STRCOL	Start column	Input	*	Decimal(3)	Range 1-132
3	ENDROW	End row	Input	*	Decimal(3)	Range 1-27
4	ENDCOL	End column	Input	*	Decimal(3)	Range 1-132
5	TEXT	Returned text	Output	*	Character	Any length
6	NBRERRRMN	Number of errors remaining	Output		Decimal(5)	
7	FORMAT	Text formatting	Input		Character(10)	
8	SELECT	Select fields	Input		Character(10)	
9	OVERLAY	Overlay stored screen	Input		Character(10)	
10	WINROW	Window start row	Input		Decimal(3)	Range 1-27
11	WINCOL	Window start column	Input		Decimal(3)	Range 1-132
12	MULTIPAGE	Multi Page Cache	Input		Character(10)	Reserved values

4.1.1.2 Parameter descriptions

STRROW Start row

Specifies the starting row where the on-screen text can be found. This is a required parameter.

The HLL API requires a decimal 3,0 packed field.

STRCOL Start column

Specifies the starting column where the on-screen text can be found. This is a required parameter.

The HLL API requires a decimal 3,0 packed field.

ENDROW End row

Specifies the ending row where the on-screen text can be found. This is a required parameter.

The HLL API requires a decimal 3,0 packed field.

ENDCOL End column

Specifies the ending column where the on-screen text can be found. This is a required parameter.

The HLL API requires a decimal 3,0 packed field.

TEXT Returned text

Specify the field that should receive the corrected text. The minimum size is governed by the area being screen scraped. A smaller size may cause problems.

If an OVERLAY is used, then this field must also contain the text to be checked.

The field size must be at least as large as the screen area captured.

NBRERRRMN Number of mistakes remaining

Specifies the name of a variable used to retrieve the number of spelling mistakes that remain in the TEXT string.

A value of 0 indicates that no spelling mistakes remain.

Each spelling mistake that the user ignores or corrects or otherwise addresses is removed from this count. Thus if there are no mistakes or if the user has ignored them all the count will still be zero.

The HLL API and command require a 5,0 packed field.

FORMAT Text formatting

Specifies the type of formatting to apply to text scraped from the screen.

The possible values are:

*NOTES Text notes are text strings continued over several fields or lines. A word can begin at the end of one line and continue on the next line. An example would be the DDS CNTFLD command where a single, long text string can be displayed over several lines. This was previously referred to as *WRAP which will still work, but is called *NOTES to distinguish the concept from DDS WORDWRP.

*LINES	Lined text is where individual text strings are on separate lines. An example would be lines of a subfile, where the words on one line cannot be moved onto the next line.

SELECT Select fields

Specifies whether the screen area (STRROW/STRCOL to ENDROW/ENDCOL) contains a single block of text or whether it contains individual fields which are to be selected.

For instance, a single screen with 3 separate fields that all start at column STRCOL can be spell checked in one go, but the words do not cross from one field to the next.

If in doubt, simply run the spell check for each field separately.

The possible values are:

*ALL	All lines within the scrape area should be checked.
*INPUT	Only lines with an underline at column STRCOL-1 should be checked.

OVERLAY Overlay stored screen

Specifies the name of a screen that should be overlayed before the text is checked.

The screen should have been previously captured with STRSPASCNC, and this parameter should specify the name of the dataarea storing the screen.

When using an overlay, the TEXT field must contain the text to be checked, and it is written to the screen.

The possible values are:

*NONE	Do not overlay the screen.
screen-name	The name of the dataarea where the screen is stored.

WINROW Position window at row

Specifies the top left position of the pop-up window.

Normally the spell checking window is positioned close to where the mistake is on-screen. This allows the spell checking window to appear at a fixed position.

The HLL API requires a decimal 3,0 packed field.

WINCOL Position window at column

Specifies the top left position of the pop-up window.

Normally the spell checking window is positioned close to where the mistake is on-screen. This allows the spell checking window to appear at a fixed position.

The HLL API requires a decimal 3,0 packed field.

MULTIPAGE Multi Page Cache

Specifies whether a page cache should be created or checked across spellchecks.

A page cache stores the F21 and F24 key strokes that replace or ignore all words in a spellcheck. This cache can be checked on subsequent spellchecks (i.e. other screenscrapes) and automatically applied. This allows 'change-all' and 'ignore-all' to effectively work across multiple pages.

Since V3R8 it now stores F6 key strokes.

Since V3R8M2 the confirmation screen is suppressed until *LAST is specified.

The possible values are:

*NONE	Do not check or create a cache.
*NEW	Create a new cache (and erase the contents of the previous cache).
*CHECK	Check and add to the current cache
*LAST	Check and add to the current cache, present confirmation screen

4.1.2 STRSPASCNQ Command and StartSpellScreenQ API

Please note that the STRSPASCNQ command and API have been replaced (using Multi-Page Caches in the normal STRSPASCN). We will support and maintain the STRSPASCNQ commands but they will not be enhanced and will not feature in V4.

The command requires several areas of screen where text can be 'scraped' and checked. The command then receives the corrected text.

The command and API are identical to STRSPASCN except that multiple screen scrapes can be identified. Each area is identified with each invocation of the command or API and the spell checks are 'queued'. On the last invocation all the spell checks are performed as one.

Common uses

Spell checking several fields of text on a user screen

Notes on usage

1. The screen scrape is done in one go, when the queue is actioned.
2. When the spell checks are queued the text variables must be unique. ie when STRSPASCNQ is invoked the pointer to the text variable is stored, when the queue is actioned the original pointers to the text areas are used.

4.1.2.1 Parameter list

1	STRROW	Start row	Input	*	Decimal(3)	Range 1-27
2	STRCOL	Start column	Input	*	Decimal(3)	Range 1-132
3	ENDROW	End row	Input	*	Decimal(3)	Range 1-27
4	ENDCOL	End column	Input	*	Decimal(3)	Range 1-132
5	TEXT	Returned text	Output	*	Character	Any length
6	NBRERRRMN	Number of errors remaining	Output		Decimal(5)	
7	FORMAT	Text formatting	Input		Character(10)	

8	SELECT	Select fields	Input		Character(10)	
9	OVERLAY	Overlay stored screen	Input		Character(10)	
10	WINROW	Window start row	Input		Decimal(3)	Range 1-27
11	WINCOL	Window start column	Input		Decimal(3)	Range 1-132
12	QUEUE	Queue Action	Input		Character(10)	Reserved values

4.1.2.2 Parameter descriptions

The parameter list is identical to STRSPASCN for the first 11 parameters and then has the 'Queue' parameter.

QUEUE Queue action

Specifies the action to perform on each invocation. Each invocation is queued (i.e. not executed) until the final invocation where the QUEUE parameter must be *IMMED.

If STRSPASCNQ is invoked once with a queue action of *IMMED it performs as if a STRSPASCN command was run.

The HLL API requires a character 10 field.

The possible values are:

*QUEUE Queue the spell check

*IMMED Perform all queued spell checks. The invocation which has a QUEUE action of *IMMED is also spell checked.

4.2 Spell Check a String

The Start Spelling Correction (STRSPACHK) command allows you to check the spelling of a string and optionally prompt the user to correct any mistakes. The command doesn't rely on the screen and the user is given no indication of the context of the string.

4.2.1 STRSPACHK Command and StartSpellString API

The command requires a text string which it checks and returns to your program. The string may be contain the original mistakes, or be corrected - this is determined by the parameters and the user.

Common uses

- Retrieve the number of mistakes in a text string
- Check a string for mistakes without user action
- Check a string for mistakes in batch
- Check database records for mistakes
- Prompt the user to correct text not on screen
- Prompt user to correct text which may not be in a fixed location on screen (e.g. subfile record)
- Spell checking without reference to screen position (e.g. webfacing)

Notes

Unlike STRSPASCN, there is no 'Confirm Acceptance' window at the end of the spell check.

4.2.1.1 Parameter list

1	TEXT	Text to check	Input	*	Character	Any length
2	LEN	Length of text	Input	*	Decimal(5)	Range 1-32767
3	NBRERRRMN	Number of errors remaining	Output	*	Decimal(5)	
4	PROMPT	Prompt user	Input		Character(10)	**YES*, *NO, *CSV, *LST
5	WINROW	Window start row	Input		Decimal(3)	Range 1-27
6	WINCOL	Window start column	Input		Decimal(3)	Range 1-132

4.2.1.2 Parameter descriptions

TEXT Text to check

Specifies the text containing the words to check. The field size can be up to 32767 characters. This is a required parameter.

The corrected text is returned in this field.

LEN Length of text

Specifies the length of the text to check. The length can be between 1 and 32767 characters. This is a required parameter.

The length is used to determine the number of characters of text which should be checked, rather than the actual size of the field.

The API requires a decimal 5,0 packed field.

NBRERRRMN Number of mistakes remaining

Specifies the name of a variable used to retrieve the number of spelling mistakes that remain in the TEXT string. This is a required parameter.

It should be a 5 position decimal variable.

A value of 0 indicates that no spelling mistakes remain. Each spelling mistake that the user ignores or corrects or otherwise addresses is removed from this count. Thus if there are no mistakes or if the user has ignored them all the count will still be zero. Any other positive number reflects the number of mistakes detected but not corrected.

PROMPT Prompt user to correct

Specifies whether the user should be prompted to correct any mistakes.

The possible values are:

*YES	Prompt the user
*NO	Do not prompt the user. This effectively returns the total number of spelling mistakes found, and could be used when user input is not desired.
*CSV	Return a comma-separated list of spelling mistakes in the TEXT field

*LST	Return a blank separated list of spelling mistakes in the TEXT field

WINROW Position window at row

Specifies the row where the top left corner of the pop-up window should start.

The possible values are:

*blank	Place the window in the centre of the screen
Number	Enter a row position between 1 and 27

WINCOL Position window at column

Specifies the column where the top left corner of the pop-up window should start.

The possible values are:

*blank	Place the window in the centre of the screen
Number	Enter a column position between 1 and 132

4.3 Spell Environment

The Override Spell Environment (OVRSPAENV) command allows you to change the local environment for the current job. The Display Spell Environment (DSPSPAENV) allows you to display the current (job) environment.

The Save Spell Environment (SAVSPAENV) command allows you to save the job settings for later retrieval with the Load Spell Environment (LODSPAENV) command. The Delete Spell Environment (DLTSPAENV) command deletes saved environments.

The Delete Spell Override (DLTSPAOVR) allows you to remove any application overrides, but keep any user preferences.

4.3.1 OVRSPAENV Command

The command is identical to CHGSPAENV which is documented on page 24 and in the help text.

OVRSPAENV takes effect across the entire job until DLTSPAOVR, LODSPAENV, DLTSPAENV or another OVRSPAENV is issued.

OVRSPAENV only affects those parameters which are specified, all other parameters are taken from previous OVRSPAENV commands or the system wide Environment.

Common uses

Change the dictionaries used just before spell checking

Change which words are accepted just before spell checking

Modify the current environment without changing all the settings

4.3.1.1 Parameter list

1	DICTIONARY	Spelling dictionaries	Input	Array of Character(10)	Max 8
2	SUGGESTION	Suggestion dictionaries	Input	Array of Character(10)	Max 8
3	ADDWORDS	Add words to dictionary	Input	Character(10)	
4	AUTOLEARN	Add mistakes to AutoChange dictionary	Input	Character(10)	
5	UIM	User Interface Manager	Input	Character(10)	
6	ACPUPR	Accept all words in upper case	Input	Character(10)	
7	ACPNUM	Accept all words with numerals	Input	Character(10)	
8	ACPUNA	Accept all unapproved words	Input	Character(10)	
9	ACPAUT	Accept all AutoChange words	Input	Character(10)	
10	BACKGROUND	Change background to blue	Input	Character(10)	
11	CONFIRM	Ask for confirmation	Input	Character(10)	

4.3.1.2 Parameter descriptions

The parameters are identical to those of CHGSPAENV which are documented on page 24 and in the help text.

4.3.1.3 API considerations

In addition to the command help, the following applies to the API:

1. Any of the parameters can be omitted.
2. Any of the special values can be specified.
3. The spelling and suggestions dictionaries can be passed into the API in two ways:
   (a) As an array with 8 rows of character(10), with one dictionary per row.
   (b) As a comma-deliminated string, e.g. COMMON,STANDARD,*USER

4.3.2 Save Spell/400 job environment (SAVSPAENV)

The Save Spell/400 environment (SAVSPAENV) command allows you to save the current job environment to a named dataarea that can be loaded at a later time with LODSPAENV.

The saved settings can be used for specific applications or users, allowing specific dictionaries to be used when entering or leaving an application.

You can display the settings that will be saved with DSPSPAENV *JOB.

Common uses

Store the dictionaries and settings used in a particular application
Store a users settings

Parameter list

ENV	Environment name	Char(4)

4.3.2.1 Parameter descriptions

ENV Environment name

Specifies the name to give to the saved environment. If the saved environment already exists it will be overwritten.

The possible values are:

environment-name The name used to store and retrieve the environment. The 4 characters are appended to 'BNBENV' to create a unique dataarea. If the dataarea already exists it is overwritten.

4.3.3 Load Spell/400 environment (LODSPAENV)

The Load Spell/400 environment (LODSPAENV) command allows you to load environment settings previously used and saved with SAVSPAENV. The saved settings overwrite the current job environment.

The saved settings can be used for specific applications or users, allowing specific dictionaries to be used when entering or leaving an application. You can display the new settings with DSPSPAENV *JOB.

Common uses

Retrieve the dictionaries and settings used in a particular application

Retrieve a users settings

Parameter list

ENV	Environment name	Char(4)

4.3.3.1 Parameter descriptions

ENV Environment name

Specifies the environment to load.

The possible values are:

*PRV The settings before the last LODSPAENV or OVRSPAENV are reloaded. This allows a program to load a special environment for an application, and when the application ends, to restore the previous settings.

environment-name A named environment previously saved with SAVSPAENV is loaded.

4.3.4 Delete Spell/400 environment (DLTSPAENV)

The Delete Spell/400 environment (DLTSPAENV) command allows you to delete the stored job environment previously saved by SAVSPAENV.

This command can 'delete' the current job environment, although you should use the DLTSPAOVR command to remove the OVRSPAENV overrides whilst keeping the UIM user preferences.

Parameter list

ENV	Environment name	Char(4)

4.3.4.1 Parameter descriptions

ENV Environment name

Specifies the environment to delete.

The possible values are:

*JOB The job settings are deleted (and system settings are used).

environment-name A named environment previously saved with SAVSPAENV is deleted.

4.3.5 DLTSPAOVR API

The command is similar to the OS/400 DLTOVR command in that it removes the previous OVRSPAENV override.

Common uses

Allows an OVRSPAENV to be limited in scope by deleting the override.

4.3.5.1 Parameter list

1	KEEPUIM Keep user preference for UIM	Input	Character(10)	

4.3.5.2 Parameter descriptions

KEEPUIM Keep user preference for UIM

Specifies if the user preference for UIM (stored in the local environment) should be kept.

The possible values are:

**YES* The user preference is kept.

*NO The user preference is deleted.

4.4 Spell/400 Screen Overlays

The STRSPASCN overlay feature allows pre-captured screens to be displayed by a trigger program to 'fake' screen scrape. Please see the 'Car Warranty' evaluation programs to better understand when this is useful:

```
GO BNBSPACW
```

Not all screens can be captured, and problems may arise with the following:

132-column wide screens
screens with pop-up windows
screens with attributes which rely on a certain device state

If you need to capture a screen that is to be used on different terminal emulators then it's best to capture it on the least capable screen (e.g. a dumb terminal). If all your terminal emulators are the same (for instance they are all Client Access) then there shouldn't be any problem with pop-up windows or attributes.

4.4.1 STRSPASCNC

The Start Screen capture (STRSPASCNC) command allows you to run an application normally, and when you press the ATTN key the screen is captured and stored in a data area.

Common uses

Store 'model' screens

4.4.1.1 Parameter list

SCREEN Name of screen captured

> Specifies the base name of the screen(s) captured. Each screen will be named with this name followed by an ascending sequence from 01 to 99.
>
> The screens are stored in data areas in library SPELL400 and can be used independently of each other. i.e. capture as many screens as you want then feel free to rename or remove any of the data areas.

4.4.1.2 Running STRSPASCNC

When you start the screen capture you will be presented with a command line. Call your application normally and when you want to store a screen press the ATTN key.

The first screen will be stored in the dataarea specified in the SCREEN parameter. Subsequent presses of ATTN will store screens in sequentially numbered dataareas.

When you have finished, F12 all the way to the command line window and then back out of that. You can work with the dataareas, delete some you don't want or rename others.

4.4.2 DSPSPASCNC

The Display Screen Capture (DSPSPASCNC) command allows you to display a captured screen from one the dataareas.

Before determining that a screen capture is suitable for every screen in your organization, it may be wise to use DSPSPASCNC to check it displays well on various types of display.

When displaying the dataarea you can press F8 to change the screen.

4.4.3 CHGSPASCNC

The Change Screen capture (CHGSPASCNC) command allows you to.change a captured screen from one the dataareas.

Common uses

Convert a real screen capture to a 'model' screen (e.g removing dates/times)

4.4.3.1 Running CHGSPASCNC

When you edit the dataarea you will be able to remove the screen attributes simply by blanking over them so be careful where you type. Of course, it's simple to duplicate a dataarea and experiment on that first.

Note that a lot of screen scraping multiple fields requires that the underline attribute remains in place to identify input-capable fields. If these are accidentally removed then the screen scrape will be misaligned or fail altogether.

4.5 Checking without Source Code

Spell/400 can check the spelling of any text on screen, and prompt the user to replace it, without any applications being changed. The only real problem is how to replace the text in the application.

Spell/400 does this with database triggers. Whenever text is spell checked it's almost inevitable that the corrected text is stored in a database file. When the data is stored or changed, a trigger invokes the spell checker - whilst the text is still on screen - and the user is prompted to correct the text. Then the trigger writes the corrected text to the database file.

4.5.1 Simple trigger

The simple trigger is program SCNTRGRPG (see SPELL400/EXAMPLES) which is added to file SPADFUP. The trigger is setup so that if the file is edited with UPDDTA then the program knows where to screenscrape.

By using the *INSERT and *UPDATE trigger methods, any text added or changed using UPDDTA is checked. This method will not work if the file is updated with any program other than UPDDTA (although the trigger can find out whether or not UPDDTA was used, and adapt the screenscrape accordingly).

4.5.1.1 Adding a trigger

Add a trigger for *INSERT (adding records). Note that the TRGTIME must be *BEFORE and ALWREPCHG must be *YES.

> ADDPFTRG FILE(SPADFUP) TRGTIME(*BEFORE) TRGEVENT(*INSERT) PGM(SPELL400/SCN TRGRPG) RPLTRG(*YES) ALWREPCHG(*YES)

And do the same for *UPDATE:

> ADDPFTRG FILE(SPADFUP) TRGTIME(*BEFORE) TRGEVENT(*UPDATE) PGM(SPELL400/SCN TRGRPG) RPLTRG(*YES) ALWREPCHG(*YES)

4.5.2 Triggers and screen overlays

The advanced use of triggers - especially using overlay screens - is probably better explained with the supplied CARWAR* (Car Warranty Application) examples. The trigger mechanism is the same, however the overlay screens allow the screen to be changed before screen-scraping.

4.6 Programming Examples

The evaluation contains several examples using the APIs and programs. The following explains some of the code and options.

4.6.1 RPG/IV procedures

Here is the STRSPACHK procedure definition in RPG/IV:
The minimum required are the text to check and how long it is:

```
D SpellCheck      PR                  ExtPgm('SPELL400/STRSPACHK')
D  TextToCheck              32767A   options(*varsize)
D  LengthOfText                 5P 0 const
```

From this you could spell check any text, for instance:

```
C              Eval     Text='This has a spelling misatke'
C              Callp    SpellCheck(Text:100)
```

However there are a few extra parameters you can take advantage of, and the full definition is:

```
D SpellCheck      PR                  ExtPgm('SPELL400/STRSPACHK')
D  TextToCheck              32767A   options(*varsize)
D  LengthOfText                 5P 0 const
D  ReturnNbrError               5P 0 options(*omit:*nopass)
D  ShowWindow                   4    const options(*omit:*nopass)
D  WindowRow                    3P 0 const options(*omit:*nopass)
D  WindowColumn                 3P 0 const options(*omit:*nopass)
```

You can still do a simple spell check, or provide some or all of the optional parameters:

```
c                   Callp     SpellCheck(Text:100:NbrRmn)
c                   Callp     SpellCheck(Text:100:NbrRmn:'*YES':StrRow)
c                   Callp     SpellCheck(Text:100:*omit:*omit:StrRow:StrCol)
```

4.6.2 RPG/IV prototypes

For convenience, all the prototypes are stored in a copy member. To use it just insert the following statement in the D-specs of your program:

```
/COPY SPELL400/EXAMPLES,APIILE
```

The procedures are called:

SpellCheckString:

```
c                   Callp     SpellCheckString(Text:100:NbrRmn:'*YES':StrRow)
```

SpellCheckScreen:

```
c                   Callp     SpellCheckScreen(7:7:14:76:Text)
```

4.6.2.1 Triggers copy member

There is also a member for triggers, which define most of the fields you need. If you use:

```
/COPY SPELL400/EXAMPLES,APITRG
```

then everything up to the *ENTRY PLIST is defined so the old fields are prefixed o_ and the new fields are prefixed n_.