I like SPF/Pro. I've been using its editing capabilities for around two decades. It's implementation is extraordinarily faithful to IBM's ISPF, yet it has all the extensions you would want to have as a Windows program...

- Dave Cole

Index

Things I Like About SPF/Pro

- First of all, its implementation is so faithful to ISPF, that the learning curve is almost nonexistent.
- It has full support for Window's file naming conventions, including long names and nearly arbitrary character content.
- It has full support for window resizing.
- It has full support for using SPLIT to create any number of edit sessions and then SWAP'ing around them. In fact, its SPLIT command is a bit easier to use that ISPF's.
- It's file I/O is amazingly fast! z/XDC's entire source content can be searched in less than 30 seconds.
- It takes advantage of Window's file caching, so that 2nd and subsequent searches finish in less that 5 seconds!
- You can define up to four sets of 12 ISPF style PF-keys onto the keyboard's 12 function keys: Bare, SHIFT'd, CTRL'd and ALT'd.
- Its profile support keys off of a file's extension.
- Normally, it processes all files as containing ASCII text; however, you can set up a profile that processes a file as EBCDIC. (But I usually don't bother with this. For our needs, ASCII's just fine.)

Problems with SPF/Pro

Sadly, Command Technologies orphaned SPF/Pro around 15 years ago. But it is a testament to the quality of its design and code that it is still highly functional so many Windows versions later.

But there are bugs:

Crash - Profile Command

A PROFILE [name] 0 command will crash SPF/Pro every time. The workaround is: **PROFILE** *[name]*;**RESET SPECIAL**.

Crash - Help Command

Another way to crash SPF/Pro is to try to use its **HELP** command. The workaround is to read the **SPFPRO.HLP** file directly. It is a text file located in **C:\Program Files (x86)\Command Technologies\SPFPRO\HELP**.

Crash - Large Files

For the most part, SPF/Pro can handle huge files (dumps, for example) just fine, but there is a bug related to file size:

- When the file being edited is longer that 64K bytes and is located on a path that is "too long" *[or maybe it's that the path contains blanks, I don't know, I haven't researched it]*, SPF/Pro will crash.
- But if you move the file to a location that has a sufficiently short path, it will open just fine. Example: I'm using SPF/Pro to edit files in **F:\XDC\git\XDC\base** with no problems.

Crash - Random

Occasionally, SPF/Pro will randomly crash for no clear reason. One thing that seems to trigger intermittent crashes is using a **FIND** command at the folder level for a very large folder (such as **F:\XDC\git\XDC\base**, for example). Several such **FIND**s will succeed just fine, but then after 8 or 10 (or so) times, SPF/Pro will fail with an Access Error.

Crash - Recovery: It's a Process!

When SPF/Pro crashes, it will crash only the current window. All swap sessions within that window will crash, but other windows will remain untouched. **Moral**:

- Don't use Swap Sessions.
- Do use multiple Windows, instead.
- When you open your first session, **do not use it!**
  - Instead, navigate it to a parking place and then just leave it alone.
  - A good parking place would be in a folder view somewhere.
  - This 1st session will be your recovery session of last resort.
  - From sad experience, I have found that it is insufficient to park the first session at the PRIMARY OPTION PANEL. Apparently, global data does not get rewritten at close time unless a session actually does something.
- Then open a 2nd session (then a 3rd, then more as needed), and do your work in those.
- Then when one of your working sessions fails, and it klotzes your global data such that new session attempts also fail:
  - Then close another session to overwrite the bad global data. That should cure the problem.
  - If the failures are not cured, then continue closing sessions one by one until the failure is cured.

If you close all sessions without curing the problem then you will have to recover from backup files.

Once or twice (because of my own carelessness) I've gotten into a situation where SPF/Pro refused to start. Eventually I recovered by trying each of the following more or less one by one in the following order:

- I used **zTree** to display **C:\ProgramData\COMMAND TECHNOLOGIES\SPFPRO\PROFILES**, sorted in date order.
- I deleting everything at and newer than the failure time point.
- I restored the deleted files from a backup I keep at **F:\product installers\Command Technology\Recovery\ProgramData**.

If that did not cure the failure, then I tried doing the same process for **C:\Program Files (x86)\ Command technologies\SPFPRO\PROFILES**.

Problem - Window Lock

Sometimes when SPF/Pro crashes, it's window will lock up, and you won't be able to move it, close it, minimize it, or anything it. Then you will have to use the Task Manager (ctrl-alt-delete) to close it. But when you look at its Tasks List, there will be at least as many SPFPro tasks as you have windows open. How to chose, how to choose... All I know is that you just start killing SPFPro tasks at random, until you finally nail the right one.

Pro Tip: !Microsoft has a cool little downloadable utility named **Process Explorer.** It has a mechanism for discovering exactly which Process owns a given window.

Problem - CPU Loop

Sometimes SPF/Pro will fall into a non-locked CPU loop, but...

- In a multi-CPU PC, this is rarely even noticeable.
- But if you do notice it, it's easy to break out of. Just press **F2** followed by **F3**...
  - **F2** is **FSPLIT**. It creates a new edit session.
  - **F3** is **END**. It ends that session.

Problem - RESET FIND

This is not implemented. (It was added to ISPF by IBM after SPF/Pro was developed.) The workaround, of course, is **FIND garbagestring**.

Problem - FIND ALL Issue

When used at the folder level, **FIND ALL** won't hurt anything, but it can give you false positives. This is annoying of course, but it's not really a big deal.

I've **not** seen instances of false negatives.

Problem - Disappearing Cursor

Sometimes the cursor will flat out disappear. To get it back:

- Just toggle the insert key,
- Or swap out of then back in to the SPF/Pro window.

Problem - Edit masks containing quotes (')

You can create an edit mask containing single quotes, but they cannot be permanently save into a profile. The next time you load the profile, all parts of the mask from the quote and rightwards will be gone. Workaround:

- Use any other character in the quote's position.
- Then when you load that profile, use **PROFILE name 9**.
- Then move the cursor to your stand-in character, and type a quote there.
- Finally issue **RESET SPECIAL** to remove the profile description messages from your display.

The quote will persist until the next time you **PROF READ** the profile.

Problem - Cut/Paste and Clipboard Errors and Unintended CRLFs

Both cut and paste have issues:

- When you use **cut** (ctrl-c or ctrl-x) to copy/move a string from an SPF/Pro edit session to the clipboard, SPF/Pro will **insist** on appending a CRLF to the string.
  - Then when you paste this string into (for example) a DOS prompt, the CRLF will cause DOS to immediately process that string before you have a chance to edit it.
  - To prevent premature execution problems, before pasting the string into a DOS prompt, I will put a garbage character on the command line, so as to cause the automatic processing of the command to fail.
  - I can then retrieve the failed command and edit it the way I want before it processes for real.
- When you use **paste** (ctrl-v) to paste a string **into** an edit session, the paste may fail (reporting CLIPBOARD ERROR) unless that string includes a CRLF. *[Weird!]*
  - So you **can't** paste a string fragment into an edit session.
  - But you **can** paste a string that spans two or more lines (whole or partial). [Go figure!]

A Workaround: A CLIPBOARD ERROR may occur when you try to paste a string into a file. However:

- It **never occurs** when pasting to the command line.
- Instead, garbage characters will be appended to the pasted text.
- So if you strip off the garbage, you're good to go.

So to get around the problem, do the following:

1. Insert === [for example] into your file at the point where you want to insert the paste.
2. Prime the command line with --> **C** === " <--.
3. Position the cursor past the ", and press **ctrl-v** to effect the paste. *[This will paste the clipboard text appended by random garbage.]*
4. Position the cursor and use **ctrl-endkey** to strip off the garbage.
5. Insert a closing " and press **enter**.

The === that you inserted into your file will be replaced by the clipboard text.

Corruption! - CHANGE ALL can destroy your files!

Don't try using the **CHANGE ALL** command at the folder level. If you do, some of the changed files will be truncated in strange and hideous ways!

The workaround is to use a folder level **FIND ALL**, and then use **CHANGE ALL** inside each individual file.

Corruption! - BEWARE OF UNDO!

Most times, it works just fine. But sometimes, instead of UNDO'ing just the most recent change, it **UNDO's EVERYTHING**! since the start of the edit session. When this happens, recovery will be either arduous or not possible. Suggestions: **Don't use UNDO!**

But if you **must** use **UNDO**:

- Do a **SAVE** first before you attempt your **UNDO**. *[Unlike ISPF, SPF/PRO doesn't fence **UNDO**s at **SAVE** boundaries.]*
- Then attempt your **UNDO**. If it works, then you're happy, and life goes on.

But if **UNDO** screws up, then you can recover as follows:

- **CANCEL**.
- Re**EDIT** and you're back to the thing you want to undo.
- So you will have to manually undo it.

But if you forgot to save first, or if what you need to manually undo is either complex or not precisely remembered, then you may be able to use **git Tortoise** to restore from the current or from any prior **git commit** point.

Things that are Different

Tab Stops:

You can create and change tab stops in SPF/Pro in pretty much the same way you would in ISPF's, but the behavior is a bit different, and that takes getting used to...

- In the Tabs Line, you define stops using asterisks placed in the column prior to where you want the stop to occur. (That's the same.)
- But tab stops do not create an unwritable character position...
  - The advantage is you can simply keep typing, no muss, no fuss.
  - The disadvantage is the stop does not limit the effect of the **erase eof** key. That will always clear to the end of the line.
  - It also does not limit the range of an **insert** action. All text to the right of an insert point will shift rightwards.
  - Ditto for the **del eof** key. All characters to the right of a delete action will shift leftwards.
  - The **backspace**'s function remains unchanged. It replaces the next leftwards character with a blank, thus causing no shift at all.
- **TAB ON ALL|STD:**
  - With **ALL**, the **tab** key will stop at a tab position regardless of whether or not the tab stop itself is blank.
  - With **STD**, the **tab** key will not stop at positions where the tab stop character location is not blank.
- *[Peter Morrison]* I use 'software' tabs in mainframe ISPF (using the ` character), SPF/Pro support them too and the support is identical to the mainframe ISPF. (I Prefer them because there are no reserved screen positions - the line expands out when you press the enter key...)

Command Retrieval

SPF/Pro maintains a retrieve ring of only 10 entries, never more than that. *[sigh]*

Rnn and RR-RR Edit Commands

These behave the same in SPF/Pro as they do in ISPF except when another compound command is pending (Examples: Mnn-A CC-CC-B). In ISPF, the compound command must be completed before it will perform the repeat commands. In SPF/Pro the Rnn and RR-RR commands will be performed even when other commands are pending. (FWIW, I like this.)

Edit Profiles

In SPF/Pro, there is a **PROFILE *[name] [n]*** command that works basically the same way as it does in ISPF:

- You can use the ***name*** operand to create any number of named profiles.
- You can use the ***n*** operand to display up to nine lines of information.
- **But** as noted earlier, there is a bug that will crash SPF/Pro if you attempt to display zero lines. (Use **RESET SPECIAL** instead.)
- You can use **TABS**, **COLS**, **BNDS** and **MASK** commands (entered at the left) to display individual lines of profile information (just like in ISPF).
- You can also use a **BNDS *begincol# endcol#*** command on the primary command line just like you can in ISPF.

When you start an edit session, the default profile that gets loaded (or created) will be the one with a name that matches the file extension of the file that you are editing.

But be aware that the creation of profile names seems to be case sensitive, while the selecting of a profile for use, appears not to be. (In other words, the developers did not seem to do a good job of managing profile names, but I haven't actually researched this thoroughly.)

So if you're in a situation where unexpected profile related things are happening, you can use **=0.7** to investigate and maybe rename or delete profiles.

The Profiles Editor (=0.7)

While the familiar **PROFILE *[NAME]*** command can be used to manage some profile controls, there is a *Profile Editor* that allows you to manage a bunch more...

=0.7 will take you to a list of all the named profiles that exist for you. You can scroll the list in normal ways using **UP**, **DOWN** and **LOCATE** (not FIND) commands.

That list is **huge!** Basically, it is an accumulation of all the profiles I have inadvertently created since the beginning of time. Feel free to purge that trash out. (You can use either =0.7's **D** command or Window's Explorer. [=0.7 reports the location of each profile.])

You can change SPF/Pro's default path for profiles by creating a suitable **SPF5PROF** environment variable. Also, you can set a bunch of default paths with **=0.8**.

As you scroll the list, you will see that some profiles will occur two or more times with names that differ only by case. If you encounter problems with profiles, deleting those instances whose names are not completely UPPERCASE, may resolve the problem.

Anyway, from this list you can Copy, Rename, Delete and **Edit** the profiles. Editing, is of particular interest...

The Edit Dialog has two pages. Pressing ENTER toggles between them. Here are my recommendations:

- **First Page:**
  - For **Record Format,** use **D**. *[Trust me]*
  - For **Max Record Length**, set it to something longer that you will ever actually need, but not so long that it will make changing all blanks prohibitive. *[I find 1000 to be a reasonable number.]*
  - *[Note, SPF/Pro always strips off all trailing blanks from its ASCII records; however, for editing purposes, it pretends that they do exist. So if you, for example, search for all blanks when the max record length is set to 64K, SPF/Pro will likely lock up trying to build its hits report, and you will probably have to force close it.]*
  - For **Character Set,** you can set **E** (EBCDIC) if you want, (it seems to work reasonably well), but then SPF/Pro will be the only tool you can use to look at the file. So I prefer **A** (ASCII).
  - For **Save on Count** and **Back on Save**, I don't bother with these. So I just leave their settings as **0** and **N**, respectively.
  - For **Overwrite Original** this is a useful setting to leave off. So I set it to **N** so that when writing out your file, it first writes to a temporary *(filename.**DMP**)* and then does delete/rename games to get the file to where you want it.
  - For **Data Shift on Change**, I just leave it as **Y**.
- **Second Page:** The following fields appear only when the **Record Format** *[on the first page]* is set to **D**.
  - For **Line Terminator**, use **SCRLF**. This will causes SPF/Pro to use CR-LFs when writing the file, but will allow it to recognize bare CRs and LFs as line terminators when reading the file.
  - For **File Terminator** you will definitely want this to be set to **N**. *[When set to Y, SPF/Pro appends a X'7F' to the end of the file which can mess up the Assembler should it make it up to the mainframe.]*
  - *[If you have saved a file under the control of a profile where File Terminator is set to Y, then after setting this to N, you will need to go back to that file and scroll to the bottom, where you will now see the X'7F' hanging out all by itself as the file's last record. You will need to delete that last record and resave the file.]*
  - For **Input Tabs to**, you can do whatever you want. Our files do not usually contain actual tab characters, so whatever you do here won't have much effect. I just leave this setting to **8**.
  - For **Output tabs at,** you definitely do not want to let this happen. Set this field to **N**.
  - For **Pad right w/blanks**, you definitely want to leave this as **N**. *[See above for a discussion involving real and apparent trailing blanks.]*
  - For **Null Protected,** this too never really comes up. I just leave this set to **N**.

When done, use **F3** to save and exit back to the Profiles Selection List.

Fixed Length vs Variable Length Lines

SPF/Pro does support enforcing both a fixed line length and a limit on variable line lengths. It's done both by profile settings and by dialog fields. My thoughts are these...

- At first blush, it might seem like a good idea to set an 80-character line length for the z/XDC source files, but for me...
  - (a) It's not necessary, because it's rare that that I would create a line that's too long, and if I do, it's pretty easy to find where it is.
  - (b) It makes **insert** actions a bit more cumbersome to perform: Insert locks when a non-blank hits the right-side wall.

Variable length records can be up to 64K characters long, but you can limit that. Typically, I'll set my profiles to limit variable lines to 1,000 characters. *[I once issued **CHANGE ALL ' ' #.** At a 64K line length limit. That's a **LOT** of blanks!]*

The Keyboard Editor (=0.k)

=0.k takes you into a keyboard editor. Here you can create and save a custom keyboard layout with whatever weird key arrangement that strikes your fancy. Sine your SPF/Pro is cloned from mine, you will find my favorite keyboard under the name **DAVECOLE**.

When you get into the editor, you will find that all of the 3270 action/edit/function key values are indicated buy tokens of the form **[key-name]**. For the most part, these are documented in **C:\Program Files (x86)\Command Technologies\SPFPRO\HELP\SPFPRO.HLP.** Just open that file in your favorite text editor (SPF/Pro, for example) and search for topic " 5.6".

[Caution: If you want to view this file with SPF/Pro, you will have to create a copy on a short path ("'C:\temp"' for example) because the file is larger than 64K and its native path name is too long.]

Several of the the **[3270-key-tokens]** are described in 5.6, but not all. You can peruse the **DAVECOLE** keyboard map for examples. But here are some notes of interest:

- SPF/Pro offers two kind of backspaces:
  - **[BACKSPACE]** moves the cursor left one character, deletes that character, and left-shifts the rightwards text one character position.
  - **[BACKSPACE-BLANK]** moves the cursor left one character, deletes that character and replaces it with a blank. The rightwards text remains unshifted.
- The **End** key that is located within the 6-key cluster located between the alphas and the number keys is called the **End-Ext** key. *[There also is an **End** key, but I don't even know where that is.]*
- The token for "Erase EOF" is named **[DEL-END-OF-DATA]**.

You can find examples of other tokens within the **DAVECOLE** keyboard definition.

Hardening Changed Profiles/PF-keys/etc (without Hardening Corruption)

When you make changes to PF-keys, profiles and various other global settings, you need to be aware of when those changes get hardened.

- Profiles are backed by individual **.prf** files that get updated immediately when profiled information is changed.
- Most other global settings are backed up in a **SPFPRO50.GBL** and other files that do not get updated until SPF/Pro is closed.
- If you have multiple SPF/Pros open, each SPF/Pro will save its changes upon being closed. So **the last man out WINS!**

Therefore...

- **Profile changes:** These get hardened immediately. When you make a change, it's done, you don't need to worry about it.
- **PF-key changes:** These do not get saved until you close SPF/Pro. So if you have multiple SPF/Pros open, you must close all but one, then make your PF-key changes in that final SPF/Pro, then close it.
- **SPF/Pro abends and other failures:** This is also a last-man-out-uh-**loses** situation. If a failure is due to a global controls corruption, then when SPF/Pro self-closes, it may or may not write the corrupted data to disk. If it does, then you're screwed **unless** you happen have another SPF/Pro window currently open.
  - Usually, if one SPF/Pro window fails, and you then close another, any corrupted global controls that the failing window may have written gets overwritten with good data.
  - **Moral:** Always keeping an unused SPF/Pro window lying around, so that you can use it to recover the Global Table when necessary.

Documentation

You can find SPF/Pro's internal documentation at **=T**. Unfortunately, the logic for displaying this information is buggy. Sometimes the chapters display, sometimes they fail.

There is a copy of the Tutorial on the web at **www.manmrk.net/tutorials/ISPF/spfproindx.htm**.

ASACCC Command

Some time ago I wrote a REXX Command for SPF/Pro named **ASACCC**. When you run it, it converts all Carriage Control Characters (except +) to appropriate spacing. Just start an edit session, and then issue **ASACCC** on the command line like any other Primary Command.

ASACCC is written in **REXX**. It uses **isredit** statements to issue edit commands to ISP/Pro.) See the figure below.)

SPF/Pro's scripting language is **REXX**. You can find a number of sample scripts at **C:\Program Files (x86)\Command Technologies\SPFPRO\REXX.**

ASACCC (Figure)

```
/************************* REXX ***************************************\
*                                                                      *
* (C):     Copyright ColeSoft Partenrs, Inc. 2009. All rights          *
*          reserved.                                                   *
*                                                                      *
* MACRO:   ASACCC - ASA Carriage Control Character implementer.        *
*                                                                      *
* PURPOSE: scans the current text file and replaces all ASA Carriage   *
*          Control Characters with appropriate codes to carry out      *
*          their intended purposes. Specifically, it does the          *
*          following:                                                  *
*              CCC Replaced by:                                        *
*              0 X'0D0A20'                                             *
*              - X'0D0A0D0A20'                                         *
*              1 C'1',132C'-',X'0D0A20'                                *
*              + (ignored)                                             *
*                                                                      *
* USAGE:   (1) Open for edit the target file.                          *
*                                                                      *
*          (2) On the command line, type ASACCC.                       *
*                                                                      *
*          (3) View/save the result.                                   *
*                                                                      *
* NOTE:    Note, the action taken for ccc='1' adds one line per        *
*          page. Consequently, if the file is actually printed, the    *
*          result will be affected.                                    *
*                                                                      *
* NOTE:    This macro depends upon the files longest record not        *
*          being longer than 133 characters (including the ASA         *
*          character).                                                 *
*                                                                      *
* NOTE:    This macro depends upon the target file not containing      *
*          any X'B6' (¶) byte codes.                                   *
*                                                                      *
\**********************************************************************/

'isredit macro'
'isredit bounds 1 135'
'isredit change all " " x"b6"'
'isredit bounds'
'isredit change all 1 "0" x"0d0a20"'
'isredit change all 1 "-" x"0d0a0d0a20"'

'isredit exclude all 1 "1---"'
'isredit change all nx 1 "1" x"2323230d0a20"'
'isredit change all x"b6" " "'
'isredit reset'

'isredit replace asaccc_temp.txt '.zfirst' '.zlast
'isredit delete '.zfirst' '.zlast
'isredit move asaccc_temp.txt'

'isredit change all 1 ###
1-------------------------------------------------------------------------------------------'
'isredit find all 1 "1---"'
'isredit (curserow,cursecol) = cursor'
if (curserow = 1) then
   'isredit change 1 "1---" " ---"'
else
   'isredit up max'

exit 0
```