ESPF/PC Full Screen Editor User's Guide Release 2.0.0

December 15th, 1987

Chuck Tribolet

TRIBLET at STLMVS1
8-543-2369
M81/A424
IBM Santa Teresa

IBM Internal Use Only

SUMMARY OF AMENDMENTS

RELEASE 2.0.1 - REVISED FEBRUARY 22, 1984.

This was a quick maintenance release. I have also updated the documentation to reflect the 2.0.0 changes.

- Modified the screen handler. A change made for 2.0.0 was causing continuous snow in color monitors.

RELEASE 2.0.0 - NO NEW MANUAL

This release included a LOT of internal changes not visible to the user. These changes lay the ground work for files wider than 72 characters, split screen, and keyboard profiles.

- ESPF will now use all available memory up to the normal 640K limit. A 128K machine can edit about 500 lines. Each 64K increment beyond 128K will allow another 500 lines.
- The change primary command no longer goes beep every time.
- Numeric scroll amount is now supported in the scroll field.
- Scroll by cursor position is now supported.
- Improved cursor positioning.
- Internal changes in preparation for split screen. Split screen is NOT yet supported.
- CAPS LOCK and NUM LOCK displayed at bottom of screen.
- Converted from the DOS keyboard routine to the BIOS keyboard routine. This has two effects visible to the user:
  - The Large plus at the right side of the screen is now a new line key. F5 continues to also be a new line, but it will go away in a future release.
  - You can now enter any ASCII character using Alt and the numeric keypad.
- An empty file will cause the screen to be filled with inserted lines.

RELEASE 1.2.4 - REVISED JULY 18TH, 1983.

No code has changed. I have however improved the documentation.

- The documentation will now SCRIPT under SSPROF. I can now get out of the business of distributing GPASS to everyone. GPASS provided a great deal of useful function. In order to allow the document to SCRIPT under SSPROF I had to write my own GML APF's to implement the best of GPASS. The hardest part about this was learning from the SCRIPT manual. Had they had even a single example, the job would have been trivial.

- I have corrected numerous minor editorial errors. There is still a long way to go (see the comments at the front of the SCRIPT input) but I think that this version is enough better than the previous one to warrant putting it on IBMPC and PCLIB. I will be working on the documentation some more before the next round of enhancements to the code because I do not have an IBM Personal Computer at the moment.

RELEASE 1.2.4

- File names are now syntax checked.

- F12 will now exit from the Edit Primary Menu back to DOS.

- COLS, MASK, BOUNDS, and TABS line commands are now supported. Please note that the TABS line command is supported, but the tab function does not use its information yet.

- Exclude (X), Block Exclude (XX), First (F), Last (L), and Show (S) line commands are supported.

- Text Enter (TE), Text Flow (TF), and Text Split (TS) line commands are supported.

- The ALL option of FIND is now supported.

- The RESET primary command now removes tabs, columns, bounds and mask lines and undoes excluded lines.

- All primary and line command abbreviations supported by SPF/370 are now supported by ESPF/PC.

- Hooks were added to support the addition of all remaining line and primary commands.

- The following minor bugs have been fixed:

  - If the 24th line on the screen is >72 characters, it overflowed onto the 25th line.
  - You could not overtype the last character of a line number.
  - If the cursor was located in a line number, FIND and CHANGE started looking on the NEXT line.

- CHANGE and FIND looked at the top and bottom of file lines.
- Insert lines contained ...... in their line number instead of ''''''.
- CHANGE would not handle a findstring longer than about 15 characters.
- ESPF/PC blew up with HEAP FULL on a 512K or larger machine.

RELEASE 1.2.3

- Substantially improved performance when reading and writing files.

- LOCATE primary command.

- TABS OUT primary command.

- The cursor will now be located in the primary command area when initially editing a file.

- A one-shot scroll amount may be entered in the primary command area. Numeric scroll amounts are supported in this area.

- Insert now acts like 370 SPF -- a line will remain in the working file only if you type something into it and lines will continue to be inserted as long as you type something into the last line of a block and leave the cursor on that line.

- Line command syntax is now the same as 370 SPF -- a line command can be entered anywhere on the line number and the count is no longer limited to a single character.

RELEASE 1.2.2

This release was a test version of 1.2.3 with limited distribution.

RELEASE 1.2.1

- Function key assignments changed to match SPF 3270 assignments in a physical way rather than a numeric way. END is now F2 (upper right corner) instead of F3.

- Color Adapter card supported in 80 character mode.

- Cursor positioning is improved.

- Added Edit option menu to allow editing more than one file without reloading the SPF module.

- Alarm sounded when typing into a protected field and when an error message is issued.

- Architecture added to support primary commands.
- Added SAVE primary command.
- Added CANCEL primary command. F13 has been removed as exit without save.
- Added FIND primary command.
- Added CHANGE primary command.
- Added RESET primary command.
- Added COPY primary command.
- The primary command area is now titled 'COMMAND' instead of 'COMMAND INPUT'.
- Fixed miscellaneous bugs:
  - Save of an empty file results in a single line that says: ******** BOTTOM OF DATA **********.
  - All line commands can be entered on the top and bottom lines of the file. Only I and A should be allowed on top line, only B on bottom line.
  - Copy on the line immediately before the Before resulted in a heap full condition.

RELEASE 1.2.0

This release was a test version of 1.2.1 with limited distribution.

RELEASE 1.1.5

This release was a test version of 1.2.0 with limited distribution.

RELEASE 1.1.4

This release was a test version of 1.2.0 with limited distribution.

RELEASE 1.1.3

First release made available within IBM.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

ACKNOWLEDGEMENTS.

I would first like to acknowledge the designer of the version of SPF that runs on System/370. They have designed an editor that pays more attention to human factors that any other two programs around. They are rightly cited outside IBM as an example of an interactive program that did it right. Where I have diverged from their design, it is either an error of ommission because I have not had the time to implement that particular widget, or a case where I found it necessary to support some particular requirement of the IBM Personal Computer.

I would also like to thank Charmaine Stansfield for proofing this document from the viewpoint of a user who know System/370 SPF well but did not have experience with the IBM Personal Computer.

I would like to thank all of those who have reported problems. A few of these problems have occurred in environments (DOS releases, memory size, etc.) that I would never have had the time try, this being a bootleg effort and all.

I would finally like to thank Walt Paul who wrote The Professional Editor and Jim Wyllie who wrote The Personal Editor. The spirit of friendly competition in which we have all worked has enhanced all three programs.

ESPF/PC USER'S GUIDE

Note: The program described in this document is IBM Internal Use Only. As an IBM asset, it is to be used only for IBM business. It may be used on an IBM Personal Computer not owned by IBM provided that use is solely for IBM business. The program and this document may be reproduced as necessary for IBM business.

Although ESPF/PC is still a subset of the SPF editor, I have found that ESPF/PC is now to the point where I feel that it is better than EDLIN in all ways except for the size of file that is editable.

I have tested the code by using it to work on itself but there are no doubt some bugs. If you run into a problem please let me now so I can fix it.

GETTING INTO ESPF/PC

The syntax of the ESPF command is:

```
ESPF [*|filespec]
```

If filespec is specified, the named file is edited. If * is specified, or no filespec is specified, a panel is presented which allows you to enter a file name.

SCREEN HANDLING

In order to minimize training requirements, the screen is handled just like a 3277. Attribute bytes take up a position on the screen.

- The BackSpace (key 14) and the left arrow (key 75) move the cursor left one space.
- The right arrow (key 77) moves the cursor right one space.
- The up arrow (key 72) moves the cursor up one row.
- The down arrow (key 80) moves the cursor down one row.
- HOME (key 71) moves the cursor to the first position of the first unprotected field on the screen. This is the primary command field when editing with ESPF/PC.
- Right tab (key 15) moves the cursor to the right until it is at the first position of an unprotected field.

- Left tab (SHIFT key 15) moves the cursor to the left until it is at the first position of an unprotected field.
- If you try to type into an protected field, the IBM Personal Computer will give a short beep.
- INS (key 82) puts you into insert mode. A second INS takes you out. There is no need to have nulls on to do an insert character --- spaces work just fine. If you try to insert too much, the IBM Personal Computer will give a short beep.
- DEL (key 83) deletes the character at the current cursor location.
- ALT 6 (ALT key 07) enters a tab character.

SUMMARY OF AVAILABLE COMMANDS

- Only files 0-72 characters wide are supported at this time. Files 73-80 characters wide will be read in but you cannot modify columns 73-80. Files wider than 80 characters will cause a warning message to be issued when the the file is read in and WILL BE TRUNCATED.
- The following primary commands are supported:
  - LOCATE n
  - TABS OUT
  - SAVE
  - CANCEL
  - {FIND|F} + [string [FIRST|NEXT|ALL]]
  - {CHANGE|C} + [from to [FIRST|NEXT|ALL]]
  - RESET
  - COPY filespec (used in conjunction with A and B line commands).
- The following line commands are supported:
  - I[nn]
  - D[nn]
  - DD
  - R[nn]
  - RR[nn]
  - M[nn]
  - C[nn]
  - MM
  - CC
  - A
  - B
  - ([nn]
  - (([nn]
  -)[nn]
  -))[nn]
  - <[nn] - This is a column shift, not a data shift.
  - <<[nn] "
  - >[nn] "

- >>[nn] "
- TS
- TF
- TE[nn]
- X[nn]
- XX
- F[nn]
- L[nn]
- S[nn]
- COLS
- MASK
- BOUNDS
- TABS

The [nn] indicates that a number may be entered with the line command.

- The big + (at right side of keyboard) (key 78) is the new line function.
- F2 (key 60) will exit after saving.
- F3 (key 61) repeats the previous FIND command.
- F4 (key 62) repeats the previous CHANGE command.
- F5 (key 63) is new line. It is preferred that you use the large + at the right of the keyboard (key 78) for this function.
- F6 (key 64) is erase end of line. This key was chosen because it it is in the same position as the same key on the 3277.
- F7 (key 65) or PgUp (key 73) will scroll closer to the top of the file. The valid scroll amounts are HALF, PAGE, MAX, CUR (cursor), or a number.
- F8 (key 66) or PgDn (key 81) will scroll closer to the bottom of the file. The valid scroll amounts are HALF, PAGE, MAX, CUR (cursor), or a number.
- F12 (Shift key 60) will exit without saving.
- You are always in 'ASIS' mode. Use CAPS LOCK if you want all caps.
- Although HEX mode is not provided, it is less necessary in ESPF/PC than in SPF/370 because:
  - Any character can be displayed uniquely.
  - Any character can be entered on the numeric keypad. Unfortunately, the DOS keyboard routine used by ESPF does not differentiate between a character entered on the numeric keypad and one entered by pressing its key. Since ESPF use some of the characters between X'00' and X'1F' for special purposes, these characters cannot be entered into the working file.
- ESPF/PC will edit a file only if it fits into memory.

INSTALLATION

I no longer distribute the ESPF source because the source and EXE files no longer fit on one diskette.

You will have received ESPF in one of several ways:

ON A DISKETTE

There is nothing further that you have to do.

OVER THE NET OR FROM THE IBMPC DISK IN YORKTOWN

I distribute ESPF only as a binary file (ESPF.EXEBIN to MVS users, ESPF EXEBIN to VM users).

DOWNLOADING USING PC3277 OR XT3277.

- Logon to VM
- Press Alt-Esc to get to PC-DOS.
- Issue: HCOPY H:ESPF EXEBIN A ESPF.EXE / BINARY

DOWNLOADING USING PC3278 (SOFTWARE FOR 3270/PC)

- I don't have access to one. Would someone PLEASE send me this information ASAP.

EXPORTED FILES

I no longer generally distribute the EXP format. If you have a requirement for it, please contact me directly and I will get a copy to you via the net.

LOADING THE B3XXX (TBH) FILE

- Logon to VM
- Convert the binary file to a TBH file by issueing:

  ```
  BINTBH ESPF EXEBIN A ESPF EXETBH A
  ```

- If you are going to download from an MVS/TSO system:
  - Send ESPF EXETBH to the TSO system using SENDFILE or XMIT.
  - Logoff VM
  - Logon to TSO
  - Receive ESPF EXETBH as ESPF.EXETBH.
- On MVS enter 'XMITH ESPF.EXETBH. On VM enter 'XMITH ESPF EXETBH'.
- Reply '/*' (MVS) or a null line (VM) when prompted to enter commands.
- When the message 'HXM002I READY TO SEND' appears, press F2, then enter 'RB ESPF.EXE' and press New Line.
- Go have a cup of coffee - It will take about 30-45 minutes at 1200 baud, 2-3 hours at 300 baud.

LEARNING TO USE ESPF

A more detailed description of the SPF editor is available in SPF/PDF for MVS: Program Reference (SC34-2089) and in SPF/PDF for VM: Program Reference (SC34-2090).

TYPES OF ESPF COMMANDS

There are two basic type of ESPF commands: primary commands and line commands.

PRIMARY COMMANDS

Primary commands are entered on the second line of the screen between where it says 'COMMAND ===>' and where it says 'SCROLL'. Primary commands usually relate to the entire file rather than a specific line or group of lines. Most primary commands do not care whether they were entered in upper or lower case. There are a few exceptions which are documented below.

LINE COMMANDS

Line commands are entered on top of the line number at the left side of the screen. The line command may be typed starting in any position in the line number. They always cause an action to be performed against the specific line or group of lines on which they are entered. In some cases, a count (number of lines to operate on or number of times to perform the operation) may be entered immediately following the line command. This count may be one or more digits. The count is computed as follows:

- If the cursor is in that line number, the count consists of everything from the line command up to but not including the cursor location.

- If the cursor is not in that line number, the count consists of everything from the line command up to and including the last changed character.

Single Line Commands: A single line command consists of a single character and causes the specified action to be performed against that one line.

Block Line Commands: A block line command consists of two characters entered on two different lines. The specified action is performed against the block beginning with the first of those two lines and ending with the second of those two lines.

A count may be included only on the first line command of a pair of block line commands.

CHANGING TEXT BY OVER-TYPING

The simplest way to change text in your file is merely overtype it on the screen. What you see is what you get.

INSERTING CHARACTERS

To insert characters, press the INS key. The word INSERT will appear at the bottom of the screen. Any characters that you type will be inserted just to the left of the cursor location. The cursor location and all characters to its right in the field will move one position to the right.

DELETING CHARACTERS

To delete the character at the cursor location, press the DEL key. All the characters to the right of the cursor in the field will move one position to the left.

SCROLLING WITHIN YOUR FILE

Since only 22 lines of your file are displayed at a time, you need to be able to move your 'window' into your file up and down. To move it up, (closer to the top of the file), press PgUp, press F7, or enter a primary command of 'UP'. To move it down, (closer to the bottom of the file), press PgDn, press F8, or enter a primary command of 'DOWN'.

If you have entered an amount in the primary command area, your window will move that amount. If this is HALF, the window will move a half a screen. If this is PAGE, the window will move one screen. If this is MAX, the window will move as far as it can (like TOP or BOTTOM in some editors).

If you have not entered an amount in the primary command area, your window will move the amount defined by the SCROLL field. If this is HALF, the window will move a half a screen. If this is PAGE, the window will move one screen. If this is MAX, the window will move as far as it can (like TOP or BOTTOM in some editors). If it is CUR, the line where the cursor is will be positioned to the top of the screen. If this is a number, the window will move that many lines.

SAVING YOUR FILE

The syntax of the save primary command is:

```
SAVE
```

To save a copy of your file without exiting, type in a primary command of SAVE. Your working copy of the file will be written back.

GETTING OUT OF ESPF WITHOUT SAVING

The syntax of the cancel primary command is:

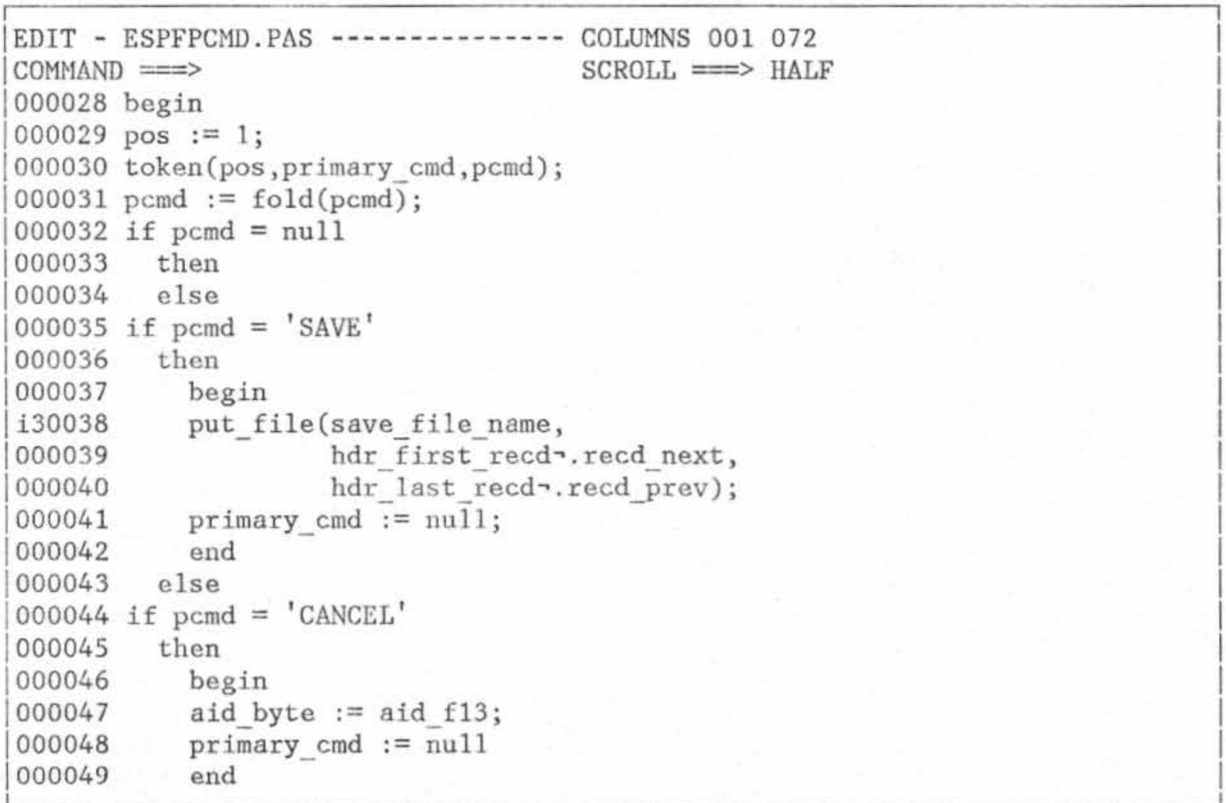
```
CANCEL
```

To exit ESPF without saving your file (for instance when you have made a major editing mistake), type in a primary command of CANCEL.

INSERTING MORE LINES

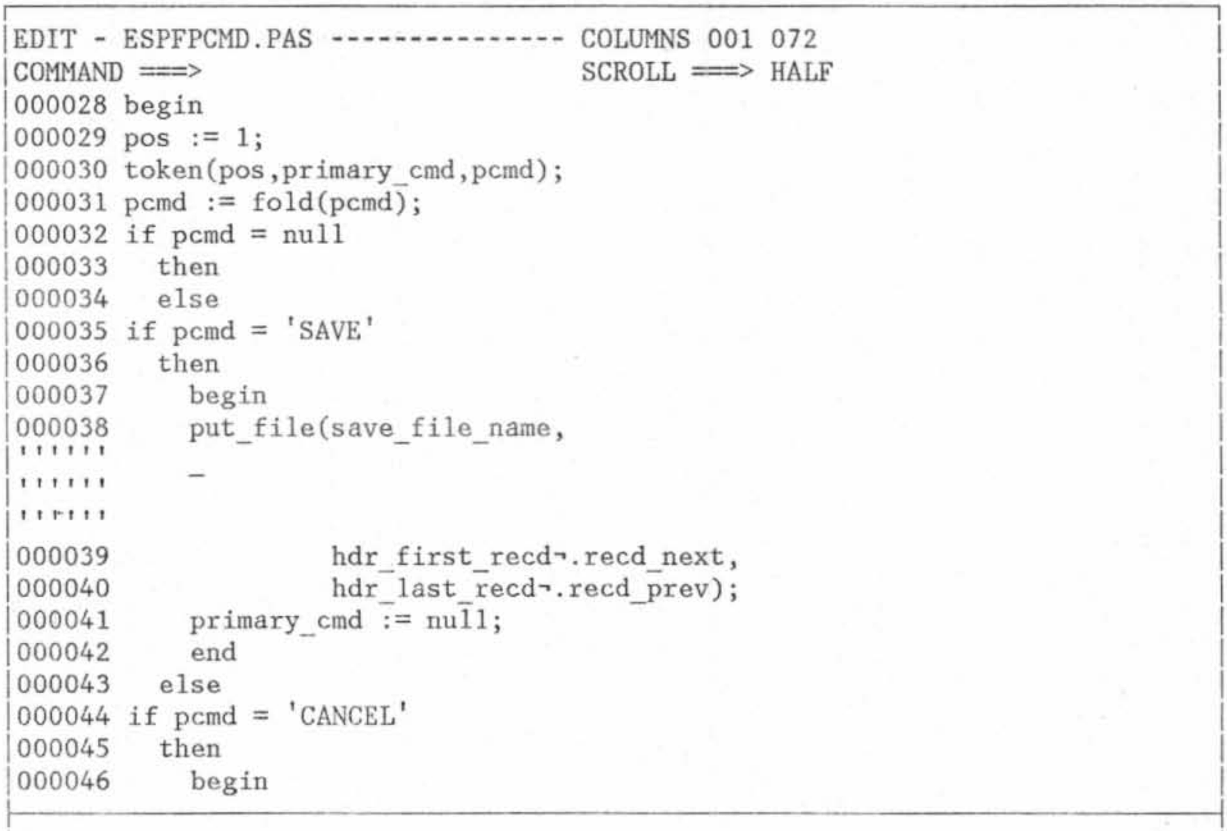
The insert line command comes in the single form only. To insert more blank lines, type a 'I' followed by the number of lines to insert on the line number after which you wish the lines inserted. ESPF will then insert that many more lines -- you can tell these lines because they will have a line number of ...... -- they will remain in your working file only until the next ENTER unless you type something on them. If you type something on the last line, and you leave the cursor there, another line will be inserted for you.

-- Figure id 'insertb' unknown -- and -- Figure id 'inserta' unknown -- show an example of how to insert more lines.



```
EDIT - ESPFPCMD.PAS --------------- COLUMNS 001 072
COMMAND ===>                        SCROLL ===> HALF
000028 begin
000029 pos := 1;
000030 token(pos,primary_cmd,pcmd);
000031 pcmd := fold(pcmd);
000032 if pcmd = null
000033   then
000034   else
000035 if pcmd = 'SAVE'
000036   then
000037     begin
i30038     put_file(save_file_name,
000039              hdr_first_recd¬.recd_next,
000040              hdr_last_recd¬.recd_prev);
000041     primary_cmd := null;
000042     end
000043   else
000044 if pcmd = 'CANCEL'
000045   then
000046     begin
000047     aid_byte := aid_f13;
000048     primary_cmd := null
000049     end
```

Figure 1. Inserting More Lines -- Before.



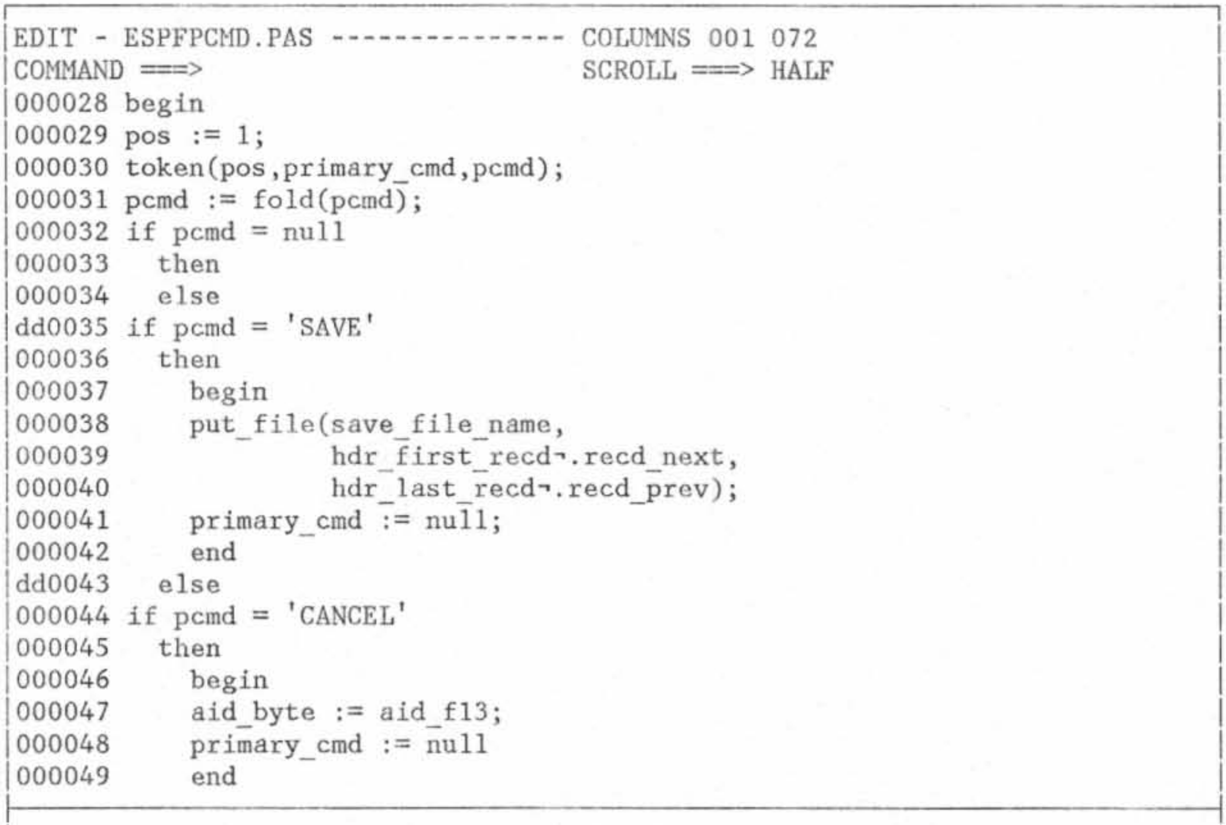
```
EDIT - ESPFPCMD.PAS --------------- COLUMNS 001 072
COMMAND ===>                        SCROLL ===> HALF
000028 begin
000029 pos := 1;
000030 token(pos,primary_cmd,pcmd);
000031 pcmd := fold(pcmd);
000032 if pcmd = null
000033   then
000034   else
000035 if pcmd = 'SAVE'
000036   then
000037     begin
000038     put_file(save_file_name,
''''''
''''''     _
''''''
000039              hdr_first_recd¬.recd_next,
000040              hdr_last_recd¬.recd_prev);
000041     primary_cmd := null;
000042     end
000043   else
000044 if pcmd = 'CANCEL'
000045   then
000046     begin
```

Figure 2. Inserting More Lines -- After.

DELETING LINES

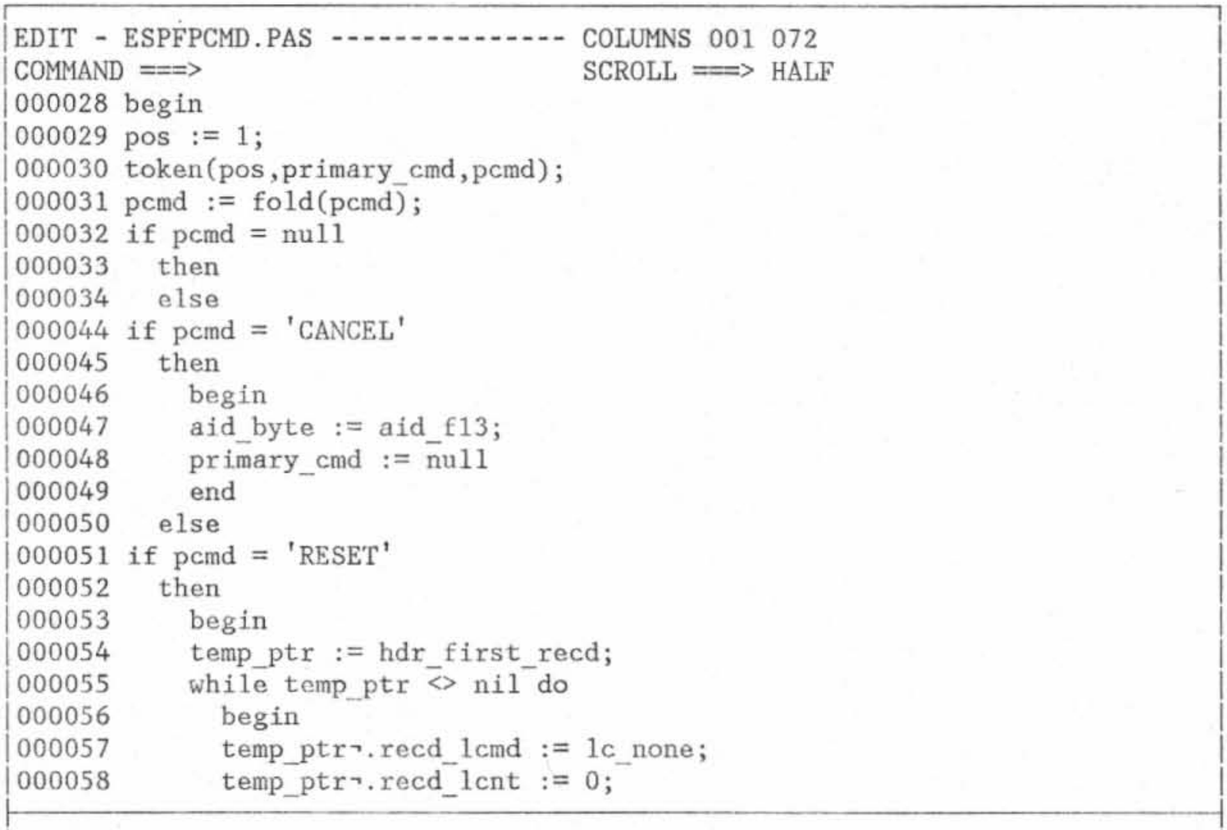
The D (delete) line command comes in both single and block forms. To use the single form, type a 'D' followed by the number of lines to delete over the first line number to be deleted. To use the block form, type 'DD' over the first line number to be deleted and another 'DD' over the last line number to be deleted.

-- Figure id 'deleteb' unknown -- and -- Figure id 'deletea' unknown -- show an example of how to delete lines.



```
EDIT - ESPFPCMD.PAS -------------- COLUMNS 001 072
COMMAND ===>                       SCROLL ===> HALF
000028 begin
000029 pos := 1;
000030 token(pos,primary_cmd,pcmd);
000031 pcmd := fold(pcmd);
000032 if pcmd = null
000033   then
000034   else
dd0035 if pcmd = 'SAVE'
000036   then
000037     begin
000038     put_file(save_file_name,
000039              hdr_first_recd¬.recd_next,
000040              hdr_last_recd¬.recd_prev);
000041     primary_cmd := null;
000042     end
dd0043   else
000044 if pcmd = 'CANCEL'
000045   then
000046     begin
000047     aid_byte := aid_f13;
000048     primary_cmd := null
000049     end
```

Figure 3. Deleting Lines -- Before.



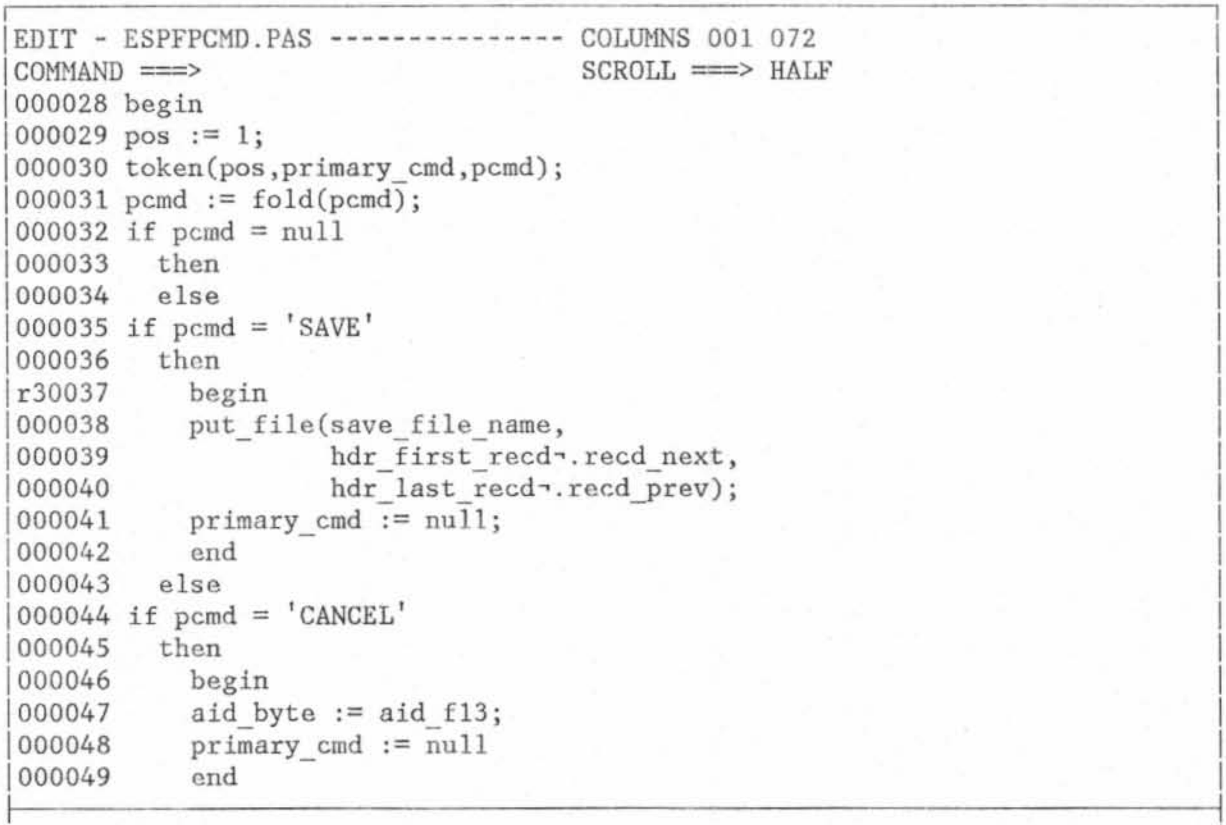
```
EDIT - ESPFPCMD.PAS --------------- COLUMNS 001 072
COMMAND ===>                         SCROLL ===> HALF
000028 begin
000029 pos := 1;
000030 token(pos,primary_cmd,pcmd);
000031 pcmd := fold(pcmd);
000032 if pcmd = null
000033   then
000034   else
000044 if pcmd = 'CANCEL'
000045   then
000046     begin
000047     aid_byte := aid_f13;
000048     primary_cmd := null
000049     end
000050   else
000051 if pcmd = 'RESET'
000052   then
000053     begin
000054     temp_ptr := hdr_first_recd;
000055     while temp_ptr <> nil do
000056       begin
000057       temp_ptr¬.recd_lcmd := lc_none;
000058       temp_ptr¬.recd_lcnt := 0;
```

Figure 4. Deleting Lines -- After.

REPEATING LINES WITHIN YOUR FILE

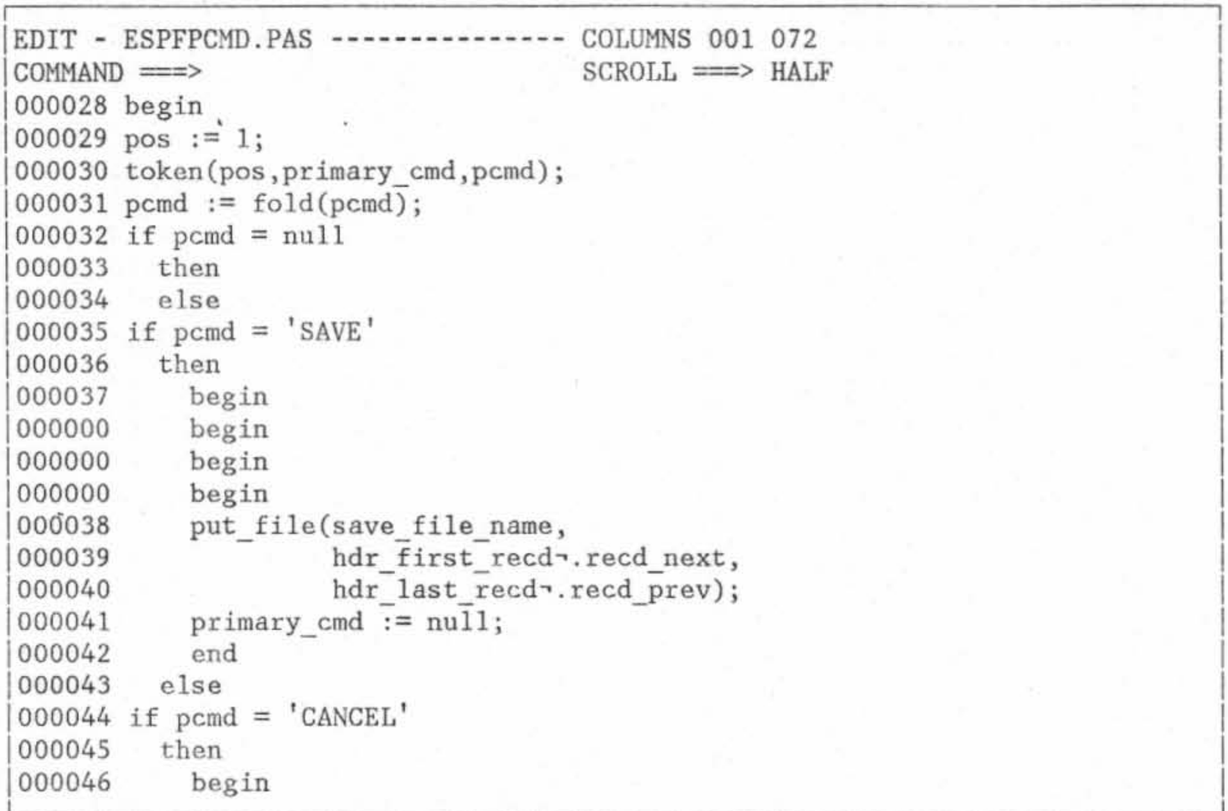
The R (repeat) line command comes in both single and block forms. To use the single form, type a 'R' followed by the number of times to repeat the line over the line number to be repeated. To use the block form, type 'RR' followed by the number of repetitions over the first line number to be repeated and another 'RR' over the last line number to be repeated.

-- Figure id 'repeatb' unknown -- and -- Figure id 'repeata' unknown -- show an example of how to repeat lines.



```
EDIT - ESPFPCMD.PAS --------------- COLUMNS 001 072
COMMAND ===>                        SCROLL ===> HALF
000028 begin
000029 pos := 1;
000030 token(pos,primary_cmd,pcmd);
000031 pcmd := fold(pcmd);
000032 if pcmd = null
000033   then
000034   else
000035 if pcmd = 'SAVE'
000036   then
r30037     begin
000038     put_file(save_file_name,
000039              hdr_first_recd¬.recd_next,
000040              hdr_last_recd¬.recd_prev);
000041     primary_cmd := null;
000042     end
000043   else
000044 if pcmd = 'CANCEL'
000045   then
000046     begin
000047     aid_byte := aid_f13;
000048     primary_cmd := null
000049     end
```

Figure 5. Repeating Lines -- Before.



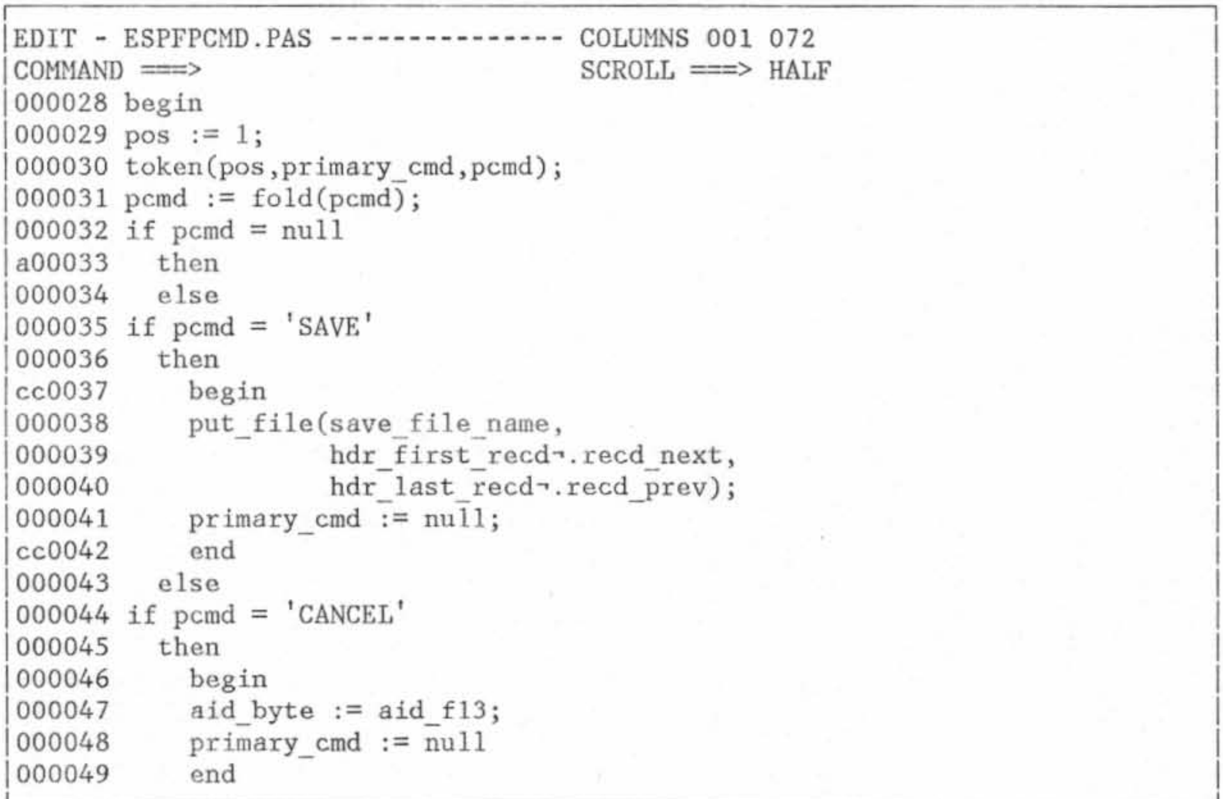
```
EDIT - ESPFPCMD.PAS --------------- COLUMNS 001 072
COMMAND ===>                        SCROLL ===> HALF
000028 begin
000029 pos := 1;
000030 token(pos,primary_cmd,pcmd);
000031 pcmd := fold(pcmd);
000032 if pcmd = null
000033   then
000034   else
000035 if pcmd = 'SAVE'
000036   then
000037     begin
000000     begin
000000     begin
000000     begin
000038     put_file(save_file_name,
000039              hdr_first_recd¬.recd_next,
000040              hdr_last_recd¬.recd_prev);
000041     primary_cmd := null;
000042     end
000043   else
000044 if pcmd = 'CANCEL'
000045   then
000046     begin
```

Figure 6. Repeating Lines -- After.

COPYING LINES WITHIN YOUR FILE

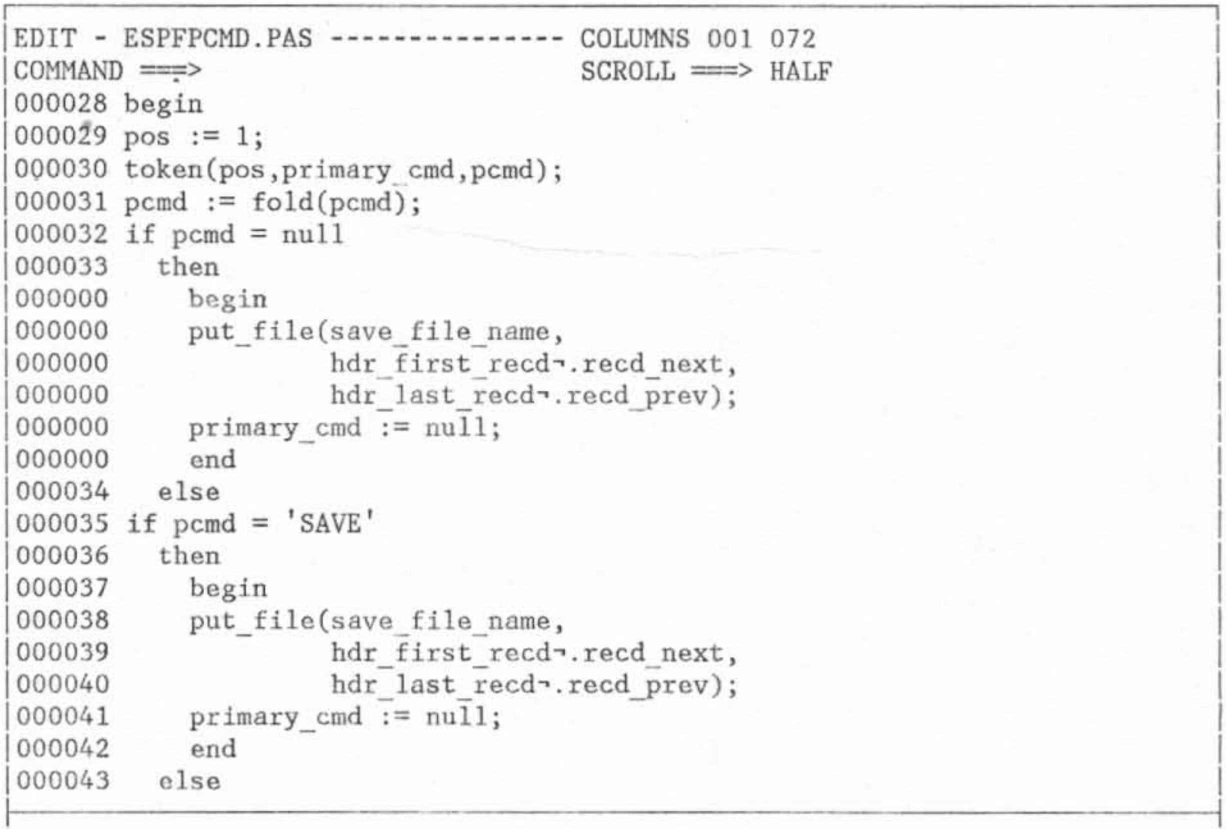
The C (copy) line command comes in both single and block forms. It is used in conjunction with the before and after line commands which have only the single form. To use the single form, type a 'C' followed by the number of lines to copy over the first line number to be copied and type an 'A' over the line number after which you wish the lines copied or type a 'B' over the line number before which you wish the lines copied. To use the block form, type a 'CC' over the first line number to be copied, another 'CC' over the last line number to be copied, and type an 'A' over the line number after which you wish the lines copied or type a 'B' over the line number before which you wish the lines copied.

-- Figure id 'copyb' unknown -- and -- Figure id 'copya' unknown -- show an example of how to copy lines.



```
EDIT - ESPFPCMD.PAS --------------- COLUMNS 001 072
COMMAND ===>                        SCROLL ===> HALF
000028 begin
000029 pos := 1;
000030 token(pos,primary_cmd,pcmd);
000031 pcmd := fold(pcmd);
000032 if pcmd = null
a00033   then
000034   else
000035 if pcmd = 'SAVE'
000036   then
cc0037     begin
000038     put_file(save_file_name,
000039              hdr_first_recd¬.recd_next,
000040              hdr_last_recd¬.recd_prev);
000041     primary_cmd := null;
cc0042     end
000043   else
000044 if pcmd = 'CANCEL'
000045   then
000046     begin
000047     aid_byte := aid_f13;
000048     primary_cmd := null
000049     end
```

Figure 7. Copying Lines -- Before.



```
EDIT - ESPFPCMD.PAS --------------- COLUMNS 001 072
COMMAND ===>                        SCROLL ===> HALF
000028 begin
000029 pos := 1;
000030 token(pos,primary_cmd,pcmd);
000031 pcmd := fold(pcmd);
000032 if pcmd = null
000033   then
000000     begin
000000     put_file(save_file_name,
000000              hdr_first_recd¬.recd_next,
000000              hdr_last_recd¬.recd_prev);
000000     primary_cmd := null;
000000     end
000034   else
000035 if pcmd = 'SAVE'
000036   then
000037     begin
000038     put_file(save_file_name,
000039              hdr_first_recd¬.recd_next,
000040              hdr_last_recd¬.recd_prev);
000041     primary_cmd := null;
000042     end
000043   else
```

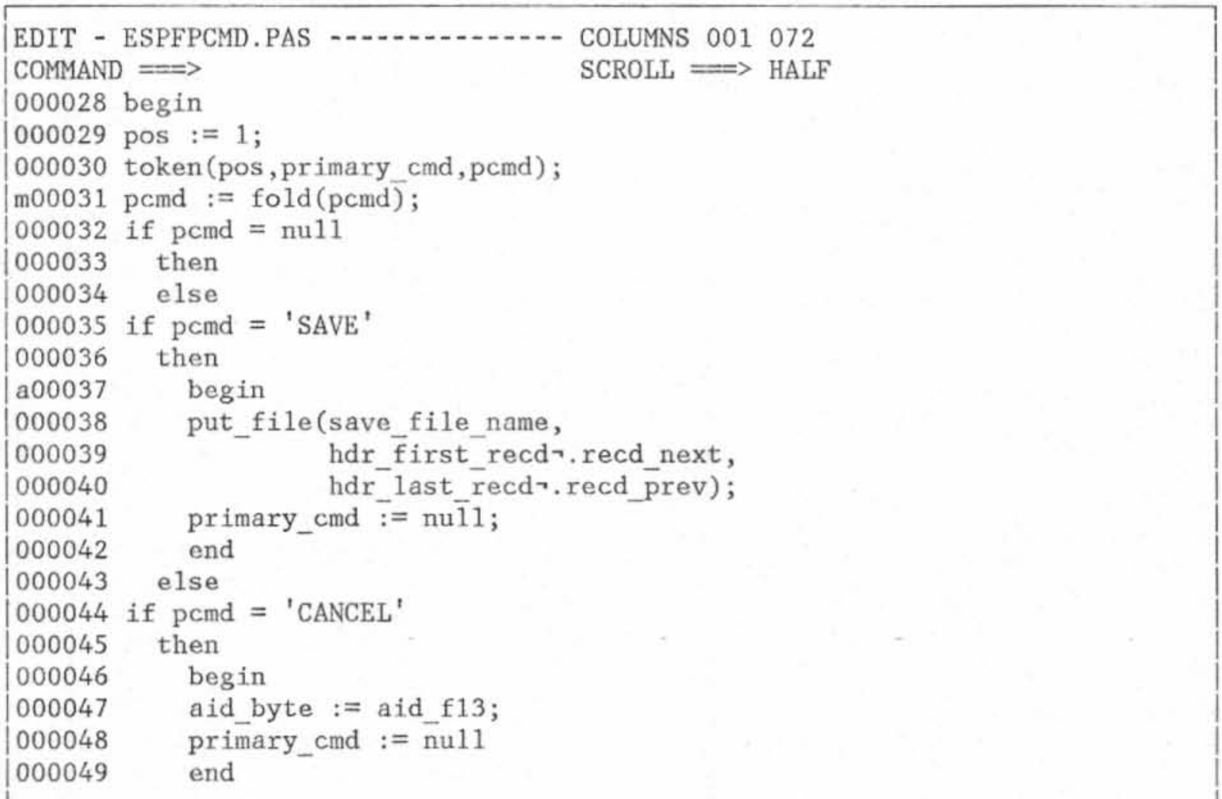
Figure 8. Copying Lines -- After.

MOVING LINES WITHIN YOUR FILE

The M (move) line command is very similar to the copy line command. The difference is that the original lines will still be there after a copy command and they will be deleted during a move command.

The move line command comes in both single and block forms. It is used in conjunction with the before and after line commands which have only the single form. To use the single form, type a 'M' followed by the number of lines to move over the first line number to be moved and type an 'A' over the line number after which you wish the lines moved or type a 'B' over the line number before which you wish the lines moved. To use the block form, type a 'MM' over the first line number to be moved, another 'MM' over the last line number to be moved, and type an 'A' over the line number after which you wish the lines moved or type a 'B' over the line number before which you wish the lines moved.

-- Figure id 'moveb' unknown -- and -- Figure id 'movea' unknown -- show an example of how to move lines.



```
EDIT - ESPFPCMD.PAS --------------- COLUMNS 001 072
COMMAND ===>                        SCROLL ===> HALF
000028 begin
000029 pos := 1;
000030 token(pos,primary_cmd,pcmd);
m00031 pcmd := fold(pcmd);
000032 if pcmd = null
000033   then
000034   else
000035 if pcmd = 'SAVE'
000036   then
a00037     begin
000038     put_file(save_file_name,
000039              hdr_first_recd¬.recd_next,
000040              hdr_last_recd¬.recd_prev);
000041     primary_cmd := null;
000042     end
000043   else
000044 if pcmd = 'CANCEL'
000045   then
000046     begin
000047     aid_byte := aid_f13;
000048     primary_cmd := null
000049     end
```

Figure 9. Moving Lines -- Before.