

SPFLite

Introduction

Welcome to SPFLite

[Top](#) [Next](#)

Who is it for?

For IBM MVS and VM developers and support people who are just comfortable with their favourite editor and hate having to 'think differently' to work with PC editors.

What's in SPFLite?

It is **NOT** a full-blown SPF dialogue system of Panels, Menus, Utility Functions, etc. It is simply the Editor portion of SPF with as many of the features and functions of the editor as I could manage to duplicate. A simple File Manager option is available which provides for file selection for Edit & Browse, as well as a File Delete and Rename function.

Although the editor is almost totally keyboard oriented, like the original SPF, some extensions have been made where they could be added as optional facilities that did not alter functioning for keyboard 'purists'.

I have tried to retain all the 'look and feel' of the original editor, but this is not, and never will be a complete clone. There are minor differences in some commands, some operands/functions are not supported, but overall I think it provides a usable editor which contains 99% of the functionality that I ever used myself. I trust that this is also true for you.

One note of caution, SPFLite is NOT suitable for processing files with a **large** number of lines (> 100,000). Although it will process such files, the internal memory management techniques used may cause performance to slow down to unacceptable levels.

Why did I bother creating SPFLite?

Well, I was just getting frustrated after trying out just about every Windows PC Editor I could lay my hands on. They were all *character stream oriented* editors and what I wanted was a *line oriented* editor like the 'good old' SPF editor I had used most of my career.

Basically, if I want a word processor, I'll use a word processor; when I want to edit source files, script files, etc. I wanted a **line** editor.

I looked for other SPF clones, and discovered that there just isn't much out there. The commercial products (e.g. CTC's version) were **way** out of my price range and secondly looked like they had been programmed by PC/Windows programmers who had simply never actually done real work on the real mainframe SPF.

So, as my wife so bluntly put it, "You've been a programmer all your life, create it yourself!" So that's where this came from.

Licensing / Distribution

SPFLite is distributed as Shareware and all copies and versions are fully functional and will remain fully functional even after the trial period has expired. The only difference after the trial period, if you have not registered the program, is that it will randomly 'nag' every few uses as it terminates. You are basically on your honour to pay for software that you find useful.

Support / Contacts

The latest release of SPFLite may be obtained at any time via the SPFLite web site:

spflite.co.nr

or from an alternate download site:

download.com

For other specific questions or support regarding SPFLite, please send an e-mail to:

SPFLite@gmail.com

Features of SPFLite

Features of SPFLite

Features

The following summarizes the SPFLite features:

- Normal 32 bit Windows application.
- Standard Long File Name support.
- Standard Windows Clipboard support (Text mode).
- User specified Windows size (Height & Width) as well as Font Name and Pitch (only fixed width fonts however).
- Support for national keyboards and alternate layouts (normal Windows Regional settings).
- UNDO support (Number of levels user specifiable).
- BROWSE command line mode, and Primary BROWSE command to help prevent inadvertent changing of data.
- Simple Macro support (Primary Commands only, simple variable substitution).
- Source Colorize support.
- HEX mode editing.
- Tabbed edit interface to support separate simultaneous edits.
- Drag / Drop support. Files can simply be dropped on an active SPFLite window to Open them.
- Optional Integrated File Manager for file selection and basic Delete/Rename functions.
- Basic Unicode file editing support.
- Print Screen options for output to Clipboard, Default Printer or Log file.

The SPF Primary commands implemented are shown below.

Note that while not listed below, all **normal** abbreviations are also supported (F/FIND, C/CHANGE, etc.):

Primary Commands

<u>BOUNDS</u>	Set edit Boundaries
<u>BPASTE</u>	Block Paste from the Windows Clipboard
<u>BROWSE</u>	Open a file for Browsing
<u>CANCEL</u>	Cancel the Edit without saving
<u>CAPS</u>	Set CAPS mode On / Off
<u>CHANGE</u>	Change one string to another within the data
<u>COPY</u>	Copy an External File into the Edit data
<u>CRETRIEV</u>	Conditional Retrieve
<u>CRLF</u>	Temporarily alter the PROFILE CRLF setting
<u>CUT</u>	CUT selected text to the Windows Clipboard
<u>DELETE</u>	Delete a selected range of lines
<u>DOWN</u>	Scroll Downward in the data
<u>EDIT</u>	Edit another file (starts another SPFLite session)
<u>ENCODING</u>	Specify the UniCode encoding to be used.
<u>END</u>	End the Edit session (Save depends on user preferences)
<u>EXCLUDE</u>	Exclude selected lines from the Edit display
<u>FIND</u>	Find specified string within the data
<u>FLIP</u>	Reverse X'd and NX'd lines
<u>HELP</u>	Display the Help file
<u>HEX</u>	Switch To / From Hex display mode
<u>HIDE</u>	Hide Excluded Lines from the Display
<u>LEFT</u>	Scroll Leftward in the data
<u>LOCATE</u>	Move the edit window to a specified line number
<u>OPEN</u>	OPEN a new file for Editing
<u>PASTE</u>	PASTE data from the Windows Clipboard
<u>PFKEY</u>	Specify commands associated with the Function Keys
<u>PRINT</u>	Print the edit file, or selected line range to the default Printer.
<u>PROFILE</u>	Set program and operational defaults

<u>RCHANGE</u>	Repeat the previous Change command
<u>REPLACE</u>	Replace an external file with all or selected data
<u>RESET</u>	Reset pending commands, etc.
<u>RETRIEVE</u>	Retrieve previous Primary commands for re-issue
<u>RFIND</u>	Repeat a previous Find command
<u>RIGHT</u>	Scroll Rightward in the data
<u>RUN</u>	Execute another command / program
<u>SAVE</u>	Save the edited data back to the original file
<u>SAVEAS</u>	Create a new file and switch the edit to the new file.
<u>SORT</u>	Sort the data on specified columns
<u>SWAP</u>	Switch between active Edit tabs
<u>TABS</u>	Set Tabs mode On / Off
<u>UNDO</u>	Undo changes since last attention key
<u>UP</u>	Scroll Upward in the data

Line Commands

The SPF Line commands implemented are:

<u>A / B</u>	Specify the line A fter or B efore which lines are to be moved / copied.
<u>BNDS</u>	Insert a Bounds line to allow specification of Edit boundaries
<u>C / CC</u>	Copy 1 or more lines, or a range of lines to another location
<u>COLS</u>	Insert a non-data COLS line to act as a Ruler
<u>D / DD</u>	Delete 1 or more lines, or a range of lines
<u>DLMS</u>	Insert a DLMS line to allow altering Word Delimiter characters
<u>F / L</u>	Specify 1 or more lines from the F irst or L ast part of an Excluded range to be made visible
<u>I</u>	Insert 1 or more temporary blank lines
<u>J</u>	Join a text line with the following line.
<u>LC / LCLC / LCC / LL</u>	Lowercase 1 or more lines, or a range of lines
<u>M / MM</u>	Move 1 or more lines, or a range of lines to another location
<u>N</u>	New Permanent Lines
<u>O / OO</u>	Specify the line, or range of lines for a Copy / Move Overlay command

<u>R / RR</u>	Replicate 1 or more lines, or a range of lines
<u>S</u>	Show indented lines from Excluded lines
<u>SC / SCSC</u>	Sentence Case 1 or more lines
<u>TABS</u>	Insert a Tabs line to allow specification of Tab boundaries for Edit
<u>TC / TCTC</u>	Title Case 1 or more lines
<u>TF</u>	Text Flow the selected paragraph maintaining current indents.
<u>TS</u>	Text split the selected line and insert blank lines between
<u>UC / UCUC / UCC / UU</u>	Uppercase 1 or more lines, or a range of lines
<u>X / XX</u>	Specify 1 or more lines, or a range of lines to be Excluded from viewing
<u>/ / (</u>	Left shift 1 or more lines, or a range of lines
<u>/ /)</u>	Right shift 1 or more lines, or a range of lines
<u>< / <<</u>	Left Data shift 1 or more lines, or a range of lines
<u>> / >></u>	Right Data shift 1 or more lines, or a range of lines

Installing SPFLite

Installing The Distribution

[Previous](#) [Top](#) [Next](#)

Installation

SPFLite is distributed as a standard compressed ZIP file. This file should be uncompressed (using standard UnZip tools, or Windows Explorer) to create the Installer executable named SPFLiteSetup.EXE. Next, simply double-click the EXE file to start the install. Follow the normal install prompts to complete the installation.

If you wish to uninstall SPFLite, an Uninstall Icon has been created during the install, simply double click this Icon to remove the program. Or you can use the standard Windows Add/Remove program support, just select SPFLite and click 'Remove'.

The first time you execute SPFLite, a default set of Preferences will be created and you will be presented with the standard Preferences Dialog where you can choose to alter or modify any of SPFLite's customization options.

Creating a Portable Version

Creating a Portable Version

[Previous](#) [Top](#) [Next](#)

SPFLite can be copied to removable media (e.g. a thumb drive) complete with its customized environment

for use on systems other than the one on which it was originally installed.

If you plan to do this, complete all customization activities **before** performing the following steps. If you are a registered SPFLite user, also do this **after** the registration is complete.

1. Copy the SPFLite.EXE file from C:\Program Files\SPFLite to the removable media..
2. If you are not sure where the SPFLite customization files are stored,
 - Start SPFLite
 - Enter PROFILE
 - The correct location will be shown at the bottom of the Preferences Dialog Box.
3. Using the directory located in 2. above, copy all the INI, CLR and SPM files from this directory to the same location on the removable media as the EXE file was copied.

Your portable version is complete, when executed **from** the removable media, all your customization will be retained.

NOTE:

If you copy SPFLite from the removable media to another system's hard drive and execute it from that hard drive, SPFLite will NOT retain your customization.

Customizing SPFLite

Customizing SPFLite

[Previous](#) [Top](#) [Next](#)

SPFLite provides a large variety of optional settings to allow you to customize it to suit your own preferences. Although a default set of values is installed with the program, you should review the following customization options as it's highly unlikely that anyone would wish to run with the standard defaults.

If this is your first time using SPFLite, you will automatically be presented with the SPFLite Preferences Dialog to allow you to choose the options you prefer.

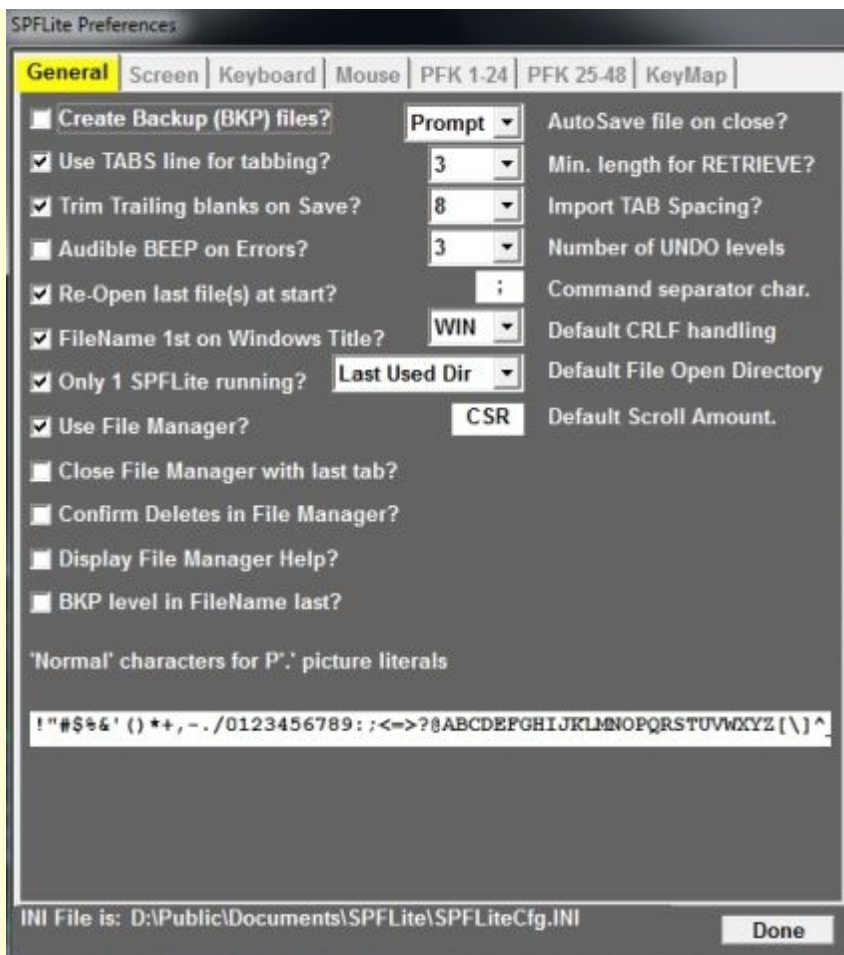
At any other time, simply enter the [PROFILE](#) (or PREF) command to reach the Preferences Dialog. The Preferences are grouped into six categories, each with its own tab. Review each tab's contents and when complete, press the **Done** button at the bottom.

Preferences - General

Preferences - General

[Previous](#) [Top](#) [Next](#)

Preferences - General Tab



Create Backup (BKP) files?

If selected (set to '**ON**'), SPFLite will create a backup copy of any Edited file by storing it with an additional level of '.BAK'. Depending on the "BKP level in FileName last?" option (below) the .BAK will be either added to the name, or inserted before the last level. e.g. A Backup of MYDATA.TXT would be saved as MYDATA.TXT.BAK or MYDATE.BAK.TXT. If the .BAK file already exists, it will be overlaid. If unselected ('**OFF**') is specified, then no Backup will be created.

Use TABS line for Tabbing?

If selected (set to '**ON**'), SPFLite will honor the tab settings specified by the TABS line commands. If set to '**OFF**' the TABS settings will be ignored. The settings are not erased, they are retained for future usage the next time this option is set to '**ON**'. Note that this specifies the **Default** setting when SPFLite is started. The TABS setting can be temporarily altered during execution with the Primary [TABS](#) command

Trim trailing blanks on Save?

If selected (set to '**ON**'), then trailing blanks on lines will be removed when saving the data, If set to '**OFF**', then trailing blanks will be retained.

Audible BEEP on Errors?

If selected (set to '**ON**'), then SPFLite will sound the standard Windows error sound when an error message is displayed or if a key is pressed in a non-typing area. If UnSelected('**OFF**'), then no sound

will occur.

Re-Open last file(s) at Start?

If Selected (set to YES), then if SPFLite is started without a filename as a command line operand, it will automatically Re-Open for edit the last set of file(s) being edited when SPFLite was terminated.

NOTE: The 'set' of files is only 'remembered' if SPFLite is closed using the Title Bar Close button (X). If tabs are closed one at a time with an END command, there will be **no** remembered file list.

Filename 1st on Windows Title?

If selected (set to Yes) then the filename of the file being edited will be positioned 1st (leftmost) in the Windows Title bar rather than last. If multiple instances of SPFLite are running, this can make switching between sessions via the task bar easier since the filename will likely be what is visible rather than the leftmost characters of "SPFLite(V4.3.2) - filename"

Only 1 SPFLite running?

If selected, multiple separate executions of SPFLite will normally be prevented. If another SPFLite is started, it will detect the one running and cause the file to be opened as a new tab within the existing SPFLite execution. The 2nd SPFLite will simply terminate. When this option is selected, a completely new SPFLite window can still be opened via the EDIT primary command in the running SPFLite.

Use File Manager?

If selected, SPFLite will activate the File Manager support. A full description of File Manager is available in [Working with the File Manager.](#)

Close File Manager with last tab?

This setting is only applicable if the previous item was activated to use the File Manager.

If selected, then an [END](#) command on the last active Edit/Browse tab will also close the File Manager tab. If not selected, then following the close of the last active Edit/Browse tab, you will be switched to the File Manager Tab.

Confirm Deletes in File Manager?

If File Manager is activated, and this item is selected, then each Delete action selected for a file will result in a confirmation prompt before the file is actually deleted. If not selected, the file will be deleted with no further prompt.

Display File Manager Help?

When File Manager is activated it will, by default, display a few lines of abbreviated Help at the bottom of the screen. Once you have used File Manager for a while, these lines can be removed to reclaim the screen 'real estate' for normal use. Use this option to turn ON/OFF the Help display.

BKP level in Filename Last?

If **Create Backup (BKP) files?** has been selected to cause BKP files to be created, then this option allows you to choose how the filename for the Backup file is to be built.

If this option is ticked (selected) then the Backup filename will be created by simply appending the characters **.BKP** to the end of the normal filename.

If this option is not ticked (de-selected) then the Backup filename will be created by simply **inserting** the characters **.BKP** prior to the filetype level of the normal filename.

e.g. For a normal filename of MYDATA.TXT, selecting this option will create MYDATA.TXT.BKP. If un-selected the backup file would be named MYDATA.BKP.TXT.

AutoSave file on Close?

Choose one of three values.

- If set to '**ON**', the Edited data will be saved automatically when an END or =X command is issued and the data has been modified.
- If '**OFF**' is specified, the data will only be saved when an explicit 'SAVE' command is issued.
- If '**PROMPT**' is specified, then if the data has been modified since loaded (or since the last SAVE) then you will be prompted if you wish to save the data before Exiting.

Min. length for Retrieve?

SPFLite saves command line entries for subsequent retrieval via the RETRIEVE command. The MinRetrieve value specifies the minimum length of a command line before it will be saved for retrieval. This prevents filling the retrieve stack with small entries which are simpler to just re-enter. Note that saved commands are also saved between SPFLite sessions.

Import TAB spacing?

SPFLite does NOT save files in Tabbed format. In this day and age of under \$1.00 / Gigabyte hard drives, compressing blanks to tabs is simply not justified to save space. If SPFLite **encounters** files with embedded tabs, it will use this value to expand the incoming tabs.

Number of UNDO levels?

You may specify the number of Undo levels which SPFLite should maintain. Specify your desired number of Undo levels (2 - 25), or specify 0 (zero) if you do not wish Undo support.

Command Separator Char.

Multiple commands may be entered on the SPFLite command line. This option specifies the character to be used to separate each command. The default is **[:]** - the semicolon.

Default CRLF handling?

Choose one of the following to direct how text line endings should be handled:

- **WIN** - stores text lines with **both** a CR and a LF character between lines. (CR/LF = Carriage Return / Line feed)
- **UNIX** -stores text lines with **only** a LF between lines.
- **MAC** - stores text lines with **only** a CR between lines.

For more information on handling of end-of-lines. See ["Handling Non-Windows Txt Files"](#) .

NOTE: **Opening** or **Copying** a file with a format different from this default will cause a warning message to be issued to make you aware of the different format of the file.

Default File Open Directory

This setting allows you to choose one of two Directories to use as the Default when SPFLite attempts to Open or Save a file.

- Working Dir

SPFLite will use the current working directory at the time it is started.

- Last Used Dir

SPFLite will use the last directory from which it Opened or Saved a file.

Default Scroll Amount

This setting allows you to specify the default scroll amount used by the UP / DOWN / LEFT / RIGHT commands for text scrolling. For each edit tab, this default amount can be overridden via the Scroll > field in the upper right corner of the Edit / Browse screen. Valid entries for this field are:

PAGE

The screen will be scrolled by the number of lines / characters visible on the screen.

HALF

The screen will be scrolled by 1/2 the number of lines / characters visible on the screen

DATA

The screen will be scrolled by the number of lines / characaters visible on the screen **minus 1**.

CSR

If the cursor is currently within the actual text area, the current line / character will be scrolled as appropriate to the edge of the current view window. If the cursor is outside the actual text area, it will act as if **PAGE** was entered.

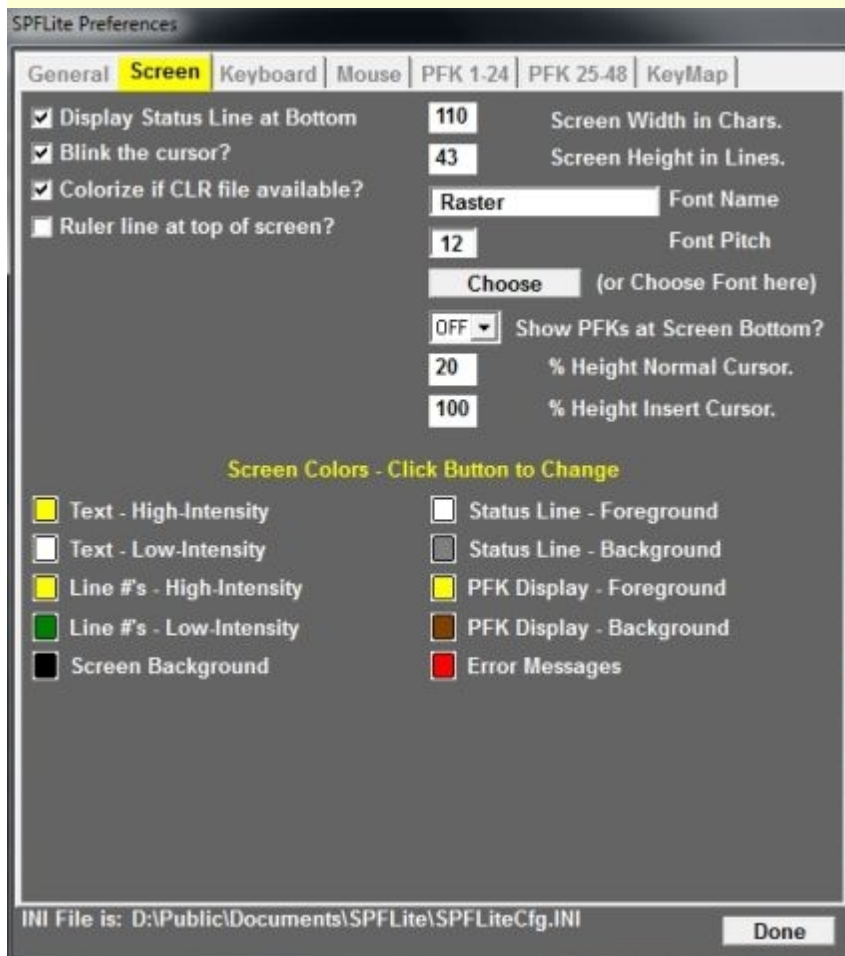
nnnn

If a numeric value it entered, then scrolling will use that value as the number of lines / characters to scroll.

'Normal' characters for P'.' Picture Literals

The characters in this text box will be considered as normal display characters when using the Picture Literal character '.' (Period). The period is used to locate characters which cannot be displayed normally on the screen. The default is the normal ASCII character set. You may alter this as needed to incorporate additional characters such as national characters, or other special characters. Simply add or remove characters as needed.

Preferences - Screen Options



Display Status Line at Bottom

If selected (**ON**) then SPFLite will display a Status Line at the bottom of the screen containing Keyboard status (Insert/Overtyp) etc. and the specific cursor location within the logical text.

Blink the Cursor?

If selected (**ON**) the cursor will blink regularly to make it easier to spot.

Colorize if CLR file available?

If selected (**ON**) then SPFLite will search for a Colorize control file to match the current edited file type, and use it to colorize the text display. If unselected (**OFF**) then no colorize will be done even if a CLR control file exists. See [Colorize Files](#) for more info on Colorize support.

Ruler line at top of screen?

If selected (**ON**) then a permanent COLS type line will be placed at the top of the screen. Doing this in

no way affects the use of COLS line commands elsewhere in the edit file.

Screen width in Chars.

Specify here your desired width for the Edit screen. Note the width includes the line number area, so a width of 80 would display only 72 characters of actual text from the file.

Screen height in Lines

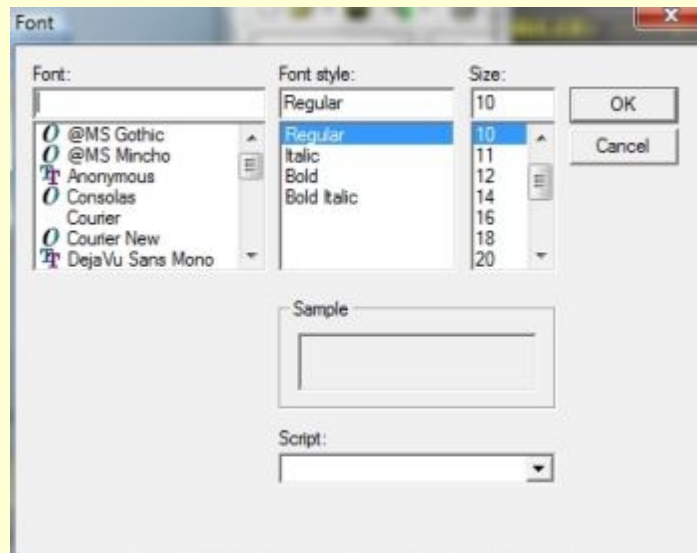
Specify the desired height of the screen, in lines.

Font Name

Font Pitch

Choose

These three related items allow you to choose what screen font will be used for the display. You can either directly specify a specific name and pitch (e.g. TERMINAL / 11) or select the **Choose** button to invoke the standard Font Selection Dialog. Note that **only fixed (non-proportional)** fonts are supported.



Show PFKs at screen bottom?

You can choose to have a summary of the PFK commands displayed at the bottom of the screen. Your choices are:

- **OFF** - No display will be done
- **12** - Display only PFK keys 1 to 12. This will occupy 2 screen lines.
- **24** - Display PFK keys 1 to 24. This will occupy 4 screen lines.

Note: The default display will show the command that will be entered (as much as will fit) but you may instead display a PFSHOW Label instead to be more meaningful. See [PFSHOW Labels](#) for info on setting the label.

% Height Normal Cursor.

% Height Insert Cursor.

These two values control the appearance of the screen cursor in Normal and Insert mode. The value given is a % of a full height Blob cursor (i.e. one which fills the character space). The value can range from 20 to 100%. Note: values below 20% tend to be either invisible, or fragmented dots.

Screen Colors - Click button to change

The bottom of this tab is used to select the colors you wish to use for the screen display. The currently selected colors are displayed in the small square box to the left of each text type. To alter a color, click the displayed color box and you will be shown the standard system color chooser dialog.

Note: If you use CLR files to control colorizing source files, and you alter the Background color setting in Preferences, then you should ensure that the Background color chosen for Scheme 0 (the default) in the colorize file be set to the same value you have chosen in Preferences. Otherwise some areas of the screen background will be shown in different colors.



Preferences - Keyboard

Preferences - Keyboard

[Previous](#) [Top](#) [Next](#)

Preferences - Keyboard



Scroll screen with Arrow Keys?

If selected (**ON**), when the cursor is moved to the edge of the actual text area, further movement will cause the screen to scroll data into view based on the cursor direction. If unselected (**OFF**), then the cursor will 'wrap' to the opposite edge of the screen, as it did on the original 3270s.

KB Starts in INSERT mode?

If selected (**ON**), then the keyboard will be in INSERT mode as the default when SPFLite is started, otherwise it will be in the normal (for a 3270) mode of overtype.

Reset INSERT mode on Attn?

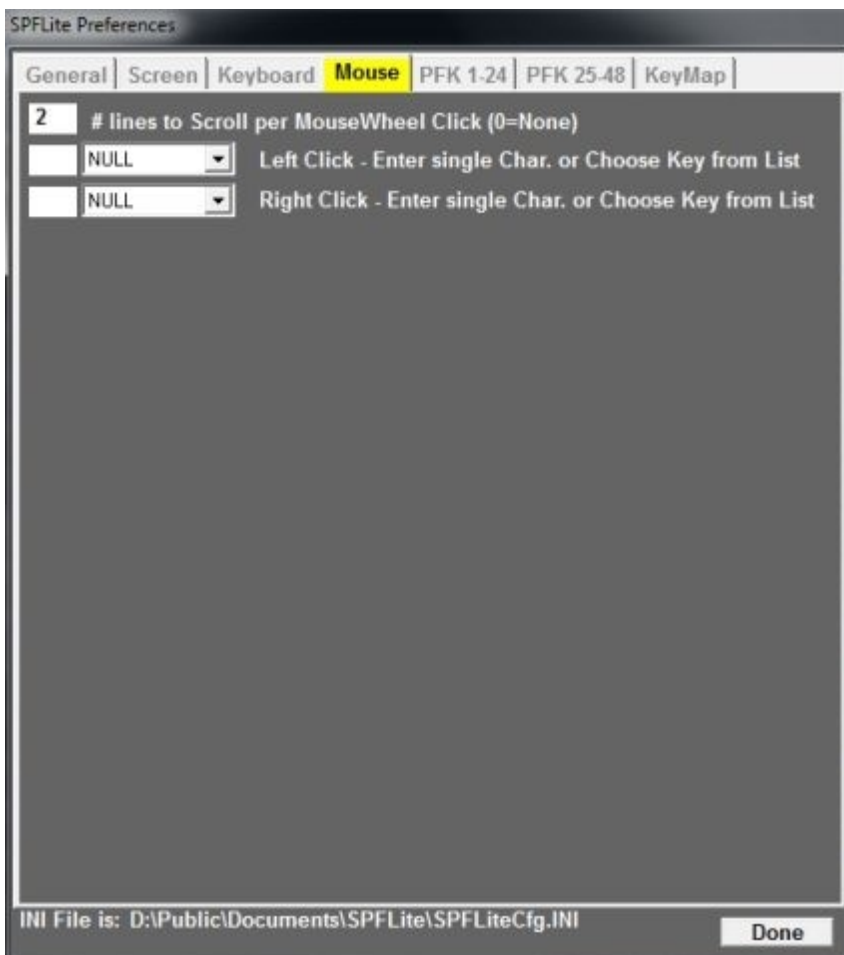
If selected (**ON**), then when any Attention key is pressed (Enter, PFK) then the Insert mode will be reset to the value specified in the previous option. If not selected (**OFF**) the INSERT status will not be altered.

Preferences - Mouse

Preferences - Mouse

[Previous](#) [Top](#) [Next](#)

Preferences - Mouse



lines to Scroll per Mouse Wheel Click (0=None)

The value entered here will be used to control the number of lines (for Vertical scrolling), or characters (for horizontal scrolling) to be performed for each 'click' of the mouse wheel. If 0 (zero) is entered, then no action is taken on MouseWheel activity.

Left/Right Click - Enter single Char. or Choose Key from List

When the left or right mouse button is clicked, the default action will be to move the current text cursor to the location of the mouse click.

These two options are used to specify **optional** activity to be performed **after** moving the cursor. For each button (left / right) you may choose one of the following:

- Enter a single character in the left-hand box. This character will be 'typed' into the location selected by the mouse click.
- Select a key action from the right-hand pull-down list. The desired 'key' will be pressed following the cursor move.

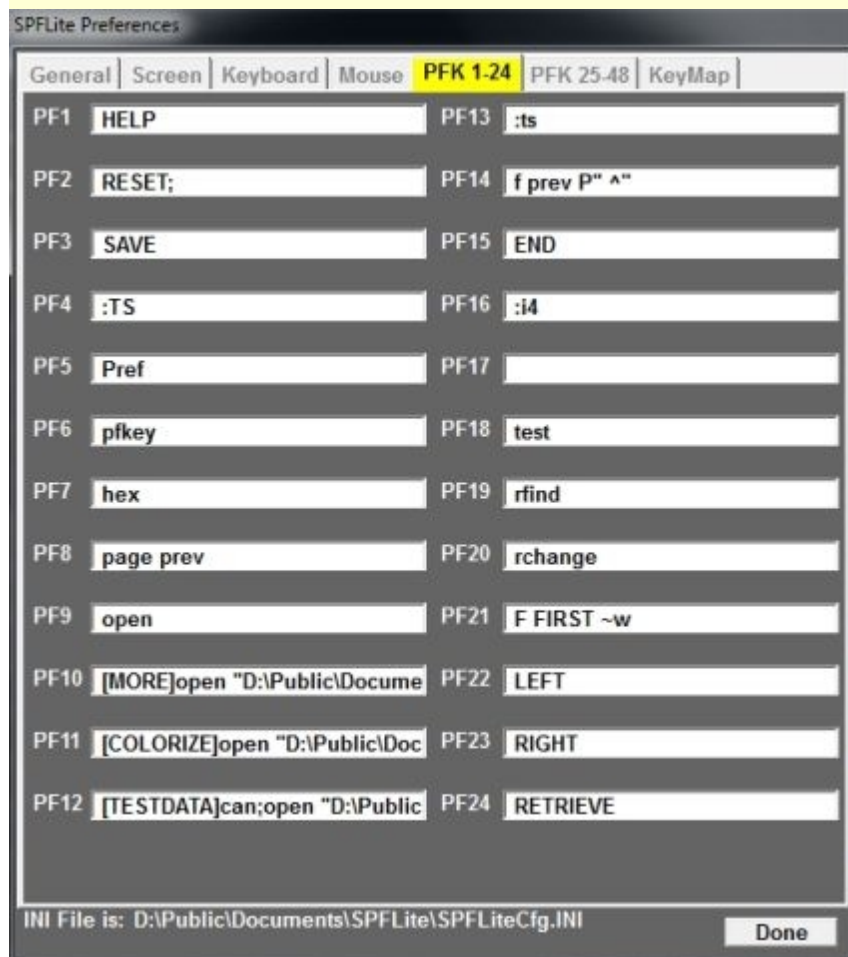
Note: If no action is desired, leave the left-hand box blank, and select **NULL** from the right-hand pull-down list.

Preferences - PFKEYS

Preferences - PFKEYS

[Previous](#) [Top](#) [Next](#)

Preferences - PFKEYS



This tab simply contains the command strings you wish to be entered for each of the PF keys. Simply alter each one as desired. There are two Preferences pages devotes to PFKEYs, with 24 PFKEYs on each tab.

PFSHOW Labels

If you select "Show PFKeys at bottom of Screen" under Preferences - Screen, then you may wish to utilize PFSHOW labels. This enables you to specify the text you wish to appear display for each key rather than simply displaying the command itself. To do this, prefix the command with your desired label, surrounded by square brackets.

Example: If PF35 above were entered as: [ONLY]x all; f all ~w

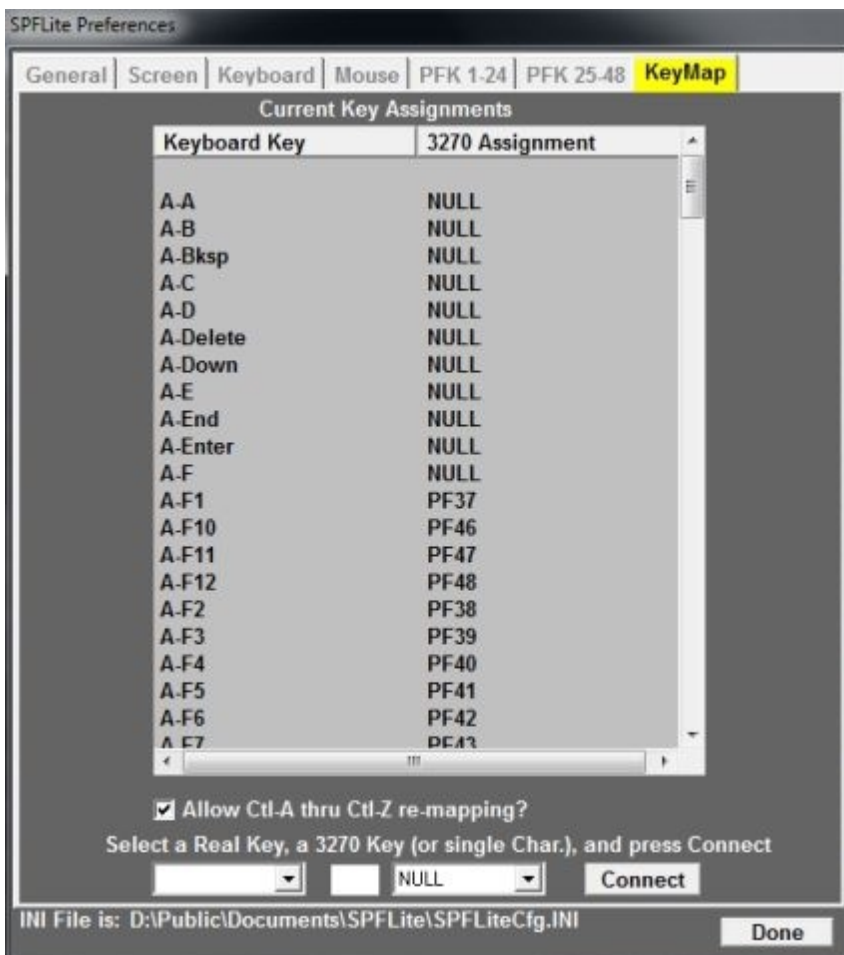
Then it would appear in the PFSHOW Display as: 35=ONLY
But would issue the command: x all;f all ~w

Preferences - KeyMap

Preferences - Key Map

[Previous](#) [Top](#) [Next](#)

Preferences - Key Map



The top half of this tab contains a display of the current keyboard keys, and the equivalent 3270 action key to which they are mapped. Scroll this as needed to examine the current assignments.

SPFLite allows you to re-map just about every possible key on the keyboard. This includes:

- The 12 Function keys in each of Normal, Shift, Control and Alt versions
- The Ctl key (left and right), the Alt key (left and right), and the Windows and Windows Menu keys, although using these is **bound** to cause you some problems with Windows itself.
- The Bksp, Delete, Down Arrow, End, Enter, Home, Insert, Left Arrow, Pg Down, Pg Up, Right arrow and Up arrow keys, in each of Normal, Shift, Control and Alt versions.
- All keys on the Numeric keyboard.
- The Ctl-A thru Ctl-Z and Alt-A thru Alt-Z keys.
- The ESC, Pause, Tab and Shift Tab keys

Using Ctl and Alt Key combinations

You are allowed to map the Ctl and Alt keys themselves. e.g. A common use is to map the Right-Ctl key as the 3270 Enter key. If you do this, be aware that it will impact your ability to re-map the combination keys such as Ctl-x or Alt-F2 since the Ctl/Alt key will be interpreted first. You MAY map only one of the two Ctl or Two Alt keys and use the other for the combination keys; this is acceptable. Just be sure to think through this aspect before charging ahead.

Allow Ctl-A thru Ctl-Z re-mapping?

If this is selected, then the Ctl-x keys may be re-mapped as 3270 function keys. If left un-ticked, the Ctl-x keys will simply enter the ASCII characters for the current code page.

Altering Key Assignments

To alter the assignment for any key, use the fields at the bottom of the tab. There are four (4) entry fields here:

Real Key

Use this pull-down list to select the keyboard key you wish to modify. When you make a selection, the display above will be scrolled to allow you to see the currently assigned definition.

Actual Text Character

If you wish the key to be treated as a normal text character, enter that character here. For special characters use the normal Alt+nnnn format. e.g. ALT-0176 = ° (degree).

3270 Key

If you wish the key to be mapped to a 3270 control key, use this pull-down list to select the desired control key and leave the previous entry field blank.

Connect

Once you have the **Real Key** and **3270 Key** set appropriately, press the **Connect** button to complete the assignment.

Repeat the key selections and Connect actions for each key you wish to customize.

Differences Between SPF Lite and Full SPF

Differences between SPFLite and Full SPF

[Previous](#) [Top](#) [Next](#)

As described under [Preferences - Keyboard](#) you may map the physical control keys to any logical SPF keys as you desire.

The following description describes the default mapping provided if you do **NOT** modify KeyMapping using the PRO Preferences Dialog.

Obviously your PC does not have a 3270 keyboard, so the following default mapping is in place for the keyboard.

- All alphabetic and numeric keys operate as normal.
- Insert / Delete / Home / Tab and Back-Tab (Shift-Tab) operate as normal
- Page Up / Page Down keys by default are mapped as function keys which generate UP and DOWN commands. i.e. you do not **have** to assign the Up and Down commands to other keys, although you may certainly do so if you wish. Default keyboard mapping of these two Keys is to map them as PF25(PgUp) and PF26(PgDn).
- The standard F1 thru F12 keys are mapped to the 3270 PF1 thru PF12 keys; the Shifted Function keys are mapped as PF13 thru PF24 keys, the Control Function keys are mapped as PF25 thru PF36 and the Alt Function keys as PF37 thru PF48.
- The 3270 keys PF13 thru PF24 are also mapped to the numeric keypad to simulate the physical PF key layout of the original 3278 terminals

KeyPad NumLk PF 13	KeyPad / PF 14	KeyPad * PF 15
KeyPad 7 PF 16	KeyPad 8 PF 17	KeyPad 9 PF 18
KeyPad 4 PF 19	KeyPad 5 PF 20	KeyPad 6 PF 21
KeyPad 1 PF 22	KeyPad 2 PF 23	KeyPad 3 PF 24

- The four arrow keys (Up, Down, Left, Right) function as normal. Please review the setting on the KeyBoard Tab in Preferences. See [Preferences - Keyboard](#) for the option which controls the handling of the arrow keys when reaching the bounds of the data area of the screen.

- Ctl-LeftArrow / Ctl-Right arrow will move the cursor to the Previous / Next 'word' when the cursor is IN the normal text area of the screen.
- The Ctl-A thru Ctl-Z keys may be left as normal data keys, or you may allow them to be re-mapped as SPF control keys. See [Preferences - Keymap](#) for more details.

The full list of available keys to re-map can be found under [Preferences - Keymap](#).

Use of the Mouse

Use of the Mouse

[Previous](#) [Top](#) [Next](#)

Since 3270 terminals had no mouse, SPFLite is basically designed to operate in the same manner, i.e. totally via the keyboard. However one cannot simply ignore the fact that SPFLite **is** running on a PC, so the following mouse support is available:

1. When the file commands ([OPEN](#), [COPY](#), [CREATE](#), [REPLACE](#)) do **not** specify the filename as part of the command operands, SPFLite will use the standard system File Open and Save dialog boxes to prompt for file locations. These Pop-Up boxes allow use of the mouse for selection purposes.
2. If keyboard scrolling is activated (see [Preferences - Keyboard](#)) the Mouse scroll wheel may be used to scroll the text window. The number of lines scrolled by each mousewheel click is set in Preferences. Scrolling is normally vertical, but holding the **Shift** key while using the Mouse Wheel will result in a Left/Right scroll.amount
3. During normal editing, the Left Mouse Button can be clicked anywhere within the screen area to move the cursor quickly to that location, and will then **optionally** issue an SPFLite keyboard key. This action is specified by the [Preferences - Mouse](#) setting.
4. The Right Mouse Button is similar to the left, it will cause the cursor to move to the selected location, and will then **optionally** issue an SPFLite key. This action is also specified by the [Preferences - Mouse](#) setting.
5. The Tabbed interface introduced with SPFLite V4.0 adds a new mouse function - switching Tab Pages. Simply left click on a Tab at the top of the screen to switch to that tab's contents.
6. During Edit, the left mouse button can be held down and dragged over the text to select a block of text. A frame will be drawn on the screen to indicate the area you have selected. Once selected the marked text can be placed on the clipboard using the [CUT](#) command.

If you have selected the incorrect data, simply release the left button and start over again.

Note:

The selected data area is restricted to what is visible on the current screen, you may not scroll the screen while selecting text.

If you want to select more data than is visible vertically, use the [Exclude](#) command ([X](#)) to 'collapse' the desired lines into the visible area.

The block select area will include excluded lines into the selected data.

Default PF Key definitions

Default PF Key definitions

[Previous](#) [Top](#) [Next](#)

Following installation, the following PF Key definitions will be defined. These can be altered at any time using the [PFKEY](#) primary command which will Open the Preferences Dialog at the PFKEYS tab.

Normal F1 - F12 Keys		Shift F1 - F12 Keys		Ctl F1 - F12 Keys		Alt F1 - F12 Keys	
PF1	HELP	PF13	HELP	PF25	UP	PF37	
PF2	RESET	PF14	RESET	PF26	DOWN	PF38	
PF3	END	PF15	END	PF27		PF39	
PF4	:TS	PF16	:TS	PF28		PF40	
PF5	RFIND	PF17	RFIND	PF29		PF41	
PF6	RCHANGE	PF18	RCHANGE	PF30		PF42	
PF7	:TF80	PF19		PF31		PF43	
PF8		PF20		PF32		PF44	
PF9		PF21		PF33		PF45	
PF10	LEFT	PF22	LEFT	PF34		PF46	
PF11	RIGHT	PF23	RIGHT	PF35		PF47	
PF12	RETRIEVE	PF24	RETRIEVE	PF36		PF48	

Working With SPFLite

Working With SPFLite

[Previous](#) [Top](#) [Next](#)

The topics following describe in general terms how to perform some basic functions in SPFLite. They are in (hopefully) logical groupings by topic; the intent is to remove these general discussions from the nitty-gritty of the various command descriptions

Starting and Ending SPFLite

Starting and Ending SPFLite

[Previous](#) [Top](#) [Next](#)

Starting SPFLite

Normally:

SPFLite is started normally by simply double-clicking its Icon which was created during the install. If you have activated the optional File Manager, you will be presented with that Tab active containing your default file list. If File Manager is **not** activated, then an edit window will open on a single tab as an empty dataset. At this point you can use the [OPEN](#) command to select a file for processing.

Drag / Drop

You can either:

- Drag and Drop a file from Explorer onto the SPFLite Icon to have the program start with that file loaded. If File Manager is activated, then it will also be Opened as the 1st tab, the Edit file tab as the 2nd tab.
- Drag and Drop a file onto an existing SPFLite window. If the current tab is (Empty) the file will be opened on that tab; if the current tab contains data, or is the File Manager tab, the dropped file will be opened in a new tab.
- A Shortcut to a data file rather than the file itself, may also be Dragged and Dropped.

Command Line Options

The normal use of the command line operand is to provide the starting filename which is to be Edited. Note that if SPFLite is started **without** a file name **and** the [Preferences - General](#) setting for re-Open Last file is ON, then SPFLite will automatically Re-Open the last set of files being edited when SPFLite was terminated.. If File Manager has been activated, it will also be opened.

However, several other options are supported:

-CLIP option.

If only the characters '**-CLIP**' (or any abbreviation such as **-C**) appear as the command line, SPFLite will operate in Clipboard Mode.

In this mode:

- The contents of the Windows Clipboard are automatically PASTEd into an empty file to become the

- working data when SPFLite starts.
- When the primary Command '**END**' is entered, the current edit data is **CUT** to the windows clipboard before termination.

Clipboard Mode is designed to provide a quick, convenient means for editing clipboard data with the SPFLite command set. It works most effectively if a separate Icon is created for this mode with the addition of the -CLIP characters.

To create a CLIP icon, simply copy the normal icon and add **-CLIP** as an operand. This can now be used to Open the clipboard contents quickly.

-BROWSE option

BROWSE mode is available to allow you to use SPFLite to browse data files without the worry of accidentally changing the data. This is particularly important if you normally run with PROFILE AUTOSAVE YES. In BROWSE mode, the SAVE and REPLACE commands are crippled as is the AutoSave function of the END command. CREATE however IS still allowed.

-BROWSE (or any abbrev. down to **-B**) may be coded alone as an operand, in which case an OPEN command would be used to select the file. Or it can be coded as **-BROWSE filename**, to initiate SPFLite with the specified filename pre-loaded.

To create a BROWSE icon, simply copy the normal icon and add **-BROWSE** as an operand. This can now be used to Drag/Drop files for BROWSE activity.

Note: There is a primary command BROWSE once SPFLite is started to invoke Browse on a file.

-INIT option

The **-INIT** option allows you to request SPFLite use a different INI file for customization. This allows you to have, for example, INI files with different screen height/width, different keyboard mapping, etc.

Simply add -INIT initname to the command line, where 'initname' is either

- the base name of the alternate INI file. For example, to start SPFLite using an INI file named MINWIN.INI the command line would be **-INIT MINWIN**. The INI file will be searched for in the normal SPFLite data path. (i.e. Documents\SPFLite)
- A fully qualified path to the INI file (e.g. D:\Data\Editors\SPFLite\MyINIFile.INI)

Example: The command line:

```
SPFLITE.EXE -B -I MYINIT MyTestFile.txt
```

Requests SPFLite start in Browse mode, using the MYINIT.INI file for customization, and to Open MyTestFile.txt as the initial working file.

Terminating SPFLite

Normally, SPFLite, like the normal SPF is terminated with a Primary END command (or =X). SPFLite will also shutdown in response to the Close button in the Title bar, along with other Windows shutdown functions (ShutDown, Logoff, etc.)

If File Manager is activated, there is also a [Preferences - General option](#) "Close File Manager with last tab" to control whether closing the last Edit tab should terminate SPFLite, or simply return to the File Manager tab.

When the Close (X) button is used, each active tab will be closed (including File Manager), with appropriate AUTOSAVE action as required.

Print Screen Functions

Print Screen Functions

[Previous](#) [Top](#) [Next](#)

SPFLite provides several variations of the Print Screen function:

Normal Print Screen Key

This will send a Text mode image of the current SPFLite screen to the Clipboard.

Control-Print Screen Key

This will send a Text mode image of only the File Text area of the current SPFLite screen to the Clipboard.

Shift-Print Screen Key

This will send a Text mode image of the current SPFLite screen to the Current Default Printer.

Alt-Print Screen Key

This will append to an SPFLite log file (creating it if not present) a Text mode image of the current SPFLite screen. The file will be created in the SPFLite Data directory (the same one where your INI, macro and colorize files are kept).

Key Mapping

As well as the normal Keyboard Print Screen key, SPFLite allows you to re-map the Print Screen functions away from the normal Print Screen key to your own choice. (e.g. Alt-P, Ctl-L, etc.) This is done by providing names for the functions under the 3270 Key Names in the [Preferences - Key Mapping](#) tab. The four key names are:

- | | |
|-------------|---------------------------------|
| PRT | - The normal Print Screen key. |
| PRTC | - The Control-Print Screen key. |
| PRTS | - The Shift-Print Screen key. |

NOTE: Normal Windows Screen Print

If you want to do a 'normal' Windows Screen Print, then before pressing the Screen Print key, you should move the Window focus **away** from the SPFLite window.

Scrolling the text

Scrolling the text

[Previous](#) [Top](#) [Next](#)

There are a variety of methods available for scrolling the Edit window to the area of text you wish to work on.

Primary Commands - Up, Down, Left and Right

These commands are typically assigned to function keys (PFKS) for easy availability. Their full syntax is available in the Primary command section ([UP](#), [DOWN](#), [LEFT](#), [RIGHT](#)). Although the amount to be scrolled can be provided as an operand to these commands, a global default is available which will be used whenever one of the commands is issued without any operand. The default is specified on the Preferences - General tab. (See [PROFILE](#)). For each individual Edit tab, this default can be overridden by using the Scroll field in the upper right corner of the Edit Tab. This override is valid only for the current Edit session (or until changed again).

Basically, using this method, you may set the scroll amount to:

PAGE	Scroll by the number of text lines on the screen (vertical) or the number of columns (horizontal)
HALF	Scroll by 1/2 the number of text lines on the screen (vertical) or 1/2 the number of columns (horizontal)
DATA	Scroll by the number of text lines on the screen minus one (vertical) or number of columns minus one (horizontal)
CSR	Cursor scrolling. Move the line / column containing the cursor to the edge of the screen, based on the direction.
nnnn	Scroll by a fixed number of lines / columns.

Arrow Keys - Up, Down, Left and Right

By default, when the cursor reached the edge of the screen, it will 'wrap' back to the opposite edge. Under Preferences - Keyboard (See [PROFILE](#)), you may activate keyboard scrolling where, instead of the wrapping described above, the screen will scroll one character / line at a time to keep the cursor in the active edit area.

Mouse Scroll Wheel

If you activate this option (Preferences - Mouse) then the mouse wheel can be used to scroll the text vertically, or horizontally if holding the Shift key at the same time. The number of characters / lines to scroll is specified in the Preferences - Mouse setting.

Primary Command - Locate

The primary command Locate (L) can be used to quickly move the top line of the screen to a specific line number. You may also jump to [Bookmarks](#) you may have established previously.

Primary Command - Find

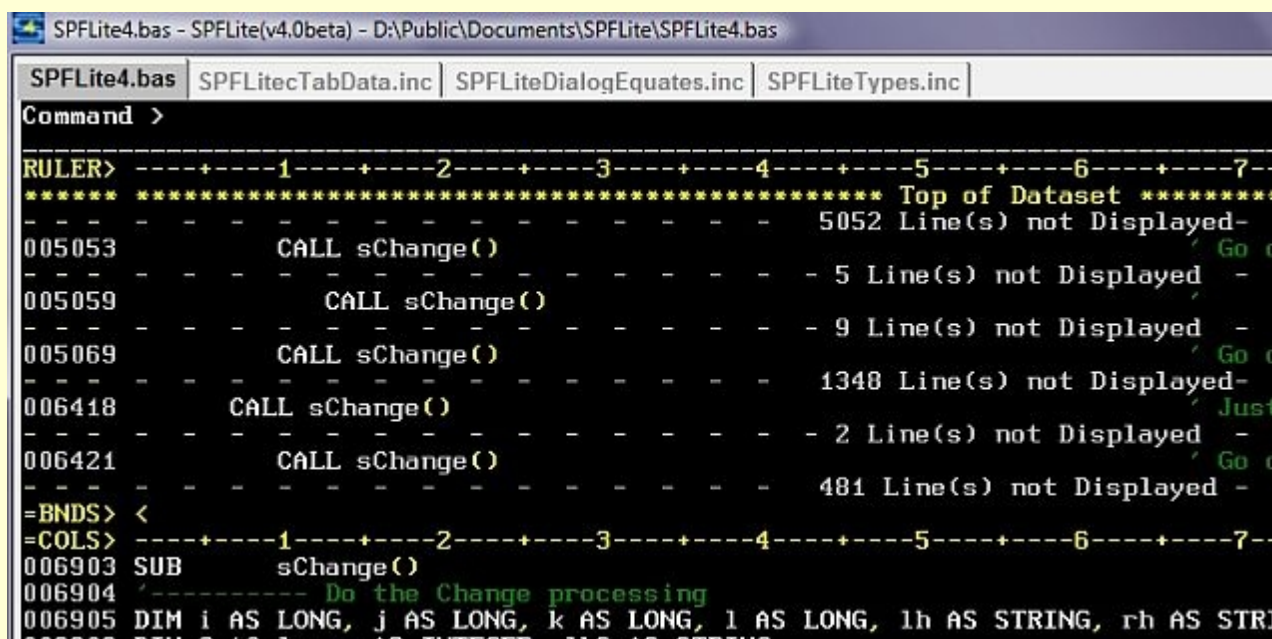
The [FIND](#) command may also be used to move the screen to a specified location based on the contents of the text data.

Working with Tab Pages

Working with Tab Pages

[Previous](#) [Top](#) [Next](#)

The SPFLite interface supports the editing / browsing of multiple files at the same time. Each open file is displayed and manipulated within a separate 'Tab Page' in the SPFLite window. The image below shows SPFLite with four Open Tabs.



```
SPFLite4.bas - SPFLite(v4.0beta) - D:\Public\Documents\SPFLite\SPFLite4.bas
SPFLite4.bas | SPFLiteTabData.inc | SPFLiteDialogEquates.inc | SPFLiteTypes.inc
Command >
RULER> -----1-----2-----3-----4-----5-----6-----7-----
***** Top of Dataset *****
005053          CALL sChange()
- - - - - 5052 Line(s) not Displayed -
- - - - - 5 Line(s) not Displayed -
005059          CALL sChange()
- - - - - 9 Line(s) not Displayed -
005069          CALL sChange()
- - - - - 1348 Line(s) not Displayed -
006418          CALL sChange()
- - - - - 2 Line(s) not Displayed -
006421          CALL sChange()
- - - - - 481 Line(s) not Displayed -
=BND> <
=COLS> -----1-----2-----3-----4-----5-----6-----7-----
006903 SUB      sChange()
006904 '----- Do the Change processing
006905 DIM i AS LONG, j AS LONG, k AS LONG, l AS LONG, lh AS STRING, rh AS STR
006906 DIM Cnt AS INTEGER, ll AS STRING
```

The filename of a file being edited (minus the full preceding path) is used as the Tab Title for each tab. When a Tab is selected (active) the tab itself is displayed with a different background color, and the full filename, including the path, is shown in the SPFLite windows title bar.

Data Isolation

The data being edited in each tab is completely separate from the data in any other tabs. However, some SPFLite program data is used in common with all tabs. This includes:

- All settings controlled by the [PROFILE](#) preferences.
- The Command retrieve stack used by the [RETRIEVE](#) command.

Switching Between Tabs

To switch between tabs you may either use the mouse to left-click on the desired tab, or you can use the SPFLite Primary command [SWAP](#) to switch to the Previous (Left) or Next (Right) tab. It would probably be normal to have this command assigned to a PFKEY for convenience. Note: Both [SWAP PREV](#) and [SWAP NEXT](#) will 'wrap' if there are no more tabs in the indicated direction.

Opening a New Tab

Without File Manager Active

When SPFLite is started, an initial Tab is opened to an (Empty) file (or the requested file if present on the command line). A subsequent [OPEN](#) command will, based on the current contents of the tab:

If the Tab is Empty

Open the requested file in the current empty Tab.

If the Tab currently contains data

Open the requested file in a new tab.

OPEN NEW

A command of [OPEN NEW](#) command will open a new tab as an (Empty) file without prompting you for a filename. Note: If you issue this command from a currently (Empty) tab, nothing will happen.

With File Manager Active

When SPFLite is started, the File Manager tab will be displayed with your default File list displayed. To open a file for Edit or Browse, simply enter the appropriate command character (**E** or **B**) next to the desired file(s). Each selected file will be opened in its own tab.

You may return to the File Manager tab at any time to select other files for processing.

All the functions described above under "Without File Manager Active" are also still available as in the past.

Closing a Tab

A tab will be closed when an [END](#) command is issued. If you are finished with the Edit of file, after saving it (or not) you may [CANCEL](#) the edit and leave the tab open as an (Empty) file. (Or close and return to File Manager if it is active)

If you click on the main window Close Button (X), SPFLite will close each tab, performing your normal [AUTOSAVE](#) activities as needed.

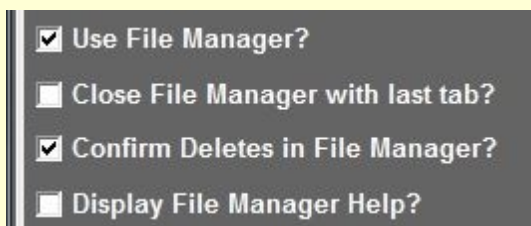
Working with the File Manager

Working with the File Manager

[Previous](#) [Top](#) [Next](#)

Description

The SPFLite File Manager is designed to provide a simple interface for file selection and basic Delete and Rename functions. It is not normally activated, you must request it via the Preferences - General settings page (See [PROFILE](#)).



The first option of course, simply turns File Manager ON or OFF.

The second option allows you to choose whether the File Manager should close when the last active Edit/Browse Tab closes (i.e. terminate SPFLite) or simply return to the File Manager Tab.

The third specifies whether or not you wish to receive a confirmation prompt when selecting a file for deletion.

File Manager initially displays a few 'reminder' lines at the bottom of the screen containing abbreviated Help. Once you have used File Manager for a while, you will probably want to turn off this Help and reclaim the screen real estate. Use the last option to do this.

Directory Monitoring

Once File Manager has displayed a directory, it will commence a monitoring of that directory for any changes, whether made by SPFLite itself or not. The next time a Tab switch back to File Manager occurs, or simply pressing Enter while still in File Manager, will cause a refreshed display if any changes have occurred.

Specifying The Criteria

The top portion of the File Manager screen contains input fields where you can specify what folder you wish to be display, what file types, and how the data should be sorted.

```
File Manager - SPFLite(v4.2)
File Manager
Command >
Default Path > D:\Public\Documents\SPFLite
Default File Types > bas;clr;inc;txt
SRT field/Direction > FILEUP
=====
C File / Path Name                                     Size LastModified
=====
NEW Empty Tab
<DIR> .. (Previous - UP)
<DIR> Colorize
<DIR> Distribution
<DIR> Fonts
<DIR> Misc Graphics
<DIR> Other Sample Sources
<DIR> SPFLite BartPE Plugin
<DIR> Test Data
$$ToDo.txt                                             175B 2009-12-15 11:36
ClrTest3.bas                                           15.9K 2009-12-03 12:50
InstallCheckList.txt                                  2.9K 2009-07-11 10:54
PBCodec.bas                                            69.9K 2009-09-27 11:54
PBCodec.inc                                           14.3K 2009-09-27 11:55
```

The Default Path should be completed with the folder name you wish to be displayed. The default File Types should contain a list of the file types you wish to show, all other file types will be skipped. Note a single asterisk '*' indicates **ALL** filetypes.

The SRT field/direction describes the Sort you wish performed. This may contain one of FILE, SIZE or DATE suffixed with UP or DOWN. e.g. FILEUP, FILEDOWN, SIZEUP, etc.

Although you may alter these at any point, or Select the Previous or any other folder, the last displayed contents when SPFLite is terminated will be 'remembered' and used as the opening display the next time SPFLite is started.

Working with the Displayed Data

Command Line

The command line will accept most SPFLite Primary commands which **make sense** in the File Manager environment. e.g. a CHANGE command will be rejected; an OPEN filename would be proper.

It will also accept a SELECT command (or **SEL** or **S**) to request a switch of Directory displayed. For an Operand you may enter:

- ..** to move UP to the parent directory
- folder** to switch to one of the lower displayed Directories listed below.

Line Select Command

Once the first detail list display is shown, you can use line command characters beside any of the displayed items to request further functions.

The line items flagged with <DIR> indicate other file folders. The only allowable action character for these is **S - Select** which will cause the File Manager list to be refreshed with the contents of the selected folder. If you select multiple folders at once, only the last (lowest on the screen) will be used.

For all other lines you may enter:

- E** Edit the file (N.B. the **S** command character will also invoke Edit)
- B** Browse the file

Basic Edit Functions

Basic Edit Functions

[Previous](#) [Top](#) [Next](#)

Basic Edit Functions

There are three basic interactions used to alter your edit file.

Direct Alteration of the Text

This is the most obvious, simply use the arrow keys to move within the data and overtype as desired. The Insert key will toggle Insert / Overtyping mode as needed. You may control a variety of visible attributes for the Insert Mode, and cursor style. See [Preferences - Screen](#) and [Preferences - Keyboard](#) for customization options. If colorization support is active, note that the display line is not re-colored to match the entered text until the next Attention key is pressed (Enter, PFnn, etc.)

If HEX mode is selected (see the [HEX](#) command), then data may also be edited in Hex mode which can be used to enter characters which are not normally available via the keyboard. Note that if you use Hex mode to insert/alter characters such as CR and LF, then any untoward results are **your** responsibility.

Primary Commands

Primary commands are those entered on the Command line at the top of the screen and are those commands which typically affect the entire file being edited. Multiple commands may be entered at one time on the command line, separated by a special character. The default for this character is the semi-colon, but may be altered via the CMDCHR option of the [Preferences - General](#).

You can use primary commands to:

- Scroll to a specific line number or [Bookmark](#)
- Find a specific line containing a search argument
- Find and change a character string
- Save the edited data or cancel without saving
- Create or Replace other files than the one being Edited
- Sort data
- Delete lines
- Undo changes made (subject to the number of maintained Undo levels, see [Preferences - General](#) for more details).

Visual Indication of A Files Modified Status

When a file has been modified since it was initially loaded, or since the last SAVE command, this will be

indicated on the Tab Title by prefixing the Filename with the characters (M).

Macros

If a command entered on the Command line is not recognized by SPFLite as one of the built-in commands, before displaying an error message to that effect, SPFLite will first check to see if the entered command is an SPFLite MACRO. A MACRO is simply a small file containing a series of SPFLite commands and stored in a file in the SPFLite data directory with a filetype of .SPM to identify it as an SPFLite Macro. Further details on using macros is provided in [Macro Support](#).

Line Commands

Line commands affect only a single line or block of lines. You enter line commands by typing them in the Line Command field on one or more lines and pressing Enter. The Line Command field is represented by a column of 6-digit numbers on the far left side of your display. When you are editing an empty line, however, the Line Command field contains quotes. This field is also used to display special lines, such as the ==CHG> flag, which indicates a line which has been altered by a Primary CHANGE command.

You can use line commands to:

- Insert or delete lines
- Repeat lines
- Rearrange lines or overlay portions of lines
- Join lines together
- Perform text paragraph entry and formatting
- Shift data
- Include or exclude lines from the display
- Control tabs and boundaries for editing

Edit Commands and PF Key Processing

When data (commands) are entered on the Command line, and a Function key is pressed rather than the ENTER key, SPF Lite concatenates the contents of the command line to the definition of the function key. The result is handled as a single command by SPFLite.

For example, when you use a Function key defined as a scroll command (UP, DOWN, LEFT, or RIGHT) the value entered on the command line becomes the **operand** of the scroll command. Entering **H** or **HALF** on the command line and pressing the Function key for scroll Down, results in a command of **DOWN HALF** being issued to scroll the screen down by half its height.

Bookmarks

Bookmarks

[Previous](#) [Top](#) [Next](#)

SPFLite allows you to place Bookmarks throughout the text lines to enable you to reposition to specific area of interest quickly. You may place as many bookmarks as you require.

A bookmark is created by entering it into the line number area preceded by a period (.) meaning you have a maximum of 5 alpha-numerics available. e.g. Labels could be **.here**, **.there**, **.main**, **.sub1** etc. The labels will be displayed in hi-intensity mode to make them easier to spot.

Once assigned, you can reposition the screen quickly to any desired bookmark by using the LOCATE primary command (or just L). e.g. **L .there**

As well, line labels can be used in place of specific absolute line numbers in those commands that support line number operands. The commands that use this form are CREATE, CUT, DELETE, LOCATE, PRINT, and REPLACE.

Two fixed named bookmarks are always available **.ZFIRST** to indicate the first data line, and **.ZLAST** to indicate the last data line.

Windows Clipboard

Windows Clipboard

[Previous](#) [Top](#) [Next](#)

The [CUT](#), [PASTE](#) and [BPASTE](#) (Block Paste) commands utilize the standard Windows Clipboard (in Text mode). This enables you to cut/paste between SPFLite and other Windows text mode applications. The CUT command can also **ADD** data to the existing data on the clipboard.

Note the [PASTE](#) command works in line mode only. i.e. it functions on full lines only.

The CUT command can function in Line Mode and in Block Mode. The default is always Line Mode unless a block of text has already been 'framed' when the CUT command is issued.

Also, when in Line Mode, the CUT command is sensitive to whether the block of lines to be CUT were selected with CC/CC or MM/MM line commands. If CC/CC is used, the lines are copied to the Clipboard and remain in the working file. If MM/MM is used, the lines are moved to the Clipboard and deleted from the working file.

Read the [CUT](#) command for a full description.

Clipboard Mode

SPFLite supports a mode known as **CLIP** mode. This mode is specified by coding the characters '-CLIP' as the command operand invoking SPFLite. In this mode, SPFLite will automatically load the contents of the Windows clipboard when started, and will also save the current edit data to the Windows clipboard on termination. See also under [Starting SPFLite](#).

Edit Boundaries

Edit Boundaries

[Previous](#) [Top](#) [Next](#)

Boundary settings control what data is affected by other line and primary commands. You can change the boundary settings by using either the [BNDS](#) line command, or [BOUNDS](#) primary command.

The Bounds column range is in effect unless you specify overriding boundaries when entering a command. Refer to the individual command descriptions for the effect the current bounds settings have.

If you do not explicitly set bounds, the editor uses the default bounds, which are the entire data line.

See the [BNDS](#) line command and the [BOUNDS](#) primary command for further information.

Creating and Replacing Data files

Creating and Replacing Data files

[Previous](#) [Top](#) [Next](#)

Use the [CREATE](#), [REPLACE](#) or [SAVEAS](#) primary commands to specify a file to be written from the data being edited. [CREATE](#) writes a new file while [REPLACE](#) rewrites a file if it exists, or creates it if it does not already exist. The [SAVEAS](#) command will create a new file and immediately switch the current Edit Tab to the newly created file. The process of creating and replacing data is very similar. However, remember that when you replace data, the original data is deleted and replaced with the new data.

There are two ways you can use [CREATE](#) or [REPLACE](#):

1. You can type either [CREATE](#) or [REPLACE](#) on the Command line, followed by the name of the file to be created or replaced. You can add line numbers that show the lines to be copied. If you omit the line numbers, you can use the [C](#) (copy) or [M](#) (move) line commands to specify which lines are to be copied or moved. Then press Enter. See the [CREATE](#) and [REPLACE](#) primary commands for further details. If no line numbers are specified on the command line, and no line commands are used to select specific lines, then the Entire edit file will be used.
2. If you omit the file name with the [CREATE](#) or [REPLACE](#) commands, the editor displays a standard Windows Save As dialog requesting the file name you want created or replaced.

When no operand at all or only a simple unqualified filename is provided for the [CREATE](#) or [REPLACE](#) commands, the default directory used for writing the file or for the file open dialog's starting directory will be determined as follows:

- If there is an active file being edited in the tab where the command is issued, then the Path for THAT Active file is used as the default for the command.
- If there is NO active file (i.e. the tab header displays (Empty):
 - If File Manager is active, the current displayed directory of File Manager will be used.

If File Manager is NOT active, then the Preferences setting for Default Open directory will be used (Last Used or Working). Remember 'Working' means the Windows working directory when SPFLite was started.

SAVEAS

SAVEAS always saves the entire current edit file, you may not specify line number ranges with the command nor use the [C/CC](#) or [MMM](#) line commands to select a line number range.

Once the new file is created, the current edit tab is switched to the newly created file.

Including another File

Including another File

[Previous](#) [Top](#) [Next](#)

While you are editing, you can copy another file into the current data by using the [COPY](#) primary command. The location at which it is to be inserted is specified with the [A](#) or [B](#) line commands.

If the [COPY](#) command is issued when there are **no** lines in the edit work area, then no [A](#) or [B](#) line command is needed.

Refer to the [COPY](#) primary command for further details.

Shifting Data

Shifting Data

[Previous](#) [Top](#) [Next](#)

When you edit data, the editor automatically shifts characters on a line to the left or right to accommodate insertions or deletions. This shifting can be either implicit or explicit. Implicit shifts occur when typing in Insert mode, deleting characters with the Del key, or when the [CHANGE](#) command string-2 length is different from the string-1 length.

Explicit shifts occur when you use the following commands:

Line commands

(Column Shift Left
)	Column Shift Right
<	Data Shift Left
>	Data Shift Right

Two columns is the default for shift operations. When shifting a block of lines more or less than the default, enter the amount on the first or last line of the block. If you enter it in both places, the line shifts only if both amounts are the same.

Differences between Column Shift and Data shift.

There are some significant differences between Column shifts and Data Shifts.

Shifting occurs within the column boundaries (either the default - the whole line) or those specified with the [BOUNDS](#) command or the [BNDS](#) line command.

Column shift moves all characters within the bounds without altering their relative spacing. Characters shifted past the bounds are deleted. That is, blanks are inserted at the bound *from* which the characters are being shifted, and the characters are deleted at the *opposite* bound. So, this shift is called a destructive shift because information shifts within column boundaries without regard to its contents, and can result in the loss of data with no error being noted.

Data Shift moves characters within the bounds and will **never** shift characters such that characters are deleted. If there are insufficient spaces at the end of the line toward which the shift is moving, then the data will only shift till the character data **reaches** the bounds location.

In **addition**, Data shift treats any data touching either bounds at the start of the shift as being unmoveable. It will look for the end (left bound) or start (right bound) of the touching 'word' and will freeze them and shift only the other space delimited words between those two boundary words.

Example:

```
BNDS=>      <                                     >
000010      Word1      Word2      Word3                                     Wordn
```

If a >10 command is entered on line 00010 the result would be

```
BNDS=>      <                                     >
000010      Word1                                     Word2      Word3      Wordn
```

As can be seen `Word1` and `Wordn` have been left alone and only the other data shifted.

Finding, Changing and Excluding Data

Finding, Changing and Excluding Data

[Previous](#) [Top](#) [Next](#)

[FIND](#), [CHANGE](#), and [EXCLUDE](#) allow you to find a specified search string, change one search string to another, or exclude a line containing a specified search string. These commands provide powerful editing functions because they operate on a complete file rather than on a single line.

The characteristics of each command follow:

- [FIND](#)** Causes all lines that it finds to be displayed, and moves the cursor (scrolling if necessary) to the first occurrence of the search string.
- [CHANGE](#)** Causes the same effect as [FIND](#), but it also has a second string operand (string-2). During a search, whenever string-1 is found, the editor replaces that string with string-2. Data to the right is shifted, if necessary.
- [EXCLUDE](#)** Causes lines that match the search to **not** to be displayed. These lines remain in the data, however. Unlike [FIND](#) and [CHANGE](#), it does not require a search string if you use the ALL operand. [EXCLUDE ALL](#) is often used with a following [FIND](#) and [CHANGE](#) because they cause excluded lines to be redisplayed. Use [RESET](#) to cause all lines to be redisplayed. Also see the [FLIP](#) command which will reverse the excluded and non-excluded data.

Specifying the Search String

The primary control for any search is the search string because it represents the value for which you are looking. Two operands, string-1 and string-2, are required for the [CHANGE](#) command to specify the new value of the string once it is found. The rules for specifying string-1 and string-2 are the same, except that if you type a single asterisk for either one, the previous value from the previous FIND or CHANGE command is used again.

The editor allows you to specify the following kinds of strings:

Simple string

Any series of characters not starting or ending with a quote (' or ") and not containing any embedded blanks.

Delimited string

Any string enclosed (delimited) by either single quotes (') or double quotes ("). The beginning and ending delimiters must be the same character.

Picture string

Any quoted string of characters, preceded by the character P, such as P ' >>>### '. The picture string provides a powerful pattern matching ability to enable searching for data of certain types, rather than by specific actual characters. The creation of Picture strings is fully covered in [Specifying A Picture String](#).

Hexadecimal string

Any quoted string of Hex characters (0123456789ABCDEF), preceded by the character X, such as X'41CF'. The string must be an even number of valid hex characters.

For example

```
FIND X'3132' would be equivalent to FIND '12'
```

Character string

Any quoted string of characters, preceded by the character C, such as C'conditions for'.

If the string is a simple or quoted string, the characters are treated as being both upper and lowercase even if caps mode is off. For example, this command:

```
find ALL 'CONDITION NO. 1'
```

successfully finds the following:

```
CONDITION NO. 1
Condition No. 1
condition no. 1
coNDitION nO. 1
```

Also, all of the following commands have the same effect:

```
FIND 'Edit Commands'
FIND 'EDIT COMMANDS'
FIND 'edit commands'
```

You must use quotes if a string contains imbedded blanks or commas, or if a string is the same as a command or keyword. You delimit strings with quotes, either ' or ". For example, to change the next occurrence of 'every one' to 'all', type:

```
Command = CHANGE 'every one' 'all'
```

Use a character string in a [FIND](#), [CHANGE](#), or [EXCLUDE](#) command if you want the search to be satisfied by an exact character-by-character match. Lowercase alphabetic characters match only with lowercase alphabetic characters, and uppercase alphabetic characters match only with uppercase.

For example, `FIND C 'XYZ'` finds the characters XYZ only, not xyz.

Effect of CHANGE Command on Column-Dependent Data

Column-dependent data is groups of non-blank source data separated by two or more blanks, such as a table. When you use [CHANGE](#) to change column-dependent data, the Editor attempts to maintain positional relationships. For instance, if you change a long word to a short word, the editor pads the short word with blanks. This padding maintains the column position of any data to the right of the change by preventing it from shifting left.

When only one blank separates words, as in most text data, padding does not occur. Changing a long word to a short word causes data to the right of the change to shift left.

Starting Point and Direction of the Search

To control the starting point and direction of the search, use one of the following operands:

NEXT	Starts at the first position after the current cursor location and searches ahead to find the next occurrence of string-1. NEXT is the default and thus is never usually coded.
ALL	Starts at the top of the data and searches ahead to find all occurrences of

FIRST

string-1. When complete, a message is issued stating the number of occurrences found. If you use this operand with CHANGE, the lines changed are marked with ==CHG> flag. The status of these lines can be changed to normal by RESET. Starts at the top of the data and searches ahead to find the first occurrence of string-1.

LAST

Starts at the bottom of the data and searches backward to find the last occurrence of string-1.

PREV

Starts at the current cursor location and searches backward to find the previous occurrence of string-1.

If you specify ALL, or FIRST, the direction of the search is forward. When you press the assigned function keys, the [RFIND](#) or [RCHANGE](#) commands find or change the next occurrence of the designated string. If you specify LAST or PREV, the direction of the search is backward. When you specify those operands, the editor finds or changes the previous occurrence of the string.

The search proceeds until the editor finds one or all occurrences of string-1, or the end of data.

If you omit the ALL operand on the CHANGE command, the editor searches only for the first occurrence of string-1 after the current cursor location. If the cursor is not in the data area of the panel, the search starts at the beginning of the first line currently displayed. Scrolling is performed, if necessary, to bring the string into view.

Depending on the direction of the search, if the string is not found between the current cursor location and the end or beginning of data, a message is displayed

Qualifying the Search String

You can specify additional characteristics of string-1 by using the operands PREFIX, SUFFIX, and WORD.

PREFIX Locates string-1 at the beginning of a word.

SUFFIX Locates string-1 at the end of a word.

WORD String-1 is delimited on both sides by blanks or other non-alphanumeric characters.

In the following example, the editor would find the noted strings only:

```
FIND 'DO'           - DO DONT ADO ADOPT 'DO' (DONT)      Finds all of them
```

```
FIND PREFIX 'DO'    - DO DONT ADO ADOPT 'DO' (DONT)      Finds DO DONT 'DO  
' (DONT)
```

FIND SUFFIX 'DO' - DO DONT ADO ADOPT 'DO' (DONT) Finds ADO

FIND WORD 'DO' - DO DONT ADO ADOPT 'DO' (DONT) Finds DO 'DO'

Delimiters Used to Determining Word / Prefix / Suffix Boundaries

In order to determine what a 'word', 'prefix' or 'suffix' actually is, SPFLite uses a set of delimiter characters which **normally** are quite sufficient to determine this. The default characters used are:

" +-*/\(){}[]:; '<>, .?&%\$#@!~`_ =

However, some computer languages allow other characters to be used in variable names (e.g. the _ underscore). This can cause **word** searches to function incorrectly. SPFLite allows you to modify this list of delimiter characters. The modified list you create will be associated with the file type being edited, and will be saved and used in future edit sessions of this file type automatically.

For information on altering these delimiters see the [DLMS Line command](#).

Column Limitations

The col-1 and col-2 operands allow you to search only a portion of each line, rather than the entire line. These operands, which are numbers separated by at least one blank, show the starting and ending columns for the search. The following rules apply:

- If you specify neither col-1 nor col-2, the search continues across all columns within the current boundary columns.
- If you specify col-1, the editor finds the string only if the string starts in the specified column.
- If you specify both col-1 and col-2, the editor finds the string only if it is entirely within the specified columns.

Excluded Line Limitations

You can limit the lines to be searched by first using the X or NX operands:

X Scan only lines that are excluded from the display.

NX Scan only lines that are not excluded from the display.

Repeating the FIND, CHANGE, and EXCLUDE Commands

The easiest way to repeat [FIND](#), and [CHANGE](#), without retyping them is to assign those commands to function keys. The defaults are:

F5/17	RFIND
F6/18	RCHANGE

The search begins at the cursor. If the cursor has not moved since the last [FIND](#), or [CHANGE](#) command, the search continues from the string that was just found. Instead of retyping string-1, you can type an asterisk to specify that you want to use the last search string. If you decide to type [RCHANGE](#) or [RFIND](#) on the Command line instead of using a function key, position the cursor at the desired starting location before pressing Enter.

All three commands share the same string-1. Therefore:

```
Command > FIND ABC
```

followed by:

```
Command > CHANGE * XYZ
```

first shows you where ABC is, and then replaces it with XYZ. However, you can do this more easily by typing:

```
Command > CHANGE ABC XYZ
```

Then press F5/17 to repeat FIND. The editor finds the next occurrence of ABC. You can either press F5/17 to find the next ABC, or F6/18 to change it. Continue to press F5/17 to find remaining occurrences of the string.

The previous value of a search string, specified by an asterisk or by use of [RFIND](#) or [RCHANGE](#), is retained until you end your editing session.

Excluding Lines

You can exclude lines from a data set using the X (exclude) line command as well as the [EXCLUDE](#) primary command.

When you are editing a program that exceeds the screen size, it is often difficult to determine whether the control structure and indentation levels are correct. Excluding lines allows you to remove one line or a block of lines from the display so that you can see the general control structure. The lines are excluded from the display, but are not deleted from the data. Excluded lines are treated as valid data lines.

The [X](#) line command can have the syntax:

X[n]

or

XX

The first form allows you to exclude one line (X) or any number of lines (Xn).

The second form allows you to exclude a block by typing [XX](#) on the first and last lines of the block of lines that you want to exclude. The first and last lines do not need to be on the same page; after typing the first XX you can scroll to the second XX.

You can enter any line command that usually operates on a single line in the line command area of the excluded lines message. For example, if you enter the D (delete) line command, the complete block of excluded lines is deleted.

Redisplaying Excluded Lines

To display all excluded lines, enter the [RESET](#) primary command. Alternatively, you can display one or more excluded lines again by entering the [F](#) (first), [L](#) (last), or [S](#) (Show) line commands, typing over the dashes in the line command area.

You can add a number following any of these line commands to cause more than one line to appear again:

F[n]

L[n]

S[n]

[FIND](#) and [CHANGE](#) also cause any excluded lines that meet the search criteria to appear again.

Hiding Excluded Lines

You may optionally choose to completely **Hide** all excluded lines. Enter the [HIDE](#) Primary command to accomplish this. All Excluded lines will be completely removed from the display. To indicate their presence, the Line Number area of the line immediately preceding the excluded lines will be Underlined to mark their presence.

To re-display these lines in normal Exclude mode, simply enter the [RESET](#) HIDE command.

Specifying A Picture String

Specifying A Picture String

[Previous](#) [Top](#) [Next](#)

A picture string in a FIND, CHANGE, or EXCLUDE command allows you to search for a particular kind of character without regard for the specific character involved.

You can use special characters within the picture string to represent the kind of character to be found, as follows:

String Meaning

P'='	Any character
P'^'	Any character that is not a blank
P'##	Any numeric character, 0-9
P'.'	Any non-numeric character
P'.'	Any non displayable character
P'@'	Any alphabetic character, uppercase or lowercase
P'<'	Any lowercase alphabetic character
P'>'	Any uppercase alphabetic character
P'\$'	Any special character, neither alphabetic nor numeric.

A picture string can include alphanumeric characters, which represent themselves, mixed with other characters.

Picture String Examples

- To find a string of 3 numeric characters:
FIND P'###'
- To find any 2 characters that are not blanks but are separated by a blank:
FIND P'^ ^'
- To find a blank followed by a numeric character:
FIND P' #'
- To find a numeric character followed by AB:
FIND P'##AB'
- To find the next character in column 72 that is not a blank:
FIND P'^' 72
- To change any characters in columns 73 through 80 to blanks:
CHANGE ALL P'=' ' ' 73 80
- To find the next line with a blank in column 1 and a character in column 2 that is not a blank:
FIND P'↵' 1

Using Picture Strings as the TO operand of a CHANGE command

Note that when Picture strings are used as the TO operand of a CHANGE, that the only allowable special string characters are '=', '<', and '>'.

Displayable Characters

The picture character "." (period) requests a search for **non**-displayable characters. Depending on the font you are using, particularly when national characters are involved, deciding what is non-displayable can be confusing. SPFLite provides you with the ability to specifically indicate **what** characters **you** wish to consider displayable. The Preferences - General settings page allows you to alter what these characters are. See [Preferences - General](#) for how this is changed.

Restricted Picture Characters

Due to the internal processing of Picture strings, several characters may NOT be coded within the Picture string.

These characters are:

. | ? + * [] \ ()

If any of these characters are present, the command will be rejected.

Word Processing Support

Word Processing Support

[Previous](#) [Top](#) [Next](#)

Word Processing

This section is a general overview of the line commands used for word or text processing: [TF](#) (text flow), [TS](#) (text split), [SC](#) (Sentence Case) and [TC](#) (Title Case)

[SC](#) and [TC](#) can be used to quickly reformat the case of a block of text. SC reformats in **sentence** format. i.e. First letter of the first word of a sentence is capitalized, all other characters are lowercased. TC reformats as a Title. i.e. The first letter of **each** word is capitalized.

[TF](#), [TS](#), assume that the data is grouped in paragraphs. A paragraph is a group of lines that begin in the same column. The first line of a paragraph is excluded from the grouping. The editor interprets any indentation or blank line as representing a new paragraph. It also recognizes common word processor control words that are used by various common scripting languages. These control words begin with a period, a colon, a '<', or an ampersand.

If you use text line commands frequently, you can assign both the TS and TF commands to function keys. Use [PFKEY](#) to open the [Preferences - PFKEYS](#) dialog to reassign the keys. For example:

F10 ==> :TS

F11 ==> :TF

Now you can split text by moving the cursor to the desired split point within a line and pressing F10. Having typed the new material, press F11 to restructure the text from the line containing the cursor to the end of the paragraph.

Formatting Paragraphs

The [TF](#) (text flow) line command formats paragraphs. It assumes that the sentences are roughly in paragraph form with a ragged right margin when it attempts to recognize groupings. TF can be followed by a number (TF72 for example) that specifies the desired right side column for the paragraph. If you do not specify a number, the right side of the panel is used.

The editor assumes that because the first line of a paragraph may be at a different indentation level than the remainder of the paragraph, the starting column of the second line is the left side of the paragraph.

When formatting paragraphs, the editor:

- Moves text so that each line contains the maximum number of words. TF limits its activity to within the limit documented above. Thus, it can be used to flow text within a border.
- Keeps any blanks between words.
- Assumes one blank between the word at the end of a line and the word on the next line except when the line ends with a period. In that case, the editor inserts two blanks.

The end of the paragraph is denoted by a blank line, a change in indentation, or the special characters period (.), colon (:), ampersand (&), or left carat (<) in the left boundary column. These special characters are used as control indicators by a variety of scripting languages.

The restructure operation removes trailing blanks on a line by using words from the following line. It does not, however, remove embedded blanks within a line. Accordingly, if one or more words in a line are to be removed, delete the words rather than type over them.

Splitting Lines

The [TS](#) (text split) line command splits a line into two lines. The cursor shows where the line is to be split. The editor moves the characters to the right of the cursor or to a new line following the original line and aligns the new line with the left side of the paragraph. As mentioned earlier, the left side of a paragraph is determined by looking for a pattern in the lines preceding or succeeding a paragraph.

One or more blank lines are inserted after the line being split, depending on what you specify when you enter the TS command. If not used, these lines will be removed (as are all Inserted lines).

Defining & Controlling Tabs

Defining Tabs

To define Tabs, first obtain a visible copy of the TABS definition line as follows:

Type [TABS](#) on any Line command area, a Tabs line will be inserted

```

000010 Data text here
=TABS>      *      *
000011 More Data text here

```

To ensure accurate positioning, you may wish to also insert a [COLS](#) line for reference

```

000010 Data text here
=COLS> ----+-----10----+-----20----+-----30----+
=TABS>      *      *
000011 More Data text here

```

On the =TABS> line enter an asterisk (*) at each position you wish to become a tab stop and press Enter.

Note that SPFLite will remember the tabs line and associate it with the type of file being edited. The next time this file type is edited, the Tabs line will be established when the file is loaded into the editor. This will enable you to have different 'favourite' tabs definitions for each type of file you edit. Note that the setting of the Primary CAPS setting is saved as well for each file type.

The editor can also be told to temporarily ignore the tabs line without altering its definition by issuing the primary command [TABS OFF](#), normal tab operation can be resumed with a [TABS ON](#) command.

PF Keys and substitution strings

PF Keys and substitution strings

[Previous](#) [Top](#) [Next](#)

SPFLite has only a simple form of macro support (see [Macro Support](#)). It does not support any of the 'normal' scripting engines (VBScript, Perl, REXX, etc.). However a variable substitution function has been 'built in' for the Macros and for PF Key definitions.

There are several 'built-in' variables that can be used in the PF Key definition string. These variables will be substituted when processing the PF Key.

Variables are identified as a tilde '~' followed by a single character. Currently, the following variables are available:

~W

Each time the Enter key, or one of the PF Keys are pressed, SPF Lite will attempt to extract, if possible, the 'word' on which the cursor is sitting. This 'word' is made available as the substitution string (~w). This enables a defined PFKEY to incorporate the 'word' into the stored command.

e.g.

```
X ALL; F ALL '~w'
```

This would be used as follows:

- Place the cursor on a word or string of interest
- Press PF8
- The display will now show only lines containing the word of interest, all other lines will be in Excluded status.

Similarly,

```
FIND FIRST '~w'
```

Which, following the same instructions, would scroll the screen to the first occurrence of the 'word' under the cursor.

~l

Similarly to ~w, SPFLite will assign the contents of the current line on which the cursor is sitting to the ~l (line) variable.

~p

This variable will be replaced by the current path to the file being edited. e.g. If editing C:\MyData\Letters\May01.txt, the ~p variable would become C:\MyData\Letters.

~f

This variable will be replaced by the current filename being edited. e.g. If editing C:\MyData\Letters\May01.txt, the ~f variable would become May01.

~x

This variable will be replaced by the current file extension of the file being edited. e.g. If editing

C:\MyData\Letters\May01.txt, the ~x variable would become txt.

Handling Non-Windows Txt files

Handling Non-Windows Txt files

[Previous](#) [Top](#) [Next](#)

Foreign Files

Text files from other systems (UNIX, Mac, etc.) may not be formatted in the same manner as normal Windows text files. Although all use normal ANSI character codes, they differ as to how text lines are separated in the file.

Windows stores text lines with **both** a CR and a LF character between lines. (CR/LF = Carriage Return / Line feed)

UNIX stores text lines with **only** a LF between lines.

Mac stores text lines with **only** a CR between lines.

SPFLite will read all three file types properly as input, and provides options to allow you to write data back in any of the three formats regardless of the form they were read in.

By default, files are always saved, or new files created, in the format specified by the [Preferences - General](#) setting **or** of the most recent primary [CRLF](#) command. Any files inserted into the edit file via the [COPY](#) command will have no effect on the default currently being used, but a warning message will be issued if the incoming format is **different** from the currently active format.

Specifically Requesting a Specific File Format

If you wish to use the [SAVE](#), [SAVEAS](#), [CREATE](#), or [REPLACE](#) commands to write a file in other than the default, use the primary [CRLF](#) command to temporarily change the mode before issuing the file command. The [CRLF](#) command supports the following:

WIN

Requests the file be written in standard Windows format.

UNIX

Requests the file be written in UNIX format.

MAC

Requests the file be written in Mac format.

Examples:

CRLF UNIX

SAVE Would save the file back in UNIX format.

CRLF MAC

CREATE NEWMACFILE.TXT Would create a new file (NEWMACFILE.TXT) in MAC format

CRLF UNIX

REPLACE TESTUNIXDATA.DAT Would replace the file (TESTUNIXDATA.DAT) in UNIX format.

Handling UniCode files.

Handling UniCode files.

[Previous](#) [Top](#) [Next](#)

Text files using UniCode are becoming more common as many of the files used to build Web sites are created in Unicode format.

SPFLite provides only basic support. That is, the editor internally is still working in ANSI only, using your default Code Page. The Unicode support consists of being able to read Unicode files, re-encoding them back to ANSI for the actual editing, and then re-encoding the data back to UniCode when writing the file out. Although this does not properly support full UniCode editing, there are still a large number of files which are still basically ANSI data, but which require UniCode encoding to be properly handled by Web applications.

Existing Files

SPFLite provide support for reading and writing Unicode files automatically. i.e. it will recognize Unicode files when they are opened, and properly translate the contents when the data is loaded into the Edit session. When the data is written (with [SAVE](#), [SAVEAS](#), [CREATE](#) or [REPLACE](#)) the data will be encoded in the same manner as it was when read.

Determining What Encoding A UniCode File Uses

Once a UniCode file is loaded, you can determine what type of encoding was used by either of the following:

- If you run normally with the Status Bar enabled at the bottom of the screen, the type of encoding used will be displayed in the Status Bar.
- If you do not use the Status Bar, then issue the Primary Command ENCODING without any operands, and a message will be displayed showing the encoding used.

Creating A New UniCode File

A new Unicode file can be created in the same manner as any other text file, with **one** exception. You must specify which 'flavor' of Unicode encoding you desire. This is done with the Primary Command [ENCODING](#). Issue this command with your desired encoding **before** writing the file.

Types of UniCode files

SPFLite supports the following three types of UniCode files:

UTF8

This format of encoding is the most 'economic' as the base ANSI characters are stored as 8 bit characters, and only special characters (above ANSI 127 are encoded as 16 bit values.

UTF16 / UTF16LE

This format of encoding uses 16 bits for each character. There are two types of UTF16, LE (Little Endian) and BE (Big Endian) which refer to the byte order of the encoded 16 bit value. The normal value for Windows based systems is UTF16/UTF16LE.

UTF16BE

This format of encoding uses 16 bits for each character. The encoded 16 bit value is stored in Big-Endian format.

Macro Support

Macro Support

[Previous](#) [Top](#) [Next](#)

SPFLite supports a limited Macro ability to allow simple entering of what would be a series of repetitive primary commands. An SPFLite macro is simply a normal text file containing a series of primary commands. The file should be created in the normal SPFLite data directory (which would normally be C:\Users\YourID\My Documents\SPFLite). It should be saved with a Filename = Macroname and with the file extension of .SPM to indicate it is an SPF Macro file.

Invoking a Macro

To invoke a macro, simply enter its name on the SPFLite command line just as you would with a normal built-in SPFLite command.

SPFLite examines all commands entered, and before rejecting a command as invalid, will check to see if an equivalent xxxx.SPM file exists, if so, it will invoke the contents of the file as a macro. Thus, with a Macro, you can create your own 'custom' commands.

Substitution Variables

The macro support allows the passing of variable data from the command line invoking the macro into the

actual commands issued by the macro. Just as internal SPFLite commands have operands, your macros can have operands. These operands operate as simple substitution variables based on their position in the command line. The 1st operand is referred to as ~1, the 2nd as ~2, and so on ~3 ~4 ~n. As well as these, any of the other substitution strings described in [PF Keys and substitution strings](#) may be used. e.g. ~p, ~f, etc.

Contents of a Macro File

A macro file can contain 3 types of command lines:

- **Comments**

Any line starting with a ';' (semi-colon) is treated as a comment and ignored.

- **NOFLUSH**

A Macro directive which prevents any following commands in the macro file from terminating the macro if an error occurs. This would typically be used when a macro contains a series of change commands, and the possibility exists that 1 or more of the Find operands may not exist. Although it applies to **ANY** following commands. See the **WKDAY** macro example below.

- **Commands**

All other commands are treated as primary SPFLite commands and should be coded as normal.

- **Prototype**

A prototype line (which **must** be the 1st line in the file) provides a sample of the Macro and its operands. The operands specified will become the defaults for each positional operand if these operands are not supplied when the macro is invoked. The Prototype line is a comment line (starting with a ; followed immediately by the macro name. Default operands follow on the remainder of the line.

Examples

- **The ONLY Macro**

```
;ONLY OperandMissing
;Display only the lines containing the provided operand
RESET
EXCLUDE ALL
FIND ALL ~1
```

If this macro were invoked on the command line with:

```
ONLY MyData
```

The following commands would be performed:

```
RESET
```

```
EXCLUDE ALL
FIND ALL MyData
```

If the macro were entered without any operands, the resulting commands would be:

```
RESET
EXCLUDE ALL
FIND ALL OperandMissing
```

And would undoubtedly result in the error message "String OperandMissing not found"

- **The EXCEPT Macro**

```
;EXCEPT OperandMissing
;Display only lines that DO NOT contain the provided oper
and
RESET
EXCLUDE ALL
FIND ALL ~1
FLIP
```

If this macro were invoked on the command line with:

```
EXCEPT MyData
```

The following commands would be performed:

```
RESET
EXCLUDE ALL
FIND ALL MyData
FLIP
```

If the macro were entered without any operands, the resulting commands would be:

```
RESET
EXCLUDE ALL
FIND ALL OperandMissing
FLIP
```

And would undoubtedly result in the error message "String OperandMissing not found"

- **The FN Macro**

```
;FN OperandMissing
;Position display to the start of a source function
;Assumes functions are coded as FUNCTION funcname() in po
sition 1
RESET
FIND FIRST "Function ~1" 1
```

If this macro were invoked on the command line with:

```
FN GetFile
```

The following commands would be performed:

```
RESET
FIND FIRST "Function Getfile" 1
```

If the macro were entered without any operands, the resulting commands would be:

```
RESET
FIND FIRST "Function OperandMissing" 1
```

And would result in "String Function OperandMissing not found"

- **The WKDAY Macro**

```
; WKDAY Convert French Weekdays to English, if present
;
NOFLUSH
change all word lundi      Monday
change all word mardi      Tuesday
change all word mercredi   Wednesday
change all word jeudi      Thursday
change all word vendredi   Friday
change all word samedi     Saturday
change all word dimanche   Sunday
reset
```

If this macro were invoked on the command line with:

```
WKDAY
```

The 7 change commands would be issued. Even if some of the French weekday names are not present, the macro will not terminate at the failing command, but would continue and issue **all 7** commands.



Command Reference

Command Reference

[Previous](#) [Top](#) [Next](#)

The Command Reference section contains the detailed Syntax for all commands. The commands are divided into two main categories, the Line Commands, which are those entered by over typing the line number area; and Primary Commands, which are those entered on the Command line at the top of the screen.

Line Commands

Line Commands

[Previous](#) [Top](#) [Next](#)

Descriptions of the following Line Commands are contained in this section:

- [A](#)

- [UC or UCUC](#)

Rules for Entering Line Commands

Rules for Entering Line Commands

[Previous](#) [Top](#) [Next](#)

You may enter a line command by one of the following:

- Type the command in the line command area and press Enter.
- Place the cursor in the data or line command field and press a function key to which the line command is assigned.

NOTE:

So that the data you type does not accidentally intermix with the previous contents of the line number area, SPFLite will, when the 1st character is typed:

- blank the entire line number area
- move the cursor to the left hand margin (Col 1)
- enter your typed character

The following rules apply to all line commands:

- You can type several line commands and make multiple data changes before you press Enter. The editor displays an error message if the line command is ambiguous. Because the line commands are processed from top to bottom, it is possible to have one error message appear that masks a later error condition. Only the first error condition found is displayed. After you have corrected that error condition, processing can continue and the next error condition, if any, is displayed. If you type a line command incorrectly, you can replace it before you press Enter by retyping it, blanking it out, or entering [RESET](#) on the Command Line.
- The TOP OF DATA line will only accept the following line commands:

I [n] Insert one or n temporary insert lines after this line.

N[n] Insert one or n permanent lines after this line.

A [n] Move or copy a line or lines one or n times after this line.

- The BOTTOM OF DATA line will only accept the following command:

B[n] Move or copy a line or lines one or n times before the last data line.

A - After Destination

A - After Destination

[Previous](#) [Top](#) [Next](#)

Syntax

A[n]

Operands

n

A number that tells the editor to repeat the associated line command a specified number of times. If the number you type is 1, the editor performs the command only once. The number applies to the associated primary commands.

Description

To specify that data is to be moved, copied, or inserted after a specific line. The A - After command is used in conjunction with one of the line commands ([C](#) - Copy or [M](#) - Move) or one of the Primary Commands [COPY](#) or [PASTE](#) to specify a destination for the command action.

1. Type A in the line command area of the line **AFTER** which the moved, copied, or inserted data is to be placed.
2. If you are specifying the destination for a line command, a number after the A line command specifies the number of times the other line command is performed. However, a number after the A command has no effect on a primary command (like [COPY](#) or [PASTE](#)).
3. Press Enter. Some of the primary commands may cause a dialogue box to be displayed if more information is needed (like file names). If so, fill in the required information and press Enter to move, copy, or insert the data. Refer to information about the specified command if you need help.
4. If no panel is displayed, the data is moved, copied, or inserted when you press Enter in step 3.

You must always specify a destination except when you are using a primary command to move, copy, or insert data into a file that is empty.

Two other line commands that are used to specify a destination are the [B](#) (before) command and the [O](#) (overlay) command.

B - Before Destination

B - Before Destination

[Previous](#) [Top](#) [Next](#)

Syntax

B[n]

Operands

n

A number that tells the editor to repeat the associated line command a specified number of times. If the number you type is greater than 1, the editor performs the command the specified number of times. If the number you type is 1, the editor performs the command only once. The number must be a positive integer. The number must be associated with a primary command.

Description

To specify that data is to be moved, copied, or inserted **BEFORE** a specific line. The B - Before command is used in conjunction with one of the line commands ([C](#) - Copy or [M](#) - Move) or one of the Primary Commands [COPY](#) or [PASTE](#) to specify a destination for the command action.

1. Type B in the line command area of the line **BEFORE** which the moved, copied, or inserted data is to be placed.
2. If you are specifying the destination for a line command, a number after the B line command specifies the number of times the other line command is performed. However, a number after the B command has no effect on a primary command (like [COPY](#) or [PASTE](#)).
3. Press Enter. Some of the primary commands may cause a dialogue box to be displayed if more information is needed (like file names). If so, fill in the required information and press Enter to move, copy, or insert the data. Refer to information about the specified command if you need help.
4. If no panel is displayed, the data is moved, copied, or inserted when you press Enter in step 3.

You must always specify a destination except when you are using a primary command to move, copy, or insert data into a file that is empty.

Two other line commands that are used to specify a destination are the [A](#) (after) command and the [O](#) (overlay) command.

BNDS - Display Bounds Line

BNDS - Display Bounds Line

[Previous](#) [Top](#) [Next](#)

Syntax

BNDS

Operands

None.

Description

The BNDS line command provides an alternative to setting the boundaries with the [BOUNDS](#) primary command; the effect on the data is the same. However, if you use both the BOUNDS primary command and the BNDS line command in the same interaction, the Primary command overrides the Line command.

To display the boundary definition (=BNDS>) line:

1. Type BNDS in the line command area of any unflagged line.
2. Press Enter.
3. The boundary definition line is inserted.

To change the BOUNDS settings:

Note: You can use the [COLS](#) line command with the BNDS line command to help check and reposition the BOUNDS settings. The COLS line command displays the column identification line.

1. Delete a < or > or >> marker. The < marker shows the left BOUNDS setting and the > marker shows the right BOUNDS setting. The >> marker indicate the Right bound is to be equal to the largest current record length, this is the default.
2. Move the cursor to a different location on the =BNDS> line.
3. Retype the deleted marker or markers. Note: The < character must be typed to the left of the > or >> character.
4. Press Enter.
5. The new BOUNDS settings are now in effect.

To revert to the default settings:

1. Display the boundary definition line. Blank out its contents with the Erase EOF key, the cursor, or the Del (delete) key.
2. Press Enter.
3. The default boundary settings will take effect.

To remove the boundary definition line from the panel:

1. You can either type [D](#) in the line command area that contains the =BNDS> flag or type [RESET](#) on the Command line.
2. Press Enter.
3. The =BNDS> line is removed from the display.

C / CC - Copy Lines

C / CC - Copy Lines

[Previous](#) [Top](#) [Next](#)

Syntax

C[n]
CC

Operands

n

The number of lines to be copied. If you do not type a number, or if the number you type is 1 on which you type C is copied

Description

To copy one or more lines within the same file:

1. Type C in the line command area of the line to be copied. If you also want to copy one or more lines that immediately follow this line, type a number greater than 1 after the C command.
2. Next, specify the destination of the line to be copied by using either the [A](#) (after), [B](#) (before), or [O](#) (overlay) line command.
3. Press Enter.
4. The line or lines are copied to the new location.

To copy a block of lines within the same file:

1. Type CC in the line command area of both the first and last lines to be copied. You can scroll (or use [FIND](#) or [LOCATE](#)) between typing the first CC and the second CC, if necessary.
2. Use the [A](#) (after), [B](#) (before), or [OO](#) (overlay) command to show where the copied lines are to be placed. Notice that when you use the block form of the C command (CC) to copy and overlay lines, you should also use the block form of the O command (OO).
3. Press Enter.
4. The lines that contain the two CC commands and all of the lines between them are copied to the new location.

To copy lines to another file:

1. Type either a [CREATE](#) or [REPLACE](#) command on the Command line.
2. Use one of the forms of the C command described previously.
3. Press Enter.
4. If you did **not** enter a filename with the [CREATE](#) or [REPLACE](#) command, then on the Save File dialog that appears, select the name of the file that you want to create or replace.
5. Press Enter.
6. The lines are copied to the file that you specified.

COLS - Display Columns

COLS - Display Columns

[Previous](#) [Top](#) [Next](#)

Syntax

COLS

Operands

None.

Description

To display the column identification (=COLS>) line, which looks like:

```
=COLS>  -1-----+-----2-----+-----3-----+-----4-----+-----
```

1. Type COLS in the line command area of any line.
2. Press Enter.
3. The column identification line is inserted.

Note: You can use the COLS line command with the [BNDS](#) line command to help check and reposition the bounds settings.

To remove the column identification line from the panel:

1. You can either type [D](#) in the line command area that contains the =COLS> flag or type [RESET](#) on the Command line.
2. Press Enter.
3. The =COLS> line is removed from the display.

SPFLite also supports a variation of [COLS](#) (referred to as RULER) which can maintain a COLS type line as the 1st data line of the screen at all times. To activate this option, use the [PROFILE](#) Screen option **Ruler Line at Top of Screen..**

D / DD - Delete Lines

D / DD - Delete Lines

[Previous](#) [Top](#) [Next](#)

Syntax

```
D[n]  
DD
```

Operands

n

The number of lines to be deleted. If you do not type a number, or if the number you type is 1, the line on which you type D is deleted

Description

To delete one or more lines:

1. Type D in the line command area of the line to be deleted. If you also want to delete one or more lines that immediately follow this line, type a number greater than 1 after the D command.
2. Press Enter.
3. The line or lines are deleted.

To delete a block of lines:

1. Type DD in the line command area of both the first and last lines to be deleted. You can scroll (or use [FIND](#) or [LOCATE](#)) between typing the first DD and the second DD, if necessary.
2. Press Enter.
3. The lines that contain the two DD commands and all of the lines between them are deleted.

DLMS - Display word Delimiters

DLMS - Display word Delimiters

[Previous](#) [Top](#) [Next](#)

Syntax

DLMS

Operands

None.

Description

The DLMS line command will insert a DLMS line display into the edit file. This line will contain the current list of characters which are treated as delimiters when SPFLite is searching for strings with the FIND / CHANGE commands **AND** one of the **WORD**, **PREFIX** or **SUFFIX** search options is chosen. The delimiters are used to determine the boundaries of a 'word'. Some computer languages extend the allowable characters which make up a 'word' to include characters such as '_' (underscore). The DLMS support allows you to customize the set of characters that are considered as delimiters to better support such languages.

Since this set of valid delimiters will vary by language, SPFLite will remember the altered character set you select and associate it with the file type being edited. It will automatically select and use this modified set each time you edit a file of that type. This is similar to the support for [TABS](#).

To display the word delimiters (=DLMS>) line:

1. Type DLMS in the line command area of any unflagged line.
2. Press Enter.
3. The word delimiter line is inserted.

To change the Word Delimiters:

1. Simply edit the characters on the line to meet your needs.
2. Ensure a blank remains embedded within the string somewhere (unless you **REALLY** don't want a space as a delimiter)
3. Press Enter.

5. The new delimiter characters are now in effect.

To remove the DLMS line from the panel:

1. You can either type [D](#) in the line command area that contains the =DLMS> flag or type [RESET](#) on the Command line.
2. Press Enter.
3. The =DLMS> line is removed from the display.

F - Display First lines

F - Display First lines

[Previous](#) [Top](#) [Next](#)

Syntax

F[n]

Operands

n

The number of lines to be redisplayed. If you do not type a number, or if the number you type is less than 1, one line is redisplayed.

Description

To redisplay the first line or lines of a block of excluded lines:

1. Type F in the line command area next to the dashed line that shows where lines have been excluded. The message in the dashed line tells you how many lines are excluded. If you want to redisplay more than one line, type a number greater than 1 after the F command.
2. Press Enter.
3. The first line or lines are redisplayed.

I - Insert Lines

I - Insert Lines

[Previous](#) [Top](#) [Next](#)

Syntax

I[n]

Operands

n

The number of blank lines to insert. If you do not type a number, or if the number you type is less than 1, one line is inserted.

Description

To insert one or more lines in a file:

1. Type `␣` in the line command area of the line that the inserted line is to follow. If you want to insert more than one line, type a number greater than 1 after the `␣` command.
2. Press Enter.
3. The line or lines are inserted.

If you type any information, even a blank character in the inserted line, the line becomes part of the data and is assigned a line number the next time you press Enter. However, if you do not type any information, the space for the new line is automatically deleted the next time you press Enter.

If you type information on the last, or only, inserted line and the cursor is still in the data portion of that line, the editor automatically inserts another line when you press Enter.

See also the [N](#) command for inserting permanent lines.

J - JOIN two lines together

J - JOIN two lines together

[Previous](#) [Top](#) [Next](#)

Syntax

`J`

Operands

`none`

There are no operands. The specified line and the immediately following line are merged.

Description

To join together a line and it's immediately following line.

1. Type `J` in the line command area of the 1st line of the pair of lines to be joined.
2. Press Enter.
3. The lines are joined.

Joining lines follows the following rules:

- The lines are treated as plain text. i.e. simple blank delimited strings. Quoted strings are not treated

specially.

- Leading blanks on a line are retained.
- Multiple embedded blanks between strings are NOT compressed to a single space.
- One space is added following the last character of the 1st line before appending the 2nd line.

Sample

```
-----1-----2-----3-----4-----5-----6-----7-----8-----9-----10-----11
J.....      Here's    some    text
.....      Some more text

would become the single line

-----1-----2-----3-----4-----5-----6-----7-----8-----9-----10-----11
.....      Here's    some    text    Some more text
```

L - Display Last lines

L - Display Last lines

[Previous](#) [Top](#) [Next](#)

Syntax

L[n]

Operands

n

The number of lines to be redisplayed. If you do not type a number, or if the number you type is less than 1, one line is redisplayed.

Description

To redisplay the last line or lines of a block of excluded lines:

1. Type L in the line command area next to the dashed line that shows where lines have been excluded. The message in the dashed line tells you how many lines are excluded. If you want to redisplay more than one line, type a number greater than 1 after the L command.
2. Press Enter.
3. The last line or lines are redisplayed.

LC / LCLC - Lowercase Lines

LC / LCLC/LCC - Lowercase Lines

[Previous](#) [Top](#) [Next](#)

Syntax

```
LC[n]  
LCLC | LCC | LL
```

Operands

n

The number of lines to be lowercased. If you do not type a number, or if the number you type is 0, the command is applied to the line on which you type LC is lowercased

Description

To convert characters on one or more lines to lowercase:

1. Type LC in the line command area of the source code line that contains the characters you want to convert. If you also want to convert characters on one or more lines that immediately follow this line, type a number greater than 1 after the LC command.
2. Press Enter.
3. The characters on the source code lines are converted to lowercase.

To convert characters in a block of lines to lowercase:

1. Type LCLC (or LCC) in the line command area of both the first and last source code lines that contain characters that are to be converted. You can scroll (or use [FIND](#) or [LOCATE](#)) between typing the first LCLC and the second LCLC, if necessary.
2. Press Enter.
3. The characters in the source code lines that contain the two LCLC commands and in all of the source code lines between them are converted to lowercase.

M / MM - Move Lines

M / MM - Move Lines

[Previous](#) [Top](#) [Next](#)

Syntax

```
M[n]  
MM
```

Operands

n

The number of lines to be moved. If you do not type a number, or if the number you type is 0, the command is applied to the line on which you type M is moved

Description

To move one or more lines within the same file:

1. Type M in the line command area of the line to be copied. If you also want to move one or more lines that immediately follow this line, type a number greater than 1 after the M command.
2. Next, specify the destination of the line to be moved by using either the [A](#) (after), [B](#) (before), or [O](#)

- (overlay) line command.
3. Press Enter.
 4. The line or lines are moved to the new location.

To move a block of lines within the same file:

1. Type MM in the line command area of both the first and last lines to be copied. You can scroll (or use [FIND](#) or [LOCATE](#)) between typing the first MM and the second MM, if necessary.
2. Use the [A](#) (after), [B](#) (before), or [OO](#) (overlay) command to show where the moved lines are to be placed. Notice that when you use the block form of the M command (MM) to move and overlay lines, you should also use the block form of the O command (OO).
3. Press Enter.
4. The lines that contain the two MM commands and all of the lines between them are copied to the new location.

To move lines to another file:

1. Type either a [CREATE](#) or [REPLACE](#) command on the Command line.
2. Use one of the forms of the M command described previously.
3. Press Enter.
4. If you did **not** enter a filename with the [CREATE](#) or [REPLACE](#) command, then on the Save File dialog that appears, select the name of the file that you want to create or replace.
5. Press Enter.
6. The lines are moved to the file that you specified. Remember, MOVEing the lines to a file DELETES them from the existing edit file.

N - Insert Permanent Lines

N - Insert Permanent Lines

[Previous](#) [Top](#) [Next](#)

Syntax

N[n]

Operands

n	The number of permanent blank lines to insert. If you do not type a number, or if the number you type is 1, only one line is inserted.
---	--

Description

To insert one or more permanent lines in a file:

1. Type **N** in the line command area of the line that the inserted line is to follow. If you want to insert more than one line, type a number greater than 1 after the **N** command.
2. Press Enter.
3. The line or lines are inserted.

The lines are inserted as permanent blank lines, unlike the **I** command, which inserts the lines as temporary blank lines. Whether you use these lines or not, they will remain as blank lines within the edit file.

O / OO - Overlay Lines

O / OO - Overlay Lines

[Previous](#) [Top](#) [Next](#)

Syntax

```
O[n]  
OO
```

Operands

n	The number of lines to be overlaid. If you do not type a number, or if the number you type is line is overlaid.
---	---

Description

The O (overlay) line command specifies the destination of data that is to be copied or moved by the **C** (copy) or **M** (move) line commands. The data that is copied or moved overlays **blanks** in an existing line of data. This allows you to rearrange a single-column list of items into multiple column, or tabular, format.

To overlay one or more lines:

1. Type either **M** or **C** in the line command area of the line that is to be moved or copied.
2. Type **O** in the line command area of the line that the moved or copied line is to **overlay**. You can type a number after the O line command to specify the number of times that the M or C line command is to be performed.
3. Press Enter.
4. The data being moved or copied overlays the specified line or lines.

To overlay a block of lines:

1. Type either **MM** or **CC** in the line command area of the first and last lines of a block of lines that is to be moved or copied. You can scroll (or use **FIND** or **LOCATE**) between typing the first command and the second command, if necessary.
2. Type **OO** in the line command area of the first and last lines that the block of lines being moved or copied is to overlay. Again, you can scroll (or use **FIND** or **LOCATE**) between typing the first OO and the second OO, if necessary.
3. Press Enter.
4. The lines that contain the two **CC** or **MM** commands and all of the lines between them overlay the lines that contain the two OO commands and all of the lines between them.

Only blank characters in the lines specified with O or OO are overlaid with corresponding characters from the source lines. Characters that are not blank are not overlaid. The overlap affects only those characters

within the current column boundaries.

The number of source and receiving lines need not be the same. If there are more receiving lines, the source lines are repeated until the receiving lines are gone. If there are more source lines than receiving lines, the extra source lines are ignored. The overlay operation involves only data lines. Special lines such as TABS, BNDS, and COLS are ignored as either source or receiving lines.

R / RR - Repeat Lines

R / RR - Repeat Lines

[Previous](#) [Top](#) [Next](#)

Syntax

```
R[n]  
RR[n]
```

Operands

n

The number of lines to be repeated. If you do not type a number, or if the number you type is line on which you type R is repeated

Description

To repeat one or more lines:

1. Type R in the line command area of the line that is to be repeated. If you want to repeat the line more than once, type a number that is greater than 1 immediately after the R command.
2. Press Enter.
3. The editor inserts a duplicate copy or copies of the line immediately after the line that contains the R command.

To repeat a block of lines:

1. Type RR in the line command area of both the first and last lines to be repeated. You can scroll (or use [FIND](#) or [LOCATE](#)) between typing the first RR and the second RR, if necessary.
2. Press Enter.
3. The lines that contain the two RR commands and all of the lines between them are repeated immediately after the line that contains the second RR command.

S - Show Lines

S - Show Lines

[Previous](#) [Top](#) [Next](#)

Syntax

S[n]

Operands

n

The number of lines to be redisplayed. If you do not type a number, or if the number you type is 1, only one line is redisplayed.

Description

The S (show line) line command causes one or more lines in a block of excluded lines to be redisplayed. The redisplayed lines have the leftmost indentation levels; they contain the fewest leading blanks.

If there are more than 2 excluded lines, and you do not type a number or if the number you type is 1, only one line is redisplayed.

Note: If you enter an S line command to display all but one line of an excluded block, then that line is also displayed. This could result in more lines being displayed than the number you requested. For example, if five lines are excluded in a block, an S4 command causes all five lines to be displayed.

To redisplay a line or lines of a block of excluded lines:

1. Type S in the line command field next to the dashed line that shows where a line or lines has been excluded. The message in the dashed line tells you how many lines are excluded. If you want to redisplay more than one line, type a number greater than 1 after the S command. If you type S3, for example, the three lines with the leftmost indentation level are displayed again. If more than three lines exist at this indentation level, only the first three are displayed.
2. Press Enter. The line or lines with the fewest leading blanks are redisplayed.

SC / SCSC - Sentence Case Text

SC / SCSC - Sentence Case Text

[Previous](#) [Top](#) [Next](#)

Syntax

SC[n]
SCSC

Operands

n

The number of lines to be converted to Sentence Case. If you do not type a number, or if the type is 1, only the line on which you type UC is uppercased.

Description

To convert characters on one or more lines to Sentence Case:

1. Type SC in the line command area of the source code line that contains the characters that you want to convert to Sentence case. To convert characters on lines following this one, type a number greater than 1 after the SC command.

2. Press Enter.
3. The characters on the source code line or lines are converted to Sentence Case.

To convert characters in a block of lines to Sentence Case:

1. Type SCSC in the line command area of both the first and last source code lines that contain characters that are to be converted. You can scroll (or use [FIND](#) or [LOCATE](#)) between typing the first SCSC and the second SCSC, if necessary.
2. Press Enter.
3. The characters in the source code lines that contain the two SCSC commands and in all of the source code lines between them are converted to Sentence Case.

Sentence Case Description

Sentence Case processing is performed by first converting **all** the text to Lowercase. Then the first letter of the first word of each sentence is Capitalized.

Note:

- The first word of the first line processed is considered to be the start of a sentence.
- The last word of a sentence is considered to be any word ending with a period (.), Exclamation point (!), or a Question mark (?).

TABS - Display TABS Line

TABS - Display TABS Line

[Previous](#) [Top](#) [Next](#)

Syntax

TABS

Operands

None.

Description

When you type TABS in the line command area, =TABS> is displayed along with any previously defined tab positions. To remove the =TABS> line, use the [D](#) (delete) line command or the [RESET](#) primary command, or end the edit session. The =TABS> line is never saved as part of the data.

The tab definitions remain in effect, even if they are not displayed, until you change them. Tab definitions are retained, and are automatically used the next time you edit the same kind of data (based on the file extension). Note, the primary CAPS setting is saved along with TABS for each file type.

See "[Defining and Using TABS](#)" for more information on using Tabs.

TC / TCTC - Title Case Text

TC / TCTC - Title Case Text

[Previous](#) [Top](#) [Next](#)

Syntax

TC[n]
TCTC

Operands

n

The number of lines to be converted to Title Case. If you do not type a number, or if the number is 1, only the line on which you type TC is uppercased.

Description

To convert characters on one or more lines to Title Case:

1. Type TC in the line command area of the source code line that contains the characters that you want to convert to Title case. To convert characters on lines following this one, type a number greater than 1 after the TC command.
2. Press Enter.
3. The characters on the source code line or lines are converted to Title Case.

To convert characters in a block of lines to Title Case:

1. Type TCTC in the line command area of both the first and last source code lines that contain characters that are to be converted. You can scroll (or use [FIND](#) or [LOCATE](#)) between typing the first TCTC and the second TCTC, if necessary.
2. Press Enter.
3. The characters in the source code lines that contain the two TCTC commands and in all of the source code lines between them are converted to Title Case.

Title Case Description

Title Case processing is performed by first converting **all** the text to Lowercase. Then the first letter of **each** word is Capitalized.

TF - Text Flow a Paragraph

TF - Text Flow a Paragraph

[Previous](#) [Top](#) [Next](#)

Syntax

TF[n]

Operands

n

The column number to which the text should be flowed. The default is the panel width. If a number greater than the current maximum record length is specified, the maximum record length is used.

Description

To flow text:

1. Type TF in the line command area of the line at which you want the text to begin flowing. If you want to specify the rightmost column position for the restructured text, type a number greater than 1 immediately after the TF command.
2. Press Enter.
3. The text is flowed from the beginning of that line to the end of the paragraph.

See "[Word Processing](#)" for more information.

TS - Text Split a line

TS - Text Split a line

[Previous](#) [Top](#) [Next](#)

Syntax

TS[n]

Operands

n

The number of blank lines to be inserted between the split lines. If you do not type a number, the default number that you type is 1, the editor inserts only one blank line.

Description

To split a line:

1. Type TS in the line command area of the line you would like to split. If you want to insert more than one blank line between the split lines, type a number greater than 1 immediately after the TS command.
2. Move the cursor to the desired split point.
3. Press Enter.
4. The line is split at the cursor location and the requested number of blank insert lines are added.

To rejoin lines, use the [TE](#) (text flow) line command. See "[Word Processing](#)" for more information.

UC / UCUC - Uppercase Lines

UC / UCUC/UCC - Uppercase Lines

[Previous](#) [Top](#) [Next](#)

Syntax

```
UC[n]  
UCUC | UCC | UU
```

Operands

n

The number of lines to be uppercased. If you do not type a number, or if the number you type is less than 1, the command will be applied to the line on which you type UC is uppercased.

Description

To convert characters on one or more lines to uppercase:

1. Type UC (or UCC) in the line command area of the source code line that contains the characters that you want to convert. To convert characters on lines following this one, type a number greater than 1 after the UC command.
2. Press Enter.
3. The characters on the source code line or lines are converted to uppercase.

To convert characters in a block of lines to uppercase:

1. Type UCUC in the line command area of both the first and last source code lines that contain characters that are to be converted. You can scroll (or use [FIND](#) or [LOCATE](#)) between typing the first UCUC and the second UCUC, if necessary.
2. Press Enter.
3. The characters in the source code lines that contain the two UCUC commands and in all of the source code lines between them are converted to uppercase.

X / XX - Exclude Lines

X / XX - Exclude Lines

[Previous](#) [Top](#) [Next](#)

Syntax

```
X[n]  
XX
```

Operands

n

The number of lines to be excluded. If you do not type a number, or if the number you type is less than 1, the command will be applied to the line on which you type X is excluded.

Description

To exclude one or more lines:

1. Type X in the line command area of the line that you want to exclude. If you want to exclude one or more lines that immediately follow this line, type a number greater than 1 immediately after the X command.
2. Press Enter.

3. The lines are excluded from the panel.

To exclude a block of lines:

1. Type XX in the line command area of both the first and last lines that you want to exclude. You can scroll (or use [FIND](#) or [LOCATE](#)) between typing the first XX and the second XX, if necessary.
2. Press Enter.
3. The lines that contain the two XX commands and all of the lines between them are excluded.

Shift Left

Shift Left

[Previous](#) [Top](#) [Next](#)

Syntax

```
( [n]  
(( [n]
```

Operands

n

A number that tells the editor how many positions to shift. If you omit this operand, the default is 2.

Description

The ((column shift left) line command moves characters on a line to the left without altering their relative spacing. Characters shifted past the current [BOUNDS](#) setting are deleted.

To column shift one line toward the left side of your display:

1. Type (in the line command area of the line to be shifted. Beside the command, type a number other than 2 if you want to shift the line other than 2 columns.
2. Press Enter.
3. The line data is shifted the requested number of positions. Characters shifted past the current [BOUNDS](#) setting are deleted.

To column shift a block of lines toward the left side of your display:

1. Type ((in the line command area of the first line to be shifted. Beside the command, type a number other than 2 if you want to shift the block of lines other than 2 columns.
2. Type ((in the line command area of the last line to be shifted. You can scroll (or use [FIND](#) or [LOCATE](#)) between typing the first ((and the second ((, if necessary.
3. Press Enter.
4. The lines that contain the two ((commands and all of the lines between them are column shifted to the left. Characters shifted past the current [BOUNDS](#) setting are deleted.

The [BOUNDS](#) setting limits column shifting. If you shift columns beyond the current [BOUNDS](#) setting, the editor deletes the text beyond the [BOUNDS](#) without displaying a warning message.

Shift Right

Shift Right

[Previous](#) [Top](#) [Next](#)

Syntax

```
) [n]  
) ) [n]
```

Operands

n

A number that tells the editor how many positions to shift. If you omit this operand, the default is 2.

Description

The `)` (column shift right) line command moves characters on a line to the right without altering their relative spacing. Characters shifted past the current [BOUNDS](#) setting are deleted.

To column shift one line toward the right side of your display:

1. Type `)` in the line command area of the line to be shifted. Beside the command, type a number other than 2 if you want to shift the line other than 2 columns.
2. Press Enter.
3. The line data is shifted the requested number of positions. Characters shifted past the current [BOUNDS](#) setting are deleted.

To column shift a block of lines toward the right side of your display:

1. Type `))` in the line command area of the first line to be shifted. Beside the command, type a number other than 2 if you want to shift the block of lines other than 2 columns.
2. Type `))` in the line command area of the last line to be shifted. You can scroll (or use [FIND](#) or [LOCATE](#)) between typing the first `))` and the second `))`, if necessary.
3. Press Enter.
4. The lines that contain the two `))` commands and all of the lines between them are column shifted to the right. Characters shifted past the current [BOUNDS](#) setting are deleted.

The [BOUNDS](#) setting limits column shifting. If you shift columns beyond the current [BOUNDS](#) setting, the editor deletes the text beyond the [BOUNDS](#) without displaying a warning message.

Shift Data Left

Shift Data Left

[Previous](#) [Top](#) [Next](#)

Syntax

```
< [n]  
<< [n]
```

Operands

n

A number that tells the editor how many positions to shift. If you omit this operand, the default is 2.

Description

The < (Data shift left) line command moves characters on a line to the left without altering their relative spacing. Characters will NOT be shifted past the current [BOUNDS](#) setting, if necessary, the number of shifted columns will be reduced.

To Data shift one line toward the left side of your display:

1. Type < in the line command area of the line to be shifted. Beside the command, type a number other than 2 if you want to shift the line other than 2 columns.
2. Press Enter.
3. The line data is shifted the requested number of positions. The shift is non-destructive, no characters will be dropped at the [Bounds](#) locations.

To column shift a block of lines toward the left side of your display:

1. Type << in the line command area of the first line to be shifted. Beside the command, type a number other than 2 if you want to shift the block of lines other than 2 columns.
2. Type << in the line command area of the last line to be shifted. You can scroll (or use [FIND](#) or [LOCATE](#)) between typing the first << and the second <<, if necessary.
3. Press Enter.
4. The lines that contain the two << commands and all of the lines between them are Data shifted to the left. The shift is non-destructive, no characters will be dropped at the [Bounds](#) locations.

The [BOUNDS](#) setting limits Data shifting. Data shifting is particularly oriented to programming source, please review [Shifting Data](#) for full details.

Shift Data Right

Shift Data Right

[Previous](#) [Top](#) [Next](#)

Syntax

```
> [n]  
>> [n]
```

Operands

n

A number that tells the editor how many positions to shift. If you omit this operand, the default is 2.

Description

The > (Data shift right) line command moves characters on a line to the right without altering their relative spacing. Characters will NOT be shifted past the current [BOUNDS](#) setting, if necessary, the number of shifted columns will be reduced.

To column shift one line toward the right side of your display:

1. Type > in the line command area of the line to be shifted. Beside the command, type a number other than 2 if you want to shift the line other than 2 columns.
2. Press Enter.
3. The line data is shifted the requested number of positions. Characters will NOT be shifted past the current [BOUNDS](#) setting, if necessary, the number of shifted columns will be reduced.

To column shift a block of lines toward the right side of your display:

1. Type >> in the line command area of the first line to be shifted. Beside the command, type a number other than 2 if you want to shift the block of lines other than 2 columns.
2. Type >> in the line command area of the last line to be shifted. You can scroll (or use [FIND](#) or [LOCATE](#)) between typing the first)) and the second)), if necessary.
3. Press Enter.
4. The lines that contain the two)) commands and all of the lines between them are Data shifted to the right. Characters will NOT be shifted past the current [BOUNDS](#) setting, if necessary, the number of shifted columns will be reduced.

The [BOUNDS](#) setting limits Data shifting. Data shifting is particularly oriented to programming source, please review [Shifting Data](#) for full details.

Primary Commands

Primary Commands

[Previous](#) [Top](#) [Next](#)

Descriptions of the following Primary Commands are contained in this section:

Note: You have the ability to create your own special commands by using the Macro facility to issue a series of Primary commands in succession. Your created commands are entered on the command line just as the built in commands. See [Macro Support](#) for details.

- [BOUNDS](#) Set EDIT Boundaries
- [BPASTE](#) Block PASTE from the Clipboard
- [BROWSE](#) Open a file for Browsing
- [CANCEL](#) Cancel the Edit without saving

○ <u>CAPS</u>	Set Keyboard CAPS mode
○ <u>CHANGE</u>	Change a data string
○ <u>COPY</u>	Copy (include) a file into the Edit area
○ <u>CREATE</u>	Create a new file from the Edit data
○ <u>CRETRIEV</u>	Conditional Retrieve
○ <u>CRLF</u>	Alter the working CR/LF handling mode
○ <u>CUT</u>	Cut selected lines to the Clipboard
○ <u>DELETE</u>	Delete specified lines
○ <u>EDIT</u>	Edit another file in a new SPFLite Window
○ <u>ENCODING</u>	Specify the type of UniCode encoding to use
○ <u>END</u>	Terminate the current file Edit
○ <u>EXCLUDE</u>	Exclude line(s) from the display
○ <u>FIND</u>	Find a string within the data
○ <u>FLIP</u>	Reverse the X and NX's lines
○ <u>HELP</u>	Display the SPFLite Help file
○ <u>HEX</u>	Switch the display between normal and HEX format
○ <u>HIDE</u>	Remove the XX indicator lines from the screen
○ <u>LOCATE</u>	Scroll the display to a specified line
○ <u>OPEN</u>	Open a file for Editing
○ <u>PASTE</u>	Paste lines from the Clipboard into the file
○ <u>PFKEY</u>	Open Preferences to allow altering PFKEY definitions
○ <u>PRINT</u>	Print all or a portion of the Edit file
○ <u>PROFILE</u>	Open the Profile Preferences dialogue
○ <u>RCHANGE</u>	Repeat the last CHANGE command
○ <u>REPLACE</u>	Replace an existing file with the current Edit data
○ <u>RESET</u>	Reset Display screen fields
○ <u>RETRIEVE</u>	Retrieve previously entered Primary commands
○ <u>RFIND</u>	Repeat the previous FIND command
○ <u>RUN</u>	Execute an external System command
○ <u>SAVE</u>	Save the current working data
○ <u>SAVEAS</u>	Save the current data as a new file, and switch to that file
○ <u>SORT</u>	Sort the current data
○ <u>SWAP</u>	Swap to a different working Tab
○ <u>TABS</u>	Turn Tab processing ON and OFF
○ <u>UNDO</u>	Undo the last screen interaction

Primary Command Notation Conventions

Primary Command Notation Conventions

[Previous](#) [Top](#) [Next](#)

Primary Command Notation Conventions

The syntax of the edit primary commands uses the following notation conventions:

UPPERCASE	Uppercase commands or operands must be spelled as shown (in either uppercase or lowercase).
lowercase	Lowercase operands are variables; substitute your own values
Brackets ([])	Operands in brackets are optional.
Stacked operands	Stacked operands show two or more operands from which you can select. If you do not choose any, the Editor uses the default operand.

Braces ({ })

Braces show two or more operands from which you must select one.

OR (|)

The OR (|) symbol shows two or more operands from which you must select one.

BOUNDS - Set Edit Boundaries

BOUNDS - Set Edit Boundaries

[Previous](#) [Top](#) [Next](#)

Syntax

```
BOUNDS [left-col {right-col|MAX} ]
```

Operands

left-col The left column boundary to be set

right-col|MAX The right column boundary to be set or the Keyword 'MAX' to indicate the maximum record length

You cannot specify the same column for both boundaries. An asterisk (*) can be used to represent the current value of the boundary.

Description

The BOUNDS primary command provides an alternative to setting the boundaries with the BNDS line command; the effect on the data is the same. However, if you use both the BOUNDS primary command and the BNDS line command in the same interaction, the primary command overrides the line command.

Note: The equivalent of the BNDS Line Command operand ">>" in the BOUNDS command is 'MAX', used as the 2nd operand.

To reset the boundaries to the default column (1 and Max Record Length)

1. On the command line, type: BOUNDS
2. Press Enter
3. The boundaries are reset to the defaults.

BPASTE - Block Paste

BPASTE - Block Paste

[Previous](#) [Top](#) [Next](#)

Syntax

```
BPASTE
```


Operands

None.

Description

BPASTE copies data from the Windows Clipboard to the Edit data. The data is handled as a 'Block' Paste. i.e. it is pasted in as an Overlay type paste at the **current cursor location**. Any current text within the bounds of the block will simply be overlaid. If need be, the line length will be extended to handle the paste. Similarly, if the block extends past the bottom of the current file, additional lines will be added to handle the paste data.

To BPaste data into a file:

Either, On the Command line, type:

```
Command > BPASTE
```

Move the cursor to the insert point (the upper left corner of the block)

Press Enter. The data is copied.

Or,

Move the cursor to the insert point (the upper left corner of the block)

Press a PFKEY to which the BPASTE command has been assigned.

BROWSE - Open a file for Browse only

BROWSE - Open a file for Browse only

[Previous](#) [Top](#) [Next](#)

Syntax

```
BROWSE      [file-name] | NEW
```

Operands

file-name The name of the file to be Browsed.

NEW If the keyword NEW is specified, a Browse Tab will be opened on an

Description

The BROWSE command loads a file into the work space for Browsing. If no file-name operand is specified, then a standard Windows File Open Dialog will be presented for you to select the file for Browsing.

In BROWSE mode, you cannot SAVE the file, nor will AUTOSAVE cause the file to be saved. If you have altered the data, and wish to save the altered copy, you may still use the CREATE command to save it to a different filename.

If the current working Tab contains data, then a new Tab will be Opened to hold the Browse session, otherwise the current Tab will be used.

Default Directory

When no operand at all or only a simple unqualified filename is provided for the BROWSE command, the default directory used for searching for the file or for the file open dialog's starting directory will be determined as follows:

- If there is an active file being edited in the tab where the command is issued, then the Path for THAT Active file is used as the default for the command.
- If there is NO active file (i.e. the tab header displays (Empty):
 - If File Manager is active, the current displayed directory of File Manager will be used.
 - If File Manager is NOT active, then the Preferences setting for Default Open directory will be used (Last Used or Working). Remember 'Working' means the Windows working directory when SPFLite was started.

CANCEL - Cancel Edit Session

CANCEL - Cancel Edit Session

[Previous](#) [Top](#) [Next](#)

Syntax

CANCEL | CAN

Operands

None.

Description

CANCEL is especially useful if you have changed the wrong data, or if the changes themselves are incorrect. To cancel changes to a file:

On the Command line, type:

Command > CANCEL

Press Enter. All lines in the current Edit session are deleted leaving you with an empty edit session. Note as well that the Undo stack is cleared so that you **cannot UNDO a CANCEL** command.

Note: If you issue [SAVE](#) and later issue [CANCEL](#), the changes you made before issuing [SAVE](#) are not canceled.

CAPS - Set Keyboard CAPS Mode

CAPS - Set Keyboard CAPS Mode

[Previous](#) [Top](#) [Next](#)

Syntax

```
CAPS { ON | OFF }
```

Operands

ON	All data from the keyboard is Uppercased regardless of the keyboard CAPS key.
OFF	All data from the keyboard is entered 'as is'.

Description

The CAPS command allows you to force CAPS mode regardless of the setting of the keyboard CAPS key. Note the setting for CAPS is saved along with the TABS value for each different filetype extension. e.g. TXT, BAS, C etc.

CHANGE - Change a Data String

CHANGE - Change a Data String

[Previous](#) [Top](#) [Next](#)

Syntax

```
CHANGE | C      string-1 string-2 [col-1 [col-2]]  
                [NEXT | PREV | ALL]
```

```
[X | NX]
[FIRST | LAST]
[WORD | PREFIX | SUFFIX]
```

Operands

string-1	The search string you want to change.
string-2	The string you want to replace string-1.
col-1	Left column of a range (with col-2) within which the string-1 value must be found. If no col-2 operand must be found starting in col-1.
col-2	The right boundary of a column range which restricts the column range for the search.
NEXT	Starts at the first position after the current cursor location and searches ahead to find the next occurrence of string-1. NEXT is the default.
PREV	Starts at the current cursor location and searches backward to find the previous occurrence of string-1.
ALL	Starts at the top of the data and searches ahead to find all occurrences of string-1.
X	Scans only lines that are excluded from the display.
NX	Scans only lines that are not excluded from the display.
FIRST	Starts at the top of the data and searches ahead to find the first occurrence of string-1.
LAST	Starts at the bottom of the data and searches backward to find the last occurrence of string-1.
WORD	Locates string-1 when it is delimited on both sides by blanks or other non-alphanumeric characters.
PREFIX	Locates string-1 at the beginning of a word.
SUFFIX	Locates string-1 at the end of a word.

Description

You can use the **CHANGE** command to change one string to another either singly, or for all occurrences. As well optional operands allow you to restrict the column boundaries within which it operates, along with other characteristics of the string searched for.

To change the next occurrence of **ME** to **YOU** without specifying any other qualifications:

On the Command line, type:

```
Command > CHANGE ME YOU
```

Press Enter. This command changes only the next occurrence of the letters **ME** to **YOU**. Since no other qualifications were specified, the letters **ME** can be:

- Uppercase or a mixture of uppercase and lowercase
- At the beginning of a word (prefix), the end of a word (suffix), or the entire word (word)
- In an excluded line or a nonexcluded line
- Anywhere within the current boundaries.

To change the next occurrence of ME to YOU, but only if the letters are uppercase:

On the Command line, type:

```
Command > CHANGE C'ME' YOU
```

Press Enter. This type of change is called a character string change (note the C that precedes the search string) because it changes the next occurrence of the letters ME to YOU only if the letters are found in uppercase. However, since no other qualifications were specified, the change occurs no matter where the letters are found, as outlined in the preceding list.

For more information, including other types of search strings, see "[Finding, Changing, and Excluding Data](#)".

The following example changes the first plus in the file to a minus. However, the plus must be the first character of a word:

```
CHANGE '+' '-' FIRST PREFIX
```

The following example changes the last plus in the file to a minus. However, the plus must be the last character of a word; and it must be found on an excluded line:

```
CHANGE '+' '-' LAST SUFFIX X
```

The following example changes the first plus that precedes the cursor position to a minus. However, the character must be a stand alone character (not part of any other word); it must be on a nonexcluded line; and it must exist within columns 1 and 5:

```
CHANGE '+' '-' PREV WORD NX 1 5
```

NOTE: When either of the CHANGE search and or replace operands is a Picture string (see [Specifying A Picture String](#)) certain search operands are restricted. i.e. the Change operands **PREV**, **LAST**, **PREFIX** and **SUFFIX** may not be used.

COPY - Include an external file

COPY - Include an external file

[Previous](#) [Top](#) [Next](#)

Syntax

`COPY` `[file-name]`

Operands

`file-name` The name of the file you wish to include in the Edit data.

Description

`COPY` adds a copy of data that already exists to the Edit data.

To copy data into an empty file:

On the Command line, type:

```
Command > COPY filename
```

The filename operand is optional. If you do not specify it the standard Windows File Open dialog will be displayed. Enter the filename in this dialog.

Press Enter. The data is copied.

To copy data into Edit data that is not empty:

On the Command line, type:

```
Command > COPY filename
```

The filename operand is optional. If you do not specify it the standard Windows File Open dialog will be displayed. Enter the filename in this dialog.

Use the [A](#) or [B](#) line command to indicate where the copied data is to be inserted. However, a number indicating that the A or B command should be

repeated cannot follow the line command.

Note: If the Edit data is not empty and you do not specify a destination with the [A](#) or [B](#) line command, a Pending message appears in the upper-right corner of the panel and the data is not copied.

Default Directory

When no operand at all or only a simple unqualified filename is provided for the COPY command, the default directory used for searching for the file or for the file open dialog's starting directory will be determined as follows:

- If there is an active file being edited in the tab where the command is issued, then the Path for THAT Active file is used as the default for the command.
- If there is NO active file (i.e. the tab header displays (Empty):
 - If File Manager is active, the current displayed directory of File Manager will be used.
 - If File Manager is NOT active, then the Preferences setting for Default Open directory will be used (Last Used or Working). Remember 'Working' means the Windows working directory when SPFLite was started.

CREATE - Create an external file

CREATE - Create an external file

[Previous](#) [Top](#) [Next](#)

Syntax

```
CREATE    [file-name] [ from-line to-line ]
```

Operands

file-name	The name of the file you wish to create.
from-line	The starting line number if only partial Edit data is to be written. This operand may also use a b
to-line	The ending line number if only partial Edit data is to be written. This operand may also use a b

Description

CREATE will use all or a specified portion of the Edit data to create a new external file. If you do not specify a full pathname as part of file-name, then the file will be created in the same directory as the file being edited. Note: If the filename already exists, then use the [REPLACE](#) command instead.

To copy the entire Edit data into a new file:

On the Command line, type:

```
Command > CREATE file-name
```

The filename operand is optional. If you do not specify it, the standard Windows File Open dialog will be displayed. Enter the filename in this dialog.

Press Enter. The Edit data is copied to the new filename.

To copy only a portion of the Edit data into a new File:

On the Command line, type:

```
Command > CREATE filename from-line to-line
```

The filename operand is optional. If you do not specify it the standard Windows File Open dialog will be displayed. Enter the filename in this dialog.

The from-line and to-line operands specify the starting and ending line numbers to be included in the new file. As an alternative to specifying these on the command line, you may use the [CC/CC](#) or [MMMM](#) line commands to indicate the desired lines. Note that the [MMMM](#) selection will cause the lines to be Deleted from the Edit data following successful completion.

Press Enter. The Edit data is copied/moved to the new filename.

File-Format

If you wish to write a file in a format other than the normal default specified by your PROFILE CRLF setting see ["Handling Non-Windows Txt Files"](#) for additional info.

Default Directory

When no operand at all or only a simple unqualified filename is provided for the CREATE command, the default directory used for searching for the file or for the file open dialog's starting directory will be determined as follows:

- If there is an active file being edited in the tab where the command is issued, then the Path for THAT Active file is used as the default for the command.
- If there is NO active file (i.e. the tab header displays (Empty):
 - If File Manager is active, the current displayed directory of File Manager will be used.
 - If File Manager is NOT active, then the Preferences setting for Default Open directory will be used (Last Used or Working). Remember 'Working' means the Windows working directory when SPFLite was started.

CRETRIEV - Conditional Retrieve

CRETRIEV - Conditional Retrieve

[Previous](#) [Top](#) [Next](#)

Syntax

CRETRIEV

Operands

None.

Description

The CRETRIEV works exactly the same as the RETRIEVE command **Except** for being cursor location dependant.

If the cursor is **on** the command line, the CRETRIEV command will act like the [RETRIEVE](#) command and bring back to the Command line the previously issued command for use to modify and re-enter. Successive CRETRIEV commands will retrieve the previous, next previous, etc. commands. The Editor keeps the last 10 unique command lines available for retrieval. Note that the list of previous commands will be saved and recalled between SPFLite sessions.

If the cursor is **NOT on** the command line, then CRETRIEV acts like the **Home** key.

See also the [Preferences - General](#) option for minimum Retrieve length.

CRLF - Alter Working CRLF Mode

CRLF - Alter Working CRLF Mode

[Previous](#) [Top](#) [Next](#)

Syntax

```
CRLF    {file-format}
```

Operands

`file-format` One of **{WIN | UNIX | MAC}** to indicate the handling of end-of-lines.

Description

[Preferences - General](#) allows specification of a **default** mode to control the handling of end-of-line characters. The CRLF command allows you to **temporarily** alter the working value for this handling **for the remainder of this edit session only** (or until another CRLF command is issued).

The operand is mandatory and must consist of one of **WIN**, **UNIX**, or **MAC**. For additional information on these values and their effect, see ["Handling Non-Windows Txt Files"](#).

CUT - Cut data to the Clipboard

CUT - Cut data to the Clipboard

[Previous](#) [Top](#) [Next](#)

Syntax

```
CUT    [ from-line to-line ]    [APPEND]
```

Operands

from-line	The starting line number to be Cut. This operand may also use a bookmark reference.
to-line	The ending line number to be Cut. This operand may also use a bookmark reference.

Description

CUT saves copies of lines from an edit session to the Windows clipboard for later retrieval by the [PASTE](#) or [BPASTE](#) commands or any other Windows application which handles Text formatted clipboard entries.

Specifying the Data to CUT

Two methods of specifying the data are possible.

- The first is the 'classic' SPF method of using line numbers. The line numbers may be specified as operands to the CUT command, or the lines may be marked by [C/CC](#) or [M/MM](#) line commands. The data CUT will be the **entire** contents of the specified lines. If the line range contains excluded lines, the excluded lines will **NOT** be included.

The lines are moved or copied from the session to the clipboard. Lines are specified by either the [C](#) (Copy) or [M](#) (Move) line commands, or [CC](#) or [MM](#) block line commands.

If the line numbers are provided as part of the command, or the [C](#) or [CC](#) line commands are used to identify the lines, the lines are copied to the clipboard. If the [M](#) or [MM](#) line commands are used to identify the lines, the lines are copied to the clipboard and deleted from the edit session (in effect, moving them).

- The second is a Windows style 'block' select using the mouse to 'drag' a frame around the text to be selected. The frame is drawn by holding the left mouse button down and dragging open a box frame around the desired text. You may only select text currently displayed on the text window. Scrolling while selecting text is **not** supported. If you require a greater vertical selection range, use

the EXCLUDE command to collapse the range so that the top and bottom of the range are both visible on the same screen. Unlike the 'classic' method above, block select **DOES** include excluded lines in the selected text.

When using block select, the from-line and to-line operands of CUT are not supported.

The optional APPEND operand indicates that you wish the selected data to be **appended** to the current clipboard contents. Without the APPEND operand the clipboard is cleared of any current contents before adding the selected lines.

See also information on CLIP mode in [Windows Clipboard](#).

DELETE - Delete selected lines

DELETE - Delete selected lines

[Previous](#) [Top](#) [Next](#)

Syntax

```
DELETE | DEL [from-line to-line] [ALL X | ALL NX]
```

Operands

ALL	The keyword ALL must be specified with X or NX to avoid unintentional mass deletes.
X NX	Specifies either all X'd lines, or all non-X'd lines are to be deleted.
from-line	The starting line number to be deleted. This operand may also use a bookmark reference.
to-line	The ending line number to be deleted. This operand may also use a bookmark reference. NOTE: An arbitrary large number (e.g. 9999999) is always taken to mean the Last Line. If such the ALL keyword becomes mandatory.

Description

There is no specific DELETE ALL command. To delete all lines:

Use the [CANCEL](#) command.

or

Delete by using the following format:

```
Command > DELETE ALL 1 9999999
```

Here are other uses of the DELETE command:

To delete all excluded lines:

```
Command > DELETE ALL X
```

To delete all not excluded lines:

```
Command > DELETE ALL NX
```

To delete all lines within a range:

```
Command > DELETE 50 250
```

EDIT - Edit another file in a new Session

EDIT - Edit another file in a new Session

[Previous](#) [Top](#) [Next](#)

Syntax

```
EDIT [ filename ]
```

Operands

None

If no operand is specified, a standard Windows File Open dialogue will be displayed from which you can locate the file to be edited.

filename

The name of a file to be edited. If a simple filename is entered, it will be looked for in the same directory as the file currently being edited. Of course, you may type any fully qualified name to edit a file in some other directory.

Description

Examples:

```
EDIT
```

Will open a standard Open File dialogue for you to choose a file to edit. Once selected, the file will be opened in a new SPFLite session.

```
EDIT OTHERTEXT.TXT
```

Will open OTHERTEXT.TXT in a new SPFLite session. The file will be looked for in the same directory as the current file being edited.

```
EDIT C:\AUTOEXEC.BAT
```

Will open the specific file in a new SPFLite session.

Default Directory

When no operand at all or only a simple unqualified filename is provided for the EDIT command, the default directory used for searching for the file or for the file open dialog's starting directory will be determined as follows:

- If there is an active file being edited in the tab where the command is issued, then the Path for THAT Active file is used as the default for the command.
- If there is NO active file (i.e. the tab header displays (Empty):
 - If File Manager is active, the current displayed directory of File Manager will be used.
 - If File Manager is NOT active, then the Preferences setting for Default Open directory will be used (Last Used or Working). Remember 'Working' means the Windows working directory when SPFLite was started.

ENCODING - Specify UniCode setting

ENCODING - Specify UniCode setting

[Previous](#) [Top](#) [Next](#)

Syntax

```
ENCODING    [ {encode-format} ]
```

Operands

`encode-format` One of **{ANSI | UTF8 | UTF16 | UTF16LE | UTF16BE}** to indicate the desired encoding (s

Description

The file will be marked to be stored in the encoding type specified as the operand.

Note: If ENCODING is entered without operands, it will display a message indicating the encoding value currently set.

SPFLite will normally process UniCode files automatically, and write the edited results back in the same format as the original file. The ENCODING command allows you to alter the desired output format.

It is also **required** when creating a new Unicode file from scratch, since all new SPFLite edit tabs are started with ENCODING set to ANSI; the ENCODING command should be issued before any SAVE or CREATE commands are issued.

See [Handling UniCode files](#) for more info on working with Unicode files.

END - End the Edit session

END - End the Edit session

[Previous](#) [Top](#) [Next](#)

Syntax

END

Operands

None.

Description

To end an edit session by using END, do one of the following:

- Enter END on the Command line, or
- Press a function key to which END is assigned. The default setting is F3.

The above two methods will process the data as described below, and close the current working tab. If this is the last active tab, then the SPFLite window will be terminated or return to the File Manager.

- Use the standard Windows Close button in the Title bar.

This option will cause END processing to be performed as described below for **ALL** active tabs. SPFLite will terminate following this.

The actions performed on the data being processed in each tab will be as follows:

- If the Tab is a [BROWSE](#) session, it will simply be closed.
- If there is no data present (the tab is (Empty), it will simply be closed.
- If data is present and no changes were made to the data (since Opening or the last actual SAVE), the tab will simply be closed.
- If the data has been changed, then the choice you made in [Preferences - General](#) for AutoSave will be performed. (i.e. Prompt, Always Save, or Never Save).as well as your choice for Automatic Backup.

EXCLUDE - Exclude lines from the Display

EXCLUDE - Exclude lines from the Display

[Previous](#) [Top](#) [Next](#)

Syntax

EXCLUDE | X string-1 [col-1 [col-2]]

[NEXT		PREV		ALL]
[FIRST		LAST]		
[WORD		PREFIX		SUFFIX]

Operands

string-1	The search string you want to control the Exclude with.
col-1	Left column of a range (with col-2) within which the string-1 value must be found. If no col-2 operand must be found starting in col-1.
col-2	The right boundary of a column range which restricts the column range for the search.
NEXT	Starts at the first position after the current cursor location and searches ahead to find the next occurrence of string-1. NEXT is the default.
PREV	Starts at the current cursor location and searches backward to find the previous occurrence of string-1.
ALL	Starts at the top of the data and searches ahead to find all occurrences of string-1.
FIRST	Starts at the top of the data and searches ahead to find the first occurrence of string-1.
LAST	Starts at the bottom of the data and searches backward to find the last occurrence of string-1.
WORD	Locates string-1 when it is delimited on both sides by blanks or other non-alphanumeric characters.
PREFIX	Locates string-1 at the beginning of a word.
SUFFIX	Locates string-1 at the end of a word.

Description

You can use the EXCLUDE command to find a search string, and exclude the line that contains the string from the display.

To exclude the next nonexcluded line that contains the letters ELSE without specifying any other qualifications:

On the Command line, type:

```
Command > EXCLUDE ELSE
```

Press Enter. Since no other qualifications were specified, the letters ELSE can be:

- Uppercase or a mixture of uppercase and lowercase.
- At the beginning of a word (prefix), the end of a word (suffix), or the entire word (word)
- Anywhere within the current boundaries.

To exclude the next line that contains the letters ELSE, but only if the letters are uppercase:

On the Command line, type:

```
Command > EXCLUDE C'ELSE'
```

Press Enter. This type of exclusion is called a character string exclusion (note the C that precedes the search string) because it excludes the next line that contains the letters ELSE only if the letters are found in uppercase. However, since no other qualifications were specified, the exclusion occurs no matter where the letters are found on a nonexcluded line, as outlined in the previous list.

FIND - Find a character string

FIND - Find a character string

[Previous](#) [Top](#) [Next](#)

Syntax

```
FIND | F      string-1 [col-1 [col-2]]  
                [NEXT | PREV | ALL]  
                [X | NX]  
                [FIRST | LAST]  
                [WORD | PREFIX | SUFFIX]
```

Operands

string-1	The search string you want to locate.
col-1	Left column of a range (with col-2) within which the string-1 value must be found. If no col-2 operand must be found starting in col-1.
col-2	The right boundary of a column range which restricts the column range for the search.
NEXT	Starts at the first position after the current cursor location and searches ahead to find the next occurrence of string-1. NEXT is the default.
PREV	Starts at the current cursor location and searches backward to find the previous occurrence of string-1.
ALL	Starts at the top of the data and searches ahead to find all occurrences of string-1.
X	Scans only lines that are excluded from the display.
NX	Scans only lines that are not excluded from the display.
FIRST	Starts at the top of the data and searches ahead to find the first occurrence of string-1.
LAST	Starts at the bottom of the data and searches backward to find the last occurrence of string-1.
WORD	Locates string-1 when it is delimited on both sides by blanks or other non-alphanumeric characters.
PREFIX	Locates string-1 at the beginning of a word.
SUFFIX	Locates string-1 at the end of a word.

Description

You can use the FIND command to locate line(s) within the file which contain a specified string.

To find the next occurrence of the letters ELSE without specifying any other qualifications:

On the Command line, type:

```
Command > FIND ELSE
```

Press Enter. Since no other qualifications were specified, the letters ELSE can be:

- Uppercase or a mixture of uppercase and lowercase.
- At the beginning of a word (prefix), the end of a word (suffix), or the entire word (word).
- In either an excluded or a non excluded line.
- Anywhere within the current boundaries.

To find the next occurrence of the letters ELSE, but only if the letters are uppercase:

On the Command line, type:

```
Command > FIND C'ELSE'
```

Press Enter. This type of search is called a character string search (note the C that precedes the search string) because it finds the next occurrence of the letters ELSE only if the letters are in uppercase. However, since no other qualifications were specified, the letters can be found anywhere in the file, as outlined in the preceding list.

For more information, including other types of search strings, see "[Finding, Changing, and Excluding Data](#)".

NOTE: When the FIND search operand is a Picture string (see [Specifying A Picture String](#)) certain search operands are restricted. i.e. the Find operands **PREV**, **LAST**, **PREFIX** and **SUFFIX** may not be used.

FLIP - Reverse X and NX Lines

FLIP - Reverse X and NX Lines

[Previous](#) [Top](#) [Next](#)

Syntax

```
FLIP
```

Operands

None.

Description

The FLIP command lets you reverse the exclude status of the lines in a file. This is useful when you have used the 'X ALL;FIND ALL xyz' command to find lines containing a string (xyz) and want to see the lines which do NOT contain the string.

HELP - Display the Help file

HELP - Display the Help file

[Previous](#) [Top](#) [Next](#)

Syntax

HELP

Operands

None.

Description

The HELP command will display the SPFLite Help file.

HEX - Change Display To/From HEX

HEX - Change Display To/From HEX

[Previous](#) [Top](#) [Next](#)

Syntax

HEX [{ ON | OFF }]

Operands

ON	The Display will switch to a Vertical HEX display of the data.
OFF	The Display will switch to a Normal Character only display of the data.

Description

The HEX command is used to switch the edit screen between the normal Character only display, to a HEX display which shows the Hex characters representing the normal character in a vertical orientation below the normal character.

In HEX mode, you may alter data either by entering characters on the 'normal' character line as usual, or you may enter valid HEX characters (0-9, A-F) in the vertical positions below the normal character line.

Note: The HEX command may be entered without any operands (On/Off) in which case it acts as a 'Toggle', switching between the two modes.

HIDE - Hide Excluded Lines

HIDE - Hide Excluded Lines

[Previous](#) [Top](#) [Next](#)

Syntax

```
HIDE  [ { ON   |   OFF } ]
```

Operands

ON	HIDE will be turned on and the indicator lines for Excluded data will also be hidden.
OFF	HIDE will be turned off and the indicator line for Excluded data will also be hidden.

Description

The HIDE command will turn on HIDE Mode. In HIDE mode, Excluded lines (XX'd), which are normally collapsed to a single indicator message line indicating how many lines are represented, will be totally removed from the DISPLAY. The lines are of course still present in the Edit work space.

To indicate the presence of these hidden lines, the line number field of the preceding line will be Underlined.

Just as with un-hidden excluded lines, line movement commands spanning these lines will include them in the requested action. And similarly, when line commands use the numeric operand version (e.g. C34 rather than paired CC -> CC) then the hidden/excluded line counts as 1 (one).

Note: The HIDE command may be entered without any operands (On/Off) in which case it acts as a

'Toggle', switching between the two modes.

LOCATE - Scroll to a specific Line Number

LOCATE - Scroll to a specific Line Number

[Previous](#) [Top](#) [Next](#)

Syntax

```
LOCATE | L line-number | bookmark-reference
```

Operands

line-number	The line number of the line you wish to appear at the top of the display. This operand may also be a bookmark reference.
-------------	--

Description

The LOCATE command positions a particular line at the top of the panel. You must specify a line number which exists, if the line number is invalid, no scrolling will occur.

OPEN - Open a file for Editing

OPEN - Open a file for Editing

[Previous](#) [Top](#) [Next](#)

Syntax

```
OPEN [ { file-name | NEW } ]
```

Operands

file-name	The name of the file to be Edited.
NEW	Open a New Tab without a file loaded (e.g. (Empty))

Description

The OPEN command loads a file into the Edit work space for Editing. If no file-name operand is specified, then a standard Windows File Open Dialog will be presented for you to select the file for Editing.

If you are currently editing a file in the Tab, the requested file (or NEW) will be opened in a new Tab.

Default Directory

When no operand at all or only a simple unqualified filename is provided for the OPEN command, the default directory used for searching for the file or for the file open dialog's starting directory will be determined as follows:

- If there is an active file being edited in the tab where the command is issued, then the Path for THAT Active file is used as the default for the command.
- If there is NO active file (i.e. the tab header displays (Empty):
 - If File Manager is active, the current displayed directory of File Manager will be used.
 - If File Manager is NOT active, then the Preferences setting for Default Open directory will be used (Last Used or Working). Remember 'Working' means the Windows working directory when SPFLite was started.

PASTE - Paste data from the Clipboard

PASTE - Paste data from the Clipboard

[Previous](#) [Top](#) [Next](#)

Syntax

PASTE

Operands

None.

Description

PASTE copies data from the Windows Clipboard to the Edit data.

To Paste data into an empty file:

On the Command line, type:

```
Command > PASTE
```

Press Enter. The data is copied.

To Paste data into Edit data that is not empty:

On the Command line, type:

Command > PASTE

Use the [A](#), [B](#) or [O/OO](#) line commands to indicate where the copied data is to be inserted. However, a number indicating that the [A](#) or [B](#) command should be repeated cannot follow the line command. The [O](#) command **can** have a count to indicate the number of lines to be overlaid.

If the Edit data is not empty and you do not specify a destination with the [A](#), [B](#) or [O/OO](#) line command, a Pending message appears in the upper-right corner of the panel and the data is not copied.

See also information on CLIP mode in [Windows Clipboard](#).

PFKEY - Set PF Key Assignment

PFKEY - Set PF Key Assignment

[Previous](#) [Top](#) [Next](#)

Syntax

PFKEY

Operands

None

Description

The PFKEY command will open the [Preferences - PFKEYS](#) Dialog to allow you to modify what command(s) are issued for each PFKey. The command may be either a Primary command, or a Line command. To distinguish line commands, they should be prefixed with a Colon (:). e.g. **:i3** would cause the insert of 3 lines.

The command line may contain the command separator character (Default ";") to cause multiple commands to be issued. If a mixture of Line and Primary commands are 'stacked' together, the Line command **MUST** be the 1st command in the line. e.g. **:a;paste**

Note: You can maintain a constant reminder of the PFK definitions on the bottom of the screen. See [Preferences - Screen](#) for details.

PRINT - Edit data to Printer

PRINT - Edit data to Printer

[Previous](#) [Top](#) [Next](#)

Syntax

PRINT [NONUM] [NX | X] [from-line to-line]

Operands

NONUM	Optional keyword requesting the Line Numbers be omitted from the listing.
NX	If NX is coded, PRINT will only print non-Excluded lines within the specified print range.
X	If X is coded, PRINT will only print Excluded lines within the specified print range.
from-line	The starting line number to be Printed. This operand may also use a bookmark reference.
to-line	The ending line number to be Printed. This operand may also use a bookmark reference.

Description

PRINT directs a hard copy listing of the file being edited to the system default printer. If no from-line / to-line operand(s) are present, the entire file will be printed. If desired, a range of line numbers may be provided as operands of the PRINT command, or the lines may be specified via the [C](#) (Copy) or [CC](#) line commands.

The listing is formatted with a page heading of the file name, and by default, the line numbers of the file will be listed in the Left margin. If desired the keyword NONUM may be entered preceeding the line number range (itself optional) to request no line numbers on the listing.

If multiple pages are required, the pages are numbered in the page footer.

Note: If no line range is given, the entire file is printed.

PROFILE - Set Editor Preferences

PROFILE - Set Editor Preferences

[Previous](#) [Top](#) [Next](#)

Syntax

PROFILE | PROF | PREF

Operands

None

Description

The PROFILE command will Display the Preferences Dialog to allow you to modify SPFLite optional settings. See [Preferences](#) for details.

RCHANGE - Repeat Change

RCHANGE - Repeat Change

[Previous](#) [Top](#) [Next](#)

Syntax

```
RCHANGE
```

Operands

None.

Description

RCHANGE repeats the change requested by the most recent [CHANGE](#) command.

You can use this command to repeatedly change other occurrences of the search string. Note: RCHANGE is normally assigned to a program function key, although you can issue it directly from the Command line.

REPLACE - Replace a File

REPLACE - Replace a File

[Previous](#) [Top](#) [Next](#)

Syntax

```
REPLACE    [file-name] [ from-line to-line ]
```

Operands

file-name	The name of the file you wish to replace.
from-line	The starting line number if only partial Edit data is to be written. This operand may also use a b
to-line	The ending line number if only partial Edit data is to be written. This operand may also use a b

Description

REPLACE will use all or a specified portion of the Edit data to replace an existing external file. If you do not specify a full pathname as part of file-name, then the same directory as the file being edited will be used as the default directory. Note: If the filename does NOT currently exist, the REPLACE command will create the file.

To copy the entire Edit data into a file:

On the Command line, type:

```
Command > REPLACE file-name
```

The filename operand is optional. If you do not specify it, the standard Windows File Open dialog will be displayed. Select the filename to be Replaced from this dialog.

Press Enter. The Edit data is copied to the filename.

To copy only a portion of the Edit data into a new file:

On the Command line, type:

```
Command > REPLACE filename from-line to-line
```

The filename operand is optional. If you do not specify it the standard Windows File Open dialog will be displayed. Select the filename in this dialog.

The from-line and to-line operands specify the starting and ending line numbers to be included in the file. As an alternative to specifying these on the command line, you may use the [CC/CC](#) or [MMMM](#) line commands to indicate the desired lines. Note that the [MMMM](#) selection will cause the lines to be Deleted from the Edit data following successful completion.

Press Enter. The Edit data is copied/moved to the filename.

File-Format

If you wish to write a file in a format other than the normal default specified by your PROFILE CRLF setting see ["Handling Non-Windows Txt Files"](#) for additional info.

Default Directory

When no operand at all or only a simple unqualified filename is provided for the REPLACE command, the default directory used for searching for the file or for the file open dialog's starting directory will be determined as follows:

- If there is an active file being edited in the tab where the command is issued, then the Path for THAT Active file is used as the default for the command.
- If there is NO active file (i.e. the tab header displays (Empty):
 - If File Manager is active, the current displayed directory of File Manager will be used.
 - If File Manager is NOT active, then the Preferences setting for Default Open directory will be used (Last Used or Working). Remember 'Working' means the Windows working directory when SPFLite was started.

RESET - Reset the Display

RESET - Reset the Display

[Previous](#) [Top](#) [Next](#)

Syntax

```
RESET | RES { CHANGE | COMMAND | EXCLUDED | LABEL | SPECIAL }
```

Operands

CHANGE

Will remove all the ==CHG> markers from yhr results of previous CHANGE commands.

COMMAND

Will remove all pending line commands.

EXCLUDED

Will restore all previously Excluded lines (X'd).

LABEL

Will remove all bookmark line labels.

SPECIAL

Will remove all special lines. i.e. COLS, TABS, BNDS

Description

The RESET primary command allows you to selectively remove the markings on data lines. The individual operands will remove **only** their respective characteristics.

If RESET is entered **without** any operands, it will remove all characteristics **except** bookmark labels.

RETRIEVE - Retrieve Previous Commands

RETRIEVE - Retrieve Previous Commands

[Previous](#) [Top](#) [Next](#)

Syntax

RETRIEVE

Operands

None.

Description

The RETRIEVE command will bring back to the Command line the previously issued command for use to modify and re-enter. Successive RETRIEVE commands will retrieve the previous, next previous, etc. commands. The Editor keeps the last 10 unique command lines available for retrieval. Note that the list of previous commands will be saved and recalled between SPFLite sessions.

See also the [Preferences - General](#) option for minimum Retrieve length.

RFIND - Repeat the Find Command

RFIND - Repeat the Find Command

[Previous](#) [Top](#) [Next](#)

Syntax

RFIND

Operands

None.

Description

RFIND repeats the find requested by the most recent [FIND](#) command.

You can use this command to repeatedly search for other occurrences of the search string. Note: RFIND is normally assigned to a program function key, although you can issue it directly from the Command line.

RUN - Execute another Program / Command

RUN - Execute another Program / Command

[Previous](#) [Top](#) [Next](#)

Syntax

`RUN` `command-line`

Operands

`command-line` The command line to invoke the new program as it would be entered to the system CMD proc

Description

The RUN command will issue the command to start the new program / command. It does not wait for completion of the program nor for successful completion status. Note: The command will be issued with the SPFLite data directory as the working directory (C:\Users\UserID\SPFLite). Any path dependant file names used in the command should take this into account.

SAVE - Save data and continue Edit

SAVE - Save data and continue Edit

[Previous](#) [Top](#) [Next](#)

Syntax

`SAVE`

Operands

None.

Description

SAVE writes the data to the same file from which it was retrieved.

If you wish to replace the data in another file, or create an entirely new file, then use the [REPLACE](#), [SAVEAS](#) or [CREATE](#) commands respectively.

NOTE: Any subsequent [CANCEL](#) command issued after a SAVE has been done will only throw away changes made since the last save, **NOT** all changes since the beginning of the entire Edit session.

file-format

If you wish to save a file in a format other than the normal default specified by your PROFILE CRLF setting see ["Handling Non-Windows Txt Files"](#) for additional info.

SAVEAS - Save data as a new file & Switch to it

SAVEAS - Save data as a new file & Switch to it

[Previous](#) [Top](#) [Next](#)

Syntax

```
SAVEAS    [file-name]
```

Operands

`file-name` The name of the file you wish to create.

Description

SAVEAS will use all of the Edit data to create a new external file. If you do not specify a full pathname as part of file-name, then the file will be created in the same directory as the file being edited. If you do not specify a file name, a normal Windows Save dialogue will be presented to allow you to select the location and file name to be created. Note: If the filename already exists, then use the [REPLACE](#) command instead.

Following creation of the new file, the current Edit Tab will be 'switched' to the newly created file. This is identical to the way 'Save As' is handled by normal windows Editors.

To copy the entire Edit data into a new file:

On the Command line, type:

```
Command > SAVEAS file-name
```

The filename operand is optional. If you do not specify it, the standard Windows File Open dialog will be displayed. Enter the filename in this dialog.

Press Enter. The Edit data is copied to the new filename and the current Tab switched to the new filename.

NOTE:

If the current Edit data has not been saved in the original file and you wish any changes to be saved there, do so before issuing the SAVEAS command, otherwise the changes will only be saved in the new file created by SAVEAS.

Scroll Commands - UP / DOWN / LEFT / RIGHT

Scroll Commands - UP / DOWN / LEFT / RIGHT

[Previous](#) [Top](#) [Next](#)

Syntax

```
{UP | DOWN | LEFT | RIGHT }  
  { number |  
    {HALF | H } |  
    {MAX | M } }
```

Operands

number	The number of columns / lines you wish to scroll.
HALF / H	Scroll half the height / width of the screen.
MAX / M	Scroll the Maximum distance to the Top/Bottom or Left/Right of the data.

Description

The four scroll commands are used to reposition the screen display to a different portion of the data being edited. Normally these commands are assigned as defaults of one or more PF keys. In this implementation, the UP/DOWN keys may not NEED to be assigned to function keys as they are always permanently assigned to the normal keyboard PageUp and PageDown keys.

Since normal PF key processing always prefixes the command line with whatever PF key is pressed, the operands are normally entered on the command line and then the PF Key is pressed.

For Example, to scroll down to the bottom of the data being edited:

- Enter MAX or M on the command line
- Press the key for DOWN (say the PageDown key)
- The resulting command issued internally would become DOWN MAX and the screen would be positioned at the bottom of the data

If no operand is entered, the following rules apply:

- If the cursor is NOT in a line command area nor in the actual data area, the screen will be scrolled one screen width/height.
- If the cursor IS within the data area, then the screen will be scrolled so that the cursor line/character becomes the screen boundary. i.e. a scroll down brings the line to the top of the screen, a scroll up brings the line to the bottom of the screen, etc. This is the equivalent of the real SPF editor scroll default of CSR.

SORT - Sort the Edit data

SORT - Sort the Edit data

[Previous](#) [Top](#) [Next](#)

Syntax

```
SORT      [criteria-1]  [criteria-2]
```

Where each criteria-n is [start-col [end-col] [A | D]]

Operands

criteria-1 The primary columns and direction on which to Sort.

criteria-2 The secondary columns and direction on which to Sort.

Each criteria-n description is as follows:

start-col The starting (leftmost) position of the Sort key.

end-col The ending (rightmost) position of the Sort key.

A | D An 'A' indicates an ascending sort, a 'D' indicates a descending sort.

Description

For all sorts, the criteria-n operands cannot specify a column range spanning more than 64 columns, this is the maximum size of each key (primary and secondary).

For SORT with no operands, the editor will default to an ascending sort on columns 1 thru 64; if only a start-col operand is provided for one of the criteria-n, then the key will be from the specified column thru that column + 63.

The A | D operand may precede or follow the column operands, or may be omitted if an ascending sort (the default) is desired.

Examples

```
SORT 1 10 A
```

This will sort the data in ascending order of columns 1 thru 10.

```
SORT 22
```

This will sort the data in ascending order of columns 22 thru 85. (22 + 63 = 85).

```
SORT 5 12 D 21 30 A
```

This will sort the data in descending order on columns 5 thru 12, secondary sort on columns 21 thru 30 in ascending order.

SWAP - Switch to Prev / Next Tab

SWAP - Switch to Prev / Next Tab

[Previous](#) [Top](#) [Next](#)

Syntax

```
SWAP [ PREV | NEXT ]
```

Operands

PREV | NEXT Switch the active Tab to either the Previous (Left) or Next (Right) tab.

Description

The active Edit Tab can be altered with the mouse by simply left-clicking on the desired Tab header at the top of the window.

The SWAP command can be assigned to a PFKEY to allow keyboard controlled switching of the active tab via the keyboard.

TABS - Turn TABS On or Off

TABS - Turn TABS On or Off

[Previous](#) [Top](#) [Next](#)

Syntax

TABS { ON | OFF }

Operands

ON	The Tab key will use tabs defined in a Tab control line.
OFF	The Tab key will only perform basic field tabbing.

Description

Normally, the Tab key simply moves from field to field. e.g. Command field to 1st line number area, line number area to data area, data area to next line number area, etc.

If Tabs have been defined with the [TABS](#) Line Command, and the [TABS ON](#) command is in effect, then the Tab key will also tab in the data area as defined by the TABS control line.

When [TABS OFF](#) is issued, no use is made of any [TABS](#) defined tabs, only basic tabbing as described above will take place.

UNDO - Undo changes

UNDO - Undo changes

[Previous](#) [Top](#) [Next](#)

Syntax

UNDO

Operands

None.

Description

Undo will 'back off' all changes since the last Attention Key was pressed. I.e. the previous Enter or PFKey.

Additional UNDO's may be done up to the maximum number of UNDO's which have been saved. See [Preferences - General](#) how to specify this value.

Sample Font Package

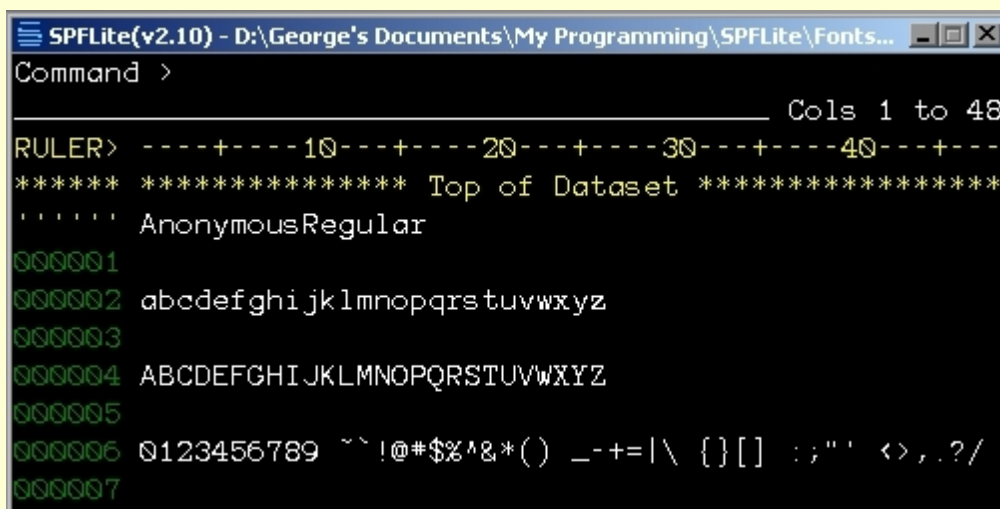
Sample Font Package

SPFLite can only utilize fixed pitch fonts for the screen. Although there are many thousands of free fonts available on the web, the vast majority of those available are proportional fonts and are thus unusable. To supplement the usual fonts available with Windows (Courier, Terminal, etc.) there is available on the SPFLite web site download page is an optional file containing a variety of freely usable fixed fonts which you may wish to use in your configuration.

Shown below are some screen shots of the fonts actually in use to help you choose. If you think one of these may be useful, simply download the `Fonts.zip` file and use the Control Panel Fonts tool to install whichever you desire. Then simply alter the Font settings in [Preferences - Screen](#) to specify the font.

Once you install the fonts, you can experiment with them within SPFLite by using [Preferences - Screen](#) to alter the font; when you close the Preferences Dialog SPFLite will dynamically alter the running window.

AnonymousRegular

A screenshot of a SPFLite window titled "SPFLite(v2.10) - D:\George's Documents\My Programming\SPFLite\Fonts...". The window displays a command prompt interface with the following text:

```
Command >
RULER> -----10-----20-----30-----40-----
***** ***** Top of Dataset *****
' ' ' ' ' ' ' ' AnonymousRegular
000001
000002 abcdefghijklmnopqrstuvwxyz
000003
000004 ABCDEFGHIJKLMNOPQRSTUVWXYZ
000005
000006 0123456789 ~`!@#$%^&*() _-+=|\ {}[] ;:' '<>.,?/
000007
```

PixelCarnage Monospace

```
SPFLite(v2.10) - D:\George's Documents\My Programming\SPFLite\Fonts...
Command >

RULER> -----10-----20-----30-----40-----
***** ***** Top of Dataset *****
PixelCarnage Monospace
000001
000002 abcdefghijklmnopqrstuvwxyz
000003 ABCDEFGHIJKLMNOPQRSTUVWXYZ
000004
000005
000006 0123456789 ~`!@#%&^&*() _-+=|\ {}[] ;:"' <>.,?/
000007
```

ProggyClean

```
SPFLite(v2.10) - D:\George's Documents\My Program...
Command >

RULER> -----10-----20-----30-----40-----
***** ***** Top of Dataset *****
ProggyClean
000001
000002 abcdefghijklmnopqrstuvwxyz
000003 ABCDEFGHIJKLMNOPQRSTUVWXYZ
000004
000005
000006 0123456789 ~`!@#%&^&*() _-+=|\ {}[] ;:"' <>.,?/
000007
```

ProggySmall

```
SPFLite(v2.10) - D:\George's Documents\My Program...
Command >

RULER> -----10-----20-----30-----40-----
***** ***** Top of Dataset *****
ProggySmall
000001
000002 abcdefghijklmnopqrstuvwxyz
000003 ABCDEFGHIJKLMNOPQRSTUVWXYZ
000004
000005
000006 0123456789 ~`!@#%&^&*() _-+=|\ {}[] ;:"' <>.,?/
000007
```

ProggySquare

```
SPFLite(v2.10) - D:\George's Documents\My Program...
Command >

RULER> -----10-----20-----30-----40-----
***** ***** Top of Dataset *****
ProggySquare
000001
000002 abcdefghijklmnopqrstuvwxyz
000003 ABCDEFGHIJKLMNOPQRSTUVWXYZ
000004
000005
000006 0123456789 ~`!@#%&^&*() _-+=|\ {}[] ;:"' <>.,?/
000007
```

QuickType Mono

```
SPFLite(v2.10) - D:\George's Documents\My Programming\SPFL...
Command >
Cols 1 to 48
RULER> -----10-----20-----30-----40-----
***** ***** Top of Dataset *****
QuickType Mono
000001
000002 abcdefghijklmnopqrstuvwxyz
000003
000004 ABCDEFGHIJKLMNOPQRSTUVWXYZ
000005
000006 0123456789 ~`!@#%&*() _-+=|\ {}[] ;:"' <>.,?/
000007
```

Raize

```
SPFLite(v2.10) - D:\George's Documents\My Programming\SPFL...
Command >
Cols 1 to 48
RULER> -----10-----20-----30-----40-----
***** ***** Top of Dataset *****
Raize
000001
000002 abcdefghijklmnopqrstuvwxyz
000003
000004 ABCDEFGHIJKLMNOPQRSTUVWXYZ
000005
000006 0123456789 ~`!@#%&*() _-+=|\ {}[] ;:"' <>.,?/
000007
```

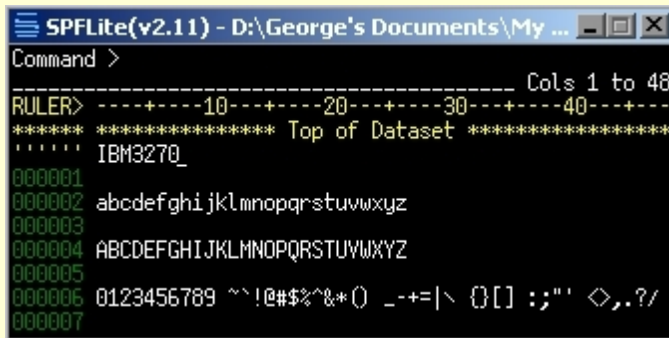
Raster

```
SPFLite(v2.10) - D:\George's Documents\My Programming\SPFLite\Fonts...
Command >
Cols 1 to 48
RULER> -----10-----20-----30-----40-----
***** ***** Top of Dataset *****
Raster_
000001
000002 abcdefghijklmnopqrstuvwxyz
000003
000004 ABCDEFGHIJKLMNOPQRSTUVWXYZ
000005
000006 0123456789 ~`!@#%&*() _-+=|\ {}[] ;:"' <>.,?/
000007
```

Triskweline

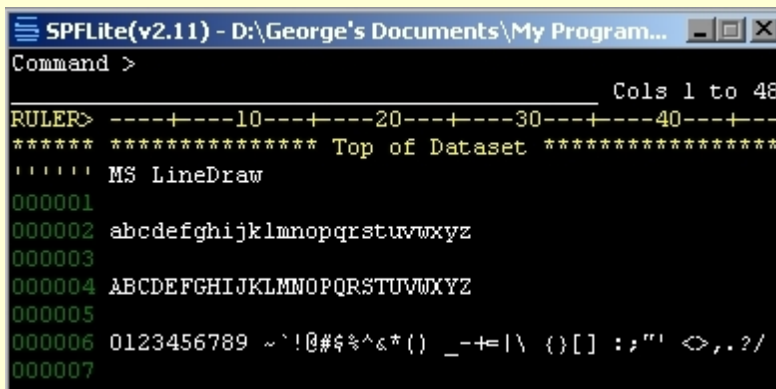
```
SPFLite(v2.10) - D:\George's Documents\My Programming\SPFL...
Command >
Cols 1 to 48
RULER> -----10-----20-----30-----40-----
***** ***** Top of Dataset *****
000001
000002 abcdefghijklmnopqrstuvwxyz
000003
000004 ABCDEFGHIJKLMNOPQRSTUVWXYZ
000005
000006 0123456789 ~`!@#%&*() _-+=|\ {}[] ;:"' <>.,?/
000007
***** ***** Bottom of Dataset *****
```

IBM3270



```
SPFLite(v2.11) - D:\George's Documents\My ...
Command >
----- Cols 1 to 48
RULER> -----10-----20-----30-----40-----
***** Top of Dataset *****
IBM3270_
000001
000002 abcdefghijklmnopqrstuvwxyz
000003
000004 ABCDEFGHIJKLMNOPQRSTUVWXYZ
000005
000006 0123456789 ~`!@#%&^*() _-+=|\ {}[] :;'" <>.,?/
000007
```

MS LineDraw



```
SPFLite(v2.11) - D:\George's Documents\My Program...
Command >
----- Cols 1 to 48
RULER> -----10-----20-----30-----40-----
***** Top of Dataset *****
MS LineDraw
000001
000002 abcdefghijklmnopqrstuvwxyz
000003
000004 ABCDEFGHIJKLMNOPQRSTUVWXYZ
000005
000006 0123456789 ~`!@#%&^*() _-+=|\ {}[] :;'" <>.,?/
000007
```

Colorize Files

Colorize Files

[Previous](#) [Top](#)

SPFLite can display source files with keywords and delimiters high-lighted in various colors to assist in the readability of source code. To do this, SPFLite requires a Colorization file which describes the particular source language (e.g. reserved words, delimiters used, comment indicators, etc.)

When the PROFILE option COLORIZE is set to YES, SPFLite will look for a Colorization file to use in the colorization process. If no matching colorization file can be found, the file will be displayed as if colorization had not been requested. Colorization is based on the **file extension** of the file being edited.

For example, if editing a file MYSOURCE.BAS (a BASIC source program), SPFLite will look in its data directory for a file called:

SPFCLr**BAS**.clr

Note the file extension (**BAS**) is inserted into the base filename of SPFCLr.Clr to create a unique name. Similarly, for a C++ source program stored with an extension of .cpp, SPFLite would look for the file SPFCLr**CPP**.clr.

Many times, source for one language is stored in a variety of file types (Header, Include, Macro, etc.) which would normally mean separate CLR files for each filetype, a real maintenance nuisance. SPFLite allows you to create a CLR file with only one line, an ALIAS entry, to point at the main CLR file to be used. In this way simple 1 line CLR files can be created for the other files, and the ALIAS command used to direct operation to the main CLR file. See the ALIAS command description below.

The SPFLite web site has a downloadable ZIP file containing a set of sample colorization files for some of the common types. Note these files are **NOT** to be considered exact and complete for **any** of the languages provided. I created these very quickly from some keyword lists located on the web and they have received little or no real testing. The only one I'm fairly sure of is SPFClrPowerBasic.clr since that is **my** language of choice.

Locate the appropriate file for your language, copy it to the SPFLite Data directory (normally C:\Users\userID\SPFLite) and rename the characters in the file name between **SPFClr** and **.Clr** to the file extension used for you source files. For example, if using PowerBasic as I do, you would copy and rename SPFClrPowerBasic.clr to SPFClrBAS.clr.

Depending on what changes you may have made to the default color selections for SPFLite in [Preferences- Screen](#), you may have to alter the color choices in the SCHEME statements to better suit your color preferences.

The structure is fairly simple (described below) and it should be fairly easy to create or update these for any language.

Should you update any of these, or create new ones for other languages, I encourage you to send them in to me; I will update the collection and make them available for all users.

Colorize File Command Syntax

Colorize files consist of either

- comments (blank lines, or lines starting with a ; in position 1)
- definition statements

Definition Statements

ALIAS filetype

The ALIAS statement (which may be the ONLY statement in a CLR file), simply points at another filetype which should be used in place of **this** CLR file. Note: other statements may be present in a file with an ALIAS statement, but they will all be ignored.

e.g. a file SPFCLRH.CLR can contain the 1 line 'ALIAS CPP' . Then when editing, say, the file MYHEADER.H, the actual CLR file that would be used would be SPFCLRCPP.CLR.

SCHEME n fg bg

You may specify up to 10 different color schemes to use in colorizing different parts of the source program. e.g. one scheme for comments, one for compiler directives, one for delimiters, etc. etc. The first operand of **SCHEME** (the n) is the scheme number and should be a single digit from 0 thru 9. Note that **SCHEME 0** is reserved for normal text which is anything **other** than what can be identified by the other definition statements. The **fg** and **bg** operands specify

the foreground(text) and background colors for this particular scheme.

The fg and bg color values may be specified as simple 'standard' colors as follows:

0	Black	4	Red	8	Gray	12	Light red
1	Blue	5	Magenta	9	Light blue	13	Light magenta
2	Green	6	Brown	10	Light green	14	Yellow
3	Cyan	7	White	11	Light cyan	15	Bright white

or it can use the full color range by specifying the color value as **&hbbggrr** where bb, gg, and rr are the HEX values respectively for Blue, Green, and Red.

A handy place to select colors and obtain this Hex value is <http://www.free-webmaster-tools.com/colorpicker.htm>.

COMMENT *n* *sn* *start* *end*

Two comment definitions are supported **COMMENT1** and **COMMENT2**.

The **sn** operand refers to the SCHEME number used to display the comment. Each **COMMENT***n* statement **may** reference different scheme's if desired, or the same scheme.

The **start** operand specifies the character (string) which indicates the beginning of comments.

The **end** operand can have one of three meanings:

1. **0** (zero) indicates the **start** string can begin anywhere in a line and the remainder of the line is a comment.
2. **n** (positive number) indicates the **start** operand is to be recognized **only** in this position and the remainder of the line is a comment. (Like the COBOL comment indicator in position 7.)
3. **string** data indicates the comment starts with the **start** operand and ends with **this** operand. A typical example of this would be the comment markers /* */ or (* *).

Note: SPFLite does **not** support multi-line comments.

QUOTED *sn*

The **sn** operand specifies the SCHEME number to be used for the display of quoted literal strings.

ESCAPECHR *x*

Literals in some languages allow embedded special characters (including the single or double quote) within a literal if they are preceded by an 'escape'

character, which basically says to ignore the next character as a delimiter. For example, if **ESCAPECHR** \ is specified, then the following string would be a valid quoted literal.

```
'What\'s the matter, didn\'t you understand'
```

DELIMS string

The **DELIMS** statement defines the delimiters which the language uses to separate operands and language elements. Normally the string would be something like **+*/=()^<>[]{};** This string typically varies mostly due to what characters are allowable in variable names where sometimes characters like **_** and **.** are allowed. Note that when the language uses relational operators like BASIC's **<>** or **>=** that the **individual** characters are what should be entered. Note that the space need not be specified, it is **always** a delimiter.

MIXEDCASE yes | no

Some computer languages are sensitive to the case of the keywords and can have duplicate keywords that have different meaning depending on their case. (e.g. void and Void in Java) If **MIXEDCASE YES** is specified, then the keyword search will be sensitive to the case of the words. If the default **MIXEDCASE NO** is specified, keywords will be searched for regardless of case.

WORD sn string

The **WORD** statement defines the reserved words for the particular language.

The **sn** operand refers to the SCHEME number used to display the word. Different scheme numbers may be used to classify reserved words into various categories.

The **string** specifies the actual reserved word. No embedded blanks are allowed, use multiple **WORDS** if needed.

NOTE: Each character entered in the **DELIMS** statement should be entered also with a **WORD** statement.

Sample Colorize File

```
; SPFLite Colorize file for DOS Batch files
; 0   Black      4   Red          8   Gray          12  Light red
; 1   Blue       5   Magenta     9   Light blue     13  Light magenta
; 2   Green      6   Brown       10  Light green    14  Yellow
; 3   Cyan       7   White       11  Light cyan     15  Bright white
;
; or specify a Full Color value as &hbbggrr where bb, gg, and rr are the
HEX
; values respectively for Blue, Green, and Red.
```



```

;      e.g the above values are equivalent to:
;
;  0      &h000000    4 &h0000C4    8   &h808080        12 &h0000FF
;  1      &h800000    5 &h800080    9   &hFF0000        13 &hFF00FF
;  2      &h008000    6 &h004080   10   &h00FF00        14 &h00FFFF
;  3      &h808000, 7 &hC4C4C4   11   &hFFFF00        15 &hFFFFFF
;
COLOR      0   7  0          Default scheme White on Black
COLOR      1   2  0          Scheme 1 - Green on Black
COLOR      2  14  0          2 - Yellow on Black
COLOR      3  15  0          3 - Bright White on Black
COLOR      4   3  0          4 - Cyan on White
COLOR      5   8  0          5 - Gray on Black
COMMENT1 1  REM   0          Either REM
COMMENT2 1  @REM  0          or @REM indicate comments
QUOTED    5                      Quoted strings use SCHEME 5
DELIMS     , ( ) =             Brackets, commas and = are delimiters
WORD       2   ,              The delimiters themselves
WORD       2   (              use scheme 2
WORD       2   )              .
WORD       2   =              .
WORD       4  aux             Reserved words
WORD       3  call            .
WORD       3  cd              .
WORD       4  com1            .
WORD       4  com2
WORD       4  com3
WORD       4  com4
WORD       4  con
WORD       3  defined
WORD       3  dir
WORD       3  do
WORD       3  echo
WORD       3  else
WORD       3  endllocal
WORD       3  equ
WORD       3  errorlevel
WORD       3  exist
WORD       3  exit
WORD       3  for
WORD       3  geq
WORD       3  goto
WORD       3  gtr
WORD       3  if
WORD       3  in
WORD       3  leq
WORD       4  lpt1
WORD       4  lpt2
WORD       4  lpt3
WORD       3  lss
WORD       3  neq
WORD       3  not
WORD       3  nul
WORD       3  pause
WORD       4  prn
WORD       3  set
WORD       3  setlocal
WORD       3  shift

```

